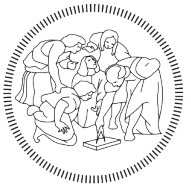

POLITECNICO DI MILANO
DIPARTIMENTO DI ELETTRONICA, INFORMAZIONE E BIOINGEGNERIA
DOCTORAL PROGRAMME IN INFORMATION TECHNOLOGY

AN ASSESSMENT OF RECENT TECHNIQUES FOR QUESTION DIFFICULTY ESTIMATION FROM TEXT

Doctoral Dissertation of:
Luca Benedetto

Supervisor:
Prof. Paolo Cremonesi

Co-supervisors:
Andrea Cappelli, PhD; Roberto Turrin, PhD

Tutor:
Prof. Francesco Amigoni

The Chair of the Doctoral Program:
Prof. Barbara Pernici

2021 – Cycle XXXIII

Abstract

In the educational domain, question difficulty estimation consists in estimating a numerical or categorical value representing the difficulty of an exam question. It is traditionally performed with manual calibration or pretesting, which have several limitations: indeed, they are either subjective or introduce a long delay between the time of question creation and when the new question can be used to assess students. Recent research tried to overcome these shortcomings by leveraging Natural Language Processing techniques to perform question difficulty estimation using as input only the textual content of the questions, which is the only information that is always available at the time of question creation. Specifically, research proceeded along two main directions: supervised and unsupervised approaches, which have peculiar advantages and limitations. This thesis explores previous literature in both research directions and evaluates several models, including novel approaches, on real world datasets coming from different educational domains. The experimental results show that model accuracy heavily depends on the characteristics of the questions under consideration and, most importantly, the educational domain: while simple models based on readability indexes and linguistic measures are generally fairly accurate on reading comprehension questions, the calibration of questions assessing domain knowledge requires more advanced models based on the attention mechanism and Transformers.

Contents

CHAPTER 1

Introduction

This Chapter introduces this thesis. We start by presenting the motivation behind it: why the estimation of question difficulty is important in the educational domain and why, in particular, we might want to perform it using the textual content of the questions (§1.1). Then, we present the main research contributions of this work (§1.2), the list of publications which it is based upon (§1.3), and its structure (§1.4).

1.1 Motivation

Recent years have witnessed an exponential growth in the availability of digital services, and the educational domain was no exception [1]. The popularity of Massive Open Online Courses increased massively, enabling hundreds of thousands of students to access online learning content and online exams [41]. Similarly to what happened in other domains, this increase in the amount of available data enabled the development of many data driven techniques to improve students' learning experience and the effectiveness of learning material. Two examples of this trend are the automatic recommendation of learning content targeted to the needs of each student [27, 79], and the development of virtual teaching assistants that can

support students by automatically answering their questions [11,12,29,38].

Another example is students' assessment, which is the task of estimating the knowledge level of the students taking an exam. Even though testing theories for students' assessment – such as Item Response Theory (IRT) [45] – had already been developed in psychometrics research and had been in use for decades (especially for high-stakes exams), they could now be used on larger pools of students. These theories make diverse assumptions but, in most of them, a crucial step is Question Difficulty Estimation (QDE), which is the task of estimating a value, either numerical or categorical, representing the difficulty of a question. Intuitively, QDE – which is also referred to as "question calibration" – can be interpreted as the analogous of students' assessment, with the difference that we consider question difficulty rather than students' skill.

The role of QDE in the educational domain is crucial, and in the following subsections we describe the reasons behind this importance (§1.1.1), the traditional methods to perform it along with their limitations (§1.1.2), and how Natural Language Processing can be leveraged to overcome such limitations (§1.1.3).

1.1.1 About the Importance of Question Difficulty Estimation

The best way to understand the importance of QDE is through some examples. The first one is Computer Adaptive Testing (CAT) [73], which consists in providing students with questions whose difficulty is targeted to their proficiency. Research showed that CAT is highly beneficial to the students' learning outcome [18] and requires a lower amount of questions to accurately assess their skill. In case of miscalibrated questions (i.e. assessment items whose difficulty has been erroneously estimated), the effectiveness of CAT is reduced massively. Indeed, according to Vygotsky's zone of proximal development [119], the range of suitable exercises for a learner is very narrow: exercises that are not challenging easily lead to boredom and stagnation, whereas overly complex exercises might result in frustration. In both cases, the learning experience is worse than if the selected questions are of appropriate difficulty.

Moreover, even considering exercises whose difficulty is suitable to the skill level of a given student, it was shown that slightly easier exercises lead to better short term engagement, whereas more difficult exercises are better for long term engagement [87]. Therefore, being able to accurately estimate the difficulty of exam questions enables developers of educational technology to focus on the desired type of engagement.

Another example of the importance of QDE are the testing theories – such as IRT – which leverage the difficulty of exam questions to numerically estimate the skill level of the students who answered them. In other words, a student that correctly answers difficult questions is assigned an estimated knowledge level higher than a student who correctly answers only easy questions. As a direct consequence, miscalibrated items affect the accuracy of students' assessment. Moreover, regardless of the testing theory that is used in a given exam, a test that is too easy or too difficult results in a limited range of scores, and such a skewed score distribution is not informative [4]. This is also the reason why, in high-stakes exams, all the questions have to pass a thorough quality control before being used to assess students, in order to keep only the ones that are informative [124].

1.1.2 Traditional Approaches to QDE and their Limitations

Traditionally, QDE is performed with either i) manual calibration [3] or ii) pretesting [69].

Manual calibration consists of having one (or more) domain experts manually selecting a numerical or categorical value representing the difficulty of each question. This can be fairly quick if performed at the time of manual question creation, but it is not scalable to large amounts of questions, and it cannot be used in the context of automatic question generation. Also, it is intrinsically subjective and it was shown that human annotations are often in disagreement with each other. Indeed, instructors already know the solutions to the assessment items and cannot always anticipate the confusion an exercise might cause for learners.

Pretesting, on the other hand, consists of estimating question difficulty based on posterior performance measures. Specifically, the questions under pretesting are deployed in an exam, as if they were standard questions, but are not used for assessment. The other questions in the exam are used to assess the students, and their answers – together with the estimated skill level – are used to calibrate the questions under pretesting. This approach leads indeed to an accurate and reliable estimation of question difficulty, which is the reason why it is generally used for high-stakes exams, but it introduces a long delay between the time of question generation and when the questions can be used to assess students. Also, it requires the new questions to be shown to students before being actually used to score them, which is in some cases undesirable, as they might be leaked or exposed too often [120].

1.1.3 Natural Language Processing for QDE

In order to overcome the limitations of traditional approaches to question calibration, recent research has attempted to leverage Natural Language Processing (NLP) to automatically estimate question difficulty at creation time. Such works are all based on the idea that question text is the only information that is always available at the time of question generation and, if we were able to perform an accurate QDE from textual content, we would overcome the need for pretesting and manual calibration, and their limitations.

In this thesis, we focus on the task of QDE from Text (QDET), evaluating and comparing different approaches proposed to address it, both the ones modeling it as a supervised task and the ones modeling it as an unsupervised task, including a novel approach proposed in this thesis for the first time.

Almost all the approaches proposed in previous research are trained in a supervised manner. Starting from a set of questions of known difficulty – generally calibrated with pretesting, which is more reliable than manual calibration – a machine learning model is trained in a supervised manner to estimate question difficulty from text. The trained model can then be used to estimate the difficulty of newly created questions (of unknown difficulty) without the need for pretesting or manual calibration. In this work, we categorize the approaches proposed in previous literature according to a taxonomy based on question characteristics, and experiment on three datasets (two being publicly available) from diverse educational domains to evaluate how different architectures perform, especially focusing on the relevance of different types of features.

Supervised QDET targets the limitations of traditional approaches to QDE, but it has some limitations of its own: crucially, it requires a large dataset of calibrated questions for training, which might hinder its effectiveness. The required amount of questions depends on the specific architecture (e.g. models based on neural networks generally require larger datasets than simple regression models), but even the simplest models require hundreds or thousands of training questions. Also, many architectures for supervised QDET leverage information related to the semantic meaning of the questions, therefore such models can only be used to calibrate questions belonging to the same educational domain as the training items. As an example, let us assume that we trained a model to calibrate math questions from their text; there is no guarantee that the same model – without retraining – would work on questions about medicine or geography, and this is

true even at a smaller scale (e.g. different mathematical topics).

Targeting these issues, some recent research experimented with unsupervised approaches to QDET. Compared to the supervised techniques, the main advantage of unsupervised approaches is that they do not require a large training set of calibrated questions, although they might require supervision in a related (but different) task. There is very limited research on unsupervised QDET, and it mainly focuses on one of three aspects: i) the readability of the question, ii) the similarity between the correct answer and the question or (in the case of multiple choice questions) between the correct choice and the distractors, or iii) the performance of Question Answering (QA) models trained to answer the questions under calibration.

In this thesis, we experiment on two real world datasets (one being publicly available) to evaluate the techniques proposed by previous research for unsupervised QDET, and propose and evaluate a novel approach.

We experiment on the supervised and the unsupervised approaches in two separate chapter because they make different assumptions and have different requirements; indeed, they cannot be always used on the same set of questions. However, to conclude this thesis, we also perform a comparison of their performance, hoping to provide some useful guidelines for practitioners and researchers addressing these tasks.

1.2 Research Contributions

In this section we detail the research contributions of this study, which are related to i) the analysis of the performance of different architectures and the importance of different features in the task of QDET, and ii) the proposal of a new approach to perform QDET in an unsupervised manner.

- **Analysis of the importance of different types of features and of the effectiveness of different architectures in the task of supervised QDET.** First, we categorize the approaches proposed in previous literature according to a taxonomy based on question characteristics. Then, we evaluate how different families of algorithms, including two approaches we proposed in previous research, perform in the task of supervised QDET using three real world datasets from different educational domains.

- **Analysis of the performance of the state of the art models to perform unsupervised QDET.** We evaluate and compare the models recently proposed for unsupervised QDET, including an approach we

proposed in previous research, using two experimental datasets from different educational domains.

- **Proposal of a new approach to perform QDET in an unsupervised manner.** We propose a novel approach for unsupervised QDET, and compare it with the previously proposed approaches.
- **Comparison of supervised and unsupervised approaches to QDET.** We compare the performance of supervised and unsupervised techniques to QDET, highlighting their strengths and weaknesses.

1.3 Publication Record

This thesis is partially based upon the following published articles (ordered by publication date).

- [10] L. Benedetto, A. Cappelli, R. Turrin, P. Cremonesi; "R2DE: a NLP approach to estimating IRT parameters of newly generated questions"; in Proceedings of the Tenth International Conference on Learning Analytics and Knowledge; 2020.
- [9] L. Benedetto, A. Cappelli, R. Turrin, P. Cremonesi; "Introducing a framework to assess newly created questions with Natural Language Processing"; in International Conference on Artificial Intelligence in Education; 2020.
- [8] L. Benedetto, G. Aradelli, P. Cremonesi, A. Cappelli A. Giussani, R. Turrin; "On the application of Transformers for estimating the difficulty of multiple choice questions from text"; in Proceedings on the 16th Workshop on Innovative Use of NLP for Building Educational Applications; 2021.
- [75] E. Loginova, L. Benedetto, D. Benoit, P. Cremonesi; "Towards the application of calibrated Transformers to the unsupervised estimation of question difficulty from text"; in Proceedings of the International Conference on Recent Advancements in Natural Language Processing (RANLP); 2021.

It is also partially based on the following survey paper, which is currently under revision for publication at the ACM Computing Surveys Journal:

- L. Benedetto, P. Cremonesi, A. Caines, P. Buttery, A. Cappelli, A. Giussani, R. Turrin; "A survey on recent approaches to Question Difficulty Estimation from text".

Considering the research activity that led to the writing of this thesis, other papers were published, even though this thesis is not directly based on them:

- [12] L. Benedetto, P. Cremonesi, M. Parenti; "A virtual teaching assistant for personalized learning"; SIR Workshop at the 27th ACM International Conference on Information and Knowledge Management (CIKM); 2018;
- [11] L. Benedetto, P. Cremonesi; "Rexy, a configurable application for building virtual teaching assistants"; in IFIP Conference on Human-Computer Interaction; 2019

1.4 Thesis Outline

- Chapter 1 introduced this thesis, describing the motivations and research contributions, as well as listing the publications that are directly or indirectly related to the work carried out for writing this thesis.
- Chapter 2 defines the concept of *question difficulty*, describing the question characteristics that affect it and presenting the theories of testing that are referred to or used in this thesis.
- Chapter 3 provides a statistics and machine learning background, introducing the techniques and models used in the rest of this document.
- Chapter 4 presents the experimental datasets which are used in this work.
- Chapter 5 focuses on supervised QDET, from presenting the relevant literature and categorizing it according to a taxonomt based on question characteristics, to describing the experimental setup and the experimental results.
- Chapter 6 focuses on unsupervised QDET, introducing the relevant literature and the newly proposed model, as well as showing the experimental results.
- Chapter 7 provides a comparison of the techniques presented in the previous two chapters for supervised and unsupervised QDET.
- Chapter 8 concludes this thesis and discusses future works.

CHAPTER 2

An Introduction to the Concept of Question Difficulty

In this Chapter we introduce the concept of *difficulty*. We start by providing the definition of question difficulty and its meaning in an educational setting (§2.1). We then describe which are the question characteristics that affect question difficulty (§2.2), and introduce the theories of testing that are most commonly used to perform the numerical estimation of question difficulty (§2.3).

2.1 The Meaning of *Question Difficulty*

As a first step towards defining the concept of *question difficulty* in the educational domain, let us start by defining *question*. In the Cambridge Dictionary, it is defined "a sentence or phrase used to find out information" but, if we narrow our focus down to the educational domain, the definition is different: "in a test or exam, a problem that tests a person's knowledge or ability". Thus, we can have an intuition of the meaning of *question difficulty* in the educational setting, which is *a quantitative measure of the skill level that is required to solve the task at hand* (i.e. correctly answer

the question).

Quantitatively measuring question difficulty is extremely challenging: indeed, it is an unobservable characteristic of exam questions – a *latent trait* – and it is not possible to perform a direct measure in order to assign it a numerical value. In this sense, it is very different from other commonly used measures: as an example, if we want to measure distance, it is sufficient to have a reference object of known length and compare the target of our measurement with the reference object. In the case of question difficulty, it is not possible to do anything like this, nor to use a reference exam and say that another exam is "N times more difficult than the reference".

Research in psychometrics proposed several techniques to numerically estimate question difficulty[1], and they are all based on the same assumption: the answers provided by a group of students (and their correctness) are used to estimate with pretesting the difficulty of the question under consideration. This implies that the estimated difficulty depends on the skill levels of the students answering the question and different difficulties might be associated to the same question, depending on the students which were considered and the exam in which the question appeared. Fortunately, these issues do not affect the usability of these techniques and the validity of the difficulties estimated with them, but they highlight the challenges of the task of difficulty estimation, which are reflected in the task of QDET.

Another important remark is the fact that there might be "several difficulties" associated to a specific exam question, each one being associated to a specific topic, skill, or cognitive process [52, 60, 127]. In these cases, students can correctly answer the question if and only if they are skilled enough in all the required skills.

Lastly, it is important to clarify here that the concept of question difficulty exists in different domains, and its definition – as well as the question characteristics that affect it – heavily depends on the domain under consideration. As an example, the question difficulty defined in a community question answering website such as *StackOverflow*[2] is different from the concept of question difficulty defined in the educational setting. In this thesis, we only focus on the educational domain and therefore consider only the difficulty as defined in that setting.

[1] Some of these theories are presented in detail in §2.3

[2] `https://stackoverflow.com/`

Question:
A 55-year-old woman with small cell carcinoma of the lung is admitted to the hospital to undergo chemotherapy. Six days after treatment is started, she develops a temperature of 38C (100.4F). Physical examination shows no other abnormalities. Laboratory studies show a leukocyte count of 100/mm3 (5% segmented neutrophils and 95% lymphocytes). Which of the following is the most appropriate pharmacotherapy to increase this patient's leukocyte count?

Options:

A. Darbepoetin	B. Dexamethasone	C. Filgrastim
D. Interferon alfa	E. Interleukin-2 (IL-2)	F. Leucovorin

Figure 2.1: *Example of a test item from the Clinical Knowledge component of the United States Medical Licensing Examination.*

2.2 Sources of Question Difficulty

In this Section we provide an overview of the question characteristics that have an impact on item difficulty, discussing both the effects of exercise content (i.e. the knowledge assessed by the question and its verbalization) and exercise format (i.e. the structure of the question).

In the previous section, we have mentioned that the definition of question difficulty can be diverse depending on the domain under consideration and there is a significant difference between educational questions and non-educational questions. However, even narrowing our focus down to the educational setting, exam questions can be very diverse from each other, and this directly influences the components which have the biggest impact on item difficulty.

As a simple example, let us consider two questions. The first one, shown in Figure 2.1, is an exam question from the Clinical Knowledge component of the United States Medical Licensing Examination[3]. Arguably, the difficulty of this question mostly depends on the specific domain knowledge which is being assessed. However, it is a Multiple Choice Question (MCQ), and its difficulty also depends on the distractors (i.e. the wrong options) that are presented to the students. Indeed, if the distractors are blatantly wrong, the question is easy, if they are closer to the correct answer, the question is more difficult. Another peculiarity of MCQ is the fact that students can pick the correct answer by random guessing, which has an influence on difficulty. Lastly, the question difficulty is also affected by its verbalization, i.e. the words that are used and their position in the text.

The second example is a question from an exam of the International English Language Testing System (IELTS)[4], shown in Figure 2.2. In this case,

[3] https://www.usmle.org/step-2-ck/
[4] https://www.ielts.org/

Reading Passage:
[...] Knowledge of chronobiological patterns can have many pragmatic implications for our day-to-day lives. While contemporary living can sometimes appear to subjugate biology - after all, who needs circadian rhythms when we have caffeine pills, energy drinks, shift work and cities that never sleep? - keeping in synch with our body clock is important.
The average urban resident, for example, rouses at the eye-blearing time of 6.04 a.m., which researchers believe to be far too early. One study found that even rising at 7.00 a.m. has deleterious effects on health unless exercise is performed for 30 minutes afterward. The optimum moment has been whittled down to 7.22 a.m.; muscle aches, headaches and moodiness were reported to be lowest by participants in the study who awoke then. [...]

Question:
What did researchers identify as the ideal time to wake up in the morning?

Answer: _____

Figure 2.2: *Example of a reading comprehension question from an IELTS exam.*

we have a reading comprehension question (i.e. the student has to infer the correct answer from the reading passage), and the answer is open (i.e. no answer options are given). If we reflect on which are the aspects that affect question difficulty, we can immediately see that there are some differences with respect to the previous example. Indeed, domain knowledge is not important – there is no "domain knowledge" in this case, at all – and there are no distractors to affect item difficulty. On the contrary, the verbalization and the readability of the question and of the reading passage have the biggest impact on question difficulty.

The two examples above show the diversity of the factors that affect question difficulty and how the importance of each factor depends on the question format and the educational domain. When dealing with question difficulty, we can identify two macro-domains: i) Language Assessment (LA), both first language and foreign language, and ii) Content Knowledge Assessment (CKA), e.g. history, medicine, which is sometimes also referred to as domain knowledge assessment. The question in the first example belongs to the CKA domain, while the second belongs to the LA domain.

In LA the difficulty comes from linguistic demands of the task and the topic being assessed along with any stimulus text, while in CKA the difficulty mostly comes from the specific topic which is being assessed and the verbalization of the question only has a secondary role. Moreover, questions in CKA are often built in order to minimize the effects of language on the difficulty, especially in high-stakes exams [123], which is obviously not the case in LA.

Regardless of the educational domain under consideration, question dif-

ficulty is determined by two components [106]: i) exercise content, which is the verbalization of the question and the knowledge assessed by it, and ii) exercise format, which is related to the structure of the question.

2.2.1 Exercise Content

Exercise content refers to the specific knowledge which is being assessed by the question, and its verbalization.

With knowledge, we indicate the topic which is assessed, both in CKA (e.g. *solution of a differential equation* in a math exam) and LA (e.g. *present perfect* in an English exam).

Verbalization, on the other hand, refers to how the question is written: its readability, whether it is unambiguous, clear, and exhaustive, as well as the complexity of the language used in the text. The relevance of these two components is different between language assessment and content knowledge assessment.

2.2.2 Exercise Format

Exercise format refers to the structure of the question; i.e. how it is presented to students. Specifically, we can categorize the questions along the following dimensions: i) format of students answers, ii) type of question, and iii) input information.

Format of Student's Answer

Considering the format of students' answers, we can distinguish between *Multiple Choice Questions (MCQ)* (such as the example in Figure 2.1) and *open answer questions* (as the example in Figure 2.2).

In MCQ, students are given the question (referred to as *stem*) and a set of possible answer choices, among which there is the correct one; the other choices are referred to as *distractors* and have the goal of inducing the student to make mistakes. The number of correct choices might vary, as well as the number of distractors, and students are generally asked to select all the correct choices in case of multiple correct options. The difficulty of MCQ depends on all three components: the stem, the correct choice(s), and the distractor(s). More precisely, research showed that the similarity between the correct choice(s) and the distractor(s) have a significant impact on question difficulty [2, 54, 118]. Also, when dealing with MCQ, students can guess the correct answer by randomly picking one or more of the options, and this influences the difficulty, as well.

Open answer questions, on the other hand, contain only the text of the exercise; thus, students are asked to provide the correct answer without having the chance to select it from a set of given options. In this case, question difficulty depends only on the question and the correct answer, as there are no distractors and the chance of randomly guessing is virtually nonexistent.

Type of Question

As for the question types, there is a variety of options which are commonly used in the educational domain. Interestingly, the widest variety of question types is observed in language assessment, while content knowledge assessment is generally limited to *interrogative* and *cloze* items. Below, we list the types of questions which are most commonly considered in the literature.

Interrogative questions, as the two examples shown in Figure 2.3, are the ones that are most commonly used. It is important to mention that they do not always end with a question mark, as visible in the example.

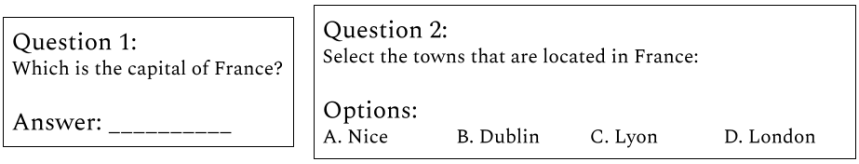

Figure 2.3: *Example of interrogative questions.*

Cloze items contain one or more gaps, each representing one or more words, and the student has to answer with the word(s) that correctly fill the gap. An example of cloze item is shown in Figure 2.4.

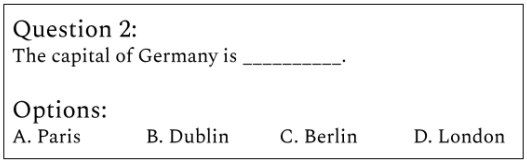

Figure 2.4: *Example of cloze item.*

C-tests are somewhat similar to cloze items, as they contain gaps which have to be filled by the student. However, the gaps do not represent whole words (or groups of words), as in cloze items, but are obtained by removing the second half of some words in the question text; the number of gaps in

each sentence/paragraph can be different and it is an exam design choice. An example of c-test is shown in Figure 2.5.

Question:
Vacc___ like penic___ and ot___ antibiotics th___ were disco___ as a dir___ result are lik___ the grea___ inventions o___ medical sci___.

Figure 2.5: *Example of c-test.*

Prefix deletion items are basically the same as c-tests, with the only difference that the first halves of the words are masked instead of the second halves; an example is shown in Figure 2.6. Cloze items, c-tests, and prefix deletion items are all form of *reduced redundancy testing* [107], which is based on the idea that natural language can be redundant thanks to contextual cues, and more advanced learners can be distinguished from beginners by their ability to deal with reduced redundancy.

Question:
___ines like ___illin and ___er antibiotics ___at were ___vered as a ___ect result are ___ely the ___test inventions ___f medical ___nce.

Figure 2.6: *Example of prefix deletion item.*

Cued Gap-Filling Item (CGFI) are used in language assessment, and have the goal of assessing the grammar knowledge of students rather than their vocabulary breadth. In CGFI, students read a short text and fill in the gap(s) using cues consisting of a single word which must be transformed to fit the context (generally verbs which have to be conjugated in the correct form), as in the example shown in Figure 2.7.

Question:
The Taj Mahal _____ (build) around 1640.

Figure 2.7: *Example of CGFI.*

In *Closest in meaning (CIM) items*, students are given a text passage and are asked to pick, from a set of possible choices, the word that is closest in meaning to a word highlighted in the text. An example of CIM item is shown in Figure 2.8.

Text: [...] The exact role of other factors is much more difficult to pinpoint; for instance, [...]

Question: The word "pinpoint" in the paragraph is closest in meaning to:
1. identify precisely 2. make an argument for 3. describe 4. understand

Figure 2.8: *Example of CIM item.*

Yes/No items have a very simple structure: students receive a list of words and have to select the ones that are real words. An example is shown in Figure 2.9.

Question:
Select the real English words:

Options:
A. Frequently B. Apply C. Positively D. Morride E. Shampoo F. Brican

Figure 2.9: *Example of Yes/No item.*

In *Vocabulary Knowledge Scale (VKS)*, students are asked to report how well they know a word and – if they report knowing it – they have to provide a synonym, a translation, or an example of the word in context [58, 110], as shown in the example in Figure 2.10.

Word: apply

Options:
1. I have not seen this word before
2. I have seen it, but I don't know what it means
3. I have seen it before and I think it means: _____ (synonym or translation)
4. I know this word. It means _____ (synonym or translation)
5. I can use this word in a sentence: _____

Figure 2.10: *Example of VKS item.*

In *Vocabulary Level Test (VLT)*, students are shown one or more definitions together with one or more target words, and they have to match the definitions with the target words [5, 84, 98]. The number of definitions and target words can vary, as shown in Figure 2.11.

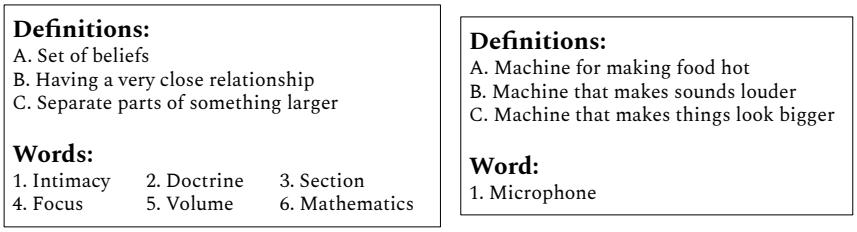

Figure 2.11: *Example of two VLT items.*

Format of Input Information

Another dimension which exam questions can be categorized along is the format of input information. Indeed, as we have seen in the first two examples in this chapter (Figure 2.1 and Figure 2.2), questions can either be i) comprehension questions or ii) knowledge questions.

Comprehension questions are given to the user together with an accompanying textual passage, and the task of the user is to find in the passage (or infer from it) the answer to the question. Two types of comprehension questions are generally used: i) *reading comprehension questions* or ii) *listening comprehension questions*, depending on whether the additional passage is written or spoken.

Knowledge questions, on the other hand, do not come with an accompanying passage, and for this reason they are sometimes referred to as standalone questions. They can be categorized in i) *text-only* questions and ii) *heterogeneous* questions, depending on whether they contain only textual information or also information of other nature. Specifically, heterogeneous questions generally contain images or tables which provide information needed to correctly answer the question.

It is important to remark here that in this thesis we use this definition of *comprehension question* and *knowledge question*, which is different from the one given in Bloom's taxonomy [13]. Indeed, Bloom's taxonomy delineates a hierarchy of cognitive-learning levels, ranging from the knowledge of specific facts to more advanced levels of synthesis, while here we exclusively categorize the questions depending on their format (i.e. whether they are provided with additional text which contains the information required to answer the question)[5].

[5]The original Bloom's taxonomy categorizes questions according to six cognitive levels, ranging from the knowledge of specific facts to more advanced levels. i) *Knowledge* "involves the recall of specifics and universals, the recall of methods and processes, or the recall of a pattern, structure, or setting". ii) *Comprehension* "refers to a type of understanding or apprehension such that the individual knows what is being communicated and can make use of the material or idea being communicated without necessarily relating it to other material or seeing its fullest implications". iii) *Application* "refers to the use of abstractions in particular and concrete situations". iv)

2.3 Theories of Testing

In the previous sections, we discussed the sources of question difficulty, but did not provide any indications of how the difficulty is numerically estimated in practice. The theories that model questions difficulty and provide a way to numerically estimate it are referred to as *theories of testing*. Several theories exists, and they all define the difficulty in different manners. Crucially, there is not one theory which is overall "better" than the others, and the process of choosing the testing theory to use is an exam design choice beyond the scope of this thesis. Arguably, the most commonly used theories are Classical Test Theory (CTT) and Item Response Theory (IRT), but it is also quite common for exam designers to use manually defined difficulties, which are not based upon any theory and are not estimated from the correctness of students' responses. Regardless of the testing theory used (if any), the difficulty can be either a continuous value or a discrete (categorical) value.

In the rest of this section, we introduce the theories and techniques that are most commonly used for modeling question difficulty: i) CTT (§2.3.1), ii) IRT (§2.3.2), and iii) manual definition (§2.3.3).

2.3.1 Classical Test Theory

Classical Test Theory (CTT) [44] is a well established testing theory that predicts outcomes of psychological testing, such as the difficulty of items or the ability of test-takers. The term "classical" refers to the contrast with modern psychometric theories such as IRT.

CTT assumes that each individual is associated with a true ability score T, which would be the expected correctness (i.e. fraction of correct answers) of an infinitely long run of repeated independent administrations of the same test. In practice, we can use the observed score X, which is the sum of the true score T and an error E:

$$X = T + E \tag{2.1}$$

where T and E are two unobservable (or latent) variables.

The major assumptions of CTT are the following:

- T and E are not correlated;

Analysis represents the "breakdown of a communication into its constituent elements or parts such that the relative hierarchy of ideas is made clear and/or the relations between ideas expressed are made explicit". v) *Synthesis* is the "putting together of elements and parts so as to form a whole". vi) *Evaluation* relates to "judgments about the value of material and methods for given purposes".

- E is normally distributed with zero mean;
- the errors of different tests are not correlated.

The concept of item difficulty in CTT is expressed by the *p-value*, which is a continuous value in the range [0; 1] (sometimes scaled to [0; 100]). The p refers to "probability" and is the fraction of correct responses in the considered population. The p-value is typically referred to as *correctness*: the higher the p-value, the easier the item is. Similarly, we can define the *wrongness* as 1 - p-value: the higher the value, the more difficult the item is.

The main limitation of CTT is the fact that it does not leverage the students' skills when estimating the item difficulty, meaning that it considers all the students as having the same skill level; its main advantage is being simple to compute and to interpret, with respect to other techniques such as IRT.

2.3.2 Item Response Theory

Item Response Theory (IRT) [45] is another well established technique that associates latent traits to both students and questions. Its simplest implementation, the one-parameter model (also referred to as the "Rasch Model" [94]), associates a skill level θ to each student and a difficulty level b to each question.

An important property of IRT is the "invariance property": the estimated latent traits do not depend on the ability distribution of test takers. In practice, this means that the difficulties estimated with IRT do not depend on the specific skill level of the pool of students used to estimate them, which is a strong advantage with respect to CTT.

The two major assumptions of IRT are that:

- the individuals are independent from one another;
- the item responses of a given individual are independent from one another.

For a given question j and its latent trait b_j, we can define the item response function (i.r.f.) which indicates the probability ($\mathrm{P_C}$) that a student i with skill level θ_i answers the question correctly. The formula of the i.r.f. is as follows:

$$\mathrm{P_C} = \frac{1}{1 + e^{-1.7 \cdot (\theta_i - b_j)}} \tag{2.2}$$

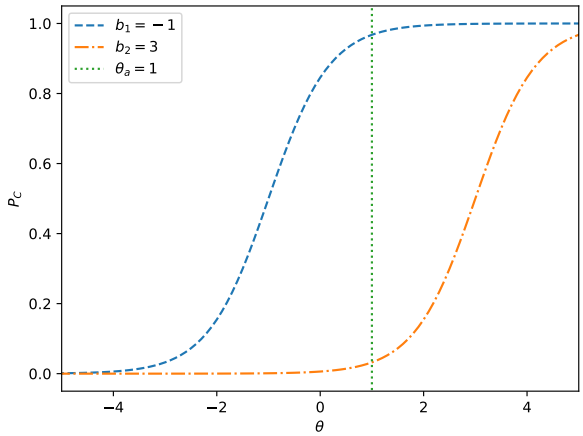

Figure 2.12: *Example of the item response functions of two questions with different difficulty.*

The background intuition is that a student with a given skill θ_i has a lower probability of correctly answering more difficult questions. If a question is too difficult or too easy (i.e. $b_j \to \infty$ or $b_j \to -\infty$), all the students will answer in the same way (i.e. $\mathrm{P_C} \to 0$ or $\mathrm{P_C} \to 1$) regardless of θ_i. This shows, from a mathematical perspective, why it is important to use only assessment items that are not too easy nor too difficult. As an example, Figure 2.12 plots the item response functions of two questions of different difficulty ($b_1 = -1$ and $b_2 = 3$). As expected, the shape of the two i.r.f. is the same, but the one related to the easier question is shifted towards the left, meaning that a student with a certain skill level (e.g. $\theta_a = 1$, as in the image) has a higher probability of correctly answering it with respect to the more difficult question.

More complex models also consider additional latent traits for questions, allowing for i.r.f. of different shapes. Specifically, we can associate to each question a discrimination a, which affects the steepness of the item response function, and a guess factor c, which represents the probability that a student correctly answers a question by guessing. The general formula of the i.r.f. is as follows (the one-parameter model is obtained by setting $c_i = 0$ and $a_i = 1$):

$$\mathrm{P_C} = c_i + \frac{1 - c_i}{1 + e^{-1.7 \cdot a_i \cdot (\theta_i - b_j)}} \tag{2.3}$$

Figure 2.13 displays the i.r.f. of three questions which have the same

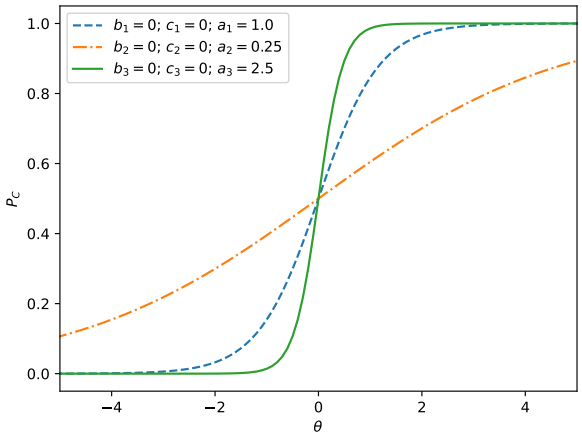

Figure 2.13: *Example of the item response functions of three questions with equal difficulty and different discrimination.*

difficulty and different discriminations: it shows that question with discrimination closer to 0 are less capable of discriminating between highly-skilled and lowly-skilled students, and therefore are less informative for estimating their knowledge level. Also, if we have a question with negative discrimination, it means that highly-skilled students are less likely to correctly answer than low-skilled students, which suggests that there might be something wrong with the question. For this reason, the discrimination is often used a quality indicator for exam questions.

Lastly, Figure 2.14 displays the i.r.f. of two questions which have the same difficulty and the same discrimination, but different guess factor; specifically, they have $c_1 = 0$ and $c_2 = 0.25$. We can observe that, considering the question with $c_2 = 0.25$, even students with a very low skill level have a chance of randomly picking the correct choice. For this reason, the guess factor is often used when modeling MCQ.

IRT models are trained using as input information the answers given by a set of students to a set of questions. Specifically, both the skill levels of the students and the difficulty of the questions are estimated via likelihood maximization, by selecting the configuration (i.e.the θs, bs, and possibly as and cs) that maximizes the probability of the observed results.

Also, with IRT, it is possible – given the responses of a student i to a set of calibrated questions $Q = q_1, q_2, ..., q_{N_q}$ – to assess the knowledge level $\tilde{\theta}_i$ of the student from the correctness of its answers. This is done

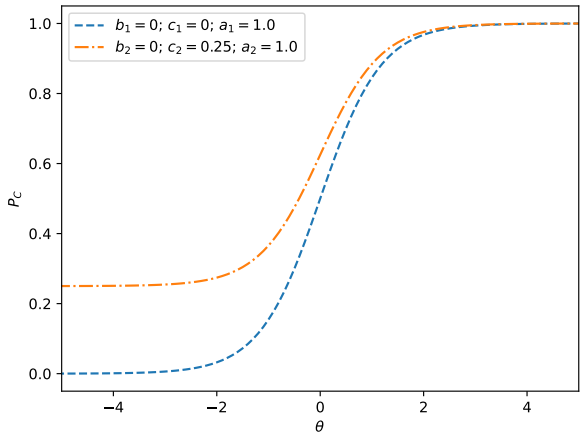

Figure 2.14: *Example of the item response functions of two questions with equal difficulty, equal discrimination, and different guess factor.*

by maximizing the results of the multiplication between the i.r.f. of the questions that were answered correctly and the complementary of the i.r.f. of the questions that were answered erroneously, as follows:

$$\tilde{\theta} = \max_{\theta} \left[\prod_{q_j \in Q_C} \frac{1}{1 + e^{-1.7 \cdot (\theta - b_j)}} \cdot \prod_{q_j \in Q_W} \left(1 - \frac{1}{1 + e^{-1.7 \cdot (\theta - b_j)}} \right) \right] \quad (2.4)$$

where Q_C and Q_W are the sets of question that were correctly and wrongly answered, respectively.

All the latent traits obtained in IRT are real values in a given range (selected at the time of model calibration). In practice, the continuous difficulty obtained with IRT is sometimes converted to discrete values, thus representing difficulty in a discrete manner.

2.3.3 Manual Definition

In many cases, discrete difficulties are obtained by converting the value obtained with either IRT or CTT into a discrete class. However, in some cases, question difficulty is not based upon any learning theories and it is just manually selected by educational experts. In all these cases – at least considering recent literature – the difficulty is modeled as a discrete variable, and the number of possible classes can vary, depending on the specific implementation.

An example of "manual definition" are CEFR[6] levels – six levels, from A1 to C2 – which are based on expert-defined rules rather than being based on the statistics of students' responses.

[6] https://www.coe.int/en/web/common-european-framework-reference-languages/table-1-cefr-3.3-common-reference-levels-global-scale

CHAPTER 3

Statistics and Machine Learning Background

In this Chapter we provide a brief introduction to the techniques and machine learning architectures that are referred to or used in this study. It is written for the reader's convenience and it is meant to provide for each technique a quick reference for the rest of this thesis, but we do not have the goal of being exhaustive. Thus, we also include references to relevant papers.

3.1 Term Frequency-Inverse Document Frequency

Term Frequency-Inverse Document Frequency (TF-IDF) [59] is a technique coming from information retrieval that can be used to represent the importance of a word (or a set of words) to a document in a corpus (i.e. a set of documents). It is based on the intuition that the importance grows with the number of occurrences of the word in the document but it is limited by its frequency in the whole corpus, meaning that words that are very frequent in all the documents are not important to any of them.

The formula used to compute the TF-IDF weight of word w in document

d belonging to the corpus $C = \{d_1, d_2, ..., d_{N_d}\}$ is the following:

$$\text{TFIDF}(w, d, C) = \text{count}(w, d) \cdot \left(\log_e \frac{N_d + 1}{\text{count}(w, C) + 1} + 1 \right)$$

where count(w, d) is the number of occurrence of w in document d, N_d is the number of documents in the corpus C and count(w, C) is the number of documents in the corpus C where w appears.

TF-IDF is used in a variety of domains and has many applications; however, in all the research covered in this study it is exclusively used as a technique to obtain features that are later used to train supervised models to perform QDET.

3.2 Recurrent Neural Networks

The Recurrent Neural Network (RNN) [95] is an artificial neural network architecture which uses sequential data, and it is most commonly used for ordinal or temporal problems (e.g. language translation). The key difference with respect to other neural networks is the fact that RNNs have a "memory", which enables them to consider information from prior steps in the sequence when dealing with current input. Therefore, the output to a given input does not depend only on the input itself (as it does for traditional neural networks), but also on the history that led to that point.

RNN are trained with backpropagation, but in a slightly different way with respect to traditional neural networks. Indeed, RNN leverage *backpropagation through time*, which is specific to sequential data. Similarly to traditional backpropagation, the model is trained by calculating the errors from the output layer to the input layer, but the difference is that – in backpropagation through time – the errors at each timestamp are summed.

This process might cause two issues, known as *vanishing gradients* and *exploding gradients*. The gradient is the slope of the loss function along the error curve; when it is too small, it keeps getting smaller, until it is too small to update the weights of the network. When that happens (vanishing gradients), the algorithms is no longer learning. On the other hand, exploding gradients occur when the gradient is too large, which creates an unstable model.

3.2.1 Bidirectional Recurrent Neural Networks

Bidirectional Recurrent Neural Networks (BRNN) [99] are a variant of standard RNN which uses future data to improve the accuracy of the current

prediction. While it is not possible in all scenarios (e.g. time series prediction), it proved very useful particularly on written text. Indeed, BRNN are capable of using all the words in a sentence to perform the task, e.g. predict a masked word, rather than using only the words that precede the one that was masked, which provides more context to the model and therefore ease the task.

3.2.2 Long Short-Term Memory

Long short-term memory (LSTM) [50] is a popular RNN architecture, that was proposed to address the issue of vanishing gradients occurring in standard RNNs. In practice, if the current prediction is affected by a previous state which is not in the recent past, standard RNN models may be unable to perform accurately, due to how they encode previous history. LSTMs address this issue by using memory cells in the hidden layers of the neural network, which contain three gates to control the flow of information which is needed to predict the output in the network.

3.2.3 Gated Recurrent Units

Gated recurrent units (GRU) [22] are a type of RNN similar to LSTM, built to address the short-term memory problem of standard RNNs. Rather than using a cell state to manage information, GRU use hidden states, and instead of the three gates used in LSTM, two gates, a reset gate and an update gate, that control how much information and which information should be retained by the model.

3.3 Word Embeddings

In Natural Language Processing (NLP), word embeddings are representations of words in the form of real-valued vectors. They encode the semantic meaning of words and are built in a way such that words which are closer in the vector space are expected to be closer in meaning. They can be obtained with diverse techniques, generally using some form of neural network. In this thesis, we use or mention three embedding techniques, which differ in size, architecture, and approach used for training: Word2Vec [81], GloVe [90] and ELMo [92].

3.3.1 Word2Vec

Word2Vec [81] is a word-based embedding model, meaning that it takes words as input and outputs word embeddings. It does not take into account

word order during training and, as a consequence, it is context independent, meaning that it outputs only one vector for each word, combining all the different meanings of that word into one vector. As a practical example, this means that the word *bank* in the two sentences *"I went to the bank for a mortgage."* and *"The river bank."* would be converted into the same word embedding. The training of the neural network is done incrementally, repeatedly iterating over a training corpus.

3.3.2 GloVe

GloVe (Global Vectors for Word Representation) [90] is similar in several ways to Word2Vec, in that it is a context independent word-based embedding model. A key difference from Word2Vec is the fact that at training time it works to fit vectors to model a word co-occurrence matrix built from the whole training corpus.

3.3.3 ELMo

ELMo (Embeddings from Language Models) [92] is quite different from both Word2Vec and GloVe: indeed, it is character-based and uses LSTMs, meaning that it takes into account word order. As a consequence, ELMo embeddings can capture the context of a word (i.e. its position in a sentence) and the same word used in different sentences will be translated into different embeddings.

3.4 Transformers

The *Transformer* [117] is a deep learning architecture originally introduced in 2017 and mostly used in NLP tasks. Transformers are built to handle sequential data, such as natural language, without requiring it to be processed in order. This allows parallelization, reducing training time and making training on large corpora easier. Also, Transformers can manage long-range dependencies as they are based on the *attention* mechanism, which is a technique that enables the model to "focus" on specific portion of the textual input, giving them more importance than the other words.

In this thesis, three Transformer-based models are referred to: BERT [26], DistilBERT [97], and XLNet [126].

3.4.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) [26] is a language model that, when introduced, reached state of the art perfor-

mance in many language tasks by applying the bidirectional training of the Transformer to language modeling.

BERT is originally trained on two tasks: i) Masked Language Modeling (MLM) and ii) Next Sentence Prediction (NSP). MLM consists in randomly masking words in the input text (one at a time) and asking the model to predict the masked words; NSP consists in giving two input sentences to the model and asking it to estimate whether the second sentence is a reasonable continuation to the first one.

One of the huge advantages of BERT is the possibility of using it for different down-stream tasks. Indeed, starting from the pre-trained model, it is possible to *fine-tune* it on desired tasks by stacking a layer on top of the original network. During fine-tuning, not only the weights of the added layer are updated, but also the internal weights of the pre-trained model in order to adapt them to the desired task, which is both more accurate than re-training the whole network (because the pre-existing knowledge is not discarded) and more efficient.

3.4.2 DistilBERT

One limitation of BERT is its being a large model, which requires many resources for training and fine-tuning. For this reason, some "lighter" version of BERT were built: in this thesis we consider DistilBERT [97], which is a language model obtained from BERT with knowledge distillation. Knowledge distillation is a compression technique in which a small model (referred to as student) is trained to reproduce the full output distribution of a larger model (referred to as teacher) [49]. With this approach, DistilBERT is able to retain – according to the authors of the original paper – 95% of BERT's performance on a language understanding task using about half the number of parameters. Also, similarly to BERT, DistilBERT can be fine-tuned on downstream tasks different from the ones it was originally trained on.

3.4.3 XLNet

XLNet [126] is a generalized autoregressive model for natural language understanding. When introduced, its main contribution was not the architecture but rather the modified training objective, which learns conditional distributions for all permutations of tokens in a sequence.

One of the training task used for BERT is masked language modeling: the model receives a sentence in which a token has been masked, and has to predict the token which was masked. In doing so, BERT uses as context

both left and right tokens, but considers only the original sentence. XLNet takes this a step further: it predicts each word in a sequence using any combination of the other words in that sequence. In practice, this means that XLNet is presented more difficult (and sometimes ambiguous) contexts to infer whether a word is or not in a sequence, and this enables it to extract more information out of the training corpus.

3.5 Calibration of Neural Networks

Large neural classification models are often capable of impressive results, but tend to be overconfident in their predictions [25]. This means that they are generally not *calibrated*: indeed, calibrating a model means aligning the posterior probabilities with the empirical likelihoods [43]. As an example, if we consider all the predictions for which a model has the confidence of 75%, then if the model is calibrated, the true accuracy is 75%.

Several techniques can be used in practice for calibrating neural models, and they have diverse advantages and weaknesses. The most commonly used ones are the following:

- *vanilla*: maximum softmax probability, which usually does not lead to calibrated classifiers [48];
- *Temperature scaling*: a posterior calibration technique using a validation set [25, 43];
- *Bayesian deep learning*, which requires alterations to the training procedure and is computationally expensive [121];
- *Ensembles*: consists in independently training M models on the entire dataset using different random initializations [66] or dropout [37,108].

Two approaches are generally used for evaluating model calibration. The most common is Expected Calibration Error (ECE) [82], which compares the confidence and the accuracy of the model. More precisely, it defines miscalibration as the difference in expectation between confidence and accuracy. Thus, ECE approximates the miscalibration by partitioning the predictions in a number M of bins and averaging the difference between the accuracy and confidence obtained in each bin. The other option are reliability diagrams, which provide a visual representation of model calibration [43]. Reliability diagrams plot the accuracy as a function of confidence; if the model is perfectly calibrated, the diagram should display the identity function. Any differences from the identity function are a signal of miscalibration.

3.6 Evaluation metrics

In this last section, we introduce the evaluation metrics that are used – or referred to – in this thesis.

3.6.1 Accuracy

Accuracy is a metric used to evaluate classification models, both binary classification and multi-label classification tasks. Considering a set of elements that have to be classified by the model, the accuracy indicates the fraction of elements that have been correctly classified over the total.

The best possible score is 1.0 and the worst possible score is 0.0.

3.6.2 Precision and Recall

Precision and Recall are two evaluation metrics used in information retrieval and binary classification. Precision is the fraction of relevant instances among the retrieved instances, while recall is the fraction of relevant instances that were retrieved. In other words, the precision indicates the ability of a model to identify only the relevant elements, while the recall indicates the ability of the model to detect all the relevant elements.

Let True Positives (TP) be the number of relevant elements that have been identified by the model, False Positives (FP) the number of not relevant elements that have been identified as relevant by the model, True Negatives (TN) the number of not relevant elements that have been classified as not relevant by the model, and False Negatives (FN) the number of relevant elements that have not been identified by the model. Using these measures, precision can be defined as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.1}$$

The recall, on the other hand, is defined as follows:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.2}$$

Even though precision and recall are both defined for binary classification, they can be used in multi-label classification tasks as well, by computing a precision score and recall score for each class.

For both metrics, the best possible score is 1.0 and the worst possible score is 0.0.

3.6.3 F1 Score

The F1 score, also known as balanced F-score or F-measure, represents a weighted average of precision and recall, according to the following formula:

$$\text{F1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3.3}$$

The best possible value is 1 and the worst possible score is 0.

Similarly to precision and recall, it is originally defined for binary classification, but it can be used in multi-class classification as well, by computing the F1-score of each class and averaging them (possibly according to a weighing average parameter).

3.6.4 Mean Squared Error

The Mean Squared Error (MSE) is an evaluation metric used in regression tasks indicating the average of the squared error between the predicted value and the true value.

Let us assume that y_i is the true value of the i-th sample and $\tilde{y}_i$ the corresponding predicted value. Then, the MSE estimated over N samples is defined as:

$$\text{MSE} = \frac{1}{N} \sum_{i=0}^{N-1} (y_i - \tilde{y}_i)^2 \tag{3.4}$$

The best possible score is 0.0 and there is no limit to the worst possible score, as the model can make arbitrarily large errors.

3.6.5 Root Mean Squared Error

Similarly to the MSE, the Root Mean Squared Error (RMSE) is an evaluation metric used in regression tasks, and it is defined as the squared root of the MSE.

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} (y_i - \tilde{y}_i)^2} \tag{3.5}$$

The best possible score is 0.0 and there is no limit to the worst possible score.

3.6.6 Mean Absolute Error

The Mean Absolute Error (MAE) is another metric used in regression tasks. It indicates the averaged error between the predicted value and the true value.

Assuming that y_i is the true value of the i-th sample and $\tilde{y}_i$ the corresponding predicted value, the MAE estimated over N samples is defined as:

$$\text{MAE} = \frac{1}{N} \sum_{i=0}^{N-1} |y_i - \tilde{y}_i| \tag{3.6}$$

Compared to the MSE and RMSE, it penalizes large errors less. The best possible score is 0.0 and there is no limit to the worst possible score, as the model can make arbitrarily large errors.

3.6.7 R2 Score

The R2 score (R^2), also referred to as coefficient of determination, represents the proportion of variance that has been explained by the independent variables in the model. It provides a measure of how well unseen samples are likely to be predicted by the model, and for this reason it is often used as the target metric while performing cross validation on regression tasks.

Since the variance considered by R^2 is dataset dependent, the values of this metric may not be meaningfully compared across different datasets. The best possible score is 1.0 and it can be negative and arbitrarily small for large errors; a constant model that always predicts the expected value regardless of the input features, would get an R^2 of 0.0.

If $\tilde{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value, the R^2 score estimated over N samples is defined as:

$$R^2 = 1 - \frac{\sum_{i=0}^{N-1}(y_i - \tilde{y}_i)^2}{\sum_{i=0}^{N-1}(y_i - Y_i)^2} \tag{3.7}$$

where $Y_i = \frac{1}{N} \sum_{i=0}^{N-1} y_i$.

3.6.8 nDCG

The normalized Discounted Cumulative Gain (nDCG) is a measure of ranking quality. It is computed by summing the true scores ranked according to the order obtained by the predicted scores, after applying a logarithmic discount, and dividing this value by the best possible score (referred to as iDCG, *ideal DCG*).

Let us define the DCG of a ranking of p elements as follows, where rel_i is the relevance of the i-th item:

$$\text{DCG} = \sum_{i=1}^{p} \frac{\text{rel}_i}{\log_2(i+1)} \tag{3.8}$$

The nDCG is then obtained as:

$$\text{nDCG} = \frac{\text{DCG}}{\text{IDCG}} \tag{3.9}$$

where IDCG is the Discounted Cumulative Gain obtained with an ideal ranking (i.e. sorting the elements according to their relevance).

The best possible score is 1.0, and the worst possible score is 0.0.

CHAPTER 4

Data collections

The data collections used for the experiments on supervised Question Difficulty Estimation from Text (QDET) (Chapter 5) and unsupervised QDET (Chapter 6) are partially overlapping. We therefore introduce them here, so that in the following chapters we can focus exclusively on the aspects that are specific to each task.

We experiment on three data collections, chosen to have a variety in the typology of questions and educational domains under consideration. Specifically, we experiment on i) *Cloud Academy*, ii) *ASSISTments*, and iii) *RACE*. *Cloud Academy* (§4.1) is a private data collection containing MCQs about cloud technologies, *ASSISTments* (§4.2) is a publicly available data collection containing questions of different type mostly about math, and *RACE* (§4.3) is a publicly available data collection containing reading comprehension MCQs. All data collections contain exclusively questions in English.

Table 4.1: *Example question from Cloud Academy.*

Role	Text
Question	*A user has launched an EBS backed EC2 instance in the US-East-1 region. The user wants to implement a disaster recovery (DR) plan for that instance by creating another instance in a European region. How can the user accomplish this?*
Correct choice	*Create an AMI of the instance and copy the AMI to the EU region. Then launch the instance from the EU AMI.*
Distractor	*Use the "Launch more like this" option to copy the instance from one region to another.*
Distractor	*Copy the instance from the US East region to the EU region.*
Distractor	*Copy the running instance using the "Instance Copy" command to the EU region.*

4.1 *Cloud Academy*

Cloud Academy Inc.[1] is an e-learning provider offering online courses about IT technologies. The *Cloud Academy* data collection used in this work is a subset of the company's data collection, generated in order to have only questions about cloud technologies (e.g. Amazon Web Services[2], Google Cloud Platform[3], and Microsoft Azure[4]). All the questions are MCQs and we have access to the text of the possible choices. An example question is shown in Table 4.1.

The data collection used in this work contains three datasets: i) *Cloud Academy_A* collects the log of interactions between students and questions, ii) *Cloud Academy_Q* contains the textual information related to the questions, and iii) *Cloud Academy_LEC* contains the transcript of some of the video lectures available on the Cloud Academy web platform about cloud technologies.

4.1.1 *Cloud Academy_A*

This dataset contains the log of interactions between students and questions; that is, it contains all the answers given by the students to the exam questions. We use this dataset only as training data for an IRT model that provides the question difficulty considered as gold standard when evaluating the models that perform QDET (i.e. the "true" difficulty). Basically, we

[1] https://cloudacademy.com/
[2] https://aws.amazon.com/
[3] https://cloud.google.com/
[4] https://azure.microsoft.com/

use this data for pretesting, in order to have a reliable estimation of question difficulty, which we use as reference at training and evaluation time.

The dataset contains 7,323,502 interactions, involving 24,696 users and 13, 603 questions. For each interaction, we have access to:

- *user id*, to uniquely identify the students;
- *item id*, to uniquely identify the questions;
- a binary *correct* label (i.e. *correct* or *wrong*);
- a *timestamp* of the interaction.

The overall correctness of the datasets – i.e. the fraction of correct answers – is 66.51%.

This dataset was built in order to have only the first answer in chronological order (i.e. the first attempt) for each student-question pair, and only items with at least 50 interactions are considered, in order to have a more accurate IRT estimation. The distribution of items per number of interactions is shown in Table 4.2.

N. of interactions (n)	Fraction of items
$50 <= n <= 100$	36.75%
$100 < n <= 200$	20.99%
$200 < n <= 500$	23.60%
$n > 500$	18.66%

Table 4.2: *Distribution of questions per number of interactions in Cloud Academy_A.*

On average, each question is answered by 304 different students (standard deviation of 365), and each student answers 114 different questions (standard deviation of 161).

4.1.2 *Cloud Academy_Q*

The *Cloud Academy_Q* dataset contains all the textual information about the questions. Specifically, it provides:

- unique *item id*, which can be used to merge this dataset with the difficulty obtained from *Cloud Academy_A*;
- text of the question;
- text of all the answer choices (in *Cloud Academy*, all the questions are MCQs).

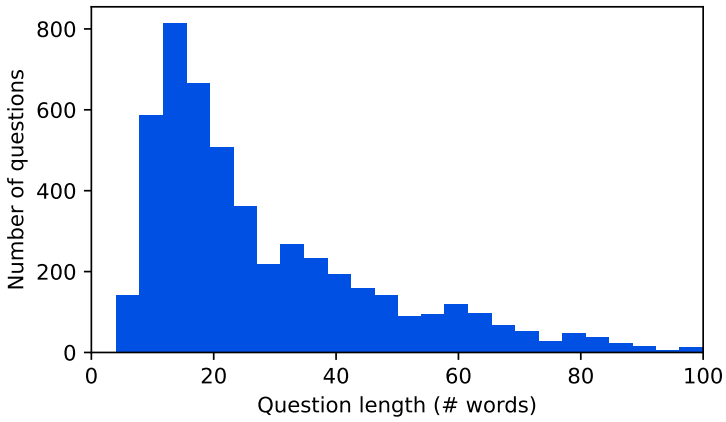

Figure 4.1: *Distribution of questions by length in Cloud Academy_Q.*

Figure 4.1 plots the distribution of questions per length. All the questions have at least four words, and the peak corresponds to lengths between 10 and 15 words. More specifically, about 10% of the questions are shorter than 10 words (included), and 75% of the questions are in the range between 10 words and 50 words. Less than 1% of the questions are made of more than 100 words.

On average, the text of the answer choices are shorter, and the average value is 6.8 words.

4.1.3 *Cloud Academy_LEC*

We leverage our access to the *Cloud Academy* data collection to retrieve a subset of the lectures about cloud technologies that are available on the platform. These are collected in the *Cloud Academy_LEC* dataset, which contains the transcript of some selected lectures about the same topics which are assessed by the questions in *Cloud Academy_Q*; an analogous dataset is not available for *ASSISTments* nor for *RACE*.

Cloud Academy_LEC is composed of 2,826,126 words, and we divide it into sentences, based on punctuation (full stop, question mark, exclamation mark) for a total of 141,306 sequences, with an average of 20 words per sequence. Table 4.3 shows the distribution of sentences per number of words.

Length (N. of words)	Fraction of sentences
len <= 10	23.24%
10 < len <= 50	72.79%
50 < len <= 100	3.70%
len > 100	0.27%

Table 4.3: *Distribution of sentences per number of words in Cloud Academy_LEC.*

Table 4.4: *Example questions from ASSISTments.*

ID	Question	Type
330	*The computer game Peter wants to buy will cost at least $50 and not more than $70. He earns $3 an hour running errands for his grandmother. Which inequality shows the number of hours, n, he will have to work to pay for the game?*	Original
326	*What is the minimum cost of the game?*	Scaffolding
327	*What is the maximum cost of the game?*	Scaffolding
328	*Write an expression that represents the amount of money Peter earns in n hours.*	Scaffolding
329	*Which inequality shows the number of hours, n, Peter will have to work to pay for the game?*	Scaffolding

4.2 *ASSISTments*

ASSISTments[5] is an online intelligent tutoring system developed by the Worcester Polytechnic Institute [35]. It provides teachers with contents from open educational resources and also gives the possibility to add new questions to the platform. When students complete assignments on the platform, it provides them immediate feedback, and sends reports on students' progress to the teachers.

As described by the authors, ASSISTments "provides instructional assistance while assessing students". In practice, this means that questions – called *problems* – can be broken down into steps: if the student does not get the *original* problem correctly, he has to answer a sequence of *scaffolding* questions that break the problem down into steps. In the current work, we consider both original and scaffolding problems for QDE from text. An example problem and the corresponding scaffolding questions are shown in Table 4.4.

Similarly to the *Cloud Academy* data collection, the *ASSISTments* data

[5] `https://new.assistments.org/`

collection contains i) a dataset with the logs of interactions between students and questions (*ASSISTments_A*), and ii) a dataset containing the textual information related to the questions (*ASSISTments_Q*). Differently from *Cloud Academy*, there is no *lectures* dataset; that is because the ASSISTments platform only provides assessment items and not online courses.

4.2.1 *ASSISTments_A*

ASSISTments_A, which is publicly available for download[6], contains the log of students' answers. We use this dataset to train the IRT model and obtain the question difficulties which are used as target values when training and evaluating the model for QDET.

The dataset contains 6,123,270 students' answers, and for each interaction we have access to several fields of information. The most interesting (for our study) are described below.

- *Problem_id* is the unique id of the question.
- *User_id*: the unique id of the student.
- The *correct* field indicates the correctness of student's answer: 1 if *correct* at the first attempt, otherwise 0. Some interactions have decimal values, which are calculated depending on the number of hints and attempts needed to correctly answer the question, but we convert them to 0, as standard IRT cannot deal with partial scores. The overall fraction of correct answer in the dataset is 67.64%.
- *Problem_type* indicates the type of problem, and there are six possible values: i) *algebra* are math expressions, ii) *choose_1* are MCQs with one correct choice, iii) *fill_in* are cloze items, iv) *open_response* questions, v) *choose_n* are MCQs with multiple correct choices, and vi) *rank* are questions requiring to rank multiple objects. Even though six types of questions exist, the vast majority of interactions (more than 99%) is from *algebra* items (57%), *choose_1* items (30%), and *fill_in* (12%).
- *Start_time* and *end_time* are timestamps indicating when the problem is shown to the student and when the student submits the answer, respectively.

[6] https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-with-affect

- The *original* field indicates whether the question is an original problem or a scaffolding problem. If a question has no scaffolding items, it is marked as a main problem. Most of the interactions in the dataset are with main problems (more than 95%), and only a small portion of the dataset contains scaffolding items.
- The *skill* attribute is sometimes used to indicate the skill associated with the question; however, it is *null* in 72% of the interactions, thus it is not really usable to model different skills.
- *Template_id* is a unique id identifying the template the question is created from, and questions generated from the same template are very similar with each other. Each question is associated to one *template_id* only, while several questions can be associated to the same *template_id*. In practice, questions with the same *template_id* are basically the same item with minor differences due to some keywords and numerical values.

We use this dataset to obtain, with IRT, the question difficulties which will be later used as gold reference evaluating the models for QDET, therefore it is important to focus on the choice of the identifier to use to distinguish the problems. Indeed, both *template_id* and *problem_id* are reasonable candidates. Considering that problems generated from the same *template_id* have very similar text and are created – by design – in order to have the same level of difficulty, we use *template_id* to distinguish the items while training the IRT model.

The raw data publicly available for download requires some preprocessing before being usable for our needs. Specifically, we consider only *first-timers*, meaning that for each student-question interaction we consider only the first answer, and we keep only the items that have been answered by at least 50 different students. This reduced the size of the dataset, and the final dimensions are shown in Table 4.5

	Raw dataset	After pre-processing
N. interactions	6,123,270	2,820,051
N. users	46,674	43,868
N. problems	179,999	55,178
N. templates	76,403	18,659

Table 4.5: *Measures of ASSISTments_A.*

The distribution of questions per number of interactions is shown in Table 4.6. On average, each question is answered by 151 students (standard

deviation of 303), and each student answers 64 different questions (standard deviation of 113).

N. of interactions (n)	Fraction of items
$50 <= n <= 100$	38.03%
$100 < n <= 200$	19.54%
$200 < n <= 500$	8.15%
$n > 500$	4.29%

Table 4.6: *Distribution of questions per number of interactions in ASSISTments_A.*

4.2.2 *ASSISTments_Q*

This dataset, which is publicly available upon request[7], contains all the textual information about the questions. Specifically, it provides:

- the *problem_id*, which can be used to merge this dataset with the difficulty obtained from *ASSISTments_A*;
- the *text* of the question.

Differently from *Cloud Academy_Q*, the text of the possible answer choices is not available. Even though this is a very large dataset, as it contains the text of almost 180,000 different *problem_id*, most of them are not usable. First of all, many texts are duplicate, and there are only 138,084 different texts. Secondly, several problems refer to content which is not available in the text (e.g. images, graphics, etc.), and therefore the information for the task of QDET would be very limited: 9.2% of the problems in the dataset have this issue. Also, some problems refer to external textbooks and are therefore unusable: indeed, the ASSISTments platform offers to instructors the possibility of manually creating custom questions, and often these questions simply refer to exercises in external textbooks, without reporting the text. Lastly, some items are incorrectly considered "questions" (as they have a *problem_id*), even though they are system messages or motivational messages. From the raw dataset, we remove all the problems which have one of the issues mentioned above.

As mentioned in the previous section, we use the *template_id* to uniquely identify the questions, and therefore we might have several similar texts associated to the same template. Previous research on this same dataset suggested that keeping all the texts would hinder the training of the model for QDET [88], therefore we keep only one text (randomly chosen) for each

[7] https://sites.google.com/site/assistmentsdata/home/assistments-problems

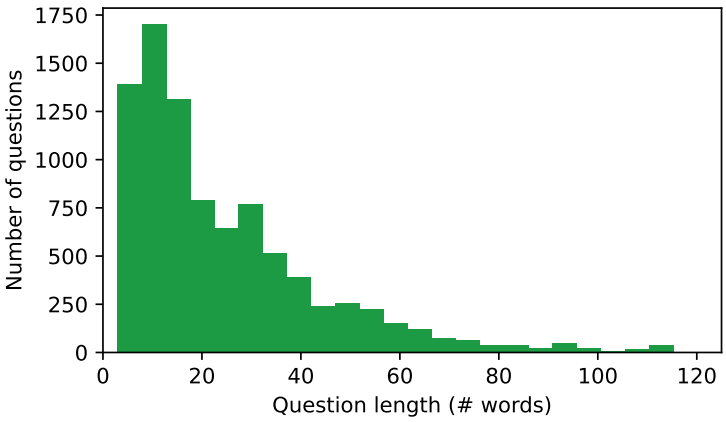

Figure 4.2: *Distribution of questions by length in ASSISTments_Q.*

template_id. In the end, after all the filtering operations, we are left with 11,393 different items.

The distribution of questions per length, which is shown in Figure 4.2, is fairly similar to *Cloud Academy*; indeed, the majority of questions is short and there is a peak for questions that are about 15 words long. Still, a significant difference is the fact that *ASSISTments* contains more questions which are very short: 27% of the questions are made of 10 or less words.

4.3 *RACE*

RACE[8] [65] is a publicly available dataset of English reading comprehension questions from middle and high school exams; it contains about 25,000 passages and up to four MCQs associated with each text, having 97,744 questions in total. Differently from *Cloud Academy* and *ASSISTments*, for *RACE* there is only one dataset which contains both the text of the questions and a manually selected difficulty.

Specifically, the dataset has the following attributes:

- *id*: unique id identifying the reading passage;
- *question_id*: unique id identifying the question;
- *article*: the text of the reading passage;
- *question*: the text of the question;

[8] `https://www.cs.cmu.edu/~glai1/data/race/`

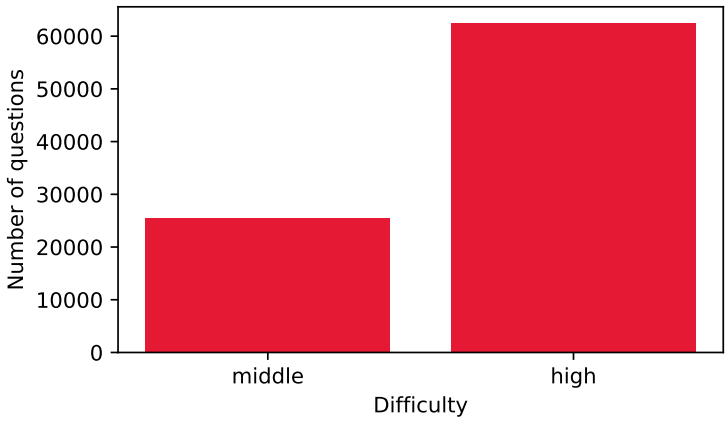

Figure 4.3: *Distribution of questions by difficulty level in RACE.*

- *options*: the text of the possible options;
- *answer*: indicates which option is the correct choice;
- *level*: the difficulty.

The *level* attribute, specifically, is a manually selected value which can be either *high* or *middle* and indicates the level of examination (high or middle school). Even though it is arguably less precise than the difficulty we can obtain for *Cloud Academy* and *ASSISTments* training an IRT model, it is an indication of the question difficulty and the authors point to the "drastic difficulty gap" between the two levels and give evidence for "higher difficulty of high school examinations" [65]. Figure 4.3 shows the distribution of question by difficulty level: the number of *high* questions in the raw dataset is considerably larger than the number of *middle* questions.

All the questions are MCQs with four possible choices, and they can be separated into interrogative or cloze items. An example question from *RACE* is shown in Figure 4.4.

Figure 4.5 presents the distribution of questions in the *RACE* dataset by question length, using number of words as indicator of length. The difference from *Cloud Academy* and *ASSISTments* is immediately visible: indeed, in this case, the questions are generally much longer, which is due to the fact that *RACE* contains reading comprehension questions, and the text of the reading passage contributes to question length.

Article:
… Bungee jumping is an activity about jumping from a tall structure while connected to a large elastic cord . The tall structure is usually a fixed object, such as a building, bridge or crane; but it is also possible to jump from a movable object, such as a hot-air balloon or helicopter …

Question:
Which of the following is NOT suitable for bungee jumping?

Options:
A. The fixed-wing aircraft
B. The helicopter
C. The hot-air balloon
D. The mobile crane

Answer: A

Figure 4.4: *Example of a question from RACE.*

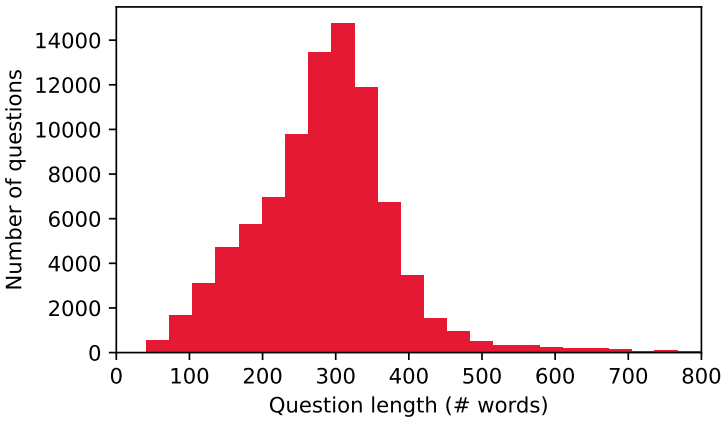

Figure 4.5: *Distribution of questions by length in RACE.*

CHAPTER 5

Supervised Question Difficulty Estimation from Text

In this Chapter, we focus on supervised Question Difficulty Estimation from text (QDET). First, we give an introduction to the task (§5.1), and provide a categorization of previous literature according to a taxonomy based on question characteristics (§5.2). We then present the recently proposed approaches, focusing separately on the language assessment domain (§5.3) and the content knowledge assessment domain (§5.4), and describe the models that are quantitatively evaluated in this study (§5.5). We describe the experimental setup (§5.6) and, by presenting the experimental results (§5.7), we evaluate how the proposed approaches (based on different features and different architectures) perform on questions of different nature. Lastly, we conclude the chapter with a recap and discussion of the experimental results (§ 5.8).

5.1 Introduction

Supervised QDET was originally proposed as a way to target the limitations of manual calibration and pretesting, which are the traditional ap-

proaches to question calibration. It consists in leveraging the textual content of questions with Natural Language Processing (NLP) techniques to automatically estimate their difficulty. Textual content is the only information that is available at the time of question creation – both for manual question creation and for automated question creation – and the idea of supervised QDET is to use such information to estimate question difficulty and thus overcome – or at least reduce – the need for pretesting and manual calibration.

In recent years, researchers proposed many approaches, which are based on diverse machine learning architectures and features. However, such approaches have been tested on a variety of question types and in the community there is a lack of a common framework for evaluating models proposed for QDET. Therefore, considering that the factors affecting question difficulty are diverse for different types of questions – as we discussed in Chapter 2 – it is not easy to understand which approaches and which features are the most effective in each scenario. Moreover, an exhaustive comparison of the proposed approaches to QDE from text is very difficult also due to the scarce number of publicly available educational datasets providing both question text and question difficulty.

In this Chapter, we evaluate several models proposed in previous research on three dataset coming from different educational domains, and analyze how the QDET performance depends on the domain and some question characteristics (i.e. question type, question format, and number of correct choices in MCQs).

We observe that, generally, the best accuracy is obtained with Transformer-based models, which are the most effective in capturing both the semantic meaning and the linguistic complexity of the questions. Nonetheless, we find that in the case of reading comprehension questions, simpler models based on linguistic features and readability indexes can perform almost as well as Transformer models at a fraction of the computational cost. We also observe that the performance of all the models depends on the questions characteristics and, specifically, Transformer models perform better on MCQs with one correct choice than MCQs with multiple correct choices, and for questions with longer texts.

5.2 Taxonomy of Literature on Supervised QDET

The research on techniques to perform QDET and to modify questions difficulty in a controllable manner has a fairly long history. However, in recent years there has been a very rapid development, which was mostly due

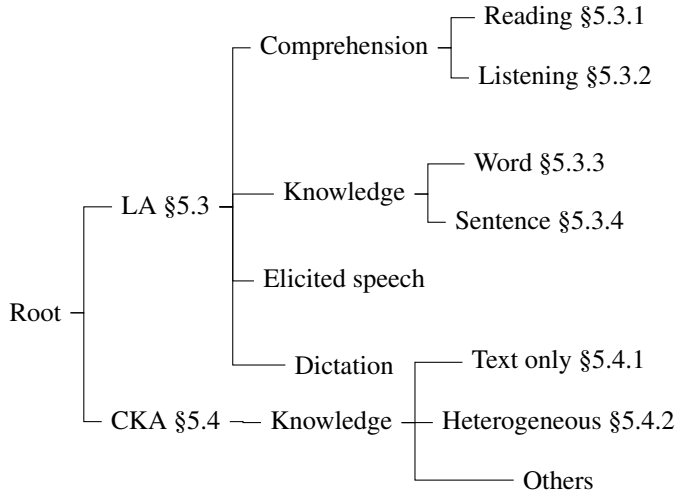

Figure 5.1: *The taxonomy, based on question format, we use for categorizing all the papers presented in this Chapter.*

to the improvements in the capabilities of NLP techniques (such as word embeddings and Transoformers), that have been reflected in the progress on QDET. Overall, there has been a shift form the usage of theoretically supported features, such as readability and word-complexity measures, towards approaches which rely upon modern NLP techniques based on machine learning.

Figure 5.1 presents the taxonomy we use for categorizing all the papers presented in this Chapter. We group the papers depending on the characteristics of the questions that the proposed models work on, since the type of question heavily affects the models that can be used in each application scenario. We provide here a brief overview of the proposed approaches and their categorization, and describe them in more detail in Section 5.3 and Section 5.4.

The first distinction is the educational domain considered by each work: specifically, we distinguish between i) Language Assessment (LA) and ii) Content Knowledge Assessment (CKA). In LA, the difficulty comes from linguistic demands of the task and topic being assessed along with any stimulus text, while in CKA the difficulty mostly comes from the topics which are being assessed.

This has an implication on the models that are developed in the two domains. Approaches developed for LA often rely upon predefined word complexity measures, which are not used in CKA. On the other hand, CKA

works leverage more frequently learnt features, such as TF-IDF and word embeddings, or end-to-end neural networks.

All the approaches presented in this Chapter address the task of QDET as a supervised problem (either classification or regression, depending on the testing theory of choice): a training set containing texts and difficulties of exam questions is used to train a model which is capable of performing QDET for previously unseen questions. In some cases, some additional textual datasets are used, generally to pre-train the model or part thereof. In such cases, the models built for LA leverage general purpose datasets (e.g. Wikipedia), while the ones built for CKA leverage datasets related to the topics that are assessed by the questions (e.g. books, lecture transcripts).

5.2.1 Language Assessment (LA)

Focusing on the approaches proposed for LA, they deal with either i) comprehension questions or ii) knowledge questions.

Comprehension questions

Models built for comprehension questions are built to leverage the additional information that is provided in the accompanying passage. Specifically, comprehension questions can be further divided into reading comprehension and listening comprehension, and this difference affects the features that are used by the models performing QDE from text. Only one recent work focused on listening comprehension questions [77]. Reading comprehension questions received more consideration in previous research, and there are three relevant works which focused on it [56, 57, 72], two of which leverage end-to-end neural networks for the task.

Knowledge questions

Considering this type of questions, many of the proposed models use fairly simple and theoretically-grounded features such as word-complexity for learners of specific languages and readability measures. No end-to-end neural networks were proposed so far and most of the works did not experiment with word embeddings or word frequency features. Knowledge questions for LA can be further divided depending on their format: some are vocabulary questions made of single words [24, 28, 101, 125], others represent whole sentences [6, 7, 34, 53, 56, 70, 86, 101, 111–113, 115].

Others

There are two types of questions which are explored in one paper only [101] and do not really fall in any of the previous categories: i) elicited speech and ii) dictation exercises. The elicited speech task evaluates reading and speaking skills of students by requiring them to produce a sentence out loud. The dictation task consists of asking the students to transcribe an audio recording, and thus evaluates both listening and writing skills.

Considering the additional information that is available (i.e. text to read and audio to listen to) these types of questions might seem to belong to the category of comprehension question. However, they do not require the students to infer the answer to a specific question from the text/audio, but only to perform a transformation from written to spoken form or vice-versa.

5.2.2 Content Knowledge Assessment (CKA)

In CKA, there is no separation between comprehension questions and knowledge questions, all items are knowledge questions. They can be categorized depending on the content of the questions and, specifically, can be divided into i) text only questions, and ii) heterogeneous questions, which contain information – such as images – that cannot be captured at text level[1].

Questions with images are quite rare and this is reflected by the fact that only three works [33, 109, 129] experimented on QDE for heterogeneous questions. Most of the research focused on text only questions, and it can be categorized depending on the type of information that is leveraged by the models. Specifically, we can distinguish between i) models that only consider the question text for the task of QDE [30], ii) models that also leverage texts coming from other sources (e.g. lecture content, books, etc.) [54,93,123,124,130], and iii) models that leverage non-textual information (e.g. knowledge components [21, 114], and others [109, 122]).

Lastly, there are two works which do not belong to any of the previous categories because they deal with specific types of questions and can be used only in the niches they were designed for. One of them [91] deals with questions whose answers are in the form of First Order Logic formulas and leverages such formulas for QDE. The other [85] performs QDE for short-answer questions and leverages the text of the students' answers instead of the question.

[1] Equations and formulas are generally considered as "text", since they can be expressed in LaTeX-like verbal format [128]: for instance, $\sqrt{1/4}$ corresponds to *\sqrt \frac 1 4*.

5.3 Literature on QDET in Language Assessment

In this section, we describe the approaches proposed for language assessment, grouping them according to the taxonomy presented above: reading comprehension questions (§5.3.1), listening comprehension questions (§5.3.2), word knowledge questions (§5.3.3), and sentence knowledge questions (§5.3.4).

5.3.1 Reading Comprehension Questions

In reading comprehension questions, the accompanying passage is an important component of question difficulty and was leveraged by all the papers addressing QDET on this type of questions. One of them [56] directly models question difficulty as the reading complexity of the reading passage, while the others [57, 72] leverage neural networks that receive as input the text of both the question and the accompanying passage. An overview of these models is shown in Table 5.1.

Paper	Year	Reading passage	Question text	Distractors	Approach
[56]	2018	✓	-	-	Reading difficulty estimation
[72]	2019	✓	✓	-	Word2Vec - LSTM - FCNN
[57]	2017	✓	✓	✓	Word2Vec - Sentence CNN - Attention - FCNN

Table 5.1: *Overview of the approaches proposed for reading comprehension questions.*

In [56] the authors assume that examinees correctly answer a reading comprehension question only if they can understand the whole textual passage, therefore they directly use reading complexity as an indicator of question difficulty. For the estimation of reading complexity, the authors adopt a measure designed for learners of English as a foreign language [55]. This can be considered a fairly simple approach, since it estimates the same difficulty for all questions associated with a certain passage. Still, the authors observe that there is a relation between question difficulty estimated with this approach and average correctness. However, the authors consider a test set containing questions of different types (reading comprehension, grammar, and vocabulary questions)[2], therefore a clear evaluation of QDE for reading comprehension questions is lacking.

In [72], the authors propose a neural model to estimate the difficulty

[2]The approaches proposed in this paper for grammar and vocabulary questions will be presented in Section 5.3.4

of Chinese reading comprehension items. The proposed model is made of three components. First of all, i) each word is transformed into a semantic vector with Word2Vec (trained on the Sinica Balanced Corpus [19]), and there is no distinction between the words of the document and the words of the question. Then, ii) the embedding vectors are input into two unidirectional LSTMs. Lastly, iii) the output of the LSTMs is input into is a Fully Connected Neural Network (FCNN) made of three layers that outputs the estimated difficulty.

The model proposed in [57] is the only one that explicitly takes into consideration the relation between the reading passage and the question. It does so by using an attention mechanism [117] to model the importance of each sentence in the reading document for a specific question. The intuition of the authors is that different questions concern different parts of the text, and by implementing a model that is capable of modeling this relation, it is possible to improve the accuracy. The proposed model is made of four components: i) input component, ii) sentence CNN (Convolutional Neural Network) component, iii) attention component, and iv) prediction component[3]. All the questions are MCQs, and the model leverages both the text of the question (i.e. the stem) and the text of the options.

First, in the input component, all the text material of a question (i.e. document, stem, and options) is converted into pretrained embeddings using Word2Vec trained on the English Gigaword dataset [40]. Then, the sentence CNN component reduces the dimensionality of the input data by applying a series of convolution and max-pooling operations. The attention component aims at finding which parts of the text are relevant for each question. In practice, there are two attentions involved in the model, both computed using cosine similarity: the first one measures the similarity between the text stimulus and the question, the second one measures the similarity between the question and the available answers. Lastly, the prediction component concatenates the two outputs of the attention components and uses a FCNN to learn the difficulty, which is modeled as a continuous value.

5.3.2 Listening Comprehension Questions

One paper [77] about QDE from text for listening comprehension questions was published in recent years. The authors focus on MCQs in English, considering four types of questions: i) "picture description": a picture and four recorded statements are presented to the student, who is asked to select the one that bests describes the image; ii) "dialogue completion": examinees

[3]The authors refer to these as "layers" instead of "components"; we change notation to clarify that these components can themselves be composed of several hidden layers.

hear a dialogue and have to complete it by selecting the best continuation from a list of candidates; iii) "conversation": examinees listen to a conversation and answer some questions about its content; iv) "monologue": examinees listen to a recorded monologue and answer some questions about its content.

Item difficulty depends on both the audio transcript and the text of the question, and indeed the proposed approach leverages both sources of information. First, the authors compute 339 raw features from the text (written and spoken) using *TextEvaluator*, an automated text complexity prediction system [83, 105]. These can be categorized into the following groups: academic vocabulary, concreteness, word familiarity, syntactic complexity, cohesion, argumentation, conversational style, and narrative structure. Then the authors experiment with several regressors for estimating item difficulty from the raw features, and show that a Random Forest regressor consistently outperforms all the other models. All the groups of features seem to bring valuable information for QDET, and the most relevant are the ones dealing with the lexical content of the item, for all item types. Specifically these features are related to vocabulary diversity, vocabulary difficulty, and the concreteness of the text [104].

An interesting finding is that the best performance was observed for "picture description" items, although the image was not considered as a feature. Considering the "monologue" items, the authors observed that the complexity of the question was more predictive than the complexity of the listening passage, which is in contrast with research about reading comprehension questions.

5.3.3 Single Word Knowledge Questions

The single word knowledge questions are all vocabulary questions. Since no information is available in addition to the target word and (possibly) some definitions, the proposed models are generally simple from an architectural point of view. It is also interesting to mention that no one of the proposed approaches leveraged the definitions, when available, for QDET. An overview of the models is shown in Table 5.2.

Neural networks are rarely used and, when they are, there is generally little focus on semantics. Indeed, [28] is the only work that leverages Word2Vec embeddings without any other features. Specifically, the author focuses on VLT, with one word and four definitions, and leverages a two-step machine learning approach for QDET. First, it computes Word2Vec embeddings of the questions, then uses a Support Vector Machine (SVM)

Paper	Year	Approach	Type of question
[24]	2015	Ortographic features and word frequencies	Yes/No, VKS, VLT
[28]	2018	Features: word2vec embeddings, model: SVM	VLT
[125]	2018	Features: word length, word frequency, utilization on the web, Age-of-acquisition, concreteness rating, number of POS tags, most frequent POS tag, word2vec embeddings, number of double consonants, number of vowels, presence of shorter homophones. Model: SVM	Yes/No, VKS, VLT
[101]	2020	Features: word length, log-likelihood from character-level language model, Fischer score. Model: weighted softmax	Yes/No

Table 5.2: *Overview of the approaches proposed for single word knowledge questions.*

regression model with linear kernel for the actual difficulty estimation. The author evaluates the model on real world questions but the dataset used for the experiments is very small (92 words, 22 being held-out for testing), which limits the significance of the findings.

The first work that evaluates the correlation between the difficulty of vocabulary questions (Yes/No, VLT, and VKS) and some textual features is [24], which found that character length and corpus frequency significantly correlate with vocabulary difficulty. However, this work did not have the task of performing difficulty prediction, and it is therefore mostly used as a starting point by more recent research.

An example is [125], in which the authors propose an approach that can be used for VKS, VLT with one word, and Yes/No items (although only for real words). The approach consists of i) computation of features related to the word difficulty level, ii) reduction of these features with Principal Components Analysis (PCA), and iii) classification with a SVM. The model uses the following features: word length, word frequency (obtained from NLTK corpora), utilization on the web (i.e. number of relevant documents retrieved by Google), Age-of-acquisition from [64], concreteness rating from [15], number of part-of-speech (POS) tags (obtained from NLTK [76] corpora), most frequent POS tag, Word2Vec embeddings, number of double consonants in the word, number of vowels, and existence of shorter homophones. The second step of the proposed approach consists of reducing the dimensionality of the data using PCA [89]: specifically, the authors reduce the dimensionality of the input data from 111 features to only 2 fea-

tures; no experiments were performed with other dimensions. Lastly, the classification is performed with an SVM model with RBF kernel.

The most recent paper in this section [101] focuses exclusively on Yes/No items. The proposed model uses three groups of features: i) character length of the target word, ii) corpus frequency, and iii) "Fischer score". While character length is straightforward to calculate, corpus frequencies may only be obtained for real words, whereas the pseudo-words found in Yes/No items inherently do not occur in corpora and therefore have no frequency value. Therefore, the authors propose a character level Markov chain language model to compute the log-likelihood of a word (or pseudo-word), and use this for the feature values instead of the corpus frequency. The character-level language model is trained on the OpenSubtitles corpus [74]. The last feature, the Fischer score of a word, is a vector representing the gradient of its log-likelihood under the language model. Conceptually, it is similar to trigrams weighted by TF-IDF [31]. The authors experiment both with a linear regression model and a weighted-softmax, and observe that linear regression appears to overfit the training data. They also find that the Fischer score features are the most useful for QDE, while character length has little impact (possibly because length is implicitly captured by the Fischer score features).

5.3.4 Sentence Knowledge Questions

Knowledge questions that are presented to students in the form of one or more sentences can be divided into three groups: i) reduced redundancy testing, ii) grammar questions, and iii) vocabulary questions.

Reduced redundancy testing

QDET for reduced redundancy testing has received a fair amount of research attention, and the proposed approaches have different levels of complexity; an overview is shown in Table 5.3.

The approach that is arguably the least complex is not based on any machine learning technique, and was proposed by Huang et al. in 2018 [56], in the same paper that deals with grammar questions and reading comprehension questions. The authors claim that the difficulty of cloze items is determined only by the difficulty of the correct answer. To estimate word difficulty, the authors use a graded word list made by an educational organization, the College Entrance Examination Center of Taiwan, which contains 6480 words in English divided into six levels of complexity. For QDET, the authors simply use the word difficulty from the aforementioned list, and

Paper	Year	Uses sentence(s)	Uses gaps	Approach	Type of question
[56]	2018	-	✓	maps missing word to difficulty using a table containing the difficulty of 6480 words in English	cloze
[53]	2019	✓	-	Features: mean token length, mean sentence length; Model: linear regression	cloze
[115]	2017	✓	✓	25 linguistic variables at passage and item level (also reduced to 6 with PCA); Model: linear regression	cloze
[34]	2019	✓	-	Shannon' entropy to assign a score to each gap based on the number of valid words that could fill the gap given the context (candidates obtained with a 5-gram language model)	cloze
[7]	2015	✓	✓	70 features related to the difficulty of the text passage, the difficulty of the target word and test parameters; model: SVM	cloze, c-tests, prefix deletion
[70]	2019	✓	✓	59 features (reduced from the 70 in [7]); models: SVM, BiLSTM, MLP	c-tests
[101]	2020	✓	✓	Features: average word length, sentence length, log-likelihood from a language model, and Fischer score; model: linear regression	cloze

Table 5.3: *Overview of the approaches proposed for reduced redundancy testing.*

observe that higher difficulty generally corresponds to lower correctness of students' answers.

Hou et al. in [53] proposed an approach for cloze items, which does not use any information about the gap but only the reading complexity of the passage. Specifically, it uses the mean token length and the mean sentence length of the textual passage to estimate question difficulty with a linear regression model. The ground truth difficulty is manually defined by human experts and there are two possible levels. Preliminary results presented in the paper show that, even though the chosen approach is arguably simple and cannot distinguish between different questions coming from the same textual passage, there is a positive correlation between the difficulty estimated with the proposed approach and the results observed in a test context.

Another paper addressing QDET of cloze items using only information from the text passage is [34], which performs a pilot study of an entropy based approach to estimate the difficulty. Specifically, the authors build on the assumption that the complexity of a gap is correlated to the number of possible answers determined by the surrounding context and the likelihood of each answer. In practice, they use Shannon's entropy [103] to assign a score to each gap based on the number of valid words that could fill in the slot given the surrounding context. As a result, gaps with many possible answers will yield higher entropy than those with fewer answers. The authors compute entropy using a 5-gram language model trained on the 1 Billion Word WMT 2011 News Crawl corpus[4] using KenLM [47], and considering only the 100 most probable words when computing the entropy of each gap (complete vocabulary has more than 82200 words). Using CEFR levels of the exams as difficulty gold standard, the authors study the correlation between the difficulty level and the entropy, and observe that indeed higher difficulty levels correspond to greater entropy.

Trace et al. in [115] study which features affect the difficulty of cloze items, and perform a regression analysis to observe the correlation between item difficulty and such features. Specifically, the authors consider 25 linguistic variables at both passage level and item level (mostly related to the number of words, sentences and syllables, and to the word frequency) and find that both passage level and item level are helpful for QDET. They also observe that three features accounted for 24% of the total variance of item difficulty: i) the frequency of the item elsewhere in the items, ii) the number of syllables per word, and iii) the number of sentences per 100 words in the passage.

The first paper addressing not only cloze tests but also c-tests and prefix deletion tests is [7], which extended previous work [6] and proposed a technique for QDET that is applicable to all three test types. Specifically, the proposed approach performs QDET with a SVM regression model and uses a subset of the features proposed in [6]: specifically, it uses 70 of the original 87 features, only the ones that can be computed for all test types. The features are related to i) the difficulty of the text passage, ii) the difficulty of the target word, and iii) test parameters. Evaluating the model on datasets containing tests in English, French, and German, the results show that there is a positive correlation between the selected features and the ground truth difficulty.

Taking inspiration from [6, 7], in [70] the authors proposed a technique to modify the difficulty of c-tests by varying the number and position of

[4] https://www.statmt.org/lm-benchmark/

the gaps. As for QDE, they evaluate a model similar to the one proposed in [7], extracting from the original set of 70 features 59 features related to: i) item dependency, ii) candidate ambiguity, iii) word difficulty, iv) and text difficulty. The regression is still performed with SVM. They evaluate the model both on the same data as [6, 7] and on a new private dataset, and obtain results in agreement with previous research. Additionally, the authors experiment with neural models for the regression component, but observe that they are outperformed by the SVM on both datasets.

Lastly, [101] proposes a linear regression model for QDET of cloze tests. The proposed model is also used for elicited speech and dictation items, and very similar to the one presented in the same paper for single word vocabulary questions. Indeed, it uses as features i) the average word length, ii) the sentence length, iii) log-likelihood obtained from a word-level unigram language model, and iv) Fischer score features. The authors evaluate the model using AUC and the CEFR level of English cloze tests as gold standard, and observe that all features are helpful for difficulty estimation. Additionally, with an ablation study, they find that the Fischer score has the biggest impact on the estimation (same observation as in the case of single word vocabulary questions).

Grammar Questions

An overview of the two approaches recently proposed for QDE of grammar questions is presented in Table 5.4.

Paper	Year	Uses sentence(s)	Uses gaps	Approach	Type of question
[56]	2018	-	✓	uses a table containing 44 pre-evaluated grammar patterns (of known difficulty); the difficulty of the question is the difficulty of the corresponding pattern	CGFI
[86]	2019	✓	✓	99 features from gap and context; then ridge regression	CGFI

Table 5.4: *Overview of the approaches proposed for grammar questions.*

In [56], the authors assume that the difficulty of a grammar question is determined by the difficulty of the grammar pattern of the correct answer. They identify 44 grammar patterns and estimate the difficulty of each one of them observing their rate of occurrence in English textbooks of different grade levels (assuming that the difficulty of the grammar pattern depends on

the grade level of the textbook in which it frequently appears). Lastly, the estimation of the difficulty of new questions consists in parsing the questions to identify the correct grammar pattern and searching the table for the corresponding difficulty.

The other paper that focused on grammar questions [86] adopted a traditional machine learning approach, modeling question difficulty with IRT. Specifically, the proposed approach consists in computing up to 99 features at gap-level (54), item-level (18), and context level (27), and using a ridge regression algorithm for difficulty estimation. Some features were directly extracted by the authors, while others were obtained from publicly available tools; for the complete list of features we refer the reader to the original paper. The authors experiment with several configurations (i.e. different subsets of the 99 features) and observe that the best results are not obtained using all of them. Indeed the authors propose a smaller model, which uses only 36 features (26 gap-level features, 4 context features, and 6 item level features). These features are selected via recursive feature elimination, which consists in recursively eliminating the least influential features. Some of the most important features for the estimation are: the tense of the verb (e.g. simple present, simple past, etc.); the presence of forms such as "used to" or "was going to" and adverbs; the word order; the frequency (from [51] and [14]), the word length, the age of acquisition [64], and the concreteness [15] of the words appearing in the question.

Vocabulary

Three papers have dealt with CIM questions [111–113]. An overview is presented in Table 5.5.

The first paper in this category [111] is a study that investigates the relations between several factors of question items in English CIM tests and the corresponding item difficulty. Specifically, the authors consider four elements: i) the target word, ii) the reading passage, iii) the correct answer, and iv) the distractors, and 10 features obtained from them. Most of these features (9 out of 10) are related to the word difficulty of the different elements (e.g. average word difficulty of the words in the reading passage), and the other feature is the number of word senses of the target word. The "word difficulty" is obtained from JACET 8000 [116], which is a list of 8000 words grouped in difficulty levels, specifically built for Japanese learners of English. The experimental results show that the number of word senses does not correlate well with the difficulty, probably because generally each word has one meaning that is much more frequent than the others. Considering the other features, the ones that correlate more with question

Paper	Year	Approach	Type of question
[111]	2017	10 features from target word, reading passage, correct answer, and distractors; studies correlation between features and difficulty.	CIM
[113]	2019	Features are reading passage difficulty, similarity between correct answer and distractors, and distractor word difficulty level. Two levels (low/high) for each of them, the number of "low" features represents the difficulty (from 0 to 3).	CIM
[112]	2020	Features are target word difficulty, similarity between correct answer and distractors, and distractor word difficulty level. Two levels (low/high) for each of them, the number of "low" features represents the difficulty (from 0 to 3).	CIM

Table 5.5: *Overview of the approaches proposed for closest-in-meaning questions. All proposed approaches leverage the text of the passage, the correct choice, and the distractors.*

difficulty are i) the difficulty of the target word, ii) the average difficulty of the words in the correct answer, and iii) the average difficulty level of the distractors.

In more recent work [113], the authors explore how three factors – related to the features mentioned above – can be leveraged to control the difficulty of CIM questions. The three factors are i) reading passage difficulty, ii) similarity between the correct answer and the distractors, and iii) distractor word difficulty level. For each of these factors, the authors only consider two levels (high and low), and the combination of levels is finally used for the task of QDE. For reading difficulty, the authors apply three well-established readability formulas to documents from two sources: *Times in Plain English*[5] to represent lower complexity English, and the *New York Times*[6] representing higher complexity English. The readability formulas used in this study are: Flesch-Kincaid Grade Level, Flesch-Kincaid Reading Ease [62], and Dale-Chall readability formula [16]. The average values obtained for the two levels of English are then considered as reference while performing QDET. For the similarity between correct answer and distractors, the authors use cosine similarity on the vectors representing the words; these vectors correspond to the frequency of the co-occurrence words within a certain window in the corpus. Finally, for the distractor

[5] http://www.thetimesinplainenglish.com/
[6] http://nytimes.com/

word difficulty level, the authors use JACET 8000. As for the estimation of question difficulty, the authors use the aforementioned factors to obtain four level of difficulty (corresponding to the number of "high" factors in the question), from "LLL" to "HHH". To evaluate the approach for QDE, the authors observe the correctness of students' answers for questions of different difficulty levels, and note that indeed there is a positive correlation between the difficulty estimated by the model and the fraction of wrong answers by the students.

The latest work [112] is an extension of [113]. Indeed, the authors have the same target of controlling item difficulty, but use slightly different features. The factors taken into consideration are: i) target word difficulty, ii) similarity between correct answer and distractors and iii) distractor word difficulty level. As before, for each factor two levels (high and low) are considered and the question difficulty is obtained from the combination of such levels (i.e. four levels). Again, for the target word difficulty and the distractor word difficulty level, the authors use JACET 8000, while the approach for computing the similarity is different from before and arguably more advanced. Indeed, the authors use GloVe [90] embeddings for calculating cosine similarity. The experimental results suggest that this approach is capable of more accurately estimating the difficulty of questions from text.

5.4 Literature on QDET in Content Knowledge Assessment

All the approaches proposed in the domain of content knowledge assessment focus on knowledge questions, and they can be categorized depending on the format of the questions they work on. Specifically, there is an important distinction between *text only* questions, whose content is only text (§5.4.1), and *heterogeneous* questions, which contain information of other types such as images and tables (§5.4.2).

5.4.1 Text Only Questions

Text only question are the ones that have received the most attention in recent years, when it comes to the task of QDET. An example of a text only question from [123], specifically an MCQ from a medical exam, can be seen in Figure 5.2. The proposed approaches can be categorized depending on the information they leverage and how they use such information, as shown in Figure 5.3.

Indeed, some approaches use only the text of the questions for the estimation [30], while others also use some additional information (which is

Question:
A 55-year-old woman with small cell carcinoma of the lung is admitted to the hospital to undergo chemotherapy. Six days after treatment is started, she develops a temperature of 38C (100.4F). Physical examination shows no other abnormalities. Laboratory studies show a leukocyte count of 100/mm3 (5% segmented neutrophils and 95% lymphocytes). Which of the following is the most appropriate pharmacotherapy to increase this patient's leukocyte count?

Options:

A. Darbepoetin	B. Dexamethasone	C. Filgrastim
D. Interferon alfa	E. Interleukin-2 (IL-2)	F. Leucovorin

Figure 5.2: *Example of text only question from [123].*

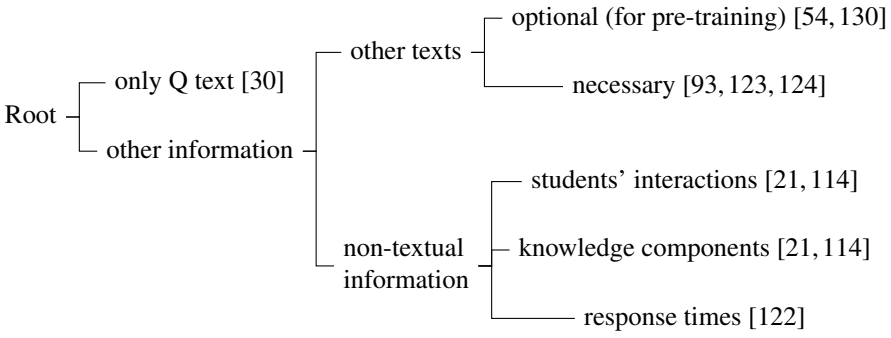

Figure 5.3: *Categorization of the approaches proposed for text only questions.*

not part of the questions themselves); importantly, the question text is always leveraged by all these approaches. The approaches based on question text can be applied in the most scenarios and have the least constraints but, on the other hand, the fact that they cannot leverage additional information might be a limitation if that is available. The works that leverage some kind of additional information can be divided into models that leverage texts from other sources, or non-textual information. As for the texts, they can be used either as data for pre-training a neural model [54, 130] or as data that is necessary for the implementation of the model [93, 123, 124]. Lastly, the non-textual information leveraged by some models can come from different sources: knowledge graphs [32, 63, 102, 118], students' interactions and knowledge components [21, 114], and response times [122].

Using only the question text

The models that leverage only question texts are the ones with the least constraints, since they can be trained using only the text and the target difficulty of each question. Only one model was proposed in previous research [30], and studies the correlation between the question difficulty (obtained with CTT) and several features obtained from the text of the questions. Specifically, the authors experiment with over a hundred linguistic indicators generated using the *Coh-Metrix* software [39] and categorised into five dimensions: i) narrativity: extent to which the item uses language comparable to everyday language; ii) syntactic simplicity: the degree to which the item is concise and makes use of simple and familiar syntactic structures; iii) word concreteness: the degree to which the vocabulary use is concrete and meaningful; iv) referential cohesion: the degree of overlap of words and ideas across sentences forming explicit connections; v) deep cohesion: the extent to which the item contains causal and intentional connectives that help the reader build connections and understand relationships and processes in the text. Additionally, the authors experiment with sentence length (number of words per sentence) and paragraph length (sentence count per paragraph) of descriptive statistics generated by *Coh-Metrix*. A linear regression model is used to estimate the difficulty of the 216 items of the experimental dataset from the aforementioned indicators, and the language variables do not seem to strongly correlate with item difficulty. The authors argue that this might be due to the fact that the question texts are fairly short, and therefore the language-related variables are not very meaningful. Also, the experimental dataset is quite small and this might have had an impact on the outcome of the research.

Additional texts

An alternative to using only the question text consists of leveraging additional resources such as books or lecture transcripts, with the constraint that such resources concern the same topics that are assessed by the questions. This is a crucial difference from language assessment, as in that case the additional resources used by models for QDET are domain-general corpora. Five works explored this area, along two different directions. On the one hand, the models proposed in [54, 130] make use of publicly available pre-trained models and leverage such additional texts only for further pre-training, therefore they can be fine-tuned on QDET and used even if such additional texts are not available. On the other hand, the models proposed in [93, 123, 124] heavily rely on such additional data and, if missing, they cannot be implemented without major modifications to the architecture. Even though it may be seen as a limitation, this approach might enable such models to extract more information from the additional texts. An overview of the proposed approaches is shown in Table 5.6.

Paper	Year	Other texts necessary	Approach
[54]	2018	-	Features: Cosine similarity between Word2Vec embeddings of stem, correct choice, and distractors; model: SVM
[130]	2020	-	multi-task BERT
[123]	2019	✓	Features: word embeddings (Word2Vec, ELMo), linguistic features, Information Retrieval-based features; model: random forest regressor
[124]	2020	✓	Same as [123]
[93]	2019	✓	Two neural networks, which estimate two components of question difficulty (recall difficulty and confusion difficulty); their estimations are then averaged.

Table 5.6: *Overview of the proposed approaches that use additional texts.*

The first paper that leverages additional textual corpora for the task of QDE from text is [54], in which the authors propose an approach built for MCQs on social sciences in Chinese. It is made of two steps: first, i) a Word2Vec model is used to obtain semantic vectors representing the question, the correct choice, and the distractors, then ii) the cosine similarities between these vectors are used as input to an SVM classifier that outputs the estimated difficulty. If such a corpus is not available, a pretrained Word2Vec model can be used but this compromise would most likely affect the accuracy of the model. The authors observe that i) there is a negative

correlation between the item difficulty and the similarity between stem and answer and ii) there is a positive correlation between the item difficulty and the similarity between the correct answer and the distractors.

Another approach that leverages additional corpora for pre-training is proposed in [130], which is a preliminary work about the effects of using multi-task BERT [26] for performing QDET; specifically dealing with English programming questions. The proposed approach i) starts from the pre-trained BERT model and ii) further pre-train it on a corpus of related documents, finally iii) it fine-tunes it on the task of QDET, modeling it as a binary classification task.

Three papers have presented approaches that leverage additional textual information and cannot be implemented if such data are not available. Two of them, written by the same team of researchers, focus on the task of QDET for MCQs in high stakes medical exams. The first one [123] directly focuses on difficulty estimation, while the second one [124] deals with item survival, which heavily depends on the difficulty (items with difficulty above or below a given threshold are not considered suitable for scoring, and thus do not "survive"). The proposed approach is divided into two steps: i) first, there is a feature engineering phase, when the input text is converted into feature arrays, then ii) the feature arrays are used as input to a regression model that performs the actual estimation of difficulty (which is modeled with CTT in the range $[0; 100]$). The features can be categorized into three groups: i) word embeddings, ii) other linguistic features, and iii) Information Retrieval (IR) features. As for the embeddings, the authors use Word2Vec and ELMo, both pretrained on a corpus of about 22M MEDLINE abstracts[7]. The linguistic features are a set of about 60 values coming from different sources: lexical features, syntactic features, semantic ambiguity features, readability formulae, cognitively-motivated features, word frequency features, and text cohesion features. Lastly, the IR features are obtained from an automated Question Answering system that is trained to respond to the item by retrieving relevant documents from the MEDLINE corpus. The authors experiment with different regression models, and observe that Random Forests are the best performing. The authors also perform an ablation study and find that all the features are helpful for the estimation, and that the IR features are, on their own, the most useful. On the other hand, embeddings and linguistic features led to comparable performance when used singularly.

The other approach to QDET using additional texts was proposed in [93], targeting MCQs in medical exams. The proposed approach is com-

[7] https://www.nlm.nih.gov/bsd/medline.html

posed of two neural networks, which are used in parallel to compute different components of question difficulty, which are later averaged to obtain a final difficulty score. The first of them, referred to as Recall Difficulty Module, receives as input i) a corpus of related medical documents, ii) the text of the questions and iii) of the correct choices, and it has the goal of estimating how difficult it is to recall the knowledge assessed by the question. The other component, named the Confusion Difficulty Module, receives as input the stem of the question and the possible choices (both the correct one and the distractors) and has the target of estimating how difficult it is to distinguish between the different choices. Finally, the two components of the difficulty are combined with a weighted average (the weight is learned and depends on the stem and the correct choice). The authors observe that, when using only the Recall Difficulty Module or the confusion difficulty module, the error is higher than when using the complete network, although the difference is not great.

Additional information, students' interactions and knowledge components

Two papers [21, 114] proposed approaches for QDET using, as additional information, the knowledge components (i.e. topics) associated to each question and the results of students' answers; an overview is presented in Table 5.7.

Paper	Year	Approach
[21]	2019	Two components: i) LSTM that receives the text of the question, ii) attention based model that capture relevance between texts and knowledge components. Then, average pooling.
[114]	2020	Pre-trained BERT to embed questions, and TextCNN to perform QDE.

Table 5.7: *Overview of the approaches that use knowledge components and students' interactions as additional information.*

The fact that such models leverage students' interactions make them unusable for QDE in the case of new items, since in that case no log of interactions is available. Indeed, although both papers proposed a model for QDE, their final target is students' performance prediction, which motivates the need for a history of previous answers.

The first of these papers [21] proposed DIRT – Deep Item Response Theory – which is a model that takes inspiration from IRT for estimating the probability that a given student correctly or wrongly answers a question, but relies on neural networks for the estimation of the IRT latent traits.

DIRT is made of three modules: i) an input module, ii) a deep diagnosis module, and iii) a prediction module. Here, we are interested only in the deep diagnosis module and, specifically, in the component that performs QDET, therefore we will not present the other components of DIRT. The model used for QDET is made of two parts, which estimate the difficulty from two different perspectives. The first one exploits semantics of question texts for the estimation, which is performed with an LSTM network that receives as input the text of the questions. The second perspective considers the width and depth of knowledge concepts, which is reflected by the relevance between question texts and knowledge concepts. In practice this is done with an attention mechanism, that captures the relationship between question texts and knowledge concepts. Lastly, an average pooling operation is performed to obtain the difficulty. Since the final target of the paper is student answer prediction, there are no experiments to directly compare the estimated difficulty with a ground truth value.

The other approach that leverages students' interactions and knowledge components for the task of QDET was proposed in [114], whose final target is knowledge tracing, consisting in modeling the evolution of students' skill levels and predicting the correctness of their answers to exam questions. In this case, the ground truth difficulty is obtained with CTT, and the model itself for QDET is fairly simple. Indeed, the authors employ a pre-trained BERT model for embedding the questions and apply a TextCNN [61] model for performing QDE.

Additional information, response times

One paper [122] has proposed a transfer learning based model for QDE of MCQs in medical exams, using question text and response times as features. The proposed model is made of an ELMo network, pretrained on the One Billion Word Benchmark [17], followed by an encoding layer added to learn the sequential information from the ELMo embeddings; the encoding layer is made of a BiLSTM. A dense layer then follows the encoding layer to convert the feature vectors to the targets through a non-linear combination of the feature vectors' elements. Considering the target, the model is first trained for response time prediction, and later fine-tuned for the task of QDE. The authors experimented with three different ELMo configurations (small, middle, and original) and various input configurations (stem only, options only, stem and options). The results indicate that transfer learning can be applied to improve the prediction of question difficulty when response time is used as pretraining and the difficulty is best predicted when using only the item stem. Contrary to the findings from [9,10,54], using the

answer options does not increase the performance of the model and actually hinders it, even though the difference is not great. Also, it is interesting to observe that, even though the best results are obtained with the larger model (i.e. ELMo original, 93.6M parameters), there is not clear correlation between the size of the models and the accuracy of the estimation: indeed, the errors obtained with ELMo middle (20.8M parameters) are generally larger than the ones obtained with ELMo small (13.6M parameters).

5.4.2 Heterogeneous Questions

Considering the publications that deal with heterogeneous questions, they all focus on questions with accompanying images [33, 109, 129] but one of them [109] also focuses on the effects that tables have on question difficulty. An overview is shown in Table 5.8.

Paper	Year	Approach
[109]	2016	Studies how the presence of images, tables, formulas, and some textual features (text length, presence of specialist terms and abstract concepts) affect the item difficulty.
[33]	2019	ResNet for extracting image representations, BERT for embedding textual content. Capsule Neural network to obtain a fixed-length vector which represents the exercise. Bayesian inference-based softmax regression classifier to perform the estimation.
[129]	2019	i) embedding of heterogeneous content (Word2Vec for texts, convolutional layers for images, fully connected layers for metadata), ii) BiLSTM, iii) self-attention, and iv) max pooling to obtain pre-trained question representations. Fine-tuning on QDE with a FCNN for the regression task.

Table 5.8: *Overview of the various approaches proposed for QDE of heterogeneous questions.*

The first work to focus on question images and their effects on the difficulty was [109], which performed a study of how some textual features and the presence of images, tables, and formulas (not their content) affect the IRT difficulty of MCQs in a scientific reasoning exam. Considering textual features, the authors takes into consideration text length, and the presence of specialist terms and abstract concepts. By studying the correlation between the aforementioned features and the IRT difficulty of the items, the authors observed that the difficulty is significantly increased by the presence of abstract concepts and specialist terms, suggesting that they might be a good predictor of question difficulty. The presence of images as well has a positive effect on item difficulty, meaning that items that contain

visual images tend to be harder to solve. This result is in contrast with previous research [71], and the authors claim that this might happen because the images in the experimental dataset are generally used to show complex scientific models and therefore the increase in difficulty might come from that, not from the images.

Being able to model the content of the images and not only their presence might be very helpful for improving the accuracy of QDE for questions containing images, which is the focus of two papers from 2019. The first of them [33] proposed an approach for predicting the difficulty of visual-textual exercises, using as input both the text and the image of each question. The authors experiment on two datasets, one containing mathematics questions and the other containing medicine related questions. The proposed model is made of two modules: i) a feature extraction module and ii) a difficulty classifier module. More precisely, the feature extraction module contains two components: a Residual Network [46] for extracting the representation of the images, and a BERT model [26] for embedding the textual content. Since the two vector representations can have different lengths, they are then fed into a Capsule Neural Network [96] to obtain a fixed-length vector which contains the unified representation of the exercise. The fixed-length representation of each exercise is then used as input to the difficulty classifier module, that is a Bayesian inference-based softmax regression classifier and performs the actual estimation of difficulty.

A different approach to perform QDE in heterogeneous questions was proposed in [129], which introduced a general pre-training method – namely QuesNet – to learn question representations that could be fine-tuned for several downstream tasks, one of them being difficulty estimation, similarly to what is done in general purpose pre-trained language models (e.g. BERT). Specifically, the paper focuses on mathematics MCQ, all containing an image. At a high level, QuesNet is made of three components: i) an embedding module, ii) a content module, and iii) a sentence module. The embedding module projects heterogeneous input content into a unified space, which enables the model to work on inputs from different sources. Specifically, the input can be i) text from the body of the question, ii) an image which is part of the question, and iii) question metadata (e.g. the knowledge components associated with a question). Text embedding is performed with Word2Vec, image embedding is done with three convolutional layers followed by activations and a max-pooling layer, the metadata embedding is performed with two fully connected layers. The content module is made of a BiLSTM which receives the concatenation of vectors produced by the embedding module. Then, the sentence module leverages

a self-attention module for aggregating the item representation vector into a sentence representation; this is done with a multi-head attention module to perform global self attention [117]. Finally, there is a max pooling layer to produce a single vector representing the heterogeneous input image. The proposed architecture is pre-trained with a two level hierarchical approach: first a masked language model is used as objective for learning low level linguistic features; then a domain oriented objective is used for learning high level domain logic and knowledge. The embedding modules are pre-trained separately: the text embedding is a Word2Vec model trained on the specific corpus, the image and metadata embeddings are fully connected neural networks pre-trained using an encoder-decoder architecture and an auto-encoder loss. Once pre-trained, the model can be fine-tuned for specific downstream tasks. Among other tasks, the authors experiment with QDET, and do so by adding a fully connected layer on top of the question embeddings.

5.5 Models

In this Section, we describe the details of the models that we experiment with in this study. We do not evaluate all the approaches proposed in previous literature, as that would be unfeasible due to both their number and the fact that many are built to leverage question characteristics that are not available in the experimental datasets we consider. Indeed, we experiment with three textual datasets – two belonging to the CKA domain and one to the LA domain – and our goal is to understand how different models perform across them. Therefore, we do not re-implement the approaches that leverage predefined measures for language learners [56] or the rigid mapping from feature to difficulty level used in [111–113]. Similarly, we do not implement all the approaches that leverage non-textual information [21,32,33,63,85,91,102,109,114,118,129], or features that cannot be computed on the type of questions that we experiment on: *TextEvaluator* features used in [77], the *Coh-Metrix* features from [30], the entropy used in [34], the gap features used in [86], the response time used in [122].

Nonetheless, even though the list of papers which are not implemented in this study might seem long, we experiment with several approaches which were used in previous research. Specifically, we experiment with:

- *linguistic features* followed by a regression or classification model, which are used in diverse forms in [24,53,115];
- *readability indexes* as features to a regression or classification model,

as in [56];

- *Information Retrieval (IR) features* as input to a regression or classification model, similarly to [101], using a model we proposed in previous research [10];
- *Word2Vec embeddings*, as in [28], which are used as input to a regression or classification model;
- *Transformer*-based neural networks, which we evaluated in previous research [8], following the examples of [130] and several other approaches that leveraged BERT or the attention mechanism [33, 57, 114];
- several *hybrid* models which merge some of the approaches listed above: linguistic and readability features [7, 9, 70]; linguistic, readability, and IR features [9]; linguistic features and word embeddings [125]; linguistic, IR features, and word embeddings [123, 124].

It is important to mention here that we do not exactly re-implement all these models, since that is in most cases unfeasible due to some missing details which might hinder the exact replication of previous experiments (e.g. model hyperparameters, training setup, etc.). Moreover, even if we could exactly re-implement all these models, we are experimenting on different datasets from the ones used in the original papers. Therefore, we do not have the goal of finding an approach which is overall the "best" but rather understand how these approaches perform on the three datasets that are available to us and whether there are some specific question characteristics which cause them to work particularly well or – on the other hand – lead to inaccurate estimations.

5.5.1 Linguistic Features

The usage of linguistic features for the task of QDET proved useful in previous research and, according to their indications, we experiment with the following features:

- *Word Count Question*;
- *Word Count Correct Choice*;
- *Word Count Wrong Choice*;
- *Sentence Count Question*;

- *Sentence Count Correct Choice*;
- *Sentence Count Wrong Choice*;
- *Average Word Length Question*;
- *Question Length divided by Correct Choice Length*;
- *Question Length divided by Wrong Choice Length*.

These features are not directly used to perform QDET, but rather we use them as input features to a regression model that performs the actual estimation of difficulty. Specifically, we experiment with a Random Forest model, which is the most commonly used model in previous research.

5.5.2 Readability Indexes

Readability indexes are measures designed to evaluate how easy a text is to understand, thus they can prove useful for estimating question properties. In particular, we use five indexes.

- The *Flesch Reading Ease* [36] gives a text a score between 1 and 100, with 100 being the highest readability score; it is computed from the average number of words per sentence and the average number of syllables per word together with some constants and coefficients.
- The *Flesch-Kincaid Grade Level* [62] approximates the reading grade level of a text and it is very similar to the Flesch Reading Ease, as it uses again the average number of words per sentence and the average number of syllables per word but different constants and coefficients.
- *Automated Readability Index* (ARI) [100] assesses the U.S. grade level required to read a piece of text; it is computed from the average number of characters per word and the average number of words per sentence.
- The *Gunning FOG Index* [42] generates a grade level between 0 and 20, which estimates the education level required to understand the text; it is computed from the average number of words per sentence and the average number of *complex* words per sentence, being complex words the ones containing three or more syllables.
- *Coleman-Liau Index* [23] is a readability formula which shows the reading level of a text; it uses number of sentences and number of letters as variables.

Some papers also experimented with the *SMOG Index* ("Simple Measure of Gobbledygook") [80], which measures how many years of education the average person needs to have to understand a text. However, we do not use it as it is best for texts of 30 sentences or more, which is not the case for the assessment items in the experimental datasets we consider in this work.

Similarly to what we do for the linguistic features, we use these indexes as input features to a Random Forest model.

5.5.3 Information Retrieval Features

Considering the features based on information retrieval techniques, we re-implement R2DE, which we proposed in a previous paper [10]. R2DE is a *Regressor for Difficulty and Discrimination Estimation* and it leverages TF-IDF for the estimation. It estimates both difficulty and discrimination, as defined in two-parameter IRT, but here we focus only on the components that perform difficulty estimation. This does not affect the efficacy of the model as the two components are separate and work in parallel, as visible in Figure 5.4. The image also shows that, in addition to having two parallel pipelines for difficulty estimation and discrimination estimation, R2DE is made of two parts: i) a feature engineering part (concatenation, preprocessing, and TF-IDF) and ii) a regression part.

Feature Engineering

The first step is the creation of a single text from the input question, and we test three alternatives (later referred to as *encodings*):

- Q_{Only} considers only the question;
- $Q_{Correct}$ appends the correct options to the question;
- Q_{All} concatenates all the options (both correct and wrong) to the question.

As an example, we can consider the MCQ *"Which is the capital city of Germany?"* with possible choices *"London"*, *"Berlin"*, *"Madrid"* and *"Paris"*. This question would be transformed, with the three encodings, in the following input texts: i) *"Which is the capital city of Germany?"*, ii) *"Which is the capital city of Germany? Berlin"*, and iii) *"Which is the capital city of Germany? London Berlin Madrid Paris"*. In all three cases, the question is converted into a single text.

As a second step, the input text is preprocessed using some standard steps of NLP: stop words and punctuation are removed, and the words are stemmed [78].

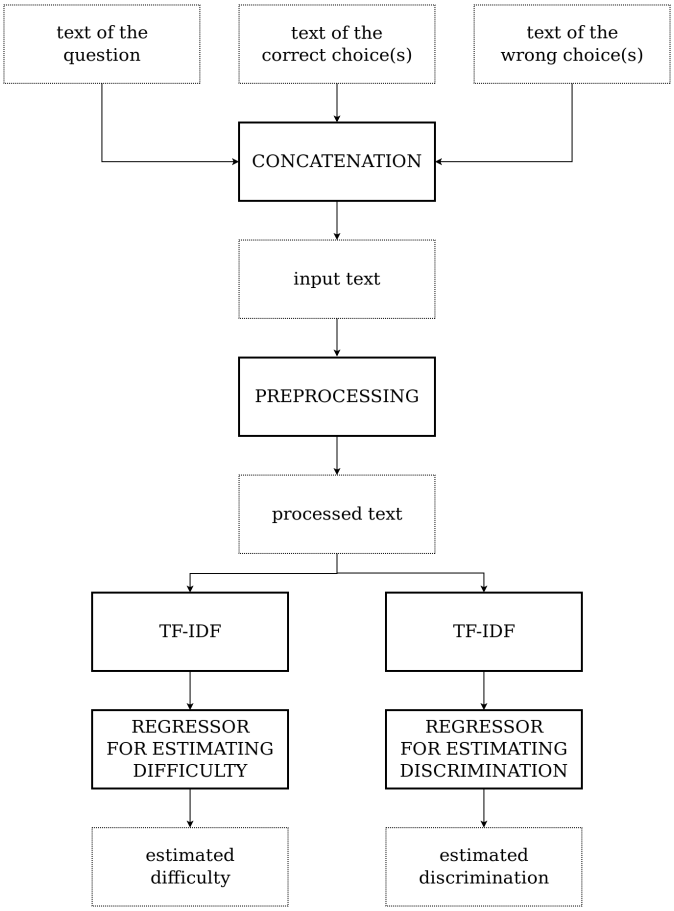

Figure 5.4: *Representation of the architecture of R2DE, from the input text to the output difficulty and discrimination.*

The third step consists in using TF-IDF to convert the input text into an array of features. TF-IDF, by default, generates vectors whose dimension is equal to the number of stemmed words from the original corpus, which would be intractable for the regression component. Therefore, we sort the features according to their number of occurrences in the corpus, and keep only the ones whose frequency is both below an upper threshold max_F and above a lower threshold min_F. The idea of this approach is to remove both the corpus-specific stop-words and the words that are so infrequent that the model is unlikely to learn them. Additionally, since the number of features obtained with this approach might still be too large, we keep at most the first N_W. All these thresholds (max_F, min_F., and N_W) are considered as parameters of the model and therefore can be chosen with cross-validation in order to be tuned for a specific dataset.

The Regression Algorithm

The regression component of R2DE is straightforward: indeed, it is made of a Random Forest regressor, whose parameters (i.e. number of trees and max depth) are chosen with cross validation. In addition to that, the encoding to use (Q_{Only}, $Q_{Correct}$, or Q_{All}) is also chosen with cross-validation.

5.5.4 Word2Vec

The majority of previous research used Word2Vec [81] as the technique for building word embeddings, therefore that is the non-contextualized word embedding technique we experiment with as well (as for contextualized embeddings, we use Transformer models). In our experiments, we use Word2Vec embeddings as input features to a Random Forest model, similarly to what we do for the other feature types.

Following the examples of previous research, we use Word2Vec arrays with dimension of 100 and we train them on the training questions of the datasets. We obtain the embedding of the whole question by averaging the embeddings of the single words.

Similarly to the approach we take for the Information Retrieval based features, we test three approaches:

- Q_{Only} considers only the question;
- $Q_{Correct}$ appends the correct options to the question;
- Q_{All} concatenates all the options (both correct and wrong) to the question.

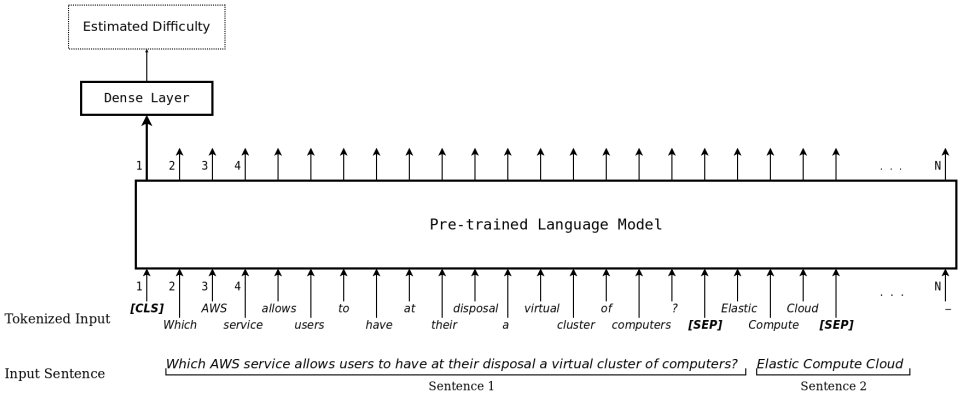

Figure 5.5: *Fine-tuning of the pre-trained Transformer language model for the task of QDET.*

5.5.5 Transformers

Transformers are pre-trained language models that can be fine-tuned to target various downstream tasks. This generally leads to better performance with shorter training times with respect to training the neural model from scratch, because it leverages the pre-existing "knowledge" of the pre-trained model. We experiment with two pre-trained language models – BERT and DistilBERT – and we fine-tune them with two different approaches, similarly to what we did in a previous paper [8]. The first approach is straightforward and consists in directly fine-tuning the pre-trained model for the task of QDET. The second approach, on the other hand, is divided into two steps: first of all, we i) perform an additional pre-training of the pre-trained model on the task of Masked Language Modeling (MLM), and then we ii) fine-tune it on the task of QDET. The difference between the two approaches is that, in the second case, we try to improve the domain knowledge of the model – by leveraging the MLM task – before the final fine-tuning on QDET, and this can lead to improved performance.

Fine-tuning for Supervised QDET

The architecture used for fine-tuning with this approach is shown in Figure 5.5. We follow the guidelines provided in the the original paper which introduced BERT [26]. Specifically, we use only the first output of the pre-trained language model and stack an additional fully connected layer on top of the network, using that as the output for QDET. This works because the first output of the pre-trained model is a special token `[CLS]` that is added in the first position of the input text and is the only one used for regression

and classification.

The number of neurons of the additional output layer depends on the specific choice of learning theory, as it depends i) on whether the difficulty is defined as a continuous or discrete variable and ii) on the number of classes in the case of discrete difficulty. In the experimental datasets we consider, the difficulty is defined either as a continuous variable or as a binary variable; therefore the additional output layer has one neuron, and the weights of the connections with the previous layer are randomly initialized. When we fine-tune the model, we update both the weights of the additional layer and the weights of the pre-trained language model.

For tokenizing and encoding the input, we use the same approach as in the original models. Specifically, we add the special token `[CLS]` in the first position of all the input samples, and all the input samples are made of pairs of sentences, separated by the special token `[SEP]`: the first sentence represents the question (after tokenization) while the second contains the (tokenized) text of the possible answer choices[8]. Similarly to what we do for R2DE and Word2Vec, we experiment with three encodings for the second sentence. Specifically, we consider: i) only the text of the question, by leaving the second sentence empty (Q_{Only}); ii) only the correct choice(s) ($Q_{Correct}$, as in the example in Figure 5.5), iii) all the possible choices, concatenating them in a single sentence (Q_{All}). In our paper [8] we also experimented another approach, considering all the possible choices and separating them with several `[SEP]` tokens (one before each choice); however, this approach performed largely worse than the others, thus we do not report it here. Most likely the model, in that case, is not capable of learning the meaning of the additional separators due to the limited number of training questions (BERT and DistilBERT, indeed, are pre-trained to use only one `[SEP]` token).

Pre-training for MLM

Masked Language Modeling (MLM) is a fill-in-the blank task, where a word of the input text is substituted by a `[MASK]` token and the model is trained to use the surrounding words to predict the word that was masked, as shown in Figure 5.6.

The second approach based on Transformers that we evaluate in this thesis consists of performing an additional MLM pre-training before the fine-tuning on QDET. This is based on the idea that the additional MLM pre-

[8] If a question is made of several sentences, the whole question is considered as "Sentence 1", and the `[SEP]` token is still used only to indicate the end of the question. We use this naming ("Sentence 1" and "Sentence 2") since it is the one used in the original paper.

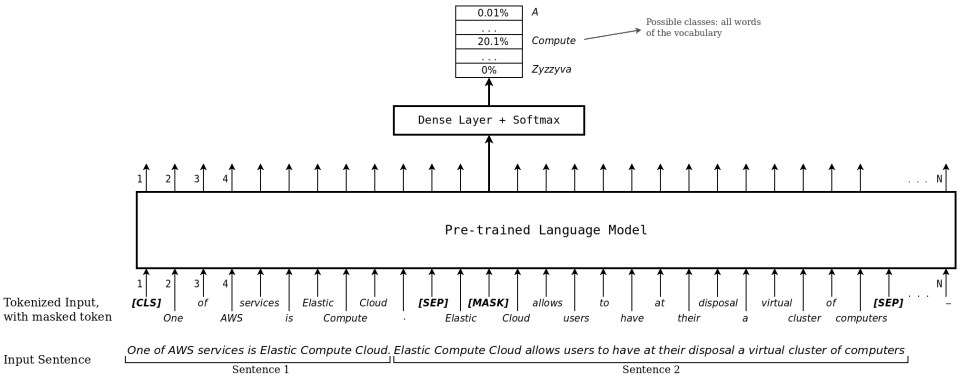

Figure 5.6: *Additional pre-training of the pre-trained Transformer language model for the task of masked language modeling.*

training can improve the domain knowledge of the models and therefore lead to better accuracy in the QDET task. The limitation of this approach is that it requires an additional dataset of documents (i.e. lectures, books, etc.) about the same topics that are assessed by the questions. Considering the data collections we experiment on in this work, this is available only for *Cloud Academy*: it is *Cloud Academy_LEC*, which contains the transcript of some of the video-lectures on the e-learning platform.

In practice, the additional MLM pretraining is performed as follows. In the available lectures, 15% of the words are randomly masked and the language model is trained to predict the masked words, sentence by sentence. The actual prediction is performed by stacking a fully connected layer and a softmax layer on top of the original pre-trained model: for each masked sentence, this additional layer consumes as input the contextual embedding corresponding to the `[MASK]` token, and tries to predict the word that should be inserted in its place. Once the additional pre-training on MLM is completed, the additional dense and softmax layers are not needed any more and therefore we remove them from the network, obtaining a pre-trained model with the same architecture as the original. Therefore we can perform the final fine-tuning for QDET with the same architecture shown in Figure 5.5. The crucial difference is that, now, all the internal weights were updated by the additional MLM pre-training.

5.5.6 Hybrid Models

As mentioned in the beginning of this Section, in addition to experimenting separately with each model, we also experiment with some hybrid approaches, which are commonly used in the literature. All the hybrid ap-

proaches considered here consist in concatenating features from two (or more) of the groups presented above, and use them as input to a single regression/classification model. Specifically, we experiment with:

- linguistic and readability features, as in [7, 9, 70];
- linguistic, readability, and IR features, as in [9];
- linguistic features and Word2Vec embeddings, as in [125];
- linguistic, IR features, and Word2Vec embeddings, similarly to [123, 124], even though they also use additional embeddings.

5.6 Experimental Setup

In this section we describe the experimental setup, which is used for all the experiments whose results are presented in the next section. First, we describe the setup and data splitting used for the experiments (§5.6.1); then we briefly reintroduce the experimental datasets (§5.6.2), focusing on the aspects that are peculiar to supervised QDET.

5.6.1 Setup for Calibration, Training, and Evaluation

All the models presented in Section 5.5 require the training and evaluation data to be in a certain format: they need the set of questions and, for each of them, the difficulty and the ground truth difficulty. The only exception are the Transformer-based models, which might require a corpus of additional documents for the additional pre-training on Masked Language Modeling (MLM).

Considering the datasets presented in the previous section, the ground truth difficulty is readily available for *RACE* only; for *Cloud Academy* and *ASSISTments* we have to estimate it from the logs of student answers. Figure 5.7 shows the experimental setup for these two datasets: first of all, we use the log of student answers to perform question difficulty estimation with Item Response Theory (IRT). Then, we split the questions into train and test set, in order to train and evaluate the models that perform QDET, which use the IRT difficulties as gold standard. We keep 80% of the questions as training questions and 20% of them as test questions; additionally, at training time, we keep 10% of the training set (i.e. 8% of the total number of questions) as evaluation which is used for hypermarameter tuning.

The setup is simpler for *RACE*, as the ground truth difficulties are already available (either 0 or 1, representing “middle” and “high” levels)

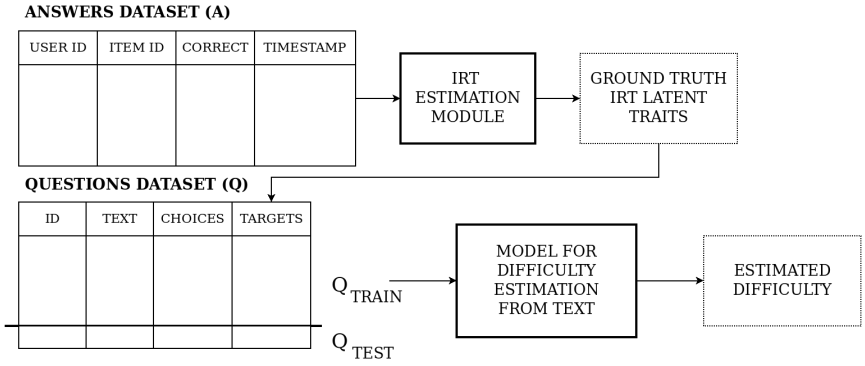

Figure 5.7: *Experimental setup for Cloud Academy and ASSISTments.*

and there is no need to estimate them from the log of student answers. Crucially, this implicates a difference in the difficulty format in the three datasets: while *ASSISTments* and *Cloud Academy* have difficulty values in the range $[-5; +5]$ (selected by us while training the IRT model), *RACE* contains categorical difficulties whose value can either be 0 or 1. As a consequence, also the metrics that are used for evaluating the models are different in the two scenarios: for *Cloud Academy* and *ASSISTments* we use regression metrics, while on *RACE* we use classification metrics.

Additionally, we also convert the IRT difficulties and the predicted difficulties of *Cloud Academy* and *ASSISTments* into categorical difficulties, in order to have a better indication of how the model performance compares between the three datasets.

5.6.2 Experimental Datasets

We already presented in detail the three data collections in Chapter 4, therefore we discuss here only the aspects that are specific of supervised QDET.

Cloud Academy

The *Cloud Academy* data collection contains three datasets.

Cloud Academy_A contains the log of students' answers to exam questions; we use it to train the IRT model and obtain, with pretesting, the gold reference difficulty which is used for training and evaluating the models.

Cloud Academy_Q contains the textual information about the questions, which is the input to all the models we evaluate. Figure 5.8 displays the distribution of training questions by IRT difficulty, obtained from the interactions in *Cloud Academy_A*. The questions are distributed according to a

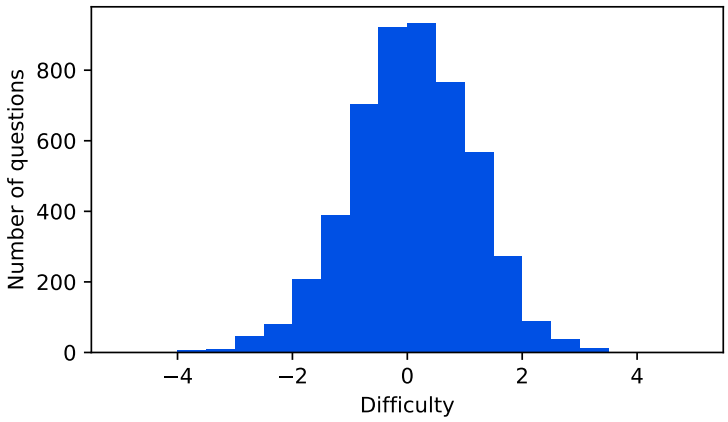

Figure 5.8: *Distribution of questions in Cloud Academy by IRT difficulty.*

Gaussian-like shape.

Cloud Academy_LEC contains the text of some online lectures available on the *Cloud Academy* platform. We use this dataset to perform some additional experiment on the Transformer-based models, to explore whether leveraging additional corpora of learning contents related to the same topics assessed by the questions can lead to improve performance in the task of QDET.

ASSISTments

Two datasets are available from ASSISTments: *ASSISTments_A* and *ASSISTments_Q*.

ASSISTments_A, similarly to the analogous *Cloud Academy_A*, contains the log of students' answers (used to obtain the "true" question difficulty), and *ASSISTments_Q* contains the question texts. No lecture dataset is available, meaning that it is not possible to perform the additional pre-training of the Transformer models.

Figure 5.9 displays the distribution of training questions by IRT difficulty, which is the target of our estimation. It is clearly visible that the questions are distributed similarly to what we observed for *Cloud Academy*, in a Gaussian-like shape, but there are two small peaks at the extremes, representing questions which were correctly (or wrongly) answered by all the students, which were not present in *Cloud Academy_A*.

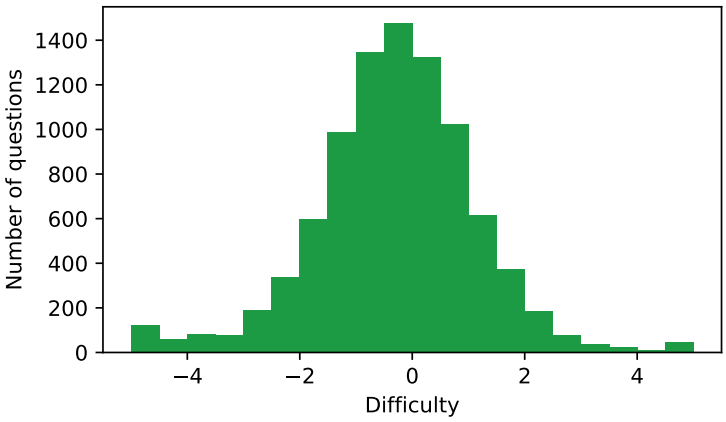

Figure 5.9: *Distribution of questions in ASSISTments by IRT difficulty.*

RACE

Only one dataset is available for *RACE*, and it contains, for each question: i) the text of the reading passage, ii) the text of the question, iii) the text of the possible options, and iv) the manually calibrated difficulty, which is a binary categorical value. For the experiments on supervised QDE, we use a reduced version of the original dataset, obtained by keeping one fourth of the questions. Specifically, in order not to have questions too similar to each other, we only keep one question for each textual passage, and the reduced dataset is unbalanced, as the original one is.

The following section presents the analysis of the experimental results carried out on the three datasets, from the comparison of the difficulties estimated with the different approaches to supervised QDET, to a more detailed analysis of how different models perform on different types of questions.

5.7 Results

This section presents the results of all the experiments that were carried out to evaluate the performance of different models in the task of supervised QDET. We start by comparing the estimated difficulties with the target values (§5.7.1), we then study the distribution of the difficulties estimated with each model (§5.7.2) and to which extent the model performance depends on question characteristics such as the question length, the number of correct choices (for MCQ), and the type of question (§5.7.3).

5.7.1 Comparison with gold standard difficulties

In this subsection, we evaluate the different models by comparing the estimated difficulties with the gold standard.

Cloud Academy

Table 5.9 presents the evaluation of all the models on the *Cloud Academy* test dataset. Each row present a different model, and each column corresponds to one metric; we use Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), nDCG, and R2 score.

It is immediately visible that QDE from text in the Content Knowledge Assessment (CKA) domain is a challenging task: indeed, some models perform even worse than the *ZeroR* baseline (which estimates for all the questions the average difficulty of the training set), and the best performing model – *BERT ($Q_{Correct}$) with MLM* – reduces the MAE of the *ZeroR* baseline of less than 7%.

Readability and *Linguistic*, which do not consider in any way the semantic meaning of the question, are outperformed by almost all the other models, which is reasonable considering that in CKA the question question difficulty derives mostly from the knowledge which is being assessed.

The only models that are outperformed by *Readability* and *Linguistic* are the Transformers models, when used in the Q_{Only} configuration; most likely, Transformers perform poorly with this configuration (three of them even have negative R2 scores) because they receive only one sentence, instead of the two sentences (separated by the special token `[SEP]`) which they are pre-trained on. This is also supported by the fact that this issue does not occur for *DistilBERT (Q_{Only}) with MLM*, which is a smaller model with respect to BERT, and therefore the number of training questions is sufficient to reduce the impact of this issue.

Considering the models that can be implemented with different input configuration (Q_{Only}, $Q_{Correct}$, and Q_{All}), we can see that the text of the answer choices is helpful for the estimation of question difficulty. However, some models performs best when all the answers choices are used (Word2Vec, DistilBERT without MLM), while other when only the correct choice(s) is used (R2DE, DistilBERT with MLM, and BERT); still, the difference is generally minor.

All the hybrid models perform fairly well, with the exception of *Readability + Linguistic*, which is outperformed by most of the other models.

The nDCG metric is always fairly high (the theoretical maximum value is 1.0), which might suggest that all these models perform really well. How-

ever, that is not really the case, and we observe that the nDCG values are overall high due to the distribution of the gold standard difficulties: indeed, the "real" difficulties are distributed with a Gaussian-like shape, meaning that many questions have similar values of difficulty and, therefore, swapping their order does not affect too much the nDCG.

Model	MAE	RMSE	nDCG	R2
ZeroR	0.8901	1.1258	0.9549	0.0000
Linguistic	0.8796	1.1064	0.9637	0.0342
Readability	0.8840	1.1143	0.9589	0.0204
R2DE (Q_{Only})	0.8572	1.0896	0.9615	0.0632
R2DE ($Q_{Correct}$)	0.8554	1.0853	0.9647	0.0706
R2DE (Q_{All})	0.8604	1.0907	0.9643	0.0614
W2V (Q_{Only})	0.8746	1.2167	0.9597	0.0401
W2V ($Q_{Correct}$)	0.8739	1.1047	0.9638	0.0371
W2V (Q_{All})	0.8643	1.0919	0.9658	0.0591
DistilBERT (Q_{Only})	0.9220	1.1708	0.9676	-0.0820
DistilBERT ($Q_{Correct}$)	0.8640	1.0955	0.9666	0.0527
DistilBERT (Q_{All})	0.8482	1.0846	0.9681	0.0714
BERT (Q_{Only})	0.8908	1.1322	0.9666	-0.0119
BERT ($Q_{Correct}$)	0.8471	1.0630	0.9685	0.1080
BERT (Q_{All})	0.8515	1.0874	0.9666	0.0667
DistilBERT (Q_{Only}) with MLM	0.8702	1.1085	0.9681	0.0301
DistilBERT ($Q_{Correct}$) with MLM	0.8434	1.0706	0.9684	0.0953
DistilBERT (Q_{All}) with MLM	0.8596	1.1011	0.9671	0.0429
BERT (Q_{Only}) with MLM	0.8952	1.1256	0.9677	-0.0001
BERT ($Q_{Correct}$) with MLM	**0.8351**	**1.0535**	0.9686	**0.1238**
BERT (Q_{All}) with MLM	0.8427	1.0762	**0.9700**	0.0857
Ling. + Read.	0.8774	1.1033	0.9639	0.0395
Ling. + Read. + R2DE (Q_{All})	0.8370	1.0563	0.9652	0.1197
Ling. + W2V (Q_{All})	0.8566	1.0796	0.9650	0.0803
Ling + R2DE (Q_{All}) + W2V (Q_{All})	0.8352	1.0583	0.9678	0.1163

Table 5.9: *Evaluation of different models on Cloud Academy, comparing the difficulty estimated from text with the target value. All evaluation metrics are computed on the held-out test set. We write in bold the best performing model (separately for each metric) and in gray the ones that perform worse than ZeroR.*

ASSISTments

Table 5.10 presents evaluation of the models on the *ASSISTments* dataset, using the same evaluation metrics as on the *Cloud Academy* dataset. In this case, the table has a smaller number of rows due to the fact that the text of the possible options is not available in *ASSISTments*, therefore the $Q_{Correct}$ and Q_{All} configurations of the models cannot be evaluated. Also, an

additional corpus is not available, therefore the MLM pre-training of BERT and DistilBERT cannot be performed.

The findings are fairly similar to *Cloud Academy*. Almost all the models perform better than *ZeroR* and in this case the difference between the best performing model – *Ling + R2DE (Q_{Only}) + W2V (Q_{Only})* – and the baseline is greater (almost 15% considering MAE).

Again, *Linguistic* and *Readability*, as well as their hybrid, are outperformed by the other models, even by the Transformer-based ones in this case, showing once again that they are not the best choice for the CKA domain.

About the Transformers, since *ASSISTments* does not provide the text of the possible options, they do not perform as well as on *Cloud Academy*, and indeed they are outperformed by *Word2Vec*, *R2DE*, and their hybrids (except for the nDCG).

Lastly, it is worth noting that, overall, all the models have worse performance on *ASSISTments* than on *Cloud Academy*, and we believe that there are two reasons for this. First of all, *ASSISTments* contains mostly mathematical questions, while *Cloud Academy* contains questions whose text is closer to natural language, possibly making them easier to model. Secondly, we can see from the larger error of the *ZeroR* baseline that the distribution of ground truth difficulties has a larger variance on *ASSISTments* than on *Cloud Academy*, which directly affects the errors made by the models.

Model	MAE	RMSE	nDCG	R2 score
ZeroR	1.1064	1.4881	0.9400	-0.0004
Linguistic	1.0655	1.4431	0.9559	0.0591
Readability	1.0545	1.4381	0.9532	0.0655
R2DE (Q_{Only})	0.9725	1.3234	0.9634	0.2087
W2V (Q_{Only})	0.9688	1.3167	0.9666	0.2166
DistilBERT(Q_{Only})	1.0035	1.3753	0.9674	0.1454
BERT (Q_{Only})	0.9758	1.3352	**0.9676**	0.1946
Ling. + Read.	1.0354	1.4100	0.9562	0.1018
Ling. + Read. + R2DE (Q_{Only})	**0.9481**	**1.2975**	0.9637	**0.2394**
Ling. + W2V (Q_{Only})	0.9561	1.3092	0.9663	0.2257
Ling + R2DE (Q_{Only}) + W2V (Q_{Only})	0.9525	1.3009	**0.9676**	0.2355

Table 5.10: *Evaluation of different models on ASSISTments, comparing the difficulty estimated from text with the target value. We write in bold the best performing models.*

RACE

Table 5.11 presents the evaluation of the models on the *RACE* dataset. In this case, since question difficulty is modeled as a binary variable, we use binary classification metrics for evaluating the models: accuracy, precision, recall, and F1-score.

The *ZeroR* baseline, in this case, predicts for all the questions the majority label (i.e. *high*). *RACE* provides both the text of the question and the text of the answer choices (as well as the reading passage), therefore we can evaluate all the input configuration of the models: Q_{Only}, $Q_{Correct}$, and Q_{All}. However, since *RACE* is a dataset of English reading comprehension questions, we do not perform the additional MLM fine-tuning: indeed, it is Language Assessment and therefore the models do not have the need of improving their domain knowledge (in contrast with *Cloud Academy* which is a CKA dataset).

Observing the models performance, there are several differences with respect to *ASSISTments* and *Cloud Academy*.

First of all, all the *R2DE* configurations are outperformed by *Readability* and *Linguistic* (as well as their hybrid *Ling.* + *Read.*); this makes sense since *R2DE* is based on TF-IDF, which focuses on specific keywords in the question to estimate its difficulty. While this approach seems to work reasonably well in CKA, since specific technical keywords might be an indication of question difficulty, the results here show that it does not work for language assessment. A similar result is visible for all the configurations of *Word2Vec*, which are outperformed by all other models except *R2DE* and *Readability*.

The Transformer based models perform the best – being *BERT* (Q_{All}) the best performing according to F1-score – and, as we observed on *Cloud Academy*, the text of the answer choices is helpful in improving the accuracy of the estimation. A significant difference with respect to *Cloud Academy*, though, is the fact that here the Transformers perform reasonably well even in the Q_{Only} configuration.

Cloud Academy and *ASSISTments*, considering categorical difficulties

Although we mainly consider continuous difficulties – obtained with IRT – for the *Cloud Academy* and *ASSISTments* datasets, we also experiment with categorical difficulties, in order to see if there are any significant differences. In order to do so, we convert the IRT difficulties and the estimated difficulties in *Cloud Academy* and *ASSISTments* as follows: the difficulty values lower than 0 are considered as easy questions while the difficulty

Model	Accuracy	Precision	Recall	F1 score
ZeroR	0.7507	0.7507	1.0000	0.8576
Linguistic	0.8225	0.8141	0.9895	0.8933
Readability	0.7988	0.7940	0.9885	0.8806
R2DE (Q_{Only})	0.7809	0.7753	0.9971	0.8723
R2DE ($Q_{Correct}$)	0.7780	0.7734	0.9962	0.8708
R2DE (Q_{All})	0.7802	0.7747	0.9971	0.8720
W2V (Q_{Only})	0.8060	0.7978	0.9933	0.8849
W2V ($Q_{Correct}$)	0.8089	0.7998	0.9942	0.8865
W2V (Q_{All})	0.8046	0.7966	0.9933	0.8842
DistilBERT (Q_{Only})	0.8585	0.8933	0.9215	0.9072
DistilBERT ($Q_{Correct}$)	0.8491	0.8408	0.9856	0.9075
DistilBERT (Q_{All})	0.8599	0.8536	0.9818	0.9132
BERT (Q_{Only})	0.8570	0.8560	0.9732	0.9109
BERT ($Q_{Correct}$)	0.8599	0.8608	0.9703	0.9123
BERT (Q_{All})	0.8721	0.8853	0.9531	**0.9180**
Ling. + Read.	0.8297	0.8201	0.9904	0.8973
Ling. + Read. + R2DE (Q_{All})	0.8254	0.8162	0.9904	0.8949
Ling. + W2V (Q_{All})	0.8405	0.8305	0.9894	0.9031
Ling + R2DE (Q_{All}) + W2V (Q_{All})	0.8405	0.8316	0.9875	0.9029

Table 5.11: *Evaluation of different models on RACE, comparing the difficulty estimated from text with the target value.*

values higher or equal than 0 are considered as difficult questions. This works as the continuous difficulties estimated with IRT are, by definition, distributed in the range $[-5; +5]$. With this conversion, we obtain two test datasets which are slightly unbalanced: indeed, *Cloud Academy* has 1324 easy questions and 929 difficult questions, while *ASSISTments* has 563 and 698 questions, respectively.

The results for both *Cloud Academy* and *ASSISTments* are shown in Table 5.12, considering the same metrics as RACE and the same models as in the previous sections, and the observations are fairly similar to what we observed when modeling difficulty as a continuous variable. Indeed, *Linguistic* and *Readability* do not perform very well, and the Transformers are the best performing models.

Comparing the classification results with RACE, we can observe that here are much worse, which might suggest that QDET in CKA is, overall, a much more difficult task than in LA.

Model	Accuracy	Precision	Recall	F1 score
Cloud Academy				
Linguistic	0.5470	0.5922	0.5863	0.5892
Readability	0.5533	0.5600	0.9058	0.6920
R2DE (Q_{Only})	0.5921	0.5983	0.8031	0.6857
R2DE ($Q_{Correct}$)	0.5976	0.6057	0.7845	0.6836
R2DE (Q_{All})	0.5929	0.5987	0.8046	0.6865
W2V (Q_{Only})	0.5565	0.5735	0.7789	0.6606
W2V ($Q_{Correct}$)	0.5628	0.5809	0.7574	0.6575
W2V (Q_{All})	0.5826	0.5975	0.7560	0.6675
DistilBERT (Q_{Only})	0.5963	0.6551	0.5716	0.6105
DistilBERT ($Q_{Correct}$)	0.6082	0.6541	0.6203	0.6368
DistilBERT (Q_{All})	0.6415	0.6614	0.7221	0.6904
BERT (Q_{Only})	0.6074	0.6385	0.6705	0.6540
BERT ($Q_{Correct}$)	0.6217	0.6657	0.6361	0.6505
BERT (Q_{All})	0.6344	0.6545	0.7191	0.6853
DistilBERT (Q_{Only}) with MLM	0.6058	0.6907	0.5214	0.5942
DistilBERT ($Q_{Correct}$) with MLM	0.6312	0.6691	0.6604	0.6647
DistilBERT (Q_{All}) with MLM	0.6494	0.6451	0.8152	**0.7202**
BERT (Q_{Only}) with MLM	0.6288	0.6402	0.7521	0.6916
BERT ($Q_{Correct}$) with MLM	0.6590	0.6796	0.7263	0.7022
BERT (Q_{All}) with MLM	0.6502	0.7011	0.6418	0.6702
Ling. + Read.	0.5620	0.5806	0.7546	0.6563
Ling. + Read. + R2DE (Q_{All})	0.6158	0.6242	0.7703	0.6896
Ling. + W2V (Q_{All})	0.5826	0.6074	0.6975	0.6494
Ling + R2DE (Q_{All}) + W2V (Q_{All})	0.6094	0.6239	0.7432	0.6783
ASSISTments				
Linguistic	0.6089	0.5638	0.2282	0.3249
Readability	0.6120	0.5676	0.2486	0.3458
R2DE (Q_{Only})	0.6383	0.6432	0.2755	0.3858
W2V (Q_{Only})	0.6529	0.6432	0.3552	0.4576
DistilBERT(Q_{Only})	0.6657	0.6014	0.5619	0.5810
BERT (Q_{Only})	0.6790	0.6229	0.5618	**0.5908**
Ling. + Read.	0.6196	0.5723	0.3068	0.3994
Ling. + Read. + R2DE (Q_{Only})	0.6454	0.6236	0.3531	0.4509
Ling. + W2V (Q_{Only})	0.6586	0.6389	0.3961	0.4890
Ling + R2DE (Q_{Only}) + W2V (Q_{Only})	0.6613	0.6509	0.3853	0.4841

Table 5.12: *Evaluation of different models on Cloud Academy and ASSISTments, comparing the difficulty estimated from text with the target value, after converting both to binary classes.*

5.7.2 Study of Question Difficulty Distribution

In this subsection we analyze the distribution of the target difficulty and of the difficulty estimated with the models evaluated in the previous subsec-

tion. This is especially relevant on *Cloud Academy* and *ASSISTments*, to understand whether the models really predict question difficulty or rather predict difficulties that are always around the mean.

Cloud Academy

Figure 5.10 presents the distribution of the target IRT difficulty (on the top left corner) and of the difficulties predicted by some of the best performing models for the *Cloud Academy* test dataset. Specifically, we consider (from left to right, top to bottom): *R2DE (Q_{All})*, *BERT ($Q_{Correct}$)*, *BERT ($Q_{Correct}$) with MLM pretraining*, *Ling. + Read. + R2DE (Q_{All})*, and *Ling + R2DE (Q_{All}) + W2V (Q_{All})*.

It is immediately visible that the output difficulty distribution for all the models is more skewed than the target one, meaning that it has lower standard deviation, and this is particularly visible for *R2DE* which mostly predicts difficulties around 0. Still, it is interesting to observe that the output difficult distributions indicate as best performing models the same models (i.e. the two BERT models) which resulted being the best ones from the error analysis as well.

ASSISTments

Figure 5.11 presents contains the same plots but for the test *ASSISTments* dataset, and considering different models. Specifically, from left to right and from top to bottom, it displays: target IRT difficulty, *R2DE (Q_{Only})*, *Word2Vec (Q_{Only})*, *BERT ($Q_{Correct}$)*, *Ling. + Read. + R2DE (Q_{Only})*, and *Ling + R2DE (Q_{Only}) + W2V (Q_{Only})*.

The observations are very similar to what we observed for *Cloud Academy*: the output difficulty distribution of all the models is more skewed than the target one, and *R2DE* is the model that performs worst from this point of view. On the other hand of the scale, *BERT ($Q_{Correct}$)* seems to be the best performing model from this analysis as well, and it even manages to reproduce the small spike for difficulties close to -5, which, interestingly, is present in the output distribution of *R2DE* as well.

5.7.3 Additional Analyses

In this subsection we study how the model performance depends on particular questions characteristics. Specifically, we consider the question type (i.e. *cloze* or *interrogative* in the datasets under consideration), the number of correct answer choices (in *Cloud Academy*), and the question length (using the number of words as indication of the length).

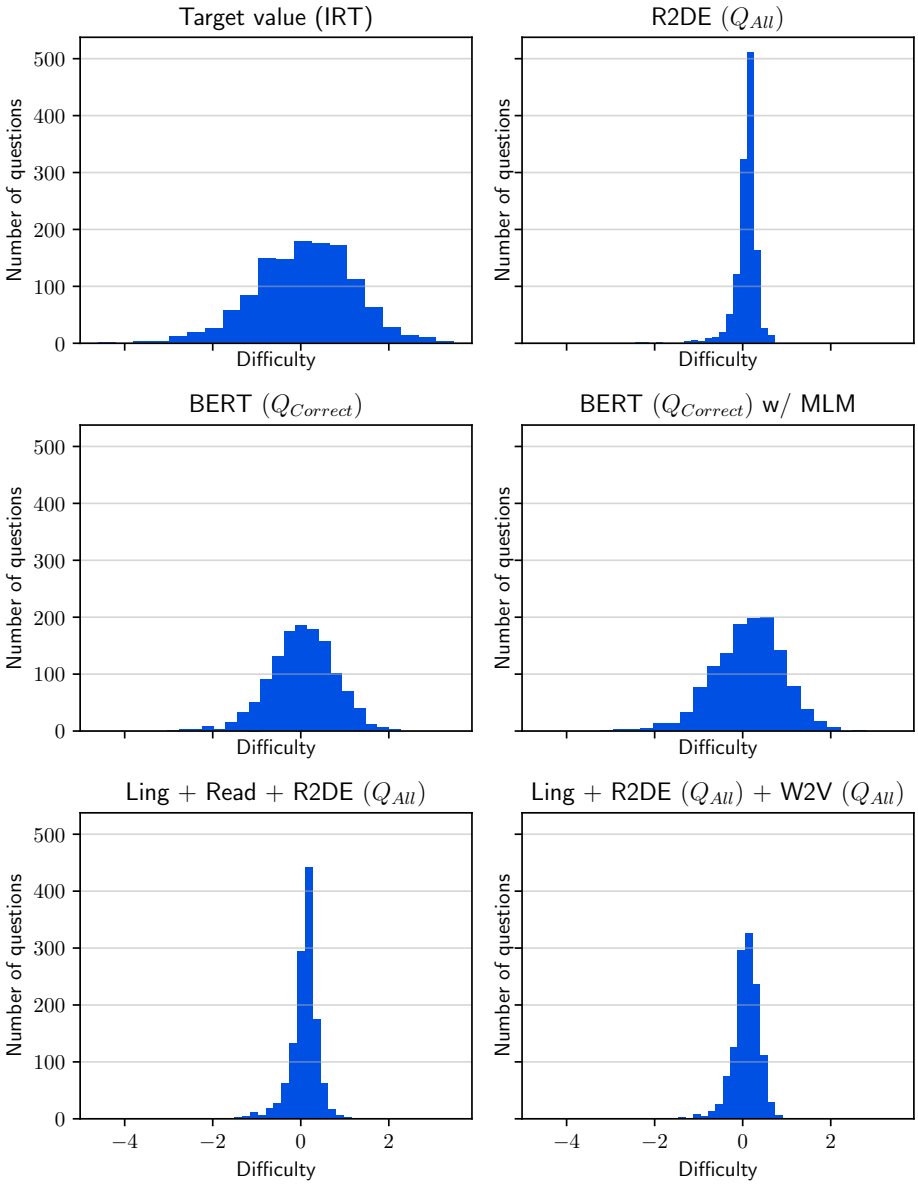

Figure 5.10: *Cloud Academy, distribution of questions by predicted difficulty.*

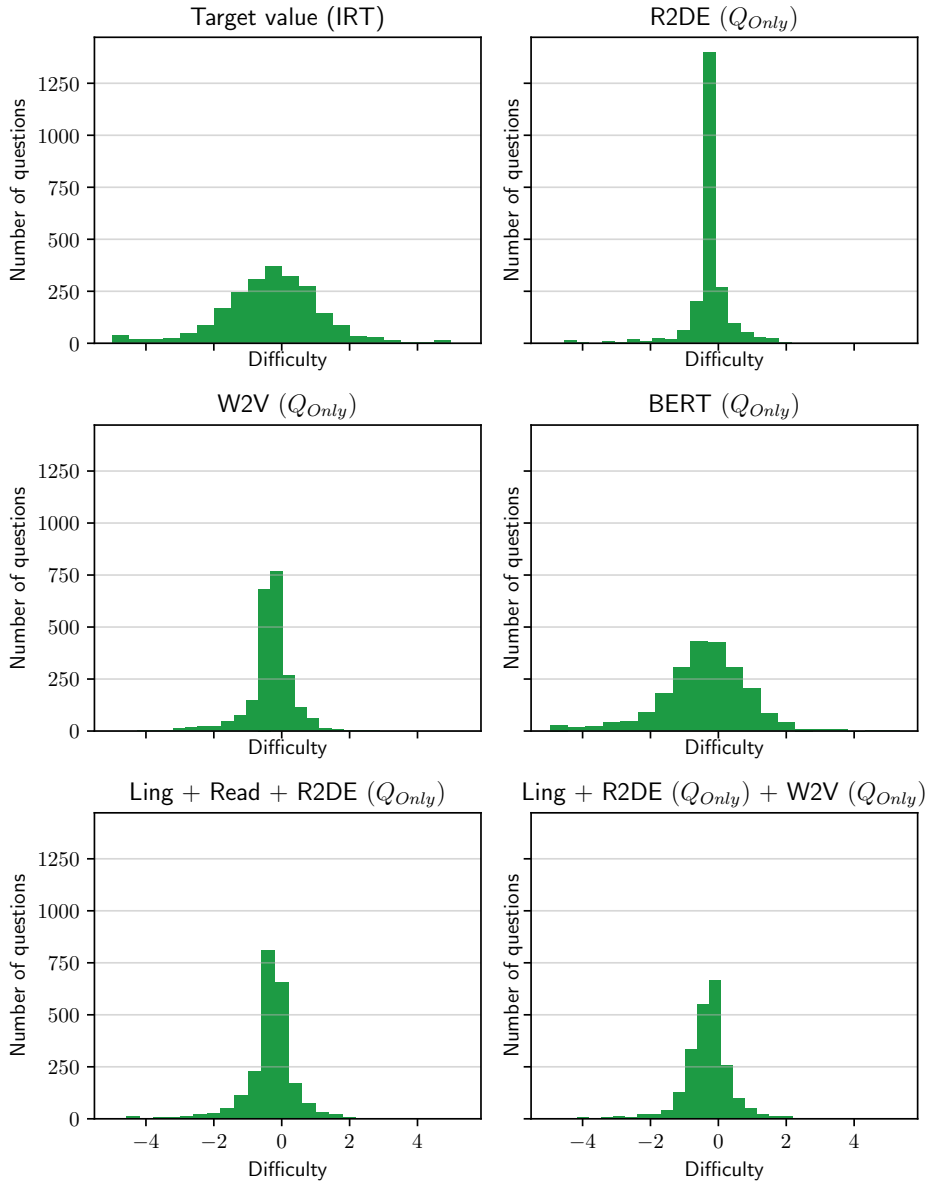

Figure 5.11: *ASSISTments, distribution of questions by predicted difficulty.*

Question type

Table 5.13 presents the evaluation of the models on *Cloud Academy*, separately for *cloze* items and *interrogative* questions; due to the lack of space, we only report MAE and R2 for each model and question type.

The first thing to notice is the fact that the two question types have different mean and standard deviation for the target IRT difficulty, and this is visible from the MAE of the *ZeroR* baseline, which is 0.96 for *cloze* items and 0.87 for *interrogative* items. As a direct consequence of this, all the models have lower MAE on *interrogative* items, with the only exceptions of *DistilBERT ($Q_{Correct}$) with MLM*, which seems to work particularly well on *cloze* items. Indeed, *DistilBERT ($Q_{Correct}$) with MLM* is the best performing model on *cloze* items, while *Ling + R2DE (Q_{All}) + W2V (Q_{All})* is the best performing on *interrogative* questions (closely followed by *BERT ($Q_{Correct}$) with MLM*). We observe that Transformers seem to perform poorly when using only the text of the question (Q_{Only}) on interrogative items, but the same behavior is not visible for cloze items. Also, the text of the answer choices is still useful for all the models that can leverage it ans, as observed in the previous experiments, some models perform best when using only the correct answer(s) while other when using all the possible choices. It is also worth noting that, even though *BERT ($Q_{Correct}$) with MLM* is the best performing model overall, when considering cloze items and interrogative items separately, it is outperformed by other approaches.

Table 5.14 compares the results on *cloze* items and *interrogative* questions for the *ASSISTments* dataset. Contrarily to the previous example, in this case the *ZeroR* baseline performs better on *cloze* items and all the other models perform accordingly, with the exception of BERT and DistilBERT. Indeed, the two Transformer models perform better on *interrogative* items and worse than the *ZeroR* baseline on the cloze items.

Lastly, Table 5.15 presents the results on *cloze* items and *interrogative* items on the *RACE* dataset. In this case there are no significant differences between the two types and, indeed, the differences seem mostly due to the different difficulty distribution between cloze items and interrogative items (which is visible from the *ZeroR* baseline).

Number of correct choices

Table 5.16 presents the MAE and R2 scores on the test set for the different models, separately considering questions with one, two, and three correct answer choices (all the MCQs have four possible options). Due to the lack of space, we only keep two decimal digits for each entry. This analysis

	cloze ($n = 233$)		*interrogative* ($n = 1032$)	
Model	MAE	R2	MAE	R2
ZeroR	0.9616	-0.0183	0.8739	-0.0013
Linguistic	0.9354	0.0218	0.8670	0.0314
Readability	0.9529	0.0008	0.8685	0.0197
R2DE (Q_{Only})	0.9334	0.0363	0.8400	0.0646
R2DE ($Q_{Correct}$)	0.9335	0.0359	0.8378	0.0740
R2DE (Q_{All})	0.9422	0.0238	0.8419	0.0655
W2V (Q_{Only})	0.9583	0.0096	0.8557	0.0422
W2V ($Q_{Correct}$)	0.9463	0.0243	0.8575	0.0345
W2V (Q_{All})	0.9288	0.0430	0.8497	0.0576
DistilBERT (Q_{Only})	1.0016	-0.1084	0.9039	-0.0816
DistilBERT ($Q_{Correct}$)	0.9111	0.0895	0.8533	0.0369
DistilBERT (Q_{All})	0.8537	0.1463	0.8470	0.0453
BERT (Q_{Only})	0.9220	0.0552	0.8837	-0.0363
BERT ($Q_{Correct}$)	0.8710	0.1418	0.8417	0.0933
BERT (Q_{All})	0.8664	0.0987	0.8482	0.0522
DistilBERT (Q_{Only}) with MLM	0.9276	0.0687	0.8571	0.0136
DistilBERT ($Q_{Correct}$) with MLM	**0.8423**	**0.1790**	0.8436	0.0671
DistilBERT (Q_{All}) with MLM	0.8751	0.1081	0.8561	0.0194
BERT (Q_{Only}) with MLM	0.9185	0.0899	0.8899	-0.0306
BERT ($Q_{Correct}$) with MLM	0.8822	0.1432	0.8244	**0.1131**
BERT (Q_{All}) with MLM	0.8735	0.1586	0.8357	0.0602
Ling. + Read.	0.9360	0.0280	0.8641	0.0366
Ling. + Read. + R2DE (Q_{All})	0.8697	0.1279	0.8296	0.1119
Ling. + W2V (Q_{All})	0.9254	0.0629	0.8411	0.0792
Ling + R2DE (Q_{All}) + W2V (Q_{All})	0.8872	0.1185	**0.8228**	0.1101

Table 5.13: *Cloud Academy, MAE and R2 scores on the test set for different types of questions.*

can be performed only on the *Cloud Academy* dataset, as in *ASSISTments* the text of the answer choices is not available and in *RACE* there is always only one correct choice. Since the focus here is on the effect of the number of correct choices, we present the results only for the models that actually leverage the text of the answer options for the prediction.

The three groups are very unbalanced: there are only 92 questions with two correct answer choices and 61 with three correct answer choices, compared to the 1112 questions with one correct choice. However, we can still make some interesting observations.

Indeed, almost all the models perform worse than the *ZeroR* baseline, considering both MAE and R2, on the questions with multiple correct choices. The only exceptions are *R2DE (Q_{All})* and the hybrid models for questions

Model	*cloze* ($n = 135$) MAE	R2	*interrogative* ($n = 2113$) MAE	R2
ZeroR	0.9893	-0.0143	1.1138	-0.0008
Linguistic	0.9648	0.0510	1.0719	0.0585
Readability	0.9738	-0.0015	1.0596	0.0677
R2DE (Q_{Only})	0.9716	0.0156	0.9725	0.2170
W2V (Q_{Only})	0.9285	0.0595	0.9713	0.2232
DistilBERT (Q_{Only})	1.0108	-0.1750	1.003	0.1597
BERT (Q_{Only})	1.0093	-0.1199	0.9736	0.2086
Ling. + Read.	0.9501	0.0621	1.0408	0.1027
Ling. + Read. + R2DE (Q_{Only})	0.9352	0.0257	**0.9489**	**0.2487**
Ling. + W2V (Q_{Only})	**0.8988**	**0.1515**	0.9597	0.2283
Ling + R2DE (Q_{Only}) + W2V (Q_{Only})	0.9053	0.1427	0.9555	0.2390

Table 5.14: *ASSISTments, MAE and R2 scores on the test set for different types of questions.*

Model	*cloze* ($n = 964$) Acc	F1	*interrogative* ($n = 407$) Acc	F1
ZeroR	0.7396	0.8503	0.7690	0.8694
Linguistic	0.8153	0.8876	0.8353	0.9030
Readability	0.7935	0.8763	0.8034	0.8853
R2DE (Q_{Only})	0.7707	0.8654	0.7985	0.8838
R2DE ($Q_{Correct}$)	0.7686	0.8642	0.7936	0.8813
R2DE (Q_{All})	0.7707	0.8654	0.7960	0.8826
W2V (Q_{Only})	0.7966	0.8784	0.8230	0.8962
W2V ($Q_{Correct}$)	0.8008	0.8805	0.8230	0.8965
W2V (Q_{All})	0.7987	0.8795	0.8132	0.8911
DistilBERT (Q_{Only})	0.8620	0.9079	0.8452	0.9014
DistilBERT ($Q_{Correct}$)	0.8485	0.9056	0.8452	0.9077
DistilBERT (Q_{All})	0.8589	0.9113	0.8599	0.9153
BERT (Q_{Only})	0.8547	0.9083	0.8599	0.9142
BERT ($Q_{Correct}$)	0.8599	0.9105	0.8574	0.9139
BERT (Q_{All})	0.8703	**0.9154**	0.8722	**0.9202**
Ling. + Read.	0.8215	0.8911	0.8427	0.9069
Ling. + Read. + R2DE (Q_{All})	0.8163	0.8883	0.8402	0.9056
Ling. + W2V (Q_{All})	0.8350	0.8982	0.8476	0.9098
Ling + R2DE (Q_{All}) + W2V (Q_{All})	0.8360	0.8985	0.8476	0.9098

Table 5.15: *RACE, Accuracy and F1 score on the test set for different types of questions. The RACE dataset also contains some title questions, such as "What is the most appropriate title for the reading passage?"; we do not include them here as there are only 21 of them in the test dataset and therefore the findings would not be reliable.*

Model	1 correct ($n = 1112$) MAE	R2	2 correct ($n = 92$) MAE	R2	3 correct ($n = 61$) MAE	R2
ZeroR	0.92	0.00	0.70	-0.02	0.58	0.00
R2DE ($Q_{Correct}$)	0.88	0.08	0.70	-0.04	0.58	-0.05
R2DE (Q_{All})	0.89	0.07	0.69	-0.02	0.59	-0.04
W2V ($Q_{Correct}$)	0.90	0.04	0.71	-0.06	0.59	0.00
W2V (Q_{All})	0.89	0.06	0.70	-0.03	0.58	0.01
DistilBERT ($Q_{Correct}$)	0.89	0.07	0.74	-0.27	0.65	-0.23
DistilBERT (Q_{All})	0.87	0.09	0.75	-0.27	0.62	-0.33
BERT ($Q_{Correct}$)	0.87	0.12	0.70	-0.11	0.59	0.00
BERT (Q_{All})	0.87	0.09	0.71	-0.12	0.73	-0.67
DistilBERT ($Q_{Correct}$) with MLM	0.86	0.12	0.75	-0.22	0.63	-0.28
DistilBERT (Q_{All}) with MLM	0.88	0.07	0.76	-0.29	0.67	-0.46
BERT ($Q_{Correct}$) with MLM	**0.85**	**0.14**	0.68	-0.09	0.64	-0.23
BERT (Q_{All}) with MLM	0.86	0.10	0.70	-0.17	0.64	-0.24
Ling. + Read. + R2DE (Q_{All})	0.86	0.13	0.67	**0.07**	0.61	-0.09
Ling. + W2V (Q_{All})	0.89	0.08	0.69	0.02	**0.57**	**0.06**
Ling + R2DE (Q_{All}) + W2V (Q_{All})	0.86	0.12	**0.66**	0.06	0.59	-0.01

Table 5.16: *Cloud Academy, MAE and R2 of different models on questions with different number of correct choices.*

with two correct choices and *W2V (Q_{All})* and *Ling. + W2V (Q_{All})* for the items with three correct choices. Interestingly, we observe this issue both for models that use the text position for the prediction (i.e. Transformers) and models that do not take into consideration the context, such as *Word2Vec* and *R2DE*. The reason for this is not clear and further research should focus on it, especially considering the limited number of test questions with multiple correct choices that are available in this dataset.

Question length

The last analysis we perform in this Chapter is a study of the correlation between question length and the model accuracy, which is shown in Figure 5.12 and 5.13. In both figures, the top-left plot represents the distribution of questions per question length and difficulty, and the support (i.e. number of questions) in each bin is represented by its color, with darker colors indicating a lower amount of questions. The other five plots in each figure represent the error of the models depending on question length and target difficulty, with darker colors indicating lower errors.

The models considered for the two datasets are different, and are the same ones considered in the previous analysis of the output difficulty dis-

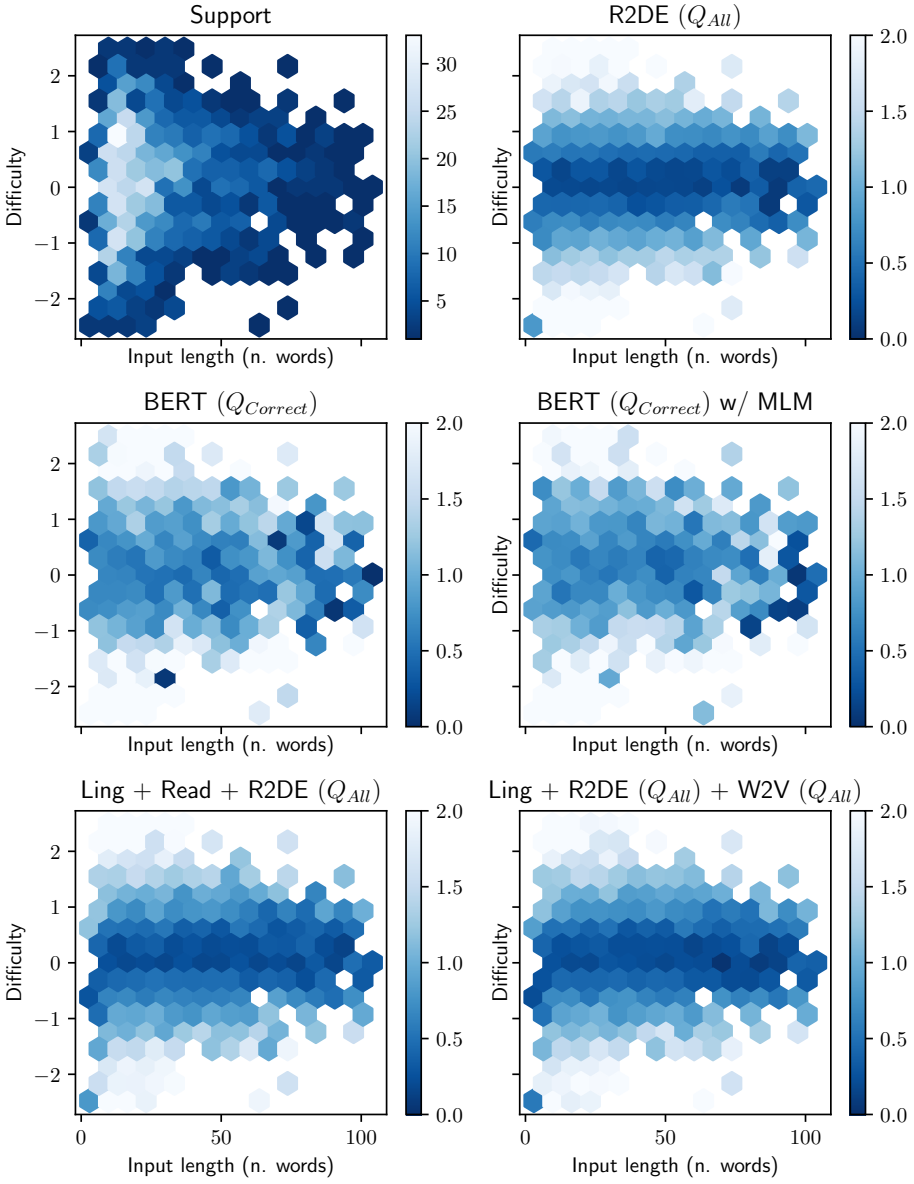

Figure 5.12: *Cloud Academy, MAE distribution by question length and difficulty.*

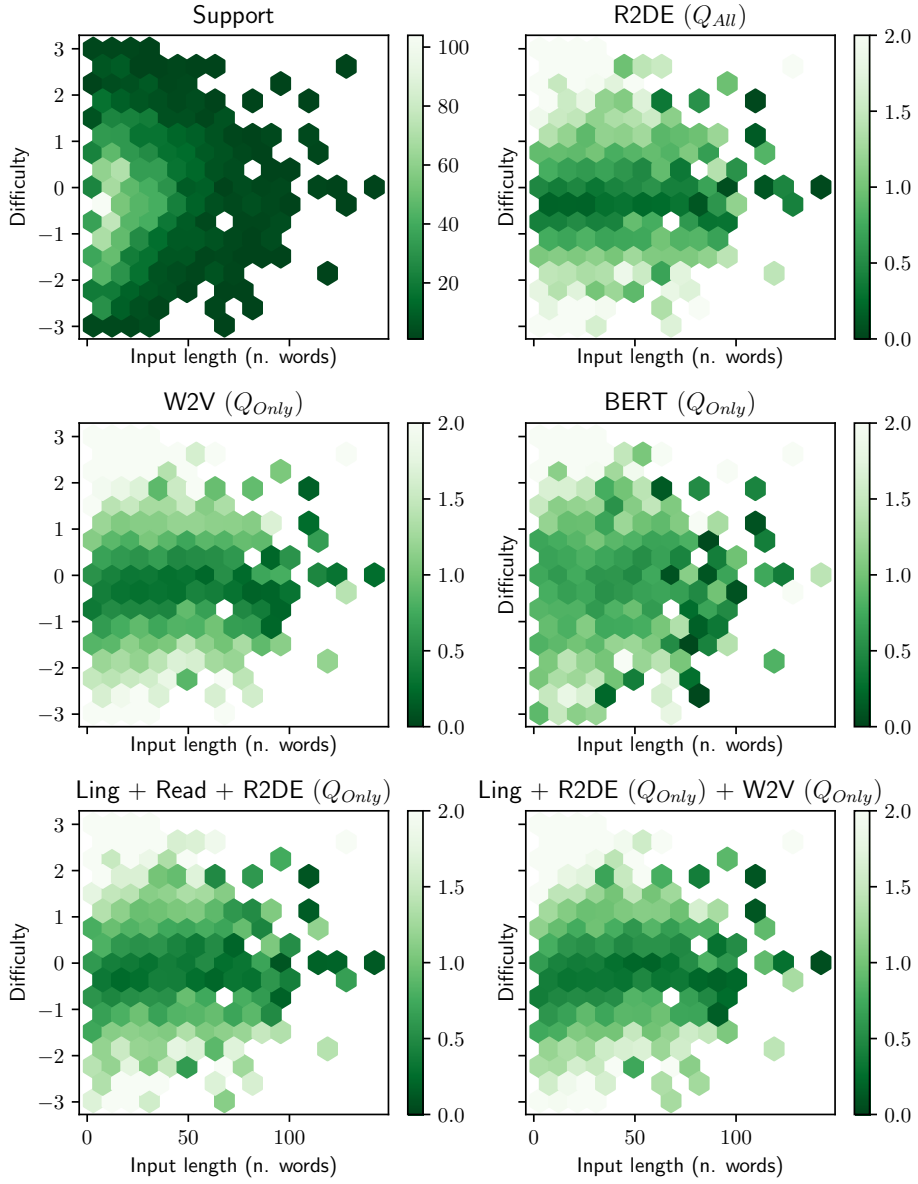

Figure 5.13: *ASSISTments, MAE distribution by question length and difficulty.*

tribution. Specifically, for *Cloud Academy* we consider (from left to right, top to bottom): *R2DE (Q_{All})*, *BERT ($Q_{Correct}$)*, *BERT ($Q_{Correct}$) with MLM pretraining*, *Ling. + Read. + R2DE (Q_{All})*, and *Ling + R2DE (Q_{All}) + W2V (Q_{All})*. For *ASSISTments*, instead, we consider (from left to right and from top to bottom): *R2DE (Q_{Only})*, *Word2Vec (Q_{Only})*, *BERT ($Q_{Correct}$)*, *Ling. + Read. + R2DE (Q_{Only})*, and *Ling + R2DE (Q_{Only}) + W2V (Q_{Only})*.

Starting from Figure 5.12, we can see that the error heavily depends on the target difficulty, with lower errors for questions whose real difficulty is closer to 0. This is partially true for all the five models shown in the plot, but for the two Transformer-based models it is less visible than for the others, showing once again that they are more capable of predicting question difficulty. Considering the relations between question length and estimation error, there are no clear correlations. However, in some cases, it seems that the models perform slightly better for longer questions, for instance looking at the line corresponding to $\texttt{Difficulty} = -0.3$ for *BERT ($Q_{Correct}$)* and *Ling. + Read. + R2DE (Q_{All})*.

Moving to Figure 5.13, which presents the same information for the *ASSISTments* dataset, we can see that the situation is fairly similar: indeed, the error seems to depend mostly on the target difficulty and *BERT (Q_{Only})* is the model that is the least affected by this; indeed, the color (which indicates the error) is fairly constant across different values of target difficulty. In this case, however, it is more visible the fact that the models generally are more accurate for longer questions: indeed, we can see that questions with longer texts (i.e. the ones towards the right end of the plots) generally have darker colors (indicating lower errors), and this is true for all the models, even though more visible for *BERT (Q_{Only})*.

5.8 Conclusions

In this Chapter we have analyzed several recently proposed approaches to supervised QDET, considering three datasets from different domains: the Language Assessment (LA) RACE dataset and the Content Knowledge Assessment (CKA) datasets *Cloud Academy* and *ASSISTments*.

We observed that the accuracy of the models depends on the type of the questions under consideration and, most importantly, on the educational domain. Specifically, we observe that in the LA domain – at least considering reading comprehension questions – readability indexes and linguistic features can capture most of question difficulty. Indeed, in LA, and particularly in reading comprehension questions, the item difficulty heavily depends on the linguistic demands of the reading passage, therefore techniques which

can capture this information are capable of an accurate difficulty estimation. Still, the same linguistic demands are captured – even more accurately – by more advanced models, such as BERT, which are generally capable of better performance. In this sense, we argue that Transformer-based models are probably the better choice from an accuracy point of view, but much simpler models, such as the ones based on readability indexes, might still be a reasonable choice in case of constraints from the computational point of view.

Unfortunately, this is not true for the CKA domain. Indeed, supervised models based on simple techniques such as linguistic features and readability indexes are not capable of accurately capturing the demands of exam questions and therefore lead to inaccurate estimations. This is because in CKA the question difficulty mostly depends on the specific topics which are being assessed by the question, and techniques that focus on language only cannot capture such information. Specifically, we observed that the Transformers are, again, the models that generally lead to the best performance for supervised QDET in the CKA domain, and are in some cases matched by techniques based on word embeddings (such as Word2Vec) or frequency based features (such as TF-IDF). In addition to that, Transformers can be pre-trained on additional documents related to the same topics as the ones assessed by the questions, and this increases their accuracy.

Also, we observed that – in the case of MCQ – having access to the text of the possible choices generally leads to improved results for the methods that can leverage it (Transformers, word embeddingd, and frequency based features), but there are some interesting observations. Indeed, we observed that, possibly due to the specific encoding we use, almost all the models are not capable of modeling questions which have multiple correct choices (i.e. they require the student to select all the correct choices)

Also, we observed that generally Transformers model lead to better results for interrogative questions rather than cloze items and the best overall performing model (*BERT ($Q_{Correct}$) with MLM*), was outperformed on both cloze items and interrogative items by other models. This suggests that, probably, training different models on specific types of questions and using those models for the QDET might lead to better results.

CHAPTER 6

Unsupervised Question Difficulty Estimation from Text

In this Chapter we focus on the task on unsupervised Question Difficulty Estimation from Text (QDET). First, we give an introduction to the task (§6.1), and present previous literature, which is fairly limited compared to the literature about supervised QDET (§6.2). Then we describe the details of the models that we experiment on, including a novel approach which was not proposed in previous research (§6.3). The two experimental datasets – one being publicly available and one being privately owned – and the experimental setup are described in §6.4. Finally, we present and discuss the experimental results in §6.5 and conclude this Chapter with §6.6.

6.1 Introduction

The task of unsupervised QDET consists in estimating the question difficulty in an unsupervised manner using only question text as input for the estimation. In this sense, it is important to distinguish it from unsupervised techniques that leverage other types of information. For instance, Item Response Theory performs QDE in an unsupervised manner – indeed,

pretesting itself consists in performing QDE in an unsupervised manner – but it does so by leveraging a log of students' answers, which is not available for newly created questions. On the contrary, the text of the question and (possibly) of the possible answer choices is always available at the time of question creation, both for manual and automated question generation, therefore unsupervised QDET can be applied to newly created assessment items.

In the previous Chapter, we have seen that the task of supervised QDET received a significant research interest as it is one of possible approaches to overcome the limitations of the traditional approaches to question calibration. Indeed, once the trained model is available, it can be used to infer the difficulty of new questions from their text, overcoming (or at least reducing) the need for pretesting and manual calibration. However, all the supervised techniques still have two major limitations: i) they require thousands of calibrated questions as a training set, and ii) they cannot perform cross-domain QDE. In other words, the training questions must assess the same topics as the new questions which the model will later be used on, limiting its applicability (e.g. when creating questions related to new courses). Crucially, these limitations are intrinsic of such approaches and cannot be addressed by just improving the estimation accuracy of the models.

Unsupervised QDET targets both limitations, and previous research was carried out along two main directions. Some works based difficulty estimation on deterministic metrics such as the readability of the questions or the similarity between the correct choice and the distractors. Others, more recently, trained Question Answering (QA) models to answer exam questions and leverage the answers of such models and their uncertainties to perform question calibration. In this Chapter, we evaluate previously proposed approaches on two real world datasets: the privately owned *Cloud Academy*, and the publicly available *RACE*. On top of studying how different approaches perform on questions from diverse domains, we also propose a novel technique and compare it with previous research. This new approach takes inspiration from IRT and its usage as a way to evaluate the "skill" of classification models and the difficulty of classification tasks (more precisely, the difficulty of different input samples which have to be classified). In practice, we train several QA models to answer the questions under calibration and instead of using their uncertainty as a proxy of question difficulty, we observe the correctness of their answers and train an IRT model on such information, as if the QA models where "students" trying to answer the questions. Finally, the question difficulty obtained in this manner can directly be used as a proxy of human perceived difficulty.

Our experimental results show that, in the LA domain, readability indexes might be sufficient to have an intuition of question difficulty, while in the CKA domain more complex models are needed. Specifically, the new approach we evaluate in this Chapter leads to the better results on *Cloud Academy*.

6.2 Related works

The literature on unsupervised QDET is fairly limited, if compared to the supervised formulation of the same task. As anticipated in the introduction to this Chapter, previous research can be categorized in two groups: i) works that base difficulty estimation deterministic measures such as readability indexes and similarity measures, and ii) research that leveraged Question Answering (QA) models.

6.2.1 Readability Indexes and Similarity Measures

As we have seen in the previous Chapter, readability indexes and similarity measures are sometimes used as input features to a model that performs QDET in a supervised manner. However, they can also be directly used for unsupervised QDET, by defining the difficulty as a function of readability and/or similarity.

In previous literature, there is only one work that models difficulty as a function of readability indexes [56]. Specifically, the authors deal with reading comprehension questions and directly define the difficulty as the readability of the reading passage, without using any testing theories. In the scenario under study in the paper – English reading comprehension questions – the difficulty estimated with this approach correlates with the actual students' performance (i.e. more difficult questions are answered with lower accuracy), suggesting that indeed it is a good indication of question difficulty. Still, it is worth noting that readability indexes cannot model, by definition, domain knowledge and, therefore, they are likely to lead to inaccurate results in the CKA domain, as we will see in this Chapter.

The other approach consists in using similarity measures, as in [54]. Specifically, three aspects can be taken into consideration as good indicators of question difficulty:

- the semantic similarity between the question and the correct answer;
- the semantic similarity between the question (or the true answer) and the reading passage, in the case of reading comprehension questions;

- the semantic similarity between the correct choice and the distractors, in the case of MCQs.

In the first two cases, the highest the similarity the easier the question, while in the third example, higher similarity leads to more difficult questions. Previous research mostly experimented with Word2Vec embeddings and cosine similarity, and the accuracy of this technique for estimating question difficulty heavily depends on the accuracy of the embedding technique used to capture the semantic meaning of the question/answer/passage and on the similarity measure that is used.

6.2.2 Question Answering Models

In previous literature, the main alternative to readability indexes and similarity measures consisted in leveraging the answers of Question Answering (QA) models. All the research carried out along this direction is based on the idea that there could be a relation between the human perceived difficulty – which is the final target in QDE – and machine perceived difficulty; therefore, being able to accurately measure machine difficulty, we might have an accurate estimate of human perceived difficulty as well.

The first paper which hypothesized such relation between the two difficulties is [123], which experimented with an Information Retrieval based QA system. However, the authors do not use the QA model directly for QDE but consider its scores as features which are given as input to a Random Forest regressor that is trained in a supervised manner, thus without experimenting on the direct usage of QA scores for unsupervised QDE.

In a previous work [75], we made that missing step and experimented on using QA models for estimating in an unsupervised manner the difficulty of assessment items, specifically using calibrated Transformers. The idea of the paper was to train a calibrated QA Transformer model to answer the exam question we wanted to calibrate, and leverage the uncertainty of the model – which is an indicator of model-perceived question difficulty – as a proxy of human perceived difficulty[1]. Calibrating the QA model was done in order to have a reliable measure of its uncertainty since, by definition, a model is calibrated only if the posterior probabilities (i.e. uncertainty of the model) are aligned with the empirical likelihood (i.e. the accuracy of the model in the QA task). Experiments on the *RACE* dataset, using the

[1]It is important to remark here that, in this Chapter, the word "calibration" might refer to one of two things. On one hand, we can refer to *question calibration*, which consists in estimating a value representing question difficulty. On the other, we might refer to *model calibration*, which means aligning the posterior probabilities with the empirical likelihoods [43].

level field as an indication of question difficulty, showed that indeed the uncertainty of such models is a reasonable indicator of question difficulty and it is more reliable than other information retrieval-based techniques, even though it is not accurate enough to be directly used to assign a difficulty to exam questions.

Other relevant research [67,68] experimented with IRT on machine learning models and observed that there is indeed a positive correlation between the difficulty estimated using the models answers and the difficulty estimated using human answers. However, the authors experiment on two tasks which are not related to the educational domain and, most importantly, do not involve questions; therefore, the question of whether this correlation holds true in education as well is still unanswered. Indeed, they experimented with the tasks of sentiment analysis and textual entailment, whose difficulty derives from sources that are very different from the ones that affect the difficulty of educational questions. In this sense, these two papers fit more neatly into the previous research on the evaluation of the performance of machine learning models with IRT [20], rather that unsupervised QDET, but are still relevant for us since they provide another hint that using IRT on machine learning models might indeed lead to an accurate estimation of human-perceived difficulty.

6.3 Models

In this Section, we describe the models used in the experiments from this Chapter. Specifically, we experiment with several models based on fours approaches: a readability-based approach (§6.3.1), a similarity based approach (§6.3.2), an approach based on the score variance of calibrated QA models (§6.3.3) and, lastly, the novel approach using IRT on QA models (§6.3.4).

6.3.1 Readability

This approach is arguably the simplest. Indeed, it consists in measuring the readability of the question under consideration with one of the readily available indexes, and using that value as an estimation of the difficulty.

We experiment with five models built with this approach, and they differ for the readability index that is used. Specifically, we consider the same readability indexes used for supervised QDET.

- The *Flesch Reading Ease* [36] gives a text a score between 1 and 100, with 100 being the highest readability score; it is computed from the

average number of words per sentence and the average number of syllables per word combined using precise constants and coefficients.

- The *Flesch-Kincaid Grade Level* [62] approximates the reading grade level of a text and it is very similar to the Flesch Reading Ease, as it uses again the average number of words per sentence and the average number of syllables per word but different constants and coefficients.
- *Automated Readability Index* (ARI) [100] assesses the U.S. grade level required to read a piece of text; it is computed from the average number of characters per word and the average number of words per sentence.
- The *Gunning FOG Index* [42] generates a grade level between 0 and 20, which estimates the education level required to understand the text; it is computed from the average number of words per sentence and the average number of *complex* words per sentence, complex words being the ones containing three or more syllables.
- *Coleman-Liau Index* [23] is a readability formula which shows the reading level of a text; it uses number of sentences and number of letters as variables.

The *Flesch Reading Ease* indicates how easy a document is to read, therefore higher values are interpreted as an indication of lower difficulties, and we define the difficulty estimated with this approach as *1 - ease*. For all the other indexes, higher values are interpreted as indication of the higher difficulties, and we directly use the value obtained from the indexes as an indication of question difficulty.

6.3.2 Similarity

We experiment with two models based on similarity, considering i) the similarity between the question and the correct answer, and ii) the similarity between the correct answer and the distractors. More precisely, we use Word2Vec embeddings to embed the question and the answer choices, and cosine similarity to measure their similarity.

When considering the similarity between the question and the correct choice, we assume that higher similarities indicate lower difficulty, and define the difficulty as *1 - similarity*. On the other hand, when considering the similarity between the correct choice and the distractors, we assume that higher similarities indicate higher difficulty, and use the value of the similarity as the direct indication of question difficulty.

In both cases, it might happen that several answer options have to be merged into a unique vector, in case of multiple correct choices or (most frequently) in case of multiple distractors. In these cases, we do so by averaging the embeddings of each answer option.

6.3.3 Score Variance of QA Models

This is the approach we proposed in [75], and it consists in leveraging the confidence of a QA model for estimating question difficulty. Specifically, when used on MCQ, it is based on the softmax scores produced by the QA model over the possible options, which indicate the probability – according to the model – of each option being the correct one. The raw softmax scores are then converted into a single numerical value by computing their variance, and assuming that larger values of variance indicate easier questions (since they indicate that the model is more certain in the estimation). The difficulty is then modeled as *1 - variance*[2].

We experiment with this approach by implementing QA models based on two Transformer architectures: i) BERT [26], and ii) DistilBERT [97]. In order to reduce miscalibration, we use the ensembling technique suggested in [66] and proceed as follows, separately for each architecture.

1. We train five instances of the architecture, and each instance is trained on the entire training dataset (randomly shuffled), with a different random initialization.

2. We pick the three best performing instances, considering the test accuracy on the QA task.

3. We build the ensembles by averaging (separately for each question) the softmax scores produced by the three instances so that each of the four answer options is assigned a single score from 0 to 1. These scores indicate the probability (according to the ensemble model) of each option being correct.

We build an ensemble for both architectures and also an "hybrid" ensemble, which is obtained with the same steps but averaging the prediction of the three best instances BERT and the three best instances of DistilBERT.

The approach originally proposed in [75] was based on calibrated ensembles but it can be used on the single instances as well. Therefore, in this

[2] In the paper, we also experimented with some alternatives, but they were outperformed by the score variance, therefore we do not consider them in this thesis. Specifically, we evaluated i) keeping only the highest softmax score, and ii) computing the difference between the highest and the second-highest softmax score, assuming in these cases as well that larger values indicate easier questions.

thesis, we also evaluate the difficulties estimated with the single instances to analyze how they compare with the calibrated ensembles.

Thus, in total, we experiment with nine models based on this approach, and they differ for the QA model whose scores are considered for difficulty estimation: the BERT ensemble, the DistilBERT ensemble, the hybrid BERT-DistilBERT ensemble, three single BERT instance, and three single DistilBERT instances.

6.3.4 IRT on QA models

This is the new approach that we experiment with in this Chapter. It consists in using the answers of several QA models to train a one-parameter IRT model that estimates question difficulty, basically considering each QA model as a student taking the exam and using IRT as it would be used on regular students.

The advantage of this approach, compared to the ones presented above, is that it is the only one that produces difficulty values which are on the desired IRT range without needing any rescaling.

In this thesis, we evaluate this approach by using the same QA models that are used for implementing the approach based on the score variance of QA models: five instances of BERT and five instances of DistilBERT. We do not consider the answers of the ensembles as they are obtained by averaging the responses of the single models and therefore violate the independence assumption of IRT.

In practice, we consider nine models based on this approach, and they differ in the QA models that are used for the IRT estimation (BERT, DistilBERT, or both), and the questions which are calibrated with IRT (train questions, test questions, or both).

6.4 Experimental setup

In this section, we describe the experimental setup used for the experiments presented in the next section. In §6.4.1 we describe the experimental dataset that are used in this Chapter, focusing on the aspects that are specific to unsupervised QDET. In §6.4.2 we present the setup for training and evaluating the QA models, which are used by two of the four approaches evaluated in this Chapter. Lastly, in §6.4.3 we describe the setup used for the evaluation of QDET.

6.4.1 Experimental datasets

In this Chapter, we experiment on *Cloud Academy* and *RACE* only. Indeed, the *ASSISTments* data collection does not provide the text of the possible answers and therefore it is not possible to train the approaches that are based on the predictions of QA models.

Cloud Academy

We use two of the datasets available in the *Cloud Academy* data collection: *Cloud Academy_A* and *Cloud Academy_Q*.

The first one of the two is used only to obtain the gold standard IRT difficulty of the questions under calibration, as we did in the experiments on supervised QDET.

Cloud Academy_Q, on the other hand, is used to train the QA models and to estimate the difficulty with the unsupervised techniques. We use the same split as in the previous Chapter: 80% of the questions are used to train the QA models and the word embeddings, and 20% of them to test them.

Starting from the test portion of *Cloud Academy_Q*, we also prepare another dataset, containing pairs of questions and a label indicating which question of the pair is more difficult (according to the gold standard IRT estimation). This was done in order to evaluate how well the unsupervised models for QDET can estimate the relative difficulty of pairs of questions (i.e. which question of the pair is more difficult).

RACE

RACE is a dataset of English reading comprehension questions from middle and high school exams; all questions are MCQ with four possible choices.

The publicly available *RACE* dataset is already split into *train*, *dev*, and *test* set: we use the train set and development set to train the QA models, and the test set to evaluate them in the QA task. All the questions in *RACE* are assigned a difficulty label, which is used to evaluate – a posteriori – the models accuracy.

Starting from the original *RACE* dataset, we also build a smaller dataset (*PairRACE_HM*), which contains a list of pairs of questions and a label indicating which question of the pair is more difficult, similarly to what we do for *Cloud Academy*.

PairRACE_HM is built from the original *RACE* dataset using the *level* label, which indicates the level of examination (high or middle school) of the question. Specifically, we prepare 2,062,096 pairs of questions, each

one containing one *middle* question and one *high* question. In all the pairs in this dataset, the two questions are related to different reading passages.

While the level is not directly a numerical estimation of the difficulty of a question itself, the authors point to the "drastic difficulty gap" between the two levels, and they also give evidence for "higher difficulty of high school examinations" [65]. Therefore, we use these labels as indication of the question difficulty level when evaluating the models on the task of unsupervised QDET. We remark that the unsupervised models do not see the labels associated with the questions at training time nor try to predict the label directly. Instead, they are evaluated on the task of Pairwise Difficulty Prediction: given a pair of question, we check whether the model labels the *high* question as being the more difficult one.

6.4.2 Training and evaluating the QA models

Two of the approaches evaluated in this Chapter for unsupervised QDET are based on QA models, therefore the first step towards implementing them consists of training and evaluating the QA models.

Considering that we are dealing with Transformers, the training is fairly straightforward. Indeed, we start from the pre-trained BERT and DistilBERT models, and fine-tune them for the task of question answering, given the text of i) the question, ii) the possible answer choices, and – for RACE – iii) the reading passage. In practice, this is done by adding a multiple-choice classification layer on top of each original model and training the network using a cross-entropy loss. Given that our main focus is not obtaining the best performing QA model on the two datasets, but rather have a variety of models to better capture and estimate question difficulty, we use different numbers of epochs and different hyper-parameters (learning rate, weight decay, and Adam epsilon) at training time. Specifically, we train five instances for each Transformer architecture, using different random initializations, different hyper-parameters, and randomly shuffling the training set, and evaluate them on the held-out test set.

As for the evaluation, we both consider the QA accuracy and the Expected Calibration Error, and compute them on both the QA test set and the QA train set.

Differently from the experimental setup in the previous Chapter on supervised QDET, here we do not perform the additional pre-training for the *Cloud Academy* dataset; and leave that exploration for future research.

An important point about the approaches based on QA models is the split used in the evaluation. We divide the experimental dataset used in this study

(*RACE* and *Cloud Academy_Q*) into a train set (80%) and a test set (20%), but this separation is needed to train and evaluate the QA models, and is not related (nor required) by the actual difficulty estimation. Therefore, in order to better understand how all the considered techniques perform, we evaluate them both on the QA train set and the QA test set.

6.4.3 Evaluating unsupervised QDET

Both datasets provide the "true" difficulty of the questions, and we leverage such information for evaluating the approaches to unsupervised QDET.

Our goal here is to investigate i) whether the machine estimations lead to a notion of difficulty that aligns well with the human one and ii) whether it can be useful in practical applications when logs of answers or calibrated questions are not available for training.

Pairwise Difficulty Prediction

First of all, since the various approaches produce difficulties on different scales and with different distributions, we evaluate their accuracy on the task of Pairwise Difficulty Prediction (PDP). Given a list of question pairs, PDP consists in evaluating whether the various approaches assign to the more difficult question a higher difficulty than to the easier question. In order to numerically evaluate model performance, we use the accuracy on the PDP task, which represents the fraction of question pairs in which the difficulty relation was labeled correctly. In order to do this, we use the two datasets of questions pairs described above.

We use this approach because the difficulties estimated with the various models produce difficulties on different scales (both different between them and different from the target scales of the experimental datasets), and rescaling them to a common scale is not always straightforward. The PDP task, on the other hand, is not affected by the fact that the target difficulties and the difficulties generated by each model are on different scales and therefore is a reasonable candidate for evaluating the models.

Ranking evaluation

Whilst the question in the RACE dataset are labeled as being either *middle* or *high*, the target difficulties of the questions in *Cloud Academy* are obtained from an IRT model trained on real students' answers and therefore have continuous values spreading along the range $[-5; +5]$.

Therefore, we can compare the target difficulty ranking with the difficulty ranking obtained with each approach. In order to do so, we rescale all

the difficulties (both estimated with the proposed approaches and the target values) to the range $[0; 1]$ and evaluate the estimations with nDCG, which is a commonly used metric for ranking evaluation.

6.5 Results

In this section, we present the results of all the models that are evaluated on the task of unsupervised QDET. First of all, we perform a preliminary evaluation of the Question Answering (QA) models, both measuring their QA accuracy and their calibration (§6.5.1). Then, we perform the evaluation on the pairwise difficulty prediction task (§6.5.2). Lastly, we show the results of the ranking evaluation (§6.5.3), and of the study of the output difficulty distribution, to see how it compares with the target difficulty distribution (§6.5.4).

6.5.1 Evaluating QA accuracy and model calibration

We present here the results of a preliminary analysis to evaluate the accuracy and the calibration of the QA models used to build two of the approaches evaluated in the rest of this section. Table 6.1 presents, for both datasets, the accuracy and the calibration of the QA models, both considering the single instances (the three best performing ones) and the ensembles, indicated by "E". For evaluating calibration, we use Expected Calibration Error (ECE).

Comparing the performance across the two datasets, it is clear that the average QA accuracy on *RACE* is much better than on *Cloud Academy*, suggesting that it is an easier dataset for the QA models. The fact that we did not any additional pre-training for *Cloud Academy* most likely has an influence on this, as that could have been a way to positively affect the performance of the model by providing some additional domain knowledge. Still, even on *Cloud Academy*, the accuracy is much better than random guessing (25%). On the other hand, focusing on model calibration, the models on *Cloud Academy* are generally better calibrated that on *RACE*, meaning that they have worse performance but are aware of that.

Focusing on the effects of the ensembles, they seem to bring more value on *RACE* than on *Cloud Academy*, both considering QA accuracy and calibration error. Indeed, in the right hand side of the table we can see that the ECE of the ensemble models on RACE is lower than all the single models, even for *BERT (E)* which is the most accurate model both on training set and on test set.

	Cloud Academy			*RACE*		
	QA Accuracy		ECE	QA Accuracy		ECE
Model	train	test	test	train	test	test
BERT	**0.4090**	**0.3752**	0.0398	0.8362	0.6184	0.1436
	0.3872	0.3476	0.0288	0.8309	0.6208	0.1393
	0.3824	0.3652	0.0459	0.8332	0.6232	0.1370
DistilBERT	0.3976	0.3526	0.0217	0.6216	0.4585	0.1042
	0.3600	0.3534	0.0353	0.6570	0.4593	0.1271
	0.3738	0.3618	0.0135	0.6803	0.4759	0.1354
BERT (E)	0.3956	**0.3752**	0.0377	**0.8531**	**0.6344**	0.0976
DistilBERT (E)	0.3794	0.3660	0.0214	0.6752	0.4743	0.0987
BERT-DistilBERT (E)	0.3942	0.3727	**0.0127**	0.8481	0.6178	**0.0274**

Table 6.1: *Evaluation of accuracy and calibration of the Transformer-based Question Answering models used for QDE from text, considering both the single instances (three for each architecture) and the ensembles (indicated by "E"). The three single instance of each architecture differ for random initialization, number of epochs, and hyper-parameters. We highlight in bold the best performing models.*

Considering *Cloud Academy*, the effect of the calibration is particularly visible for the hybrid ensemble, as it is the one with the lowest ECE. We believe that the decrese for the other ensembles is not as visible as on RACE because the ECE of the single instances is fairly low and thus more difficult to reduce.

6.5.2 Evaluating QDET on the Pairwise Difficulty Prediction Task

In this subsection we evaluate the approaches to QDET on the task of Pairwise Difficulty Prediction (PDP), starting from *PairRACE_HM* e then moving on to *Cloud Academy*.

PairRACE_HM

The results of all the approaches on the PDP task on *PairRACE_HM* are shown in Table 6.2. Each row indicates a different model, and the columns indicate i) the approach, ii) the specific components of the model (e.g. which QA models or readability index are being considered), and the PDP accuracy, separately for the QA train set and the QA test set.

It is important to remark that the train set and test set indicate the splits that are used for training the QA models; still, we present the results separately to understand whether the models built upon QA systems perform differently on the questions used for training them and on the held out test questions. The other models – i.e. the ones that are not built upon QA

models – do not see any differences between the train questions and the test questions, but we still plot the results separately to compare them with the other models.

		PDP Accuracy	
Approach	Components	QA Train	QA Test
ZeroR	-	0.5000	0.5000
Readability	Flesch Reading Ease	0.7202	0.7917
Readability	Flesch-Kincaid Grade Level	0.7305	0.8022
Readability	Automated Readability Index	0.7304	0.8132
Readability	Gunning FOG Index	**0.7410**	**0.8245**
Readability	Coleman-Liau Index	0.7264	0.8104
Similarity	W2V - question-correct choice(s)	0.4967	0.4993
Similarity	W2V - correct choice(s)-wrong choice(s)	0.4842	0.4514
Score variance	BERT	0.6209	0.5970
Score variance	BERT	0.6161	0.6049
Score variance	BERT	0.6251	0.6020
Score variance	DistilBERT	0.6080	0.5851
Score variance	DistilBERT	0.6140	0.5779
Score variance	DistilBERT	0.6113	0.5801
Score variance	BERT (E)	0.6205	0.5984
Score variance	DistilBERT (E)	0.6123	0.5802
Score variance	BERT-DistilBERT (E)	0.6163	0.5827
IRT on QA models	Train BERT	0.6443	-
IRT on QA models	Train DistilBERT	0.5550	-
IRT on QA models	Train BERT-DistilBERT	0.5419	-
IRT on QA models	Test BERT	-	0.7075
IRT on QA models	Test DistilBERT	-	0.6215
IRT on QA models	Test BERT-DistilBERT	-	0.5962
IRT on QA models	Train-Test BERT	0.6443	0.7044
IRT on QA models	Train-Test DistilBERT	0.5570	0.6155
IRT on QA models	Train-Test BERT-DistilBERT	0.5604	0.6226

Table 6.2: *Evaluation of all the approaches on the task of PDP, on PairRACE_HM. For each approach, we consider several models, implemented using different components, and separately evaluate the accuracy on the QA train dataset and the QA test dataset. We gray out the models that perform worse than the random baseline and write the best preforming ones in bold.*

Going through the table from top to bottom, we evaluate the following approaches, which are the same one described in §6.3.

- ZeroR baseline, which randomly picks the most difficult question of the pair.
- Readability indexes, separately considering Flesch Reading Ease, Flesch-

Kincaid Grade Level, Automated Readability Index, Gunning FOG Index, Coleman-Liau Index.

- Similarity-based approaches, considering i) the similarity between the Word2Vec embeddings of the question and of the correct choice(s), and ii) the similarity between the Word2Vec embeddings of the correct choice(s) and of the wrong choice(s).

- Score variance of the softmax scores of QA models, and we present both the results obtained with the single instances (the three lines with BERT and DistilBERT as "components") and the results obtained using the ensembles (the three lines with "(E)").

- IRT on the answers of QA models; considering several configurations for this approach as well: specifically, we evaluate if there are any differences training the IRT model i) on the questions used for training the QA model, ii) on the questions used for testing the QA model, or iii) on the union of the two sets ("Train-Test" in the table). Additionally, for each configuration, we consider i) only the answers of the BERT models, ii) only the answers of the DistilBERT models, and iii) the answers of all the QA models (we separately consider the answers of the models and not the answers of the ensemble built from them). This is the approach is the only one that cannot always be evaluated on all the questions, as it depends on the questions that are considered for training the IRT model: for instance, if we consider only the answers of the QA models to the test questions, we have no information about the difficulty of the train questions.

Comparing the evaluation of the different models with the ZeroR baseline, we can see that all the models perform better than it, with the exception of the similarity-based approach. We believe that the Word2Vec embeddings used for these experiments are not capable of accurately capturing the semantic meaning of the questions and therefore their complexity, and more advanced embeddings might lead to improve results; but it might also be the case that the similarity itself is not a good proxy of question difficulty for reading comprehension questions.

Considering the other models, the readability based approach is clearly the best performing one and, interestingly, its accuracy does not depend too much on the specific index which is being used, which is an indication that all the readability indexes provide a reasonably accurate proxy of question difficulty. This is reasonable, as *RACE* contains reading comprehension

questions and, in this type of questions, the difficulty is known to be heavily dependent on the complexity of the reading passage.

As for the approach built upon the score variance of QA models, there is no clear correlation between the ECE and the PDP accuracy nor the QA accuracy and the PDP accuracy. Indeed, the model built upon the better calibrate QA model (*BERT-DistilBERT (E)*) is outperformed in the PDP task by most of the models built upon single instances, both on the QA train set and the QA test set. In this sense, even though this approach based on the score variance of QA models consistently perform better than the random baseline, we find that it is not clear how – and if – measures on the QA models (such as accuracy and calibration) can be used to pick the model that will perform better for QDET.

Finally, considering the approach that perform QDET by training an IRT model on the responses of QA models, we can see that it is generally the second best performing approach (outperformed by the readability-based one). For both the training set and the test set, the approach based on BERT leads to the better results – with minor differences depending on whether both the train set and the test set or only one of them is used to train the IRT model – and this was somewhat unexpected since the variety of answers is more reduced than using both the answers of the BERT models and the DistilBERT models. Lastly, another relevant observation is the fact that this approach consistently performs better on the test set, which was not the case for the approach based on the score variance.

Cloud Academy (IRT)

Table 6.3 present the same analysis for the *Cloud Academy* dataset; the approaches that are being evaluated are the same (built with components trained on the *Cloud Academy* dataset) and the same metric (PDP accuracy) is used for the evaluation. Again, we separately consider the QA train questions and the QA test questions.

Comparing the table with the results obtained on *PairRACE_HM*, some significant differences are immediately visible. First of all, the readability indexes in this case perform in par with (if not worse than) the ZeroR baseline, meaning that they cannot be used in any way as a proxy of question difficulty. This makes sense, since *Cloud Academy* is a content knowledge assessment dataset, and therefore the difficulty of the questions mostly comes from the domain knowledge which is being assessed, which is not captured by the readability indexes.

Approach	Component(s)	PDP Accuracy QA Train	QA Test
ZeroR	-	0.5000	0.5000
Readability	Flesch Reading Ease	0.4938	0.4807
Readability	Flesch-Kincaid Grade Level	0.5067	0.4949
Readability	Automated Readability Index	0.5039	0.4926
Readability	Gunning FOG Index	0.5051	0.4916
Readability	Coleman-Liau Index	0.5053	0.4888
Similarity	W2V - question-correct choice	0.4352	0.4393
Similarity	W2V - correct choice-wrong choice	0.5725	0.5770
Score variance	BERT	0.5240	0.5236
Score variance	BERT	0.5355	0.5264
Score variance	BERT	0.5255	0.5201
Score variance	DistilBERT	0.5298	0.5187
Score variance	DistilBERT	0.5180	0.5027
Score variance	DistilBERT	0.5237	0.5210
Score variance	BERT (E)	0.5317	0.5254
Score variance	DistilBERT (E)	0.5265	0.5187
Score variance	BERT-DB (E)	0.5332	0.5254
IRT on QA models	Train BERT	0.6107	-
IRT on QA models	Train DistilBERT	0.5968	-
IRT on QA models	Train BERT-DistilBERT	0.5658	-
IRT on QA models	Test BERT	-	**0.6349**
IRT on QA models	Test DistilBERT	-	0.6047
IRT on QA models	Test BERT-DistilBERT	-	0.5773
IRT on QA models	Train-Test BERT	**0.6130**	0.6328
IRT on QA models	Train-Test DistilBERT	0.5971	0.6087
IRT on QA models	Train-Test BERT-DistilBERT	0.5639	0.5768

Table 6.3: *Evaluation of all the approaches on the task of PDP, on Cloud Academy. For each approach, we consider several models, implemented using different components, and separately evaluate the accuracy on the QA train dataset and the QA test dataset. We gray out the models which perform worse than the random baseline, and write the best performing model in bold.*

Another difference is the performance of the similarity-based approach: indeed, while the similarity between the embedding of the question and the embedding of the correct choice(s) perform worse than the random baseline, the similarity between the correct choice(s) and the wrong choice(s) reaches an accuracy of 57%, much better than on *RACE*. Most likely, it is capable of capturing – at least partially – the semantic meaning of the answer choices and this lead to improved results.

Moving on to the approach which leverages the score variance of QA models, we can see that it performs better that the random baseline, but

worse than on *RACE*. Most of the observations from the previous analysis still hold true: the PDP accuracy is always better on the QA train set than on the QA test set, and the ensembles do not bring clear improvements over the models based on single instances, meaning that the ECE is not a clear indicator of which models will lead to the best PDP performance. Indeed, considering this approach, the best performance is obtained with *BERT* on both the QA train set and the QA test set.

Lastly, we can see that the *IRT on QA models* approach consistently performs better than the others, and all the models implemented with this approach, regardless of the chosen components, perform better than the other approaches, with the only exception of *Train BERT-DistilBERT*, which is outperformed by one of the similarity-based models. Similarly to what we observed on *RACE*, the better performance is consistently obtained on the test questions, and the models based on the QA answers of BERT models lead to the better performance.

6.5.3 Evaluating with ranking metrics

The target difficulty label used to evaluate the model performance on *Cloud Academy* is obtained by training an IRT model on the answers of real students to exam questions. As a consequence, the whole difficulty ranking of questions is available, and we can use it to evaluate the difficulty ranking obtained with the models for QDET. Unfortunately, this is not possible on *RACE*, since it does not provide a difficulty ranking but only a binary label that indicates question difficulty.

Since all the models produce difficulties on different scales and different from the target one ($[-5; +5]$), we rescale all the difficulties (including the target) to the range $[0; 1]$ with min-max normalization (separately for each model). Then, we use to nDCG to evaluate the difficulty rankings, by comparing the true ranking with the ranking obtained from the various approaches.

The results of this evaluation are shown in Table 6.4, which has the same structure as the two tables presented above but shows the ranking nDCG rather than the PDP accuracy. Again, we separately consider the questions which were considered as training set and test set for training the QA models.

Approach	Component(s)	nDCG QA Train	nDCG QA Test
ZeroR	-	0.9641	0.9503
Readability	flesch reading ease	0.9632	0.9492
Readability	flesch kincaid grade level	0.9633	0.9505
Readability	automated readability index	0.9636	0.9518
Readability	gunning fog index	0.9630	0.9497
Readability	coleman liau index	0.9641	0.9524
Similarity	W2V - question-correct choice	0.9634	0.9489
Similarity	W2V - correct choice-wrong choice	0.9647	0.9494
Score variance	BERT	0.9669	0.9538
Score variance	BERT	0.9681	0.9537
Score variance	BERT	0.9668	0.9523
Score variance	DistilBERT	0.9675	0.9562
Score variance	DistilBERT	0.9666	0.9504
Score variance	DistilBERT	0.9675	0.9550
Score variance	BERT (E)	0.9677	0.9551
Score variance	DistilBERT (E)	0.9671	**0.9564**
Score variance	BERT-DB (E)	**0.9685**	0.9551
IRT on QA models	Train BERT	0.9651	-
IRT on QA models	Train DistilBERT	0.9650	-
IRT on QA models	Train BERT-DistilBERT	0.9651	-
IRT on QA models	Test BERT	-	0.9514
IRT on QA models	Test DistilBERT	-	0.9511
IRT on QA models	Test BERT-DistilBERT	-	0.9517
IRT on QA models	Train-Test BERT	0.9651	0.9514
IRT on QA models	Train-Test DistilBERT	0.9650	0.9511
IRT on QA models	Train-Test BERT-DistilBERT	0.9651	0.9517

Table 6.4: *Evaluation of the difficulty ranking obtained with all the approaches on Cloud Academy, using nDCG as evaluation metric. The nDCG is computed by comparing the difficulty ranking produced by each model with the target difficulty ranking (obtained with IRT from the answers of real students). For each approach, we consider several models, implemented using different components, and separately evaluate the nDCG on the QA train dataset and the QA test dataset. We highlight in bold the best performing models and gray out the ones that perform worse than the ZeroR baseline.*

The first thing that catches the eye is probably the fact that all the approaches and all configurations lead to fairly high values of nDCG (score 1.0 would indicate a perfect ranking), which is due to the distribution of the "true" IRT difficulties, as we observed in Chapter 5 while evaluating the models proposed to perform QDET in a supervised manner. Indeed, the target difficulty is distributed as a Gaussian, meaning that many questions have similar scores and, therefore, swapping their order does not affect too

much the nDCG; indeed, even the *ZeroR* baseline (i.e. assigning 0.5 as score to all the questions) reaches fairly high nDCG. Still, there are some observations that are worth making.

First of all, we can see that the readability indexes still are outperformed by basically all the other approaches and by the *ZeroR* baseline as well, which is in agreement with the results from the previous section.

The similarity based approaches, as well, are now outperformed by both the *score variance* approach and the *IRT on QA models* approach, regardless of the configuration, and even by the *ZeroR* baseline, in most cases.

The only two approaches that are consistently better than the baseline are the two approaches leveraging the answers of QA models, and there is an interesting difference with respect to the previous results on PDP accuracy. Indeed, here the *score variance* approach leads to the best results, both on the training questions and on the test questions, and this can be observed fairly consistently across the different models based on that approach.

6.5.4 Distribution of estimated difficulty

The last analysis we perform in this Chapter is a study of the distribution of the estimated difficulty and a comparison with the distribution of the "true" IRT difficulty estimated from real students' interactions. Again, we consider separately the questions that were used as QA train set and QA test set. Given that each approach generates difficulties on a different scale, we rescale them to a common scale (in the range $[-5; +5]$), which is the known range of the true IRT difficulties.

We do not plot the distribution of all the models for all the approaches, but rather pick one model for each one (considering the best performing in the PDP task), and two for the *IRT on QA models* approach, since it is the approach that is proposed for the first time in this thesis.

Specifically, we consider the following approaches:

- *Readability - Automated Readability Index*;
- *Similarity - Word2Vec: correct choice-wrong choice*;
- *Score variance - BERT-DistilBERT Ensemble*;
- *IRT on QA models - Train-Test BERT*;
- *IRT on QA models - Train-Test BERT-DistilBERT*.

QA Train set

Figure 6.1 plots the difficulty distribution of the items from the train set; we have on the top-left corner the true IRT difficulty distribution and in the other plots the approaches to unsupervised QDET.

It is immediately visible that all models have distributions very different from the target, meaning that the difficulties generated by them could not be directly used for calibrating items in an exams. Still, it is worth highlighting some differences. Indeed, the similarity approach and the score variance approach lead to difficulties that are too spiked towards the extremes, while the other three approaches generate distributions that are more spread across the range of possible values. Specifically, we believe that the distribution of the *IRT on QA models* are particularly promising: indeed, there are spikes for very low difficulties and very high difficulties, but these are most likely due to the number of QA models used to train the IRT model. Indeed, those two spikes correspond to questions that are correctly answered or wrongly answered by all the QA models under consideration, and increasing the number of QA models – as well as diversifying their capabilities – could help improve difficulty estimation.

QA Test set

Figure 6.1 plots the difficulty distributions obtained from the same models on the test set.

Overall, the observations are very similar to what we observed on the train set. The main difference is the fact that, here, the spikes for high difficulties generated by the *IRT on QA models* approach are higher than in the previous case, which is due to the fact that the QA accuracy is lower on the test set than on the train set. Still, we believe that the same countermeasure of having a larger number of QA models with a variety of architectures (and capabilities) could address this limitation. Figure 6.2.

6.6 Conclusions

The results presented in this Chapter support the idea that it is possible to estimate the human-perceived difficulty in an unsupervised manner, which have as main advantage over the supervised approaches the fact that it is possible to predict the relative difficulty of questions without needing any calibrated questions or logs of students' answers. Indeed, it is sufficient to have access to the corpus of questions and possibly, for the approaches based on QA models, to an additional corpus of learning materials.

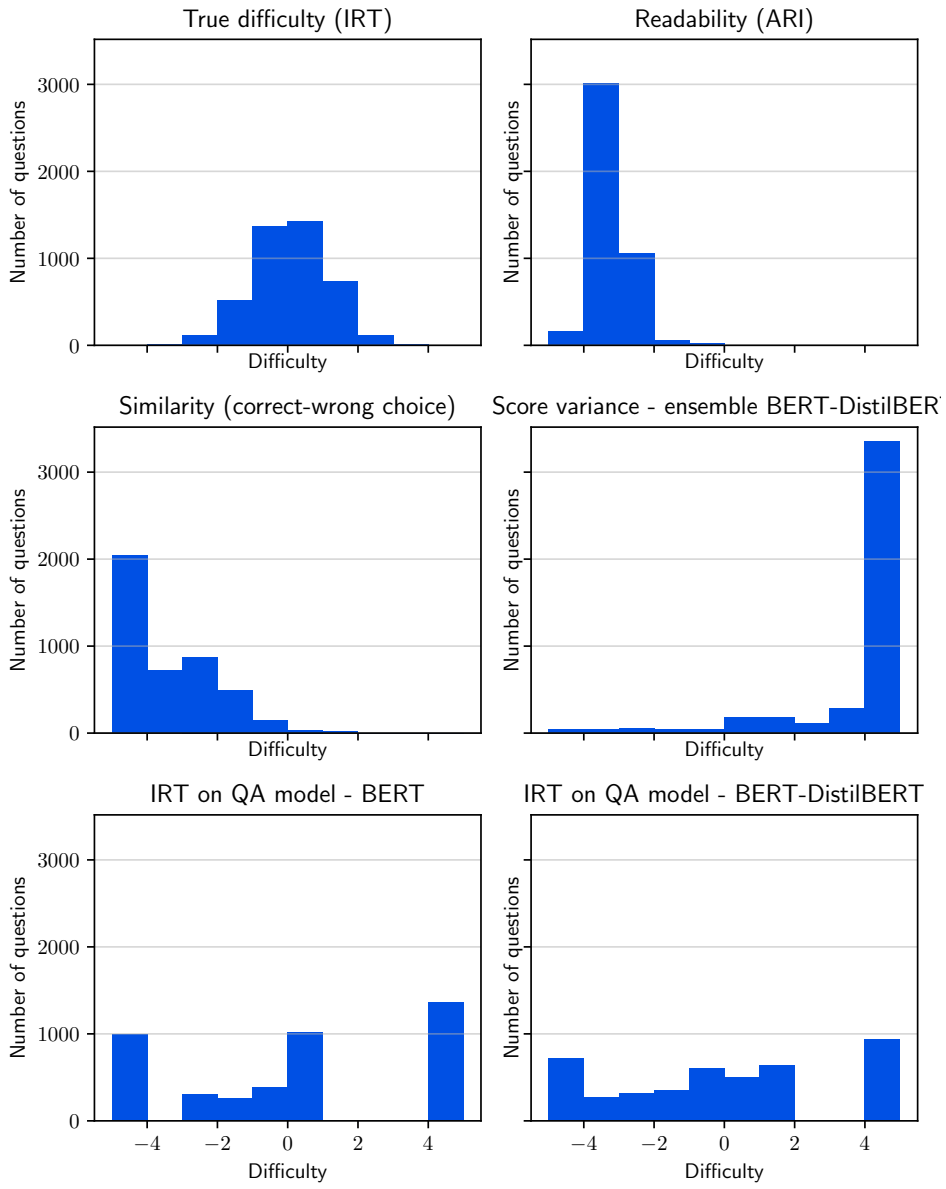

Figure 6.1: *Comparison of the difficulty distribution obtained with various models with the target difficulty distribution, considering the QA train set of Cloud Academy.*

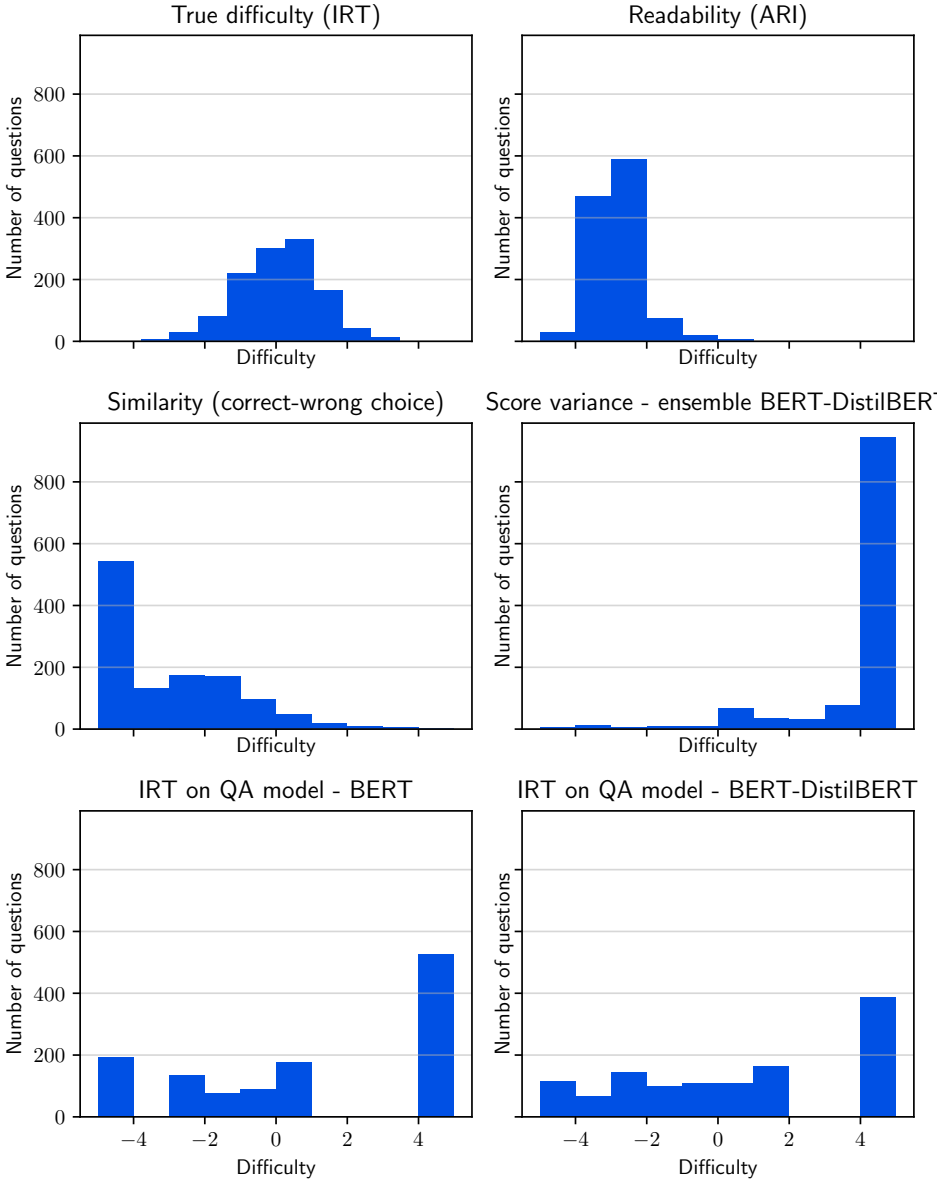

Figure 6.2: *Comparison of the difficulty distribution obtained with various models with the target difficulty distribution, considering the QA test set of Cloud Academy.*

The experimental results show that the choice of model heavily depends on the educational domain under consideration, similarly to what we observed for the supervised approaches. Indeed, we in the language assessment domain the readability indexes are a good proxy of question difficulty, and perform on their own even better than the techniques based on Question Answering (QA) models, at least considering the reading comprehension questions available in *RACE* in which the difficulty is known to be heavily dependent on the characteristics of the reading passage. On the contrary, as we observed for supervised QDET, readability indexes are not capable of producing accurate results in the content knowledge assessment domain. Indeed, in that case the question difficulty depends only partially on the verbalization and the linguistic characteristics of the questions, and therefore readability indexes lead to poor results, often worse than a random baseline. In this scenario, we observe that the best performing approach is to leverage the answers of QA models, which are trained to answer the questions, and use those answers to train an IRT model, mimicking pretesting with real students.

Also, the approach of leveraging, with IRT, the answers of QA models for QDET is the only approach that leads to output difficulty distributions that are comparable with the target distribution, and not too skewed towards extreme values. Still, we experiment with a limited number of QA models, and this approach is most likely to be capable of better performance if used on a larger number of models (and with a larger variety in their QA accuracy). A first step towards this would be leveraging the additional pre-training on the task of Masked Language Modeling (as we did for the experiments on supervised QDET) to obtain models that are (most likely) more accurate in the QA task and would therefore be helpful for the unsupervised estimation of question difficulty.

CHAPTER 7

A Brief Comparison of Supervised and Unsupervised QDET

In the previous Chapters, we have seen how different models proposed for supervised and unsupervised QDET perform, and how their performance depends on the educational domain under consideration and the characteristics of the questions which are being calibrated. In this Chapter, we briefly compare the two approaches and argue about the advantages and disadvantages of each of them. We start by discussing the real world applicability of these approaches (§7.1), and then perform a numerical comparison of the best performing supervised and unsupervised approaches to QDET (§7.2).

7.1 Real World Applicability

The first thing to consider when discussing the actual applicability of the techniques proposed in recent research to QDET are the assumptions that are made by the supervised and unsupervised models, respectively, and what these assumptions imply on the possible application scenarios. Indeed, the two approaches perform different assumptions and therefore cannot always be used on the same pools of questions.

Supervised models require a large pool of calibrated training questions, which must assess the same topics as the questions that the trained model will later be used on This is a limitation for two main reasons: first of all, such training data is not always available or expensive to obtain (as it needs pretesting); secondly, even when it is possible to pretest some training questions, it is necessary to label them with the correct skill (i.e. topic) which is assessed by them, in order not to build a training dataset which contains questions that assess different topics.

Unsupervised models, on the other hand, do not require a training dataset of calibrated items, which is an significant advantage. However, some of them, especially the ones which are seemingly the best performing on content knowledge assessment questions (e.g. *IRT on QA models*), need a large pool of training questions for a related task (e.g. question answering), which is not always available. For instance, considering the data collections used in this study, we were not able to train two of the unsupervised models on *ASSISTments*. Still, some unsupervised techniques, such as the ones based on readability indexes, can be used without any initial training and seem to be reasonably accurate in specific domains.

A huge disadvantage of unsupervised techniques with respect to the supervised ones is the fact that they do not always output question difficulties which are on the same scale as the difficulties which are currently used in the exam, and the task of converting them is not always straightforward. Indeed, as we have seen in the previous Chapter, rescaling the output difficulties to the desired range is often not sufficient, as the output distribution results very different from the target one. Supervised models, on the other hand, are specifically trained to output difficulties in the desired distribution and, even though they tend not to use the whole range of possible output values but rather perform predictions closer to the average value, they still have output distributions that are closer to the target one.

7.2 Numerical Comparison

In this Section, we perform a numerical comparison of the difficulties obtained with the supervised and the unsupervised techniques on *Cloud Academy*.

In order to keep the analysis short and focus only on the most important aspects, we do not consider all the supervised and unsupervised models but only the ones which seemingly lead to the better estimations, but still considering two different approaches for both categories. Specifically, we consider i) *R2DE* $Q_{Correct}$ and ii) *BERT* $Q_{Correct}$ *with the additional Masked Language Modeling pretraining* as supervised techniques, and i) *IRT on*

Type	Model	nDCG (Test)
ZeroR	-	0.9503
Sup.	R2DE $Q_{Correct}$	0.9647
Sup.	BERT $Q_{Correct}$ with MLM	0.9686
Unsup.	score variance (BERT-DistilBERT Ensemble)	0.9551
Unsup.	IRT on QA models (Train-Test BERT)	0.9514

Table 7.1: *nDCG on the Cloud Academy test set of the two best performing supervised and the two best performing unsupervised models for QDET.*

QA models (Train-Test BERT) and ii) *score variance of QA models (BERT-DistilBERT Ensemble)* as unsupervised techniques.

To compare the performance of the four models, we evaluate them on two tasks. First, we evaluate the ranking capabilities of the models by measuring the nDCG obtained when comparing the estimated difficulty distribution with the target one. Then, we compare the performance of the four models on the task of Pairwise Difficulty Prediction (PDP). Since the supervised approaches require a training dataset of calibrated questions, we evaluate the fours models only on the test set. We use *Cloud Academy* because it provides, differently from *RACE*, a complete ranking of the questions and not categorical difficulty values.

Table 7.1 presents the nDCG obtained by the four models on the test set. The nDCG for the two supervised models is larger, meaning that they perform better than the unsupervised ones, which is somewhat expected. Indeed, the supervised model can leverage more information and therefore are capable of generating a difficulty ranking which is, overall, closer to the target ranking. The absolute difference is not major, but considering the relative improvement over the *ZeroR* baseline, we can observe that the difference between the supervised and unsupervised approaches is still significant. Indeed, the best unsupervised approach improves the nDCG of the random baseline by 0.5%, and the best supervised approach improve it by 1.9%.

Moving the focus to the accuracy on the PDP task, which is shown in Table 7.2, we can see some notable differences. Indeed, in this case, the better performing model is the unsupervised approach based on the IRT difficulty obtained from QA models, and the supervised approaches outperform only the unsupervised approach based on the variance of softmax scores. All models perform better than the random baseline, which has accuracy of 0.50, with improvements from 5.1% (the unsupervised model based on score variance) to 26.5% (the unsupervised model based on IRT

Type	Model	PDP Accuracy (Test)
ZeroR	-	0.5000
Sup.	R2DE $Q_{Correct}$	0.5803
Sup.	BERT $Q_{Correct}$ with MLM	0.5892
Unsup.	score variance (BERT-DistilBERT Ensemble)	0.5254
Unsup.	IRT on QA models (Train-Test BERT)	0.6328

Table 7.2: *Pairwise Difficulty Prediction (PDP) accuracy on the Cloud Academy test set of the two best performing supervised and the two best performing unsupervised models for QDET.*

on QA models). This difference with respect to the results observed in the previous table is somewhat surprising, but it most likely due to the different nature of the two metrics which are being considered. Indeed, in PDP we do not take into consideration the specific value of difficulty, as long as the relation between the two is correct (i.e. the more difficult question is correctly identified), therefore the known behavior of the *IRT on QA models* approach, which sometimes lead to very high or very low difficulties (as shown in Figure 6.1 and Figure 6.2 in the previous Chapter) is not heavily penalized. On the contrary, the nDCG takes into consideration both the positional ranking and the difficulty value assigned to each question, which we believe is the reason behind this difference in the two metrics.

Still, the fact that the performance of the unsupervised model are better than the supervised models according to this one metric is a suggestion that it is a promising approach that, with some improvements, might indeed lead to a reliable alternative to supervised QDET for the scenarios in which the latter is not usable.

CHAPTER 8

Conclusions

8.1 Discussion

In this study, we have assessed the techniques proposed in recent research for Question Difficulty Estimation from Text (QDET), both the ones modeling it as a supervised task and the ones modeling it as an unsupervised task. We observed that these techniques, overall, are still not accurate enough to produce a final question difficulty that could be directly used in exam questions, but can already be helpful in several ways.

Considering educational settings that require pretesting, the techniques evaluated in this study could be used as an initial estimation of question difficulty which would be more accurate than a predefined value independent of the textual content. The educational settings leveraging manual calibration of exam questions, as well, could benefit from the application of these techniques for an initial estimation of question difficulty to suggest to the content curators. For instance, by highlighting the automatically estimated difficulty in case of a large difference from the manually selected value, it might point to questions which have been erroneously calibrated by the content curator.

Considering the two categories of approaches – supervised and unsu-

pervised – it seems that supervised approaches are better at producing a difficulty which is adapted to the current difficulty distribution. However, unsupervised approaches seem to provide a decently accurate overall ranking of question difficulties, with the important limitation that they are not aligned to the target difficulty distribution and therefore have to be rescaled or filtered in some way, which might not always be straightforward. Still, considering that these two categories are generally used in different scenarios, since they have different requirements and constraints, in this study we mostly focused on the two categories separately, only performing a preliminary comparison between them.

We have experimented on questions of different nature and coming from different educational domains, and observed that the choice of model to use heavily depends on the nature of the questions under consideration, which is in agreement with previous research.

Starting from the supervised approaches and, specifically, the Language Assessment (LA) domain, we observed that – at least considering reading comprehension questions – readability indexes and linguistic features can capture most of question difficulty, and this is true both for supervised approaches and unsupervised techniques. Indeed, in LA, and particularly in reading comprehension questions, the item difficulty heavily depends on the linguistic demands of the reading passage, therefore techniques which can capture this information are capable of an accurate difficulty estimation. Still, the same linguistic demands are captured – even more accurately – by more advanced models, such as BERT, which are generally capable of better performance. In this sense, we argue that Transformer-based models are probably the better choice from an accuracy point of view, but much simpler models, such as the ones based on readability indexes, might still be a reasonable choice in case of constraints from the computational point of view.

Unfortunately, this is not true for the Content Knowledge Assessment (CKA) domain. Indeed, supervised models based on simple techniques such as linguistic features and readability indexes are not capable of accurately capturing the demands of exam questions and therefore lead to inaccurate estimations. This is because in CKA the question difficulty mostly depends on the specific topics which are being assessed by the question, and techniques that focus on language only cannot capture such information. Specifically, we observed that the Transformers are, again, the models that generally lead to the best performance for supervised QDET in the CKA domain, and are in some cases matched by techniques based on word embeddings (such as Word2Vec) or frequency based features (such as TF-

IDF). In addition to that, Transformers can be pre-trained on additional documents related to the same topics as the ones assessed by the questions, which increases their accuracy.

Still focusing on the supervised techniques, we observed that – in the case of MCQ – having access to the text of the possible choices generally leads to improved results for the methods that can leverage it (Transformers, word embeddingd, and frequency based features), but there are some exceptions. Indeed, we observed that, possibly due to the specific encoding we use, Transformer models are not capable of accurately modeling questions which have multiple correct choices (i.e. they require the student to select all the correct choices), and are outperformed by the models based on TF-IDF and Word2Vec, which do not take into consideration the context and therefore are not affected by the specific positioning of words.

Moving our focus to unsupervised QDET, the observations are only partially different. Indeed, we observe that in LA the readability indexes are a good proxy of question difficulty, and perform on their own even better than the techniques based on Question Answering (QA) models. However, as we observed for supervised QDET, readability indexes are not capable of producing accurate results in the CKA domain. Indeed, in this case the best results are obtained with techniques that leverage the answers of QA models which are trained to answer the questions under calibration to train an IRT model, mimicking pretesting with real students.

In case of MCQ in the CKA domain, the semantic similarity between the correct answer and the distractors also seemed to provided a reasonably accurate indication of question difficulty, but it is generally outperformed by the IRT on QA models approach. In the case of reading comprehension questions, on the other hand, the semantic similarity did not seem to lead to accurate results.

8.2 Future Works

In this thesis, we have performed an assessment of recently proposed techniques for QDET, and there are several research directions which we believe might be worth exploring in the future.

Exploration of techniques to estimate the confidence of the models in their predictions. We have seen that for both supervised and unsupervised QDET – and especially for the latter – the models were in some cases capable of accurate predictions and, in other occasions, completely missed the correct difficulty value. In this sense, having a reliable estimation of model uncertainty – in other words, having calibrated models – could be

very helpful in understanding which are the questions for which we can trust the model predictions and which predictions, on the other hand, are not reliable. In some cases, such as the supervised binary difficulty prediction on *RACE*, this could be done with the traditional approach to model calibration (see §3.5), but for most of the approaches assessed in this thesis, such as the unsupervised estimation based on the IRT estimation obtained from QA models, it is not as straightforward.

Extension of the experiments on the usage of IRT on QA models for the unsupervised estimation of question difficulty. In Chapter 6 we have seen that the application of a one-parameter IRT model to the answers of QA models might lead to a good approximation of question difficulty, and this is the only unsupervised approach that produces an estimated difficulty distribution at least partially similar to the target distribution (see §6.1 and §6.2). However, we have been experimenting with a limited number of QA models (six, at most), which is a very limited number compared to the requirements that are recommended for a one-parameter IRT model (indeed, when estimating the "true" IRT difficulty using real students' answers we keep only the questions that were answered by at least 50 students). A direct continuation of this work, thus, would be the expansion of the experiments on this technique for unsupervised QDET, using more models and possibly with a larger variety of architectures.

Rescaling of estimated difficulty to match the target distribution. In a real world scenario where the difficulty estimated from text are supposed to be used together with some pre-existing and already calibrated questions, it is important to have an output difficulty distribution that matches the target one.

Considering the supervised models, they are supposed to produce output difficulties which are in agreement with the target difficulty distribution, but we observed that it is not always the case. Indeed, all the models tend to produce predictions which do not differ too much from the average difficulty, even though they move towards the correct direction (i.e. difficulty questions are labeled as being more difficult that average, and easy questions on the other hand as being less difficulty than average). Future research might explore the possibility of rescaling the predicted difficulty in order to better match the difficulty distribution of the training items.

The same issue is also present, with even more importance, in the case of unsupervised QDET: in this thesis, we performed a simpler rescaling to the desired difficulty range, but we believe that it might not be the best choice. Future work might explore techniques to adapt the distribution of the predicted difficulty to the distribution of pre-existing exam questions.

Bibliography

[1] I Elaine Allen and Jeff Seaman. *Online report card: Tracking online education in the United States.* ERIC, 2016.

[2] Tahani Alsubait, Bijan Parsia, and Ulrike Sattler. 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, 2013.

[3] Yigal Attali, Luis Saldivia, Carol Jackson, Fred Schuppan, and Wilbur Wanamaker. Estimating item difficulty with comparative judgments. *ETS Research Report Series*, 2014(2):1–8, 2014.

[4] Lyle F Bachman et al. *Fundamental considerations in language testing*. Oxford university press, 1990.

[5] David Beglar and Alan Hunt. Revising and validating the 2000 word level and university word level vocabulary tests. *Language testing*, 16(2):131–162, 1999.

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

[7] Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. Candidate evaluation strategies for improved difficulty prediction of language tests. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–11, 2015.

[8] Luca Benedetto, Giovanni Aradelli, Paolo Cremonesi, Andrea Cappelli, Andrea Giussani, and Roberto Turrin. On the application of transformers for estimating the difficulty of multiple-choice questions from text. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 147–157, 2021.

[9] Luca Benedetto, Andrea Cappelli, Roberto Turrin, and Paolo Cremonesi. Introducing a framework to assess newly created questions with natural language processing. In *International Conference on Artificial Intelligence in Education*, pages 43–54. Springer, 2020.

[10] Luca Benedetto, Andrea Cappelli, Roberto Turrin, and Paolo Cremonesi. 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, 2020.

[11] Luca Benedetto and Paolo Cremonesi. Rexy, a configurable application for building virtual teaching assistants. In *IFIP Conference on Human-Computer Interaction*, pages 233–241. Springer, 2019.

[12] Luca Benedetto, Paolo Cremonesi, and Manuel Parenti. A virtual teaching assistant for personalized learning. *arXiv preprint arXiv:1902.09289*, 2019.

[13] Benjamin Samuel Bloom. Taxonomy of educational objectives: The classification of educational goals. *Cognitive domain*, 1956.

[14] Marc Brysbaert and Boris New. Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior research methods*, 41(4):977–990, 2009.

[15] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3):904–911, 2014.

[16] Jeanne Sternlicht Chall and Edgar Dale. *Readability revisited: The new Dale-Chall readability formula*. Brookline Books, 1995.

[17] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[18] Chih-Ming Chen, Hahn-Ming Lee, and Ya-Hui Chen. Personalized e-learning system using item response theory. *Computers & Education*, 44(3):237–255, 2005.

[19] Keh-Jiann Chen, Chu-Ren Huang, Li-Ping Chang, and Hui-Li Hsu. Sinica corpus: Design methodology for balanced corpora. In *Proceedings of the 11th Pacific Asia Conference on Language, Information and Computation*, pages 167–176, 1996.

[20] Yu Chen, Telmo Silva Filho, Ricardo B Prudencio, Tom Diethe, and Peter Flach. β^3-irt: A new item response model and its applications. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1013–1021. PMLR, 2019.

[21] Song Cheng, Qi Liu, Enhong Chen, Zai Huang, Zhenya Huang, Yiying Chen, Haiping Ma, and Guoping Hu. Dirt: Deep learning enhanced item response theory for cognitive diagnosis. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2397–2400, 2019.

[22] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[23] Edmund B Coleman. On understanding prose: some determiners of its complexity. *NSF Final Report GB-2604. Washington, DC: National Science Foundation*, 1965.

[24] Brent Culligan. A comparison of three test formats to assess word difficulty. *Language Testing*, 32(4):503–520, 2015.

[25] Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, 2020.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, 2019.

[27] Hendrik Drachsler, Katrien Verbert, Olga C Santos, and Nikos Manouselis. Panorama of recommender systems to support learning. In *Recommender systems handbook*, pages 421–451. Springer, 2015.

[28] Yo Ehara. Building an english vocabulary knowledge dataset of japanese english-as-a-second-language learners using crowdsourcing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[29] Bobbie Eicher, Lalith Polepeddi, and Ashok Goel. Jill watson doesn't care if you're pregnant: Grounding ai ethics in empirical studies. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 88–94, 2018.

[30] Yasmine H El Masri, Steve Ferrara, Peter W Foltz, and Jo-Anne Baird. Predicting item difficulty of science national curriculum tests: the case of key stage 2 assessments. *The Curriculum Journal*, 28(1):59–82, 2017.

[31] Charles Elkan. Deriving tf-idf as a fisher kernel. In *International Symposium on String Processing and Information Retrieval*, pages 295–300. Springer, 2005.

[32] Ainuddin Faizan and Steffen Lohmann. Automatic generation of multiple choice questions from slide content using linked data. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, pages 1–8, 2018.

[33] Jiansheng Fang, Wei Zhao, and Dongya Jia. Exercise difficulty prediction in online education systems. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 311–317. IEEE, 2019.

[34] Mariano Felice and Paula Buttery. Entropy as a proxy for gap complexity in open cloze tests. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 323–327, 2019.

[35] Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *User modeling and user-adapted interaction*, 19(3):243–266, 2009.

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

[37] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 1050–1059, 2016.

[38] Ashok K Goel and Lalith Polepeddi. Jill watson: A virtual teaching assistant for online education. Technical report, Georgia Institute of Technology, 2016.

[39] Arthur C Graesser, Danielle S McNamara, Max M Louwerse, and Zhiqiang Cai. Coh-metrix: Analysis of text on cohesion and language. *Behavior research methods, instruments, & computers*, 36(2):193–202, 2004.

[40] David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34, 2003.

[41] Innovation Great Britain. Department for Business and Skills (BIS). The maturing of the mooc: literature review of massive open online courses and other forms of online distance learning. 2013.

[42] Robert Gunning et al. Technique of clear writing. 1952.

[43] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.

[44] Ronald K Hambleton and Russell W Jones. Comparison of classical test theory and item response theory and their applications to test development. *Educational measurement: issues and practice*, 12(3):38–47, 1993.

[45] Ronald K Hambleton, Hariharan Swaminathan, and H Jane Rogers. *Fundamentals of item response theory*. Sage, 1991.

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[47] Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197, 2011.

[48] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[49] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[50] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[51] Sebastian Hoffmann, Stefan Evert, Nicholas Smith, David Lee, Ylva Berglund-Prytz, et al. *Corpus linguistics with BNCweb-a practical guide*, volume 6. Peter Lang, 2008.

[52] John L Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–185, 1965.

[53] Jue Hou, Koppatz Maximilian, José María Hoya Quecedo, Nataliya Stoyanova, and Roman Yangarber. Modeling language learning using specialized elo rating. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 494–506, 2019.

[54] Fu-Yuan Hsu, Hahn-Ming Lee, Tao-Hsing Chang, and Yao-Ting Sung. Automated estimation of item difficulty for multiple-choice tests: An application of word embedding techniques. *Information Processing & Management*, 54(6):969–984, 2018.

[55] Yi-Ting Huang, Hsiao-Pei Chang, Yeali Sun, and Meng Chang Chen. A robust estimation scheme of reading difficulty for second language learners. In *2011 IEEE 11th International Conference on Advanced Learning Technologies*, pages 58–62. IEEE, 2011.

[56] Yi-Ting Huang, Meng Chang Chen, and Yeali S Sun. Development and evaluation of a personalized computer-aided question generation for english learners to improve proficiency and correct mistakes. *arXiv preprint arXiv:1808.09732*, 2018.

[57] Zhenya Huang, Qi Liu, Enhong Chen, Hongke Zhao, Mingyong Gao, Si Wei, Yu Su, and Guoping Hu. Question difficulty prediction for reading problems in standard tests. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[58] Syeda Asima Iqbal and Syeda Anila Komal. Analyzing the effectiveness of vocabulary knowledge scale on learning and enhancing vocabulary through extensive reading. *English Language Teaching*, 10(9):36–48, 2017.

[59] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

[60] Henry F Kaiser. The application of electronic computers to factor analysis. *Educational and psychological measurement*, 20(1):141–151, 1960.

[61] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[62] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch, 1975.

[63] Vishwajeet Kumar, Yuncheng Hua, Ganesh Ramakrishnan, Guilin Qi, Lianli Gao, and Yuan-Fang Li. Difficulty-controllable multi-hop question generation from knowledge graphs. In *International Semantic Web Conference*, pages 382–398. Springer, 2019.

[64] Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. Age-of-acquisition ratings for 30,000 english words. *Behavior research methods*, 44(4):978–990, 2012.

[65] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

[66] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

[67] John P Lalor, Hao Wu, and Hong Yu. Building an evaluation scale using item response theory. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 648. NIH Public Access, 2016.

[68] John P Lalor, Hao Wu, and Hong Yu. Learning latent parameters without human response patterns: Item response theory with artificial crowds. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2019, page 4240. NIH Public Access, 2019.

[69] Suzanne Lane, Mark R Raymond, and Thomas M Haladyna. *Handbook of test development*. Routledge, 2015.

[70] Ji-Ung Lee, Erik Schwan, and Christian M Meyer. Manipulating the difficulty of c-tests. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 360–370, 2019.

[71] SC Leong. On varying the difficulty of test items. In *32nd Annual Conference of the International Association for Educational Assessment, Singapore*, pages 21–26, 2006.

[72] Li-Huai Lin, Tao-Hsing Chang, and Fu-Yuan Hsu. Automated prediction of item difficulty in reading comprehension using long short-term memory. In *2019 International Conference on Asian Language Processing (IALP)*, pages 132–135. IEEE, 2019.

[73] Wim J Linden, Wim J van der Linden, and Cees AW Glas. *Computerized adaptive testing: Theory and practice*. Springer, 2000.

[74] Pierre Lison and Jörg Tiedemann. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, 2016.

[75] Ekaterina Loginova, Luca Benedetto, Dries Benoit, and Paolo Cremonesi. Towards the aplication of calibrated transformers to the unsupervised estimation of question difficulty from text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, September 2021.

[76] Edward Loper and Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70, 2002.

[77] Anastassia Loukina, Su-Youn Yoon, Jennifer Sakano, Youhua Wei, and Kathy Sheehan. 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, 2016.

[78] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[79] Nikos Manouselis, Hendrik Drachsler, Riina Vuorikari, Hans Hummel, and Rob Koper. Recommender systems in technology enhanced learning. In *Recommender systems handbook*, pages 387–415. Springer, 2011.

[80] G Harry Mc Laughlin. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646, 1969.

[81] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 3111–3119, 2013.

[82] Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 2015, page 2901. NIH Public Access, 2015.

[83] Diane Napolitano, Kathleen M Sheehan, and Robert Mundkowsky. Online readability and text complexity analysis with textevaluator. In *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Demonstrations*, pages 96–100, 2015.

[84] ISP Nation and P Teaching. Learning vocabulary. *New Zealand Language Teacher*, 9(1):10–11, 1983.

[85] Ulrike Padó. 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, 2017.

[86] Irina Pandarova, Torben Schmidt, Johannes Hartig, Ahcène Boubekki, Roger Dale Jones, and Ulf Brefeld. Predicting the difficulty of exercise items for dynamic difficulty adaptation in adaptive language tutoring. *International Journal of Artificial Intelligence in Education*, 29(3):342–367, 2019.

[87] Jan Papoušek, Vít Stanislav, and Radek Pelánek. Impact of question difficulty on engagement and learning. In *International Conference on Intelligent Tutoring Systems*, pages 267–272. Springer, 2016.

[88] Thanaporn Patikorn, David Deisadze, Leo Grande, Ziyang Yu, and Neil Heffernan. Generalizability of methods for imputing mathematical skills needed to solve problems from texts. In *International Conference on Artificial Intelligence in Education*, pages 396–405. Springer, 2019.

[89] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[90] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[91] Isidoros Perikos, Foteini Grivokostopoulou, Konstantinos Kovas, and Ioannis Hatzilygeroudis. Automatic estimation of exercises' difficulty levels in a tutoring system for teaching the conversion of natural language into first-order logic. *Expert Systems: The Journal of Knowledge Engineering*, 33(6):569–580, 2016.

[92] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, 2018.

[93] Zhaopeng Qiu, Xian Wu, and Wei Fan. Question difficulty prediction for multiple choice problems in medical exams. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 139–148, 2019.

[94] Georg Rasch. *Probabilistic models for some intelligence and attainment tests*. Danish Institute for Educational Research, 1960.

[95] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[96] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3859–3869, 2017.

[97] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[98] Norbert Schmitt, Diane Schmitt, and Caroline Clapham. Developing and exploring the behaviour of two new versions of the vocabulary levels test. *Language testing*, 18(1):55–88, 2001.

[99] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[100] RJ Senter and Edgar A Smith. Automated readability index. Technical report, CINCINNATI UNIV OH, 1967.

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

[102] Dominic Seyler, Mohamed Yahya, and Klaus Berberich. Knowledge questions from knowledge graphs. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 11–18, 2017.

[103] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[104] Kathleen M Sheehan, Michael Flor, and Diane Napolitano. A two-stage approach for generating unbiased estimates of text complexity. In *Proceedings of the Workshop on Natural Language Processing for Improving Textual Accessibility*, pages 49–58, 2013.

[105] Kathleen M Sheehan, Irene Kostin, Diane Napolitano, and Michael Flor. The textevaluator tool: Helping teachers and test developers select texts for use in instruction and assessment. *The Elementary School Journal*, 115(2):184–209, 2014.

[106] Günther Sigott. How fluid is the c-test construct. *Der C-Test: Theorie, Empirie, Anwendungen The C-Test: Theory, Empirical Research, Applications*, pages 139–146, 2006.

[107] Bernard Spolsky. Reduced redundancy as a language testing tool. 1969.

[108] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 2377–2385, 2015.

[109] Jurik Stiller, Stefan Hartmann, Sabrina Mathesius, Philipp Straube, Rüdiger Tiemann, Volkhard Nordmeier, Dirk Krüger, and Annette Upmeier zu Belzen. Assessing scientific reasoning: A comprehensive evaluation of item features that affect item difficulty. *Assessment & Evaluation in Higher Education*, 41(5):721–732, 2016.

[110] Fredricka Stoller and William Grabe. Implications for l2 vocabulary acquisition and instruction from l1 vocabulary research. *Second language reading and vocabulary learning*, pages 24–45, 1993.

[111] Yuni Susanti, Hitoshi Nishikawa, Takenobu Tokunaga, and Obari Hiroyuki. Item difficulty analysis of english vocabulary questions. In *Proceedings of the 8th International Conference on Computer Supported Education*, pages 267–274, 2016.

[112] Yuni Susanti, Takenobu Tokunaga, and Hitoshi Nishikawa. Integrating automatic question generation with computerised adaptive test. *Research and Practice in Technology Enhanced Learning*, 15:1–22, 2020.

[113] Yuni Susanti, Takenobu Tokunaga, Hitoshi Nishikawa, and Hiroyuki Obari. Controlling item difficulty for automatic vocabulary question generation. *Research and practice in technology enhanced learning*, 12(1):1–16, 2017.

[114] Hanshuang Tong, Yun Zhou, and Zhen Wang. Exercise hierarchical feature enhanced knowledge tracing. In *International Conference on Artificial Intelligence in Education*, pages 324–328. Springer, 2020.

[115] Jonathan Trace, James Dean Brown, Gerriet Janssen, and Liudmila Kozhevnikova. Determining cloze item difficulty from item and passage characteristics across different learner backgrounds. *Language Testing*, 34(2):151–174, 2017.

[116] Toshihiko Uemura and Shinichiro Ishikawa. Jacet 8000 and asia tefl vocabulary initiative. *Journal of Asia TEFL*, 1(1):333–347, 2004.

[117] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[118] Ellampallil Venugopal Vinu et al. A novel approach to generate mcqs from domain ontology: Considering dl semantics and open-world assumption. *Journal of Web Semantics*, 34:40–54, 2015.

[119] Lev S Vygotsky. Mind in society: The development of higher mental processes (e. rice, ed. & trans.), 1978.

[120] Walter D Way. Protecting the integrity of computerized testing item pools. *Educational Measurement: Issues and Practice*, 17(4):17–27, 1998.

[121] Paul Westermann and Ralph Evins. Using bayesian deep learning approaches for uncertainty-aware building energy surrogate models. *Energy and AI*, 3:100039, 2021.

[122] Kang Xue, Victoria Yaneva, Christopher Runyon, and Peter Baldwin. 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, 2020.

[123] Victoria Yaneva, Peter Baldwin, Janet Mee, et al. 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, 2019.

[124] Victoria Yaneva, Peter Baldwin, Janet Mee, et al. Predicting item survival for multiple choice questions in a high-stakes medical exam. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6812–6818, 2020.

[125] Hua Yang and EUM Suyong. Feature analysis on english word difficulty by gaussian mixture model. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 191–194. IEEE, 2018.

[126] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.

[127] Keith A Yeomans and Paul A Golder. The guttman-kaiser criterion as a predictor of the number of common factors. *The Statistician*, pages 221–229, 1982.

[128] Yu Yin, Zhenya Huang, Enhong Chen, Qi Liu, Fuzheng Zhang, Xing Xie, and Guoping Hu. Transcribing content from structural images with spotlight mechanism. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2643–2652, 2018.

[129] Yu Yin, Qi Liu, Zhenya Huang, Enhong Chen, Wei Tong, Shijin Wang, and Yu Su. Quesnet: A unified representation for heterogeneous test questions. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1328–1336, 2019.

[130] Ya Zhou and Can Tao. Multi-task bert for problem difficulty prediction. In *2020 International Conference on Communications, Information System and Computer Engineering (CISCE)*, pages 213–216. IEEE, 2020.