10

Extracting Linguistic Signal From Item Text and Its Application to Modeling Item Characteristics

Victoria Yaneva, Peter Baldwin, Le An Ha, and Christopher Runyon

1. Introduction

One novel application of natural language processing (NLP) in assessment that has received growing interest is the modeling of item characteristics using predictors extracted from item text. Often, these attempts to capitalize on the relationship between text features and item characteristics occur in advance of pretesting, when response data have not yet been collected and the item text is the only available information about the item. It follows that the predicted item characteristics of greatest interest will be those that can *increase* the efficiency of pretesting or *reduce* its various negative side effects. In the case of efficiency – and pressure for efficiency gains will only increase with advances in automatic item generation – improvements can be made either by predicting items' probability of survival (i.e., the probability of satisfying the statistical criteria for use in scoring; Ha et al., 2019; Yaneva et al., 2020) or by eliminating (or reducing) the necessity for pretesting altogether through item difficulty prediction (Benedetto et al., 2020; Kurdi, 2020; Leo et al., 2019; Xue et al., 2020). With respect to reducing the negative aspects of pretesting, it has been shown that the timing variability of test forms can be reduced by predicting the time demands of pretest items prior to form assembly (Baldwin et al., 2021). For these activities and others, researchers have found that variables extracted from item text can predict item characteristics better than several baselines; yet the practical importance of these gains has not been convincingly demonstrated in all cases. As a result, the application of NLP to prediction problems in educational measurement remains an active and exciting area for research.

Because of its potential to illuminate and inform test development, the understanding of relationships between ancillary item data generally and various item characteristics has been of long-standing interest to assessment specialists. NLP has expanded and enriched the universe of ancillary data in novel ways, but despite this interest, it has not been widely used for this purpose. For example, except for Baldwin et al. (2021), the studies cited here were published in NLP venues, illustrating the limited exposure these methods have within educational measurement and identifying potential methodological and knowledge gaps. In this chapter, we address some of these gaps by providing an overview of several well-known NLP approaches

DOI: 10.4324/9781003278658-14

for representing text and demonstrating how these representations can be used to solve practical measurement problems. This twofold purpose also structures the chapter.

More specifically, our overview of text representation methods starts with a summary of traditional linguistic features, moves on to introduce non-contextualized word embeddings,[1] and then concludes with a nontechnical primer on contextualized embeddings. These descriptions are targeted to readers with no background in NLP. The second part of the chapter provides an empirical illustration of these approaches by outlining the process of predicting item characteristics for multiple-choice questions (MCQs) accompanied by various relevant findings. In this context, several practical considerations are highlighted, including: the choice of pretraining data and model architecture, the encoding of different levels of dependencies, and the constraints imposed by model interpretability.

2. Representing Item Text

As mentioned, in this section we introduce three different classes of ancillary data that can be extracted from an item's text and explain how these data can be used to predict item characteristics. What we might call *ancillary* or *collateral data* in this context are generally referred to as *features* in the NLP literature. Next, these categories are presented in the following order: *human-engineered linguistic features*, *non-contextualized embeddings*, and *contextualized embeddings*, which also follows the order of their increasing abstraction (and, likewise, their chronological development). This overview is brief, merely intending to introduce those readers unfamiliar with NLP to the main approaches to text representation. For a detailed, NLP-focused review, we refer the reader to Pilehvar and Camacho-Collados (2020).

2.1 Human-Engineered Linguistic Features

Early approaches to text processing relied heavily on linguistic information extracted through human-engineered features. This extraction process requires both: (1) an initial hypothesis that a given feature will covary with a variable of interest (e.g., the hypothesis that *average noun phrase length* is related to the readability of text passages); and (2) the necessary NLP tools and resources for extracting the predictor (e.g., a *parser* and *part-of-speech tagger* that can separate a given text into relevant subparts and identify which parts constitute noun phrases).[2] Other examples of human-engineered features include *number of polysemous words*,[3] which is intended to capture semantic ambiguity; and *age of acquisition*, which is meant to capture the familiarity subjects (e.g., students) are expected to have with a given word at a given age. There are many others. Linguistic features can capture different levels of linguistic processing such as lexical, syntactic, semantic, and discourse, and they have been used to predict item difficulty in the context of reading and listening comprehension exams (e.g., Choi & Moon, 2020; Loukina et al., 2016). Beyond reading exams, linguistic features have also been shown to predict item difficulty more generally as well as the average time required to respond to different MCQs (Baldwin et al., 2021).

The extraction of linguistic features is highly reliant on NLP resources. To measure *polysemy*, first, ontologies are needed that encode semantic relationships (e.g., WordNet; Miller, 1995); to measure *age of acquisition*, normed word lists are needed (e.g., MRC psycholinguistic database; Coltheart, 1981); and so on. As can be expected, early approaches to extracting linguistic features were constrained by the availability and coverage of these kinds of resources, which were both costly and slow to develop.

Despite these challenges, the hypothesis-driven approach (where a feature is extracted only because of a researcher's hypothesis that it may have predictive power) has been successfully applied to many practical problems and is especially useful when a given application calls for interpretable features. For example, this approach allows the researcher not only to extract highly predictive features, but also to exclude ones that should not be used (e.g., text length when predicting essay scores) and have better control over model bias. This advantage of linguistic features, however, is also their limitation: *because* they are hypothesis driven, linguistic features may not always capture the most important or relevant predictors a given dataset has to offer for a given problem. For a data-driven approach, we instead must turn to a new paradigm in NLP research: dense word vector representations, also known as *word embeddings*.

2.2 Word Embeddings: Theoretical Background

The notion of word embeddings has its origins in the *distributional hypothesis*, which states that words occurring in the same contexts tend to have similar meanings (Harris, 1954). This hypothesis was later immortalized by Firth (1957) as: 'You shall know a word by the company it keeps'. A well-known illustration of this phenomenon is an experiment by McDonald and Ramscar (2001), who placed nonce words such as *wampimuk* in different contexts – e.g., 'He filled the *wampimuk* with the substance, passed it around and we all drunk some' and 'We found a little, hairy *wampimuk* sleeping behind the tree'. When presented in these contexts, *wampimuk* was consistently understood by the study participants to refer to some type of container for holding liquid or an animate creature, respectively.

The distributional hypothesis has important implications for the computational processing of language, since context can be represented numerically by encoding word co-occurrences in large collections of texts (*corpora*). In other words, if we can encode a sufficiently large number of contexts for a given word (or subword[4]), we can infer its semantic, syntactic, or pragmatic[5] properties without having to rely on external resources such as ontologies. While this was only a theoretical possibility a few decades ago, it is now practically feasible thanks to two advances: the accumulation of large amounts of electronically stored text data, which allows a sufficient number of co-occurrences to be encoded, and developments in parallel computing, which provide the computational power needed to process these large datasets. These developments were further aided by advances in deep neural network models that made it possible to condense high-dimensional and sparse co-occurrence vectors into dense vectors with fewer dimensions. These dense vectors are sometimes called *dense vector representations* but, more often, are referred to as *embeddings*. You can think of an embedding as the location of a word in an *n*-dimensional vector space, and it follows that its semantic properties can be inferred based on other nearby words in this space. The high predictive power of dense vector representations for many NLP tasks was first demonstrated by early embedding types such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), which we discuss next.

2.3 Non-Contextualized Embeddings

Early embedding types are sometimes described as *non-contextualized embeddings* because they do not fully capitalize on differences in context. So, the word *hard* in 'I read Simon's book, which was hard' and 'Simon hit me with his book, which was hard' is represented by a common embedding. This limitation was later addressed by contextualized embeddings (described in the next section), in which the various uses of a given polyseme such as 'hard' are encoded separately. Nevertheless, non-contextualized embeddings are less demanding computationally and have been productively used to solve many problems.

Creating non-contextualized embeddings can be divided into two distinct stages: *data generation* and *model training*. We illustrate this process with Word2Vec.[6] During the dataset generation stage, neighboring words for each word in the corpus (i.e., the training data such as Google News or PubMed articles) are identified. For a *sliding window* (or *context window*) with *size 2*, a given word's neighbors are the two words preceding it and the two words following it. So, for example, suppose our corpus contains the preprocessed[7] sentence, 'You shall know word by company it keeps'. Data generated for the input words *know* and *word*, with a sliding window size of 2, would look as shown in Table 10.1.

Eventually, during the model-training stage (described later in this section), these data are used as input for a neural network tasked with predicting the value in the Target column (i.e., whether or not the input and output words are neighbors – sometimes called the *label*); however, note that here the target values are all 1, and so before this can be done, additional data are needed. To address this, output words are added to the dataset that are randomly sampled[8] from the vocabulary (a process called *negative sampling* that works by contrasting signal with noise). These sampled words are *not* a given input word's neighbors and so their target values are all 0, as shown in Table 10.2.

This procedure generates a large dataset of word co-occurrences (and non-co-occurrences) without relying on manual annotation or external resources, as in the case of extracting linguistic features described earlier.

Data generation is followed by the model-training stage, which begins with the creation of two matrices that are first initialized with random numbers: an embedding matrix, which will store the embeddings of the input words, and a context matrix, which will store the embeddings of the output (context) words. These are $m \times n$ matrices, where m is the number of words in the corpus vocabulary and n is the desired number of dimensions for the embeddings (e.g., 300).[9]

Both matrices are initialized with random numbers and then updated during training as follows. For each word in the dataset, the model takes one positive sample and some number[10] of

Table 10.1 Example (Partial) Training Data Samples (Before Applying Negative Sampling)

Input Word	**Output Word**	**Target** (*Are the Input and Output Words Neighbors?*)
know	you	1
know	shall	1
know	word	1
know	by	1
word	shall	1
word	know	1
word	by	1
word	company	1

Table 10.2 Example Training Data Samples for the Input Word 'Know' With Negative Sampling

Input Word	**Output Word**	**Target** (*Neighbors?*)
know	you	1
know	shall	1
know	word	1
know	by	1
know	aardvark	0
know	aarhus	0
know	. . .	0
know	truck	0

Table 10.3 Training Sample for the Input Word ‘Know’

	Input Word	Output Word	Target (*Neighbors?*)
Positive Sample	know	you	1
Negative Sample	know	aardvark	0
	know	truck	0

negative samples. Table 10.3 illustrates this for the input word ‘know’ with the positive sample ‘you’ and two negative samples, ‘aardvark’ and ‘truck’.

These samples correspond to four embeddings: one from the embedding matrix (for the input word ‘know’) and three embeddings from the context matrix (for the output words ‘you’, ‘aardvark’, and ‘truck’). The similarity between each input word and output word then can be quantified by the dot product for each input word and output word embedding pair. Each of these three dot products then is transformed into a value ranging between zero and one using the sigmoid function:

$$p(neighbor\,|\,w,c) \sim \sigma\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{c}\right) = \frac{1}{1+e^{-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{c}}}$$

where $p(c\,|\,w)$ is the (estimated) probability that an input word w and an output word c (the context) are neighbors; $\sigma(\cdot)$ is the sigmoid function; and $\mathbf{w}^{\mathrm{T}}\mathbf{c}$ is the dot product between the embedding for the input word (from the embedding matrix) and the embedding for the output word (from the context matrix).

For this small sample of training data, this process produces three probabilities – one for each input word/output word pair. Of course, the model is still untrained (the embedding and context matrices are still in their initial random state), and so these probabilities are, at this point, meaningless (in fact, even when they mean something, these probabilities will not be of any importance to us – we care only about the embeddings matrix from the hidden layer). As a next step, then, these probabilities are subtracted from their target value (1 for neighbors and 0 otherwise), yielding errors. This produces an error vector, which then is used to adjust the embedding weights for the input and output words in the embedding and context matrices, respectively. As this iterative process is repeated (the number of iterations may depend on computational resources), the predictions and embeddings gradually improve. After training is complete, each input word has a Word2Vec embedding in the embedding matrix with a fixed number of dimensions. Provided the training set is comprehensive enough (only words in the training set will have embeddings), Word2Vec embeddings pretrained in this way can be used as predictors for various other tasks. These tasks include predicting item characteristics, which is possible when items with similar meanings (as captured by their embeddings through encoding similarities in context) are similar with respect to the item characteristic of interest.[11]

For many NLP tasks, non-contextualized word embeddings have been shown to perform as well or better than human-engineered linguistic features, without requiring annotated corpora or external resources. For example, Word2Vec embeddings and linguistic features produced comparable results when predicting item difficulties (Ha et al., 2019) and average response times (Baldwin et al., 2021) for clinical MCQs. Nevertheless, as noted, non-contextualized embeddings like Word2Vec and GloVe fail to account for, well, *context*; and in the absence of context, some meaning cannot be represented by a single embedding per word. Recent advances have addressed this shortcoming by using *contextualized* word embeddings.

2.4 Contextualized Embeddings

Most current contextualized word embeddings are produced using large models with millions of parameters known as *transformer models*.[12] Given the complexity of these models, a detailed explanation of how they work is outside of the scope of this chapter (for a more in-depth description, see Devlin et al., 2018; Wolf et al., 2020). Here, we focus on the output from these models and how it can be used for the task of predicting item characteristics.

Several well-known contextualized embedding models have been developed, including ELMo (Peters et al., 2018), GPT-2 (Radford et al., 2019), and GPT-3 (Brown et al., 2020). However, perhaps the most widely used among these at the time of this writing is BERT (Bidirectional Encoder Representations from Transformers; Devlin et al., 2018). The BERT model is pretrained on a vast collection of texts and is freely available to download.[13] In the original work, there are two versions of BERT: *BERT Large*, which performs best in a number of benchmarking tasks but has more than 300 million parameters to train; and a lighter version known as *BERT Base* (merely 110 million parameters) that performs slightly worse on some benchmarks but requires far less computational power.

In the context of predicting item characteristics, models like BERT can be trained on either generic or domain-specific texts. When medical text embeddings are desirable, for example, a corpus such as MEDLINE abstracts,[14] can be used to generate the predictive word embeddings for the task. Once trained, the model can be *fine-tuned* for the specific data and task of interest – e.g., on a given set of test items and their item characteristics, respectively. Fine-tuning typically involves adding an extra layer on top of the original deep neural network and then retraining the model on the desired task (e.g., on predicting item difficulty). In this way, the internal weights of the pretrained model are updated without discarding the knowledge gained from the original model training. Alternatively, as in the case of our experiments later in this chapter, the pretrained model can be used to generate embeddings for a dataset of interest *without* fine-tuning. In these cases, the generated embeddings are used as input for training familiar machine learning models such as regression, random forests, support vector machines, and so on. Several studies suggest that the latter approach may be more successful for some applications (e.g., Caron et al., 2020; Conneau et al., 2017; Conneau & Kiela, 2018); however, as we will describe, this was not our only motivation here.

Unlike models such as Word2Vec described earlier, BERT is trained on *two* tasks that produce *two* different types of embeddings: one for tokens (e.g., words), and one for sentences:

- *Token embeddings: Token* refers to both words and subwords. The algorithm first breaks all words into subwords and then reconstructs words from the smaller units, which gives the model the capability to handle vocabulary that is *not* included in the training data. Task one, then, is to predict each token in the training corpus by those tokens appearing before or after it, with the goal of encoding as much context as possible. Recalling the previous example, unlike non-contextualized embeddings, here the ambiguous word *hard* would be represented differently depending on whether it means 'difficult' or 'solid' in a given context. Moreover, although embeddings are at the individual word (or subword token) level, they can be pooled to represent a document (which could be an item, for example).
- *Sentence embeddings:* The goal of task two is to determine the sequence of two sentences (e.g., *does sentence B follow sentence A?*). Trained this way, the model produces embeddings for entire sentences rather than individual words or subwords, which may be beneficial for tasks where this larger context contains relevant information. In the case of test items, each item comprises several BERT sentence embeddings (depending on the number of sentences in the item) that can be pooled to form the final item embedding.

Given BERT's popularity, variations have been developed to improve aspects of the original model. For example, DistilBERT reduces the size of a BERT model by 40%, which requires significantly less computational power to train, 'while retaining 97% of its language understanding capabilities and being 60% faster' (Sanh et al., 2019). Other models such as RoBERTa (Liu et al., 2019) improve the hyperparameter tuning of the original BERT, leading to better performance on several benchmarking NLP tasks. There are versions of BERT that have been pretrained on domain-specific texts, such as Clinical BERT (Alsentzer et al., 2019), which is trained on clinical texts. Given the separation of the training and fine-tuning discussed earlier, domain-specific pretrained models allow researchers to focus on the modeling aspect of a given problem without requiring the computational power and access to data typically necessary for robust model pretraining. Finally, other transformer-based architectures include DeBERTa (He et al., 2020), Electra (Clark et al., 2020), and many others.

In the context of predicting item characteristics, contextualized embeddings have shown promising results. Ha et al. (2019) compare different predictors including linguistic features, Word2Vec embeddings, and ELMo embeddings for the task of predicting item difficulty for clinical MCQs. The results outperformed several simple baselines, and ELMo performed best among these three predictor classes (although the best results overall were obtained by the combination of linguistic features *and* ELMo, indicating that the signals they encode complement rather than completely overlap each other). Outside of the medical domain, Benedetto et al. (2021) found that BERT and DistilBERT were more successful than ELMo and several other baseline approaches in predicting item difficulty for math and IT MCQs.

2.5 Other Predictors of Item Characteristics

The previous sections gave a short overview of different ways to extract and represent linguistic information from item text that can be used to predict item characteristics for MCQs. In addition to these, there have been several other NLP-related approaches for predicting item characteristics. These approaches have so far been less successful than the ones already described based on linguistic features and embeddings; however, for greater context, a short description of these is given next.

The first approach predicts item difficulty using TF-IDF (Term Frequency–Inversed Document Frequency) representation using a tool called R2D2 (Benedetto et al., 2020). TF-IDF is a well-known early approach to text representation that relies on sparse co-occurrence vectors of words in a document. Later experiments by the same authors show that R2D2 is outperformed by contextualized embeddings such as BERT (Benedetto et al., 2021).

Another approach reported in Ha et al. (2019) used the output of an automatic question-answering system[15] for predicting item difficulty. This approach was based on the hypothesis that there is a positive relationship between the difficulty of questions for humans and their difficulty for machines. An information retrieval–based automated question-answering system was applied to a set of MCQs, and the retrieval scores from that system were used as predictors for item difficulty. While subsequent experiments showed that these predictors had low utility in the final models, exploiting question-answering systems in this way may be a promising direction for future work should the aforementioned hypothesis be true.

3. Experiments

The previous sections provided a brief overview of different approaches for extracting linguistic information from item text for the task of predicting item characteristics. In this section, we illustrate the approaches outlined earlier through several experiments for predicting different item characteristics.

Table 10.4 Example of a Practice Item

Item Stem	A 16-year-old boy is brought to the emergency department because of a 2-day history of fever, nausea, vomiting, headache, chills, and fatigue. He has not had any sick contacts. He underwent splenectomy for traumatic injury at the age of 13 years. He has no other history of serious illness and takes no medications. He appears ill. His temperature is 39.2°C (102.5°F), pulse is 130/min, respirations are 14/min, and blood pressure is 110/60 mm Hg. On pulmonary examination, scattered crackles are heard bilaterally. Abdominal examination shows a well-healed midline scar and mild, diffuse tenderness to palpation. Which of the following is the most appropriate next step in management?
Item Options	a. Antibiotic therapy* b. Antiemetic therapy c. CT scan of the chest d. X-ray of the abdomen e. Reassurance

Note: The asterisk denotes the correct answer, also known as item key.
Source: www.usmle.org/sites/default/files/2021-10/Step2_CK_Sample_Questions.pdf

3.1 Data

Data were collected between 2010 and 2015 and comprised approximately 19,000 pretest MCQs from the Step 2 Clinical Knowledge component of the United States Medical Licensing Examination (USMLE), an exam sequence taken by medical doctors as a requirement for licensure in the United States. Each exam included unscored pretest items that were presented alongside scored items. Test-takers had no way of knowing which items were scored and which were unscored pretest items. On average, each item was answered by 335 first-time examinees who were medical students from accredited[16] U.S. and Canadian medical schools.

An example test item from this exam is shown in Table 10.4. All items tested medical knowledge and were written by experienced item writers following a set of guidelines that specified a standard structure and prohibited the use of verbose language, extraneous material not needed to answer the item, information designed to mislead the test-taker, and grammatical cues such as correct answers that are longer or more specific than other options. Standards for style were also imposed, including consistent vocabulary and consistent formatting and presentation of numeric data.

Several item characteristics were computed for these items based on the responses received during pretesting. These included:

- *P-value:* The proportion of correct responses for a given item computed as:

$$p_i = \frac{\sum_{n=1}^{N} u_n}{N}$$

  where p_i is the p-value for item i, u_n is the 0–1 score (incorrect-correct) on item i earned by examinee n, and N is the total number of examinees in the sample.
- *Mean response time:* The average time (measured in seconds) that examinees spent viewing an item.
- r_b : The biserial correlation coefficient between examinees' responses on the given item and examinees' total test score. For a given item i, this may be calculated as follows:

$$r_{b_i} = \frac{(\mu_+ - \mu_X)}{\sigma_X}(p_i / y)$$
,

where μ_+ is the mean examinee test score *for those examinees responding correctly to item i*; μ_X is the mean examinee test score for all examinees; σ_X is the standard deviation of these scores; p_i retains its meaning from earlier; and y is the ordinate of the standard normal curve at the z-score associated with p_i. Equivalently, r_{b_i} may be expressed as the product of the Pearson product moment coefficient (between the examinee item score and test score) and

$$y^{-1}\sqrt{p_i(1-p_i)}.$$

3.2 Predictors

We report results based on each of the three types of predictors described in Section 2: human-engineered linguistic features, non-contextualized embeddings, and contextualized embeddings. The human-engineered linguistic features used here include several levels of linguistic processing and are summarized in Table 10.5. A complete list, including details on their computation, is found in Baldwin et al. (2021) and Yaneva et al. (2021).

Word2Vec (300 dimensions) was used for non-contextualized embeddings as described in Section 2.

Several models for contextualized embeddings were investigated. These included BERT Base and BERT Large (Devlin et al., 2018; described in Section 2) trained on the BooksCorpus (800 million words; Zhu et al., 2015) and English Wikipedia (2,500 million words). In addition to the BERT Base models trained on generic data, we also use a BERT Base model trained on clinical text from PubMed Central[17] and MEDLINE abstracts[18] (as in Gu et al., 2020). Training two separate BERT Base models on two types of data – generic and biomedical – allows for direct comparison of the effects of data domain for model performance on our task.

Finally, we also report results using RoBERTa (Liu et al., 2019), which is based on a more advanced architecture. This model was trained on the biomedical data described earlier to evaluate the effects of model architecture on performance. As mentioned in Section 2, RoBERTa represents a version of BERT with improved hyperparameter tuning.

Table 10.5 Summary of the Human-Engineered Linguistic Features by Level of Linguistic Processing

Linguistic Processing Level	Feature Count	Examples
Lexical	5	Word count; Average word length in syllables; Complex word count
Syntactic	29	Part of speech (POS) count; Average sentence length; Average number of words before the main verb Passive-active ratio
Semantic	11	Polysemic word count; Average senses for nouns; Average senses for verbs
Readability[a]	7	Flesch Reading Ease; Flesch-Kincaid grade level; Automated Readability Index; Gunning Fog
Cognitive[b]	14	Concreteness ratings; Imageability ratings; Familiarity ratings
Frequency[c]	10	Average word frequency; Words not in the first 2,000 most common words; Words not in the first 4,000 most common words
Cohesion	5	Temporal connectives count; Causal connectives count; Referential pronoun count
Specialized Clinical Features	8	Unified Medical Language System Metathesaurus terms count [d]

[a] See Dubay (2004) for formula definitions.

[b] *Source:* MRC Psycholinguistic Database (Coltheart, 1981).

[c] *Source:* British National Corpus (Leech et al., 2014).

[d] UMLS; Number of terms in an item that appear in the UMLS Metathesaurus (Schuyler et al., 1993).

3.3 Analysis

Several models for predicting each of the three item characteristics – p-value, mean response time, and biserial correlation (r_b) – were constructed using the three classes of ancillary data just described (linguistic features, non-contextualized embeddings, and contextualized embeddings).[19] For each of these models, 80% of the data were used for model training and the remaining 20% was used as a test set. Item characteristics were estimated using pretest data, and these empirical values were treated as truth for the purpose of model evaluation. Root Mean Squared Error (RMSE) was calculated for each model's predicted item characteristics.

To allow comparisons between the linguistic features and embeddings, the prediction step was not part of the embedding architecture but rather was done separately using several regressor algorithms from Python's *scikit-learn* library (Pedregosa et al., 2011): *linear regression*, *support vector regressor (SVR)*, *elastic net*, and *random forests (RF)*.[20] For elastic net, the alpha value was varied as a study condition and included 0.01, 0.03, and 0.05; likewise, for RF, number of trees was varied as study condition and included 100, 200, 300, and 400. Elastic net and RF were selected for their variance reduction ability in datasets with large numbers of input features.

Predictions were compared with a ZeroR baseline. This baseline is computed by taking the mean of the dependent variable (i.e., p-value, mean response time, or r_b) for the training set and treating it as the predicted value for the items in the test set. Predictions would need to outperform this baseline to be considered potentially useful.

3.4 Results

The results for modeling p-value, mean response time, and r_b are presented in Figures 10.1, 10.2, and 10.3, respectively. As can be seen, the results show that the values of the variables extracted from item text contain signal that can be used to predict item characteristics. The p-value and mean response time parameters were predicted with a significant improvement over the ZeroR baseline (especially mean response time): RMSE of .218 compared to .241 for p-value and RMSE of 23.3 compared to 32.9 for mean response time. The r_b predictions were less successful – only showing a small improvement over baseline (RMSE of .152 compared to .159) – making it the most challenging parameter to model among the parameters reported here.

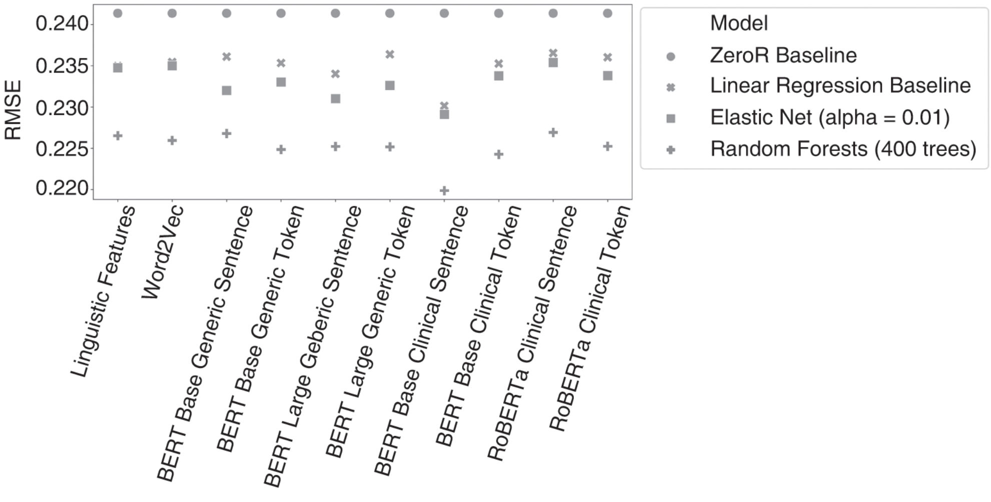

Figure 10.1 Results from various predictors and models for modeling p-value.

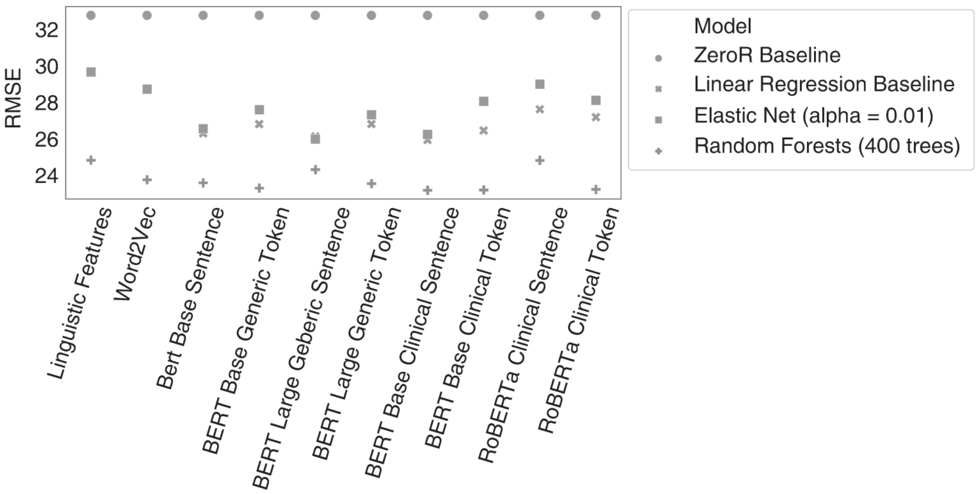

Figure 10.2 Results from various predictors and models for modeling mean response time.

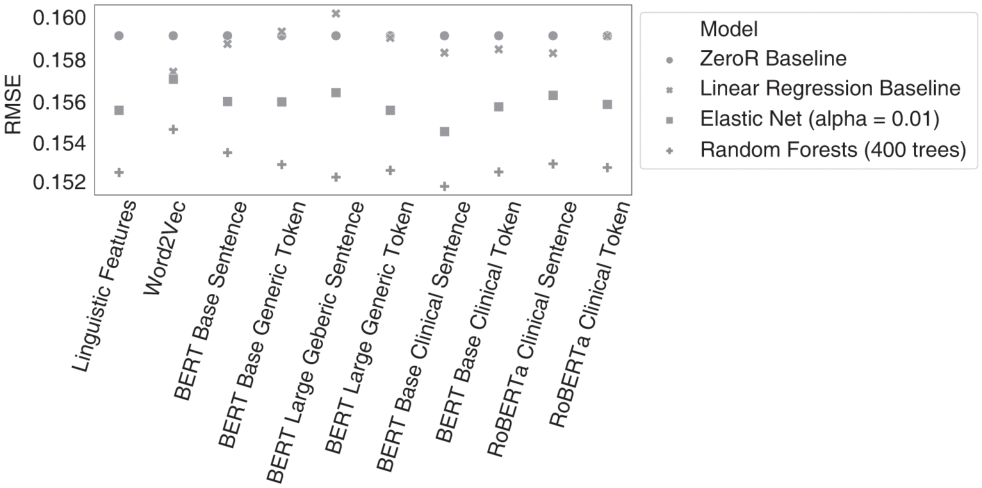

Figure 10.3 Results from various predictors and models for modeling r_b.

The sentence embeddings from the BERT Base model trained on clinical data performed best for predicting p-value and were among the best performing models for predicting response time and r_b. In contrast, models trained on generic data generally performed better with token embeddings than with sentence embeddings. The other model trained on clinical data, RoBERTa, did not perform as well as BERT Base. More than choice of model or training data, however, the regressor algorithm had the greatest effect on prediction quality, with random forests consistently outperforming all other models on all three tasks.[21]

The implications of these results for the different ways to extract signal from item text are discussed in the next section.

4. Discussion

This chapter set out to achieve two goals: to provide an overview of NLP approaches to item representation and to illustrate these approaches in the context of predicting item characteristics for clinical MCQs. The overview traced the transition from human-engineered linguistic features to non-contextualized and eventually contextualized embeddings; and the experimental results demonstrated that in general, small gains were associated with these historical developments. The experiments consistently showed that sentence embeddings from the BERT Base model trained on clinical data (BERT Base Clinical Sentence) were the best performing configuration for all tasks (although the improvements over some models were not always statistically significant).

It is conceivable that training models on generic data suffered because not all specialized medical terms from the clinical MCQs were present in the training corpus. While contextualized embeddings – like those produced by the BERT model – can process out-of-vocabulary data through subword modeling, this technique requires more computational resources[22] and may lead to lower performance. For example, as Gu et al. (2020) note, subword modeling of common medical terms such as *naloxone* first requires breaking naloxone into subword units (e.g., [na, ##lo, ##xon, ##e]) and then modeling it through these subwords. This is avoided when the training data comes from the biomedical domain, where common biomedical vocabulary such as *naloxone* are likely to be present. Gu et al. (2020) show that domain-specific pretraining using biomedical data can substantially outperform pretraining using generic data. Moreover, they note that even biomedical data alone outperforms generic + biomedical data, and they hypothesize that *the two domains are so different that negative transfer may occur if the representations are first learned on the generic data (i.e., performance may be hurt if the knowledge learned from the generic data does not apply to the specialized domain)*. The results presented here add more evidence in support of the hypothesis that domain-specific data result in superior performance. This may not be the case for predicting item characteristics in other assessment domains. Nevertheless, it highlights the use of domain-specific versus generic data as an important choice to investigate when pretraining models.

The results suggest that there is no straightforward way of deciding which architecture is best for a given task, as bigger and more robustly optimized models were not necessarily the best-performing ones. This was evident with both generic and biomedical training data, where comparisons between BERT Base and BERT Large – trained on the same generic data – did not provide clear evidence in support of the larger model; and comparison between BERT Base and RoBERTa, which were both trained on biomedical data, did not favor RoBERTa despite the additional hyperparameter tuning. This suggests that model size and parameter optimization are not necessarily the barrier preventing improvements. It is therefore advisable that researchers experiment with various architectures and make their final selection based on empirical results.

In terms of the type of encoded context, the fact that the sentence embedding from the BERT Base model trained on clinical data outperforms the token embedding from the same model shows that predicting item characteristics benefits from encoding larger context and longer dependencies as opposed to the shorter ones, characteristic of token embeddings. Since the BERT model is a bidirectional encoder[23] (as suggested by its name), it also shows the importance of capturing context from not only the prior tokens, but also the token that follows. It is interesting to observe that the sentence embeddings from RoBERTa perform consistently worse than those from BERT (and from RoBERTa token embeddings). This can be explained with a modification introduced in RoBERTa on how sentence embeddings are encoded. As described in Section 2, the BERT architecture is trained on two tasks: next token prediction and next sentence prediction. The authors of RoBERTa note that the next-sentence-prediction task, originally designed to improve performance on tasks that require reasoning about the relationships between pairs of sentences, could be removed, and that sentence embeddings can be generated without this objective. The superior performance of the BERT Base sentence embeddings suggests that reasoning about the relationships between pairs of sentences, learned from domain-specific data, is important to the task of predicting item characteristics.

One important reason for the comparative underutilization of embeddings in educational measurement is that the field is traditionally concerned with model interpretability, and embeddings offer very little information about the contribution of specific variables. While model interpretability is undoubtedly crucial for some applications, the trade-off between interpretability and accuracy may be more balanced in the area of predicting item characteristics (in fact, one may argue that accuracy is all that matters for improving pretesting or evaluating automatically generated items). Therefore, predicting item characteristics is one assessment area that can take better advantage of more sophisticated text representations such as embeddings. Should model interpretability be the focus, linguistic features, while generally not the highest-performing predictors, can provide greater insight into the relationships between various interpretable characteristics of items and various outcomes of interest. A good example of the value that linguistic features add beyond predictive performance is a study that uses linguistic features extracted from MCQs to gain insight into the cognitive complexity associated with answering the items and its relationship to item text (Yaneva et al., 2021). Apart from use cases where interpretability matters, our experiments support the now widely accepted idea that contextualized embeddings perform better than linguistic features and non-contextualized embeddings for many tasks.

The framework presented here has several limitations that merit discussion, mainly related to approaches that could have improved results but that were not shown here. These include combining different predictor types within a single model, or experimenting with other types of embedding models, including ones specifically developed for biomedical text. We also could have fine-tuned the models as opposed to using the embeddings they produced as input for a prediction model like random forests. While further research into each of these areas could be beneficial, the goal of this chapter was more modest: we merely focused on providing an overview of the most widely used, accessible, and well-known strategies in the field of NLP. This choice was motivated in part by the rapid pace of NLP research, which suggests that even bigger and more successful models for text representation will be in use by the time this chapter is published. It is our hope that a more foundational introduction to NLP approaches – particularly low-interpretability embeddings models – and their potential application to assessment problems will be more valuable to a non-NLP audience than a showcase of the latest language models. Readers are advised to view this chapter as a framework for item text representation rather than as a source of guidance about the most recent models and architectures.

The practical significance of these results will depend on specific use cases. For example, Baldwin et al. (2021) show that the use of the aforementioned approaches to predict item response time can help improve exam fairness. The results from this study indicate that if forms are assembled considering predicted response times for newly developed pretest items, overall timing variability for test forms can be reduced by 2 to 4 minutes. In contrast, the practical value of predicted p-values and Rb parameters has not yet been demonstrated, and the current results are most useful as an exploration of the type of predictive power different features or representations have with a view to optimizing the results.

One area for future research relates to recent advances in automated question answering such as those utilizing the T5 transformer model (Raffel et al., 2019). As was noted in Section 2, it may be that items that are more difficult for humans to answer also may be more difficult for machines to answer, and this relationship could be used to predict item difficulty and response time. Even if improvements in automated question answering do not lead to improvements in item parameter modeling on their own, predictors related to machine performance could potentially complement, rather than overlap with, the other approaches described in this chapter. While not yet explored in depth, combining signals from multiple sources in this way may be a promising direction for future research in modeling item characteristics.

5. Conclusion

This chapter discussed the evolution of text representation and demonstrated the use of three types of representations – linguistic features, non-contextualized word embeddings, and contextualized token and sentence embeddings – for the tasks of predicting p-values, mean response times, and r_b correlations for nearly 19,000 clinical MCQs. The empirical results suggested that, at least within the domain of clinical MCQs, it is beneficial to pre-train models on biomedical text, and that when this is done, encoding larger context and longer dependencies can improve results. Using larger models or ones with improved hyperparameter tuning does not necessarily lead to improved predictions and so, ideally, a range of architectures should be experimented with before selecting the best-performing one for a given problem. The task of modeling item parameters is one way in which the field of assessment can capitalize on the advances in text representation that have recently transformed the field of NLP and its numerous applications in our everyday lives.

Notes

[1] Word embeddings, which are described in greater detail in Section 2.2, refer to several techniques for representing words based on their usage such that words with similar meanings have similar representations.

[2] In most cases, the development of accurate NLP tools for the extraction of specific linguistic features first requires the availability of specific NLP resources. For example, part-of-speech-taggers are tools that automatically identify the parts of speech of words in a text, but their development and efficacy depend on the availability of resources like the Penn Treebank (Marcus et al., 1993), which contain large numbers of words manually labeled with their corresponding part of speech.

[3] Words with more than one meaning.

[4] Words may be divided into various subcomponents or *subwords*. For example, 'readable' can be divided into 'read' and 'able'.

[5] How meaning is constructed in specific contexts.

[6] Here, we describe the process using the skip-gram architecture with negative sampling, which generally works well with large datasets; however, its (in some sense) inverse architecture, *continuous bag of words*, also can be used.

[7] Note that we have skipped the articles 'a' and 'the' during the preprocessing stage, where certain stopwords such as 'a', 'an', and 'the' are removed. This is optional depending on the application; however, in many tasks, stopwords are not highly informative for context.

[8] The sampled distribution is sometimes called the *noise distribution*. Mikolov et al. (2013) suggest using the unigram distribution raised to the power of ¾, which reflects each word's frequency in the corpus, as the noise distribution.

[9] The number of embeddings is usually defined empirically through trial and error.

[10] Mikolov et al. (2013) propose 2–5 for large samples.

[11] In the case of items, word embeddings must be pooled together. This can be done, for example, using element-wise averages, minimums, or maximums for all vectors (i.e., average-pooling, min-pooling, and max-pooling, respectively).

[12] We note that some of the earlier contextualized embeddings such as ELMo (Peters et al., 2018) are not generated using transformer models.

[13] https://github.com/google-research/bert

[14] www.nlm.nih.gov/medline/medline_overview.html

[15] Automated Question Answering is an NLP application, where the goal is to develop systems that can automatically answer various types of questions, including open-ended ones in reading comprehension exams, fora, or search queries; MCQs; true or false questions, etc.

[16] Accredited by the Liaison Committee on Medical Education (LCME).

[17] www.ncbi.nlm.nih.gov/pmc/

[18] www.nlm.nih.gov/medline/medline_overview.html

[19] As noted earlier, we use the embeddings as input to classic machine learning algorithms rather than finetuning BERT and RoBERTa models. This is done for two reasons: (1) preliminary experiments showed that this approach leads to better results for our data; and (2) this allows for fairer comparisons between models (especially with linguistic features, which cannot be fine-tuned).

[20] Regressor algorithms were used with default parameters.

[21] Performance improved with increases in the number of trees up to 400; further increases did not lead to additional meaningful gains (e.g., p-value RMSE of .216 with 1,000 trees compared to .218 with 400 trees).

[22] This limitation can be addressed if the necessary computational resources are available, which may not always be the case in practice for many research institutions.

[23] Models that learn information from left to right and from right to left.

References

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W. H., Jin, D., Naumann, T., & McDermott, M. (2019). Publicly available clinical BERT embeddings. *arXiv*. https://arxiv.org/abs/1904.03323

Baldwin, P., Yaneva, V., Mee, J., Clauser, B. E., & Ha, L. A. (2021). Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*, *58*(1), 4–30.

Benedetto, L., Aradelli, G., Cremonesi, P., Cappelli, A., Giussani, A., & Turrin, R. (2021, April). *On the application of Transformers for estimating the difficulty of multiple-choice questions from text* (pp. 147–157). Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications.

Benedetto, L., Cappelli, A., Turrin, R., & Cremonesi, P. (2020, March). *R2DE: A NLP approach to estimating IRT parameters of newly generated questions* (pp. 412–421). Proceedings of the Tenth International Conference on Learning Analytics & Knowledge.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *arXiv*. https://arxiv.org/abs/2006.09882

Choi, I. C., & Moon, Y. (2020). Predicting the difficulty of EFL tests based on corpus linguistic features and expert judgment. *Language Assessment Quarterly*, *17*(1), 18–42.

Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv*. https://arxiv.org/abs/2003.10555

Coltheart, M. (1981). The MRC psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A*, *33*(4), 497–505.

Conneau, A., & Kiela, D. (2018). Senteval: An evaluation toolkit for universal sentence representations. *arXiv*. https://arxiv.org/abs/1803.05449

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv*. https://arxiv.org/abs/1705.02364

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*. https://arxiv.org/abs/1810.04805

DuBay, W. H. (2004). *The principles of readability*. Online Submission. DoE.

Firth, J. R. (1957). A synopsis of linguistic theory 1930–1955. In *Studies in linguistic analysis* (pp. 1–32). Philological Society. Reprinted in Palmer, F. R. (Ed.). (1968). *Selected papers of J. R. Firth 1952–1959*. Longman.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., & Poon, H. (2020). Domain-specific language model pretraining for biomedical natural language processing. In *ACM transactions on computing for healthcare (HEALTH)*. ACM.

Ha, L. A., Yaneva, V., Baldwin, P., & Mee, J. (2019, August). *Predicting the difficulty of multiple choice questions in a high-stakes medical exam* (pp. 11–20). Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications.

Harris, Z. (1954). Distributional structure. *Word*, *10*(23), 146–162.

He, P., Liu, X., Gao, J., & Chen, W. (2020). DeBERTA: Decoding-enhanced BERT with disentangled attention. *arXiv*. https://arxiv.org/abs/2006.03654

Kurdi, G. R. (2020). *Generation and mining of medical, case-based multiple choice questions*. The University of Manchester.

Leech, G., & Rayson, P. (2014). *Word frequencies in written and spoken English: Based on the British national corpus*. Routledge.

Leo, J., Kurdi, G., Matentzoglu, N., Parsia, B., Sattler, U., Forge, S., Donato, G., & Dowling, W. (2019). Ontology-based generation of medical, multi-term MCQs. *International Journal of Artificial Intelligence in Education*, *29*(2), 145–188.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *arXiv*. https://arxiv.org/abs/1907.11692

Loukina, A., Yoon, S. Y., Sakano, J., Wei, Y., & Sheehan, K. (2016, December). *Textual complexity as a predictor of difficulty of listening items in language proficiency tests*. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers.

Marcus, M., Santorini, B., & Marcinkiewicz, M. (1993). Building a large annotated corpus of English: The Penn treebank. *Computational linguistics-Association for Computational Linguistics (Print)*, *19*(2), 313–330.

McDonald, S., & Ramscar, M. (2001). *Testing the distributional hypothesis: The influence of context on judgements of semantic similarity* (Vol. 23, No. 23). Proceedings of the Annual Meeting of the Cognitive Science Society.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 3111–3119.

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Pennington, J., Socher, R., & Manning, C. D. (2014, October). *Glove: Global vectors for word representation* (pp. 1532–1543). Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). *Deep contextualized word representations* (pp. 2227–2237). Proceedings of NAACL-HLT.

Pilehvar, M. T., & Camacho-Collados, J. (2020). Embeddings in natural language processing: Theory and advances in vector representations of meaning. *Synthesis Lectures on Human Language Technologies*, *13*(4), 1–175. Morgan & Claypool Publishers.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv*. https://arxiv.org/abs/1910.10683

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv*. https://arxiv.org/abs/1910.01108

Schuyler, P. L., Hole, W. T., Tuttle, M. S., & Sherertz, D. D. (1993). The UMLS metathesaurus: Representing different views of biomedical concepts. *Bulletin of the Medical Library Association*, *81*(2), 217.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., & Rush, A. M. (2020, October). *Transformers: State-of-the-art natural language processing* (pp. 38–45). Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.

Xue, K., Yaneva, V., Runyon, C., & Baldwin, P. (2020, July). *Predicting the difficulty and response time of multiple choice questions using transfer learning* (pp. 193–197). Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications.

Yaneva, V., Ha, L. A., Baldwin, P., & Mee, J. (2020, May). *Predicting item survival for multiple choice questions in a high-stakes medical exam* (pp. 6812–6818). Proceedings of the 12th Language Resources and Evaluation Conference.

Yaneva, V., Jurich, D., Ha, L. A., & Baldwin, P. (2021, April). *Using linguistic features to predict the response process complexity associated with answering clinical MCQs* (pp. 223–232). Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). *Aligning books and movies: Towards story-like visual explanations by watching movies and reading books* (pp. 19–27). Proceedings of the IEEE International Conference on Computer Vision.