POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

AUTOMATIC SHORT ANSWER GRADING USING TRANSFORMERS

HADI ABDI GHAVIDEL
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Février 2021

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

AUTOMATIC SHORT ANSWER GRADING USING TRANSFORMERS

présenté par **Hadi ABDI GHAVIDEL**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Benoit OZELL, président
Amal ZOUAQ, membre et directrice de recherche
Quentin CAPPART, membre

DEDICATION

Dedicated to my parents, sisters, nieces and all whom I have an eternal love for ...

ACKNOWLEDGEMENTS

Natural language processing (NLP) and computational linguistics (CL) have become a part of my life and soul since 2011. I was happy when I entered this field and I'm happier now that I'm in this community. During the past years of my NLP and CL life, I learned a lot and am still learning now. I gained a lot of knowledge from people in this community. Among these people, one of them was considerably influential. Here, I would like to express my deep appreciation for all knowledge, supervision and support I received from this person. She is my most vibrant supervisor who is Dr. Amal Zouaq.

RÉSUMÉ

L'évaluation des réponses courtes en langage naturel est une tendance dominante dans tout environnement éducatif. Ces techniques ont le potentiel d'aider les enseignants à mieux comprendre les réussites et les échecs de leurs élèves. En comparaison, les autres types d'évaluation ne mesurent souvent pas adéquatement les compétences des élèves, telles que les questions à choix multiples ou celles où il faut combler des espaces. Cependant, ce sont les moyens les plus fréquemment utilisés pour évaluer les élèves, en particulier dans les environnements de cours en ligne ouverts (MOOCs). La raison de leur emploi fréquent est que ces questions sont plus simples à corriger avec un ordinateur. Comparativement, devoir comprendre et noter manuellement des réponses courtes est une tâche plus difficile et plus longue, d'autant plus en considérant le nombre croissant d'élèves en classe. La notation automatique de réponses courtes, généralement abrégée de l'anglais par ASAG, est une solution parfaitement adaptée à ce problème. Dans ce mémoire, nous nous concentrons sur le ASAG basé sur la classification avec des notes nominales, telles que correct ou incorrect. Nous proposons une approche par référence basée sur un modèle d'apprentissage profond, que nous entraînons sur quatre ensembles de données ASAG de pointe, à savoir SemEval-2013 (SciEntBank et BEETLE), Dt-grade et un jeu de données sur la biologie. Notre approche utilise les modèles BERT Base (sensible à la casse ou non) et XLNET Base (seulement sensible à la casse). Notre analyse subséquente emploie les ensembles de données GLUE (General Language Understanding Evaluation), incluant des tâches de questions-réponses, d'implication textuelle, d'identification de paraphrases et d'analyse de similitude textuelle sémantique (STS). Nous démontrons que celles-ci contribuent à une meilleure performance des modèles sur la tâche ASAG, surtout avec le jeu de données SciEntBank. Nous montrons par la suite que les modeles de type Transformers, tels que BERT et XLNET, surpassent (ou égalent) les approches de pointe basées sur l'ingénierie d'attributs. Nous indiquons en outre que les performances de notre modèle BERT augmentent considérablement lorsque nous raffinons un modèle ASAG après un entraînement sur la tâche d'implication textuelle avec le jeu de données MNLI, par rapport aux autres modèles entraînés sur d'autres jeux de données provenant de GLUE.

ABSTRACT

Assessment of short natural language answers is a prevailing trend in any educational environment. It helps teachers to understand better the success and failure of students. Other types of questions such as multiple-choice or fill-in-the-gap questions don't provide adequate clues for evaluating the students' proficiency exhaustively. However, they are common means of student evaluation especially in Massive Open Online Courses (MOOCs) environments. One of the major reasons is that they are fairly easy to be graded. Nonetheless, understanding and marking manually short answers are more challenging and time-consuming tasks, especially when the number of students grows in a class. Automatic Short Answer Grading, usually abbreviated to ASAG, is a highly demanding solution in this current context. In this thesis, we mainly concentrate on classification-based ASAG with nominal grades such as correct or not correct. We propose a reference-based approach based on a deep learning model on four ASAG state-of-the-art datasets, namely SemEval-2013 (SciEntBank and BEETLE), Dt-grade and Biology dataset. Our approach is based on BERT (cased and uncased) and XLNET (cased) models. Our secondary analysis includes how GLUE (General Language Understanding Evaluation) tasks such as question answering, entailment, paraphrase identification and semantic textual similarity analysis strengthen the ASAG task on SciEntBank dataset. We show that language models based on transformers such as BERT and XLNET outperform or equal the state-of-the-art feature-based approaches. We further indicate that the performance of our BERT model increases substantially when we fine-tune a BERT model on an entailment task such as the GLUE MNLI dataset and then on the ASAG task compared to the other GLUE models.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

ASAG	Automatic Short Answer Grading
BERT	Bidirectional Encoder Representations from Transformers
BLEU	Bilingual Evaluation Understudy
BOW	Bag of Word
CNN	Convolutional Neural Network
ELMo	Embeddings from Language Models
GLUE	General Language Understanding Evaluation
LDA	Latent Dirichlet Allocation
LRS	Lexical Resource Semantics
LSA	Latent Semantic Analysis
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
MNLI	Multi-Genre Natural Language Inference
MOOCs	Massive Open Online Courses
MRPC	Microsoft Research Paraphrase Corpus
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
POS	Part of Speech
QNLI	Question Natural Language Inference
RNN	Recurrent Neural Network
RTE	Textual Entailment Recognition
SOTA	State-of-the-art
SQuAD	The Stanford Question Answering Dataset
STS-B	Semantic Textual Similarity Benchmark
TF-IDF	Term Frequency-Inverse Document Frequency
XLNET	Extra Long Network

CHAPTER 1 INTRODUCTION

In this chapter, we describe the motivation for our work in section 1.1. Then, we explain our goal and major contributions in sections 1.2 and 1.3. The last section of this chapter is devoted to the description of our thesis outline.

1.1 Motivation

Over the past years, the load of online education increased to a significant extent due to the emergence of MOOCs (Massive Open Online Courses) [4]. Since 2020, this has been increased in consequence of COVID-19 (Coronavirus disease 2019) global pandemic as the universities changed temporarily the process of teaching and learning into a distance mode[1]. This heightens the need to improve the way institutions, universities and schools are:

- imparting knowledge to a large number of people as there is an ongoing exponential increase of demand in the provision of higher education[2]
- providing feedback (formative and summative), especially for open responses
- organizing the communication among participants (for example discussion forums, teamwork channels such as Slack channel[3] or Discord channel[4], etc.)

In this thesis, we target the provision of feedback for open responses and design a pipeline to evaluate these types of responses.

There are several types of questions for evaluating the student's knowledge. Some examples of these questions are multiple-choice questions, fill-in-the-gap questions and open-ended questions [5]. Within the MOOCs platforms, the evaluation of student knowledge is mostly based on multiple-choice questions [4]. These types of questions limit the evaluation of students' understanding only to a selection of an answer from several possible alternatives [6]. Similarly, the fill-in-the gap questions bear the same consequences. However, open-ended questions have been proved to provide teachers with a more accurate and detailed understanding of how a student comprehends domain-specific knowledge [7]. Grading a large number of answers for multiple-choice or fill-in-the-gap questions is quite straightforward

[1] https://globalnews.ca/news/6935364/coronavirus-canadian-university-fall-classes/
[2] https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=3710001801
[3] https://slack.com/
[4] https://discord.com/

when compared to the open-ended questions. Therefore, more complex automatic methods are necessary for grading the open-ended types. These methods include machine learning and NLP (natural language processing) algorithms, which eases the process of evaluating a large number of students using a wide variety of questions in different domains.

According to Alvarado [8], the evaluation of open-ended questions includes several bottlenecks. Some of the most important are subjective grading schemes, extension of domain-specific grading pipelines to other domains and finally understanding questions and answers written in natural language. All these bottlenecks exist in essay type or short type of student answers, which highlights the importance of automated assessment of open-ended questions. The challenge becomes even harder in automatic short answer grading (ASAG) where the length of the answer is short and the answer may lack enough context for an adequate understanding of the text.

1.2 Goal

Generally, there are two types of approaches to solve an ASAG task. They are called *response-based* and *reference-based* approaches. According to [9], the response-based approach only considers student answers while the reference-based approach examines the relationship between student answer and teacher-provided answer.

Similar to [10], we consider the ASAG as a textual entailment task. We assume that the student's correct answer (which is called *text or premise* in the entailment framework) implies the meaning of the reference answer (which is called *hypothesis* in the entailment framework). In this work, we adopt the reference-based approach based on [9] in which we attempt to find a relation between each student answer and teacher-provided reference (also called model[5]) answer.

According to Jurafsky and Martin [11], language models are highly useful in several NLP tasks such as speech recognition, machine translation, etc. In particular, neural language models has recently aided the performance of the textual entailment tasks ([12, 13]). In this thesis, we propose to learn the entailment between the student answer and the model answer using two modern language models namely BERT (Bidirectional Encoder Representations from Transformers) [12] and XLNET (Extra Long Network) [13]. Our experiments are based on ASAG datasets that contain nominal grades such as correct, incorrect, etc. In other words, we consider ASAG as an automated classification task. In this sense, our task is a supervised machine learning problem with gold labels (nominal grades) for training an ASAG model.

[5] We use reference answer and model answer interchangeably throughout the whole thesis

As we consider the ASAG task as textual entailment and it involves understanding, we also explore how transfer learning from highly robust understanding tasks help us achieve better performance in ASAG. Alternatively stated, we reuse the models trained for these tasks to improve generalization for the ASAG task. In this regard, we use the top-performing tasks within GLUE (General Language Understanding Evaluation)[6] benchmark which is a group of tasks mainly concentrated on natural language understanding. The tasks are MNLI (Multi-Genre Natural Language Inference), SQuAD (The Stanford Question Answering Dataset), STS-B (Semantic Textual Similarity Benchmark), QNLI (Question Natural Language Inference), MRPC (Microsoft Research Paraphrase Corpus) and RTE (Textual Entailment Recognition). Overall, we fine-tune the trained GLUE models on the ASAG task.

We seek to answer the following research questions:

1. How do modern language models perform on the ASAG task?
2. How do NLP tasks and transfer learning impact the ASAG task?

1.3 Contributions

The major contributions of our research are as follows:

- We show that modern language models considerably exceed the SOTA (state-of-the-art) results even though we carry out the task of grading without taking advantage of any type of human engineered features.
- We show that transfer learning based on GLUE has a fairly favourable impact on the ASAG task, especially when our language model is pre-trained on the GLUE inference task which is MNLI.

1.4 Outline of the Thesis

The structure of the thesis is as follows: In chapter 2, we provide a background for our thesis and review previous works conducted in ASAG. In chapter 3, we propose our methodology and give the details on the datasets, the experimental setup and our evaluation measures. Then, we present and analyze our results while comparing them to the SOTA in chapter 4. We discuss how our pipelines boost ASAG and make an error analysis in chapter 5. In this chapter, we also describe our conclusion, limitations and future research avenues.

[6] https://gluebenchmark.com/tasks

CHAPTER 2 BACKGROUND AND RELATED WORK

Throughout this chapter, we describe a background and SOTA for ASAG. Firstly, we present a brief overview of the types of assessments in MOOCs. Then, we define the concept of short answer and how it is graded. Thirdly, we summarize the ASAG SOTA and discuss the top performing systems. In the sections 2.5 and 2.6, we describe language models and transfer learning. Finally, we present a summary of the chapter.

2.1 MOOCs Assessments

Evaluation of students' learning progress is an indispensable part of any type of learning process. Both students and teachers benefit from assessment. Students become aware how their ongoing learning progress is. Teachers analyze whether or not the objectives of the course are met well.

In general, assessment is carried out in two forms: formative and summative assessments. According to [14, 15], formative assessment includes the provision of immediate feedback during the different phases of the course instruction to enhance the effective learning process. The second form is summative assessment for which the students receive cumulative feedback at the end of the course. This feedback is considered as the teacher's most final judgment.

In a broad sense, both these assessments include different types of questions such as multiple-choice questions or open-ended questions. In MOOCs, most commonly known example of both formative and summative assessments is a set of multiple-choice questions as they are fairly easy to be graded at large scale thus less labor-intensive. Meanwhile, open-ended questions such as short-answer questions, discussion platforms and peer assessments are quite appropriate for demonstrating the highly abstract layers of learner's knowledge [16]. The comprehensive evaluation of large number of students using these types of questions is rather costly both in terms of heavy computation and enormous expenses. In this regard, Page [17] appreciated the importance of building computerized open examinations which significantly reduce the aforementioned costs. In section 2.2.1 and section 2.2.2, we will define precisely what we mean by open-ended questions (with a focus on short answer types).

2.2 Automatic Short Answer Grading

In this section, we clarify the definition of short answer and explain how ASAG is carried out in general.

2.2.1 Short Answer

Based on the definition provided by Burrows et al. [5], the current thesis focuses on the recall questions which motivated the students to cast their mind back to what they have learned in the class and to write their answers in natural language. These types of questions fall into the higher level of learning in Bloom's taxonomy of learning objectives [18] to natural language.

In the current thesis, we consider the above-mentioned definition of the short answers with the following distinctive characteristics [5]:

- The elements of the question such as words and phrases should not help the student to guess the answer
- The length of the answer should not exceed the approximate length of a paragraph.
- The evaluation of the answer should be based on content of the answer rather than the form of writing.

2.2.2 ASAG

The overall structure of ASAG is depicted in Figure 2.1. We decomposed the ASAG task into different components. ASAG generally takes as input a question, a reference answer and a student answer.

As explained in section 1.2 of chapter 1, two main representation schemes have been observed for ASAG in the literature [19]: the response-based and reference-based representation. These two representations can be exploited in a supervised learning or unsupervised learning approach whose aim is to determine if the answer is correct or not correct. Inspired by the definition given by Goodfellow et al. [20], supervised ASAG can be defined as a learning algorithm which learns to relate the student answers with some grades (labels) when provided with a dataset of inputs and outputs. In many cases, the grades are provided by a human expert. However, the task is still called supervised learning even when the grades are collected in an automated manner. Following a distinction made between supervised and unsupervised machine learning task by Goodfellow et al. [20], unsupervised ASAG can be defined as the extraction of information from student answers for which there is no need for a teacher to provide a grade. This information helps clustering the answers to be graded by the teacher.

ASAG task is not always a binary classification task. More fine-grained label categories might be used as well, which makes ASAG task an n-way task such as 2 way, 3 way and 5

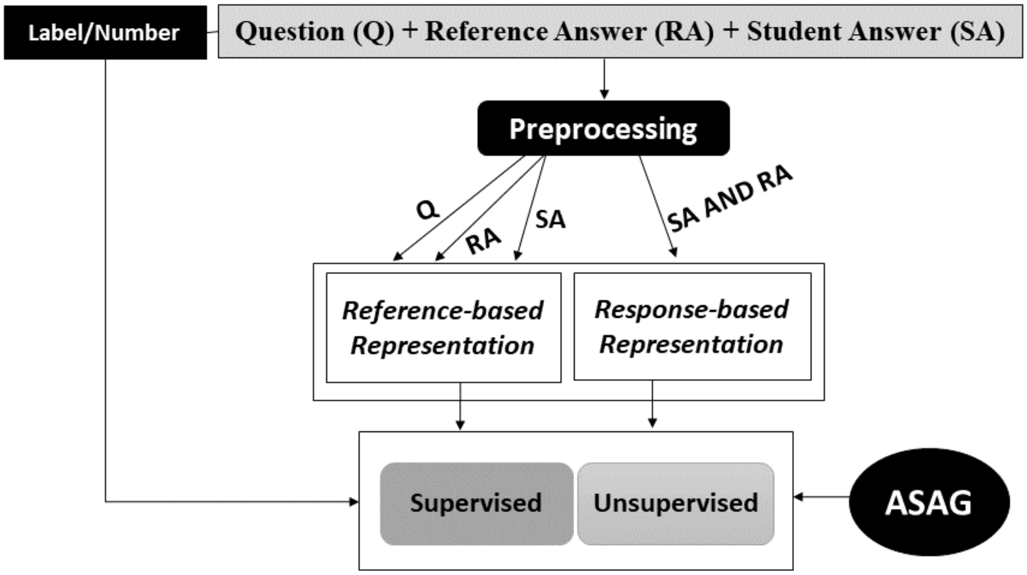
Label/Number
Question (Q) + Reference Answer (RA) + Student Answer (SA)
Preprocessing
Q
RA
SA
SA AND RA
Reference-based Representation
Response-based Representation
Supervised
Unsupervised
ASAG

Figure 2.1 ASAG general architecture (Q (question), RA (reference answer) and SA (student answer))

way task [1]. In other words, the output label is assigned from among the n labels which are available in the dataset. In the sense of classification, it should be noted that ASAG is a single-output classification task which differs from multi-label (multi-output) classification. Alternatively stated, the student answer cannot be considered both correct and partially correct at the same time. This also implies that the ASAG labels are always mutually exclusive.

Sometimes reference answers are considered as positive/correct examples in a supervised grading setting. Alvarado et al. [21] took this approach in the response-based framework. In some other settings, only the student answer is considered in ASAG. Finally, the performance of the grading models is evaluated using standard machine learning measures.

In the current thesis, we adopt the reference-based approach through which we consider only the student and reference answers. We don't conduct any type of external data preprocessing (for example lemmatization) as our models (which will be explained in the next sections) are self-contained. Throughout the thesis, we use n way terminology to categorize the ASAG task. Overall, we conduct 2 way, 3 way, 4 way and 5 way ASAG.

2.3 Analysis of ASAG SOTA

In this section, we firstly describe the trends of ASAG. Then, we report a summary of our analysis of ASAG research papers. We discuss in detail the different feature categorizations, methods and tools for feature extraction and ASAG, and the datasets and evaluation metrics used for the analysis of the ASAG performance.

Burrows et al. [5] analyzed 35 ASAG systems in depth through a broad classification of what they called ASAG eras. According to this review, there are five eras (trends) for ASAG systems. Each era is defined as follows:

- Concept mapping (e.g. [22]): In this era, similar concepts between the student and reference answers are identified and enumerated. According to Burrows et al. [5], the most obvious limitation for these approaches is that they don't grade the student answers for which there is no unique reference answer for the question.

- Information extraction (IE) (e.g. [23]): Based on Burrows et al. [5] definition, the alternative term for *information* is *fact*. The facts are specific ideas the teacher seeks to find in the student answers. In this regard, the facts are searched for in the student answers and compared against a template which is provided by the teacher. In other words, structured data is extracted from the student answers. As highlighted by Hasanah et al. [24], IE-based ASAG systems try to match students and reference answers based on seven main techniques: parse tree matching (e.g. [25]), regular expression matching (e.g. [26]), Boolean phrase matching (e.g. [27]), syntactic pattern matching (e.g. [28]), syntactic-semantic pattern matching (e.g. [29]), semantic word matching (e.g. [30]) and LRS (Lexical Resource Semantics) representation matching (e.g. [31]).

- Corpus-based methods (e.g. [32]): In these methods, features such as the frequency of ngrams are extracted from the answers and then are used by similarity measures such as BLEU (for example [33]) in the grading process. For example, Mohler and Mihalcea [32] used corpus-based similarity features based on Latent Semantic Analysis (LSA) in their grading pipeline.

- Machine learning (e.g. [34]): As the name suggests, the representations such as bag-of-words and n-grams are extracted from the ASAG datasets through NLP techniques and then merged into a grade by machine learning models. As a result, these models produce grades for the student answers.

- Evaluation [35]: As Burrows et al. [5] described, this era includes the systems which are implemented on a shared dataset for the sake of taking part in a competition. The

SemEval-2013 dataset [1] we use in the current thesis falls into the shared datasets of this era.

In the next subsections, we report our analysis of 50 papers based on our designed framework for analyzing the literature of ASAG: ASAG data, answer/question features, ASAG methods and ASAG evaluation measures in the following subsections. Here, we explain how we selected the papers in the following paragraphs.

Initially, we looked for English ASAG papers with no time limit in Scopus[1], Web of Science[2] and Google Scholars[3]. For further papers, we also relied on the Google Metrics website[4], which provides top-ranked (h5 index > 24) journals and conferences in Engineering and Computer Science-Education Technology. We included the journals and conferences on Educational Data Mining[5] and Learning Analytics and Knowledge[6] because they are seemingly important for our analysis. Finally, we filled in the missing papers from the literary surveys [19, 36].

Considering the following inclusion and exclusion criteria, we ended up with 50 papers from 2003 to 2020:

- *Inclusion criteria:* Studies that analyzed learner answers using artificial intelligence, machine learning and natural language processing techniques;
- *Exclusion criteria:* We do not consider language learning questions like essay writing or reading comprehension questions. In these questions, students' language skills are evaluated rather than content understanding skills. We have also not considered computer-assisted grading where the teacher is provided with the preprocessed students' answers for grading.

2.3.1 Answer/Question Features

Based on our analysis, several features are extracted from student answers, reference answers and questions. We classify them into four groups: statistical features, syntactic features, semantic features and similarity features.

[1] https://www.scopus.com/home.uri

[2] https://www.webofknowledge.com

[3] https://scholar.google.com/

[4] https://scholar.google.com/citations?view$_op = top_venueshl = en$

[5] https://educationaldatamining.org/

[6] https://www.solaresearch.org/

Statistical Features

According to [37], text statistics can describe each student answer and some ratio between student answer, questions and reference answers. The description includes the length of an answer (e. g. the ratio of the number of words in the student response to that in the reference answer [38]), and spelling errors (for example the number of spelling errors normalized by the text length in [39]). The ASAG text statistics can also be based on the corpus of students' answers as a whole. We can use it to identify the most salient components in students' answers. The example features include frequency (e.g. [40]) and TF-IDF (e.g. [34]). The papers including these features considered the use of vector space models based on word or character n-grams to represent answers, weighted by TF or TF-IDF.

Syntactic Features

Syntactic features include POS (part of speech) or dependency relations. Dependency parses were used along with part-of-speech tags to derive lexical resource representations in [41]. In papers such as [42, 43], similar patterns of dependency relations are searched for in student answers and in reference answers. POS features were used in several papers. For example in [44], five POS features (Verb, Noun, Adjective, Adverb and Other) were used to calculate the maximum similarity of each word embedding in the reference answer with each word embedding of the student answer.

Semantic Features

By semantic features, we mean ontology-based and corpus-based features.

- *Ontology-based Features:*

We observed the utilization of two types of ontologies in ASAG systems: lexical ontologies and formal ontologies. Among our papers, the percentage of synonym tokens (using WORDNET [45] as a lexical ontology) is calculated between student and reference answers for example in [41]. For general or domain ontologies, Alvarado et al. [21] annotated students answers with DBpedia [46] URIs using TAGME [47] or DBpedia Spotlight [48] and used the bag of the obtained concepts as a vector representation of the answers.

Besides word categories or lexical related elements, some approaches adopted a more complete semantic representation. In papers like [41] and [49], a syntax-semantic interface based on lexical resource semantics (LRS) [50] was extracted from the reference answer, the student answer and the question. According to Richter and Salien [50], LRS is defined as underspecified semantic formalism and can be automatically derived given POS and dependency tags. Overall, the LRS underspecified characteristics are useful for analyzing student's ill-formed and ambiguous sentences. For instance, in [41], the similarity is calculated when there is an overlap LRS representation of the reference answer and the student answer and the question.

Other approaches relied on frames based on knowledge bases like FrameNet [51], VerbNet [52] and PropBank [53]. Frame-based representations were found in the papers like [41] and [54]. For example in [54], these representations are used for measuring the similarity as the overlap of the predicates and their associated semantic arguments.

- *Corpus-based Features:*

Corpus-based features designate features extracted from a complete corpus of students' answers (and reference answers) rather than from individual answers and are generally represented as a vector space model or with word-document matrices. For example, word-answer matrices were used in [55] to represent student and reference answers.

Overall, corpus-based features can be represented explicitly or implicitly. In the explicit representations, each index of a vector is mapped to a word or n-gram feature. For example, the traditional bag-of-words approach is represented as a high-dimensional vector whose dimension is equal to the vocabulary size of the corpus that is analyzed. In the implicit representations, features emerge from approaches such as LSA [56], LDA [57] and recent dense vector-based representations such as word2vec [58] or GloVe (based on ratios of probabilities from the word-word co-occurrence matrix) [59] and vectors indices cannot be mapped directly to specific words in the corpus.

Finally, transformer-based architectures [60] were first used in 2018. These architectures enable the use of pre-trained language models such as BERT [12] and XLNET [13] and produce contextual word embeddings that can be used as robust representations of students' answers. In fact, BERT-like architectures can be used to fine-tune pre-trained embeddings on a corpus of students' answers and were among the top achievers in the ASAG task [61].

Similarity Features

In similarity features, students' answers and reference answers are compared with some sort of semantic distance metric and these metrics represent semantic similarity features. Inspired by [62], we categorize ASAG similarity measures into the following types:

- *Character-based Similarity:*

Character-based similarities calculate string-based differences between student answers and reference answers. However, instead of relying on words, they consider characters. For example, [63] used Damerau-Levenshtein distance [64], Jaro algorithm [65, 66], Jaro–Winkler distance [67], Needleman-Wunsch algorithm [68], Smith-Waterman algorithm [69] and character n-grams (called q-grams) similarity [70]. Character n-grams were also used in soft cardinality algorithm [71] by Jimenez et al. [72], where the authors smoothed the classical cardinality through controlling the degree of softness and hardness and assigning weights to the tokens using Dice coefficient.

- *Knowledge-based similarity:*

These metrics rely on ontologies to compute distance measures such as shortest path [55], Leacock and Chodorow [73], Lesk [74], Wu and Palmer [75], Resnik [76], Lin [77], Jiang Conrath [78], or Hirst St. Onge [79]. For example, [55] used each of these similarity metrics in a formula shown below:

$$Sim(R,S) = \frac{1}{2}(\frac{\sum_{w\varepsilon(R)}(MaxSim(w,R) \times idf(w))}{\sum_{w\varepsilon(R)} idf(w)} + \frac{\sum_{w\varepsilon(S)}(MaxSim(w,S) \times idf(w))}{\sum_{w\varepsilon(S)} idf(w)}) \tag{2.1}$$

In Formula 2.1, R stands for the reference answer and S for student answer. Using one of the metrics mentioned above, MaxSim calculates the maximum similarity between each word in text 1 (for example reference answer) and all the words in text 2 (for example student answer). Finally, idf is the total number of answers divided by the number of answers which contain a specific word. Overall, the similarity score has a value between 0 and 1 with 1 indicating identical text segments. This formula is computed only between words with the same syntactic category (POS).

- *Corpus-based similarity:*

In corpus-based similarity, the corpus is processed to create vector representations of students and reference answers, which are then used in similarity calculation formulas using distance measures such as Cosine, Euclidean, Manhattan, etc. In fact, creating the elements of the answer vectors is not restricted to the text of the answers. For example, paper [63] used LSA technique to build the text vectors for the aforementioned distance measures.

In BOW-based similarity, the answers are represented as the set of words without the consideration of their order. Then, they are plugged into the distance measures. For example, the measures adopted from the machine translation evaluation field were used, such as BLEU (bilingual evaluation understudy), which is used in papers like [33] to calculate the percentage of n-grams from the student answer to be present in the reference answers or the text summarization evaluation measure ROUGE, which is used as a feature in [80] to compare the student answer and the reference answer.

- *Embedding-based similarity:*

In this type of similarity, the vectors are embeddings that are learned within neural network architectures (for example word2vec [58]) or through by constructing a co-occurrence matrix (for example GLoVE [59]). Sultan et al. [81] calculated the cosine similarity using the embeddings-based (3400-dimensional Baroni word embeddings [82]) and used it as feature in both regression type and classification type of ASAG.

2.3.2 ASAG Methods

Based on their selected papers, Roy and Narahari [36] categorized the ASAG techniques into NLP, Information Extraction, Machine Learning, Document Similarity and Clustering. According to the findings of Galhardi and Brancher [37], most ASAG systems were based on supervised algorithms (classification and regression) rather than unsupervised ones. In this section, we categorized the ASAG methods based on our selected papers.

In the supervised tasks, we either train a classification model with a nominal scale or a regression model with a ratio scale. In the unsupervised tasks, the similarity between student answer (sometimes alongside the question) and the reference answer is calculated and then scaled into numbers like 0, 1, 2, etc. Similarity algorithms which fall into unsupervised type of ASAG, contains Cosine similarity and Text-text similarity (formula 2.1). Both these algorithms were used in [32, 55]. The details are provided in the explanation for Formula 2.1 in subsection 2.3.1.

It should be noted that we didn't find any type of semi-supervised ASAG among our papers. Inspired by Jason Bowie's grouping of algorithms[7], we further categorized the algorithms used in ASAG into 8 categories. As Table 2.1 shows, 20 algorithms have been used in our selected papers. In this subsection, we explain each group of the supervised ASAG methods with the related example papers.

- ***Regression algorithms*** include both classification and regression algorithms. In the regression type, these algorithms estimate the numerical value (e.g. [32]). The general

[7] https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Table 2.1 Categories of algorithms, and example algorithms used in ASAG

Categories	Algorithms
Regression Algorithms	Least Square Regression [83], Logistic Regression, Isotonic Regression, Multinomial Logistic Regression classifier
Instance-based Algorithms	k-Nearest Neighbor, Support Vector Machines
Regularization Algorithms	ridge regression model
Decision Tree Algorithms	J48, C4.5
Ensemble Algorithms	random forest, Stacking
Bayesian Algorithms	Naive Bayes
Deep Learning	LSTM, BERT, XLNET, Bidirectional LSTM, ConNet, Deep Belief Networks

formula for regression algorithm is as follows:

$$y = b + wx \tag{2.2}$$

In formula 2.2, b is the bias term and w is the coefficient for a single input value (x). While in the classification type, the probability of class membership is estimated and the class (label) with the highest probability is assigned to an answer (e.g. [21, 80]). When formula 2.2 is plugged into logistic function, Logistic Regression is calculated for the inputs. For example, Alvarado et al. [21] used ontology-based features in a response-based Logistic Regression algorithm within Biology dataset [2].

- ***Instance-based algorithms*** (also called winner-take-all methods or memory-based learning) simply compare test data with the train data (e.g. [21,84]). In the very simple form, the similarity is calculated between each student answer (x) in the test set and all the student answers (x_i) in the train set. Then, the first k most similar distances are taken and the final grade (class) is assigned to the one with highest frequency among these k examples. More complex algorithms of this type include Support Vector Machine algorithm. Alvarado et al. [21] tested the same types of ontology-based features using this algorithm.

- ***Regularization algorithms*** are the extensions of the linear regression algorithms that reduce the inaccuracy (minimizing the cost function) based on the complexity of the models. For example, objective function is minimized using l2 regularization in ridge regression model. In other words, the following penalty term (where λ is regularization strength, b is bias term and w_j are coefficients) is added to the loss function.

$$\lambda \sum_{j=1}^{p} |w_j| \tag{2.3}$$

Sultan et al. [81] used several features such as TFIDF and cosine similarity of embeddings in Ridge Regression Model.

- ***Decision tree algorithms*** build a model of decisions based on the attributes (inference rules r_1, r_2,.., r_n) of train data (e.g. [21, 81]). Alvarado et al. compared Decision Tree approaches with Instance-based algorithms while Sultan et al. [81] compared them with Ridge Regression Model (Regularization algorithms).

- ***Ensemble Algorithms*** consist of multiple base models (also called estimators) that are separately trained and the predictions of each are combined to make the final prediction (e.g. [8, 80]). For instance, several trees (t_1, t_2,.., t_i) are built in random forest algorithm and the output grade (class) is considered as the most frequent grade (class) of the grades (classes) of the individual trees. In this regard, Archand and Kumar [80] conducted feature space exploration using Random Forest (among several other algorithms). Their features are semantic and similarity features discussed in section 2.3.1.

- ***Bayesian algorithms*** utilize Bayes' Theorem through assuming conditional independence between the features and considering the grades (classes). In ASAG, these assumptions belong to the different grades (e.g. [21, 84]). Given the grade (class) y and feature vectors (x_1, x_2,.., x_n) for each answer regarding the label, the decision about the grade (class) is outputted using the following formula (used in Naive Bayes algorithm) in terms of Maximum A Posteriori (MAP) estimation:

$$\hat{y} = \underset{y}{argmax}\ P(y) \prod_{i=1}^{n} P(x_i|y) \tag{2.4}$$

As a broad comparison to other approaches discussed above, Alvarado et al. [21] used Naive Bayes algorithm.

- ***Deep learning algorithms***, inspired by biological neural networks, train an ASAG model through a number of connected layers which include nodes (neurons) (e.g. [61, 85, 86]). In a general sense, a deep learning algorithm learns a non-linear function for either classification or regression using features (x_1, x_2,.., x_n) and the target grade (class) y. For example, Kumar et al. [86] used word2vec [58] in Siamese bidirectional LSTMs.

2.3.3 ASAG Datasets

Table 2.2 Common datasets used in ASAG (based on our collection)

Data	Domain	Respondents' Level	NO of Answers	Grade
CSD	computer science course (Data Structure)	Undergraduate	630	Ratio
XCSD	computer science course (Data Structure)	Undergraduate	2273	Ratio
Dt-grade	Physics	junior level college students	900	Nominal
SemEval	more than 15 different science domains: SciEntBank (various domains) + BEETLE (electricity)	Basic	10000 + 3000	Nominal
Kaggle	science, biology and ELA (English Language Arts)	Not specified	Approx. 18000	Ratio
FOSS	life science, physical science, earth and space science, scientific reasoning, technology	3rd-6th grade students	15400	Nominal
Biology Dataset	human biology	first-year undergraduate	15758	Nominal

Roy and Narahari [36] analyzed short answer datasets (including reading comprehension questions and non-English datasets). Based on their collection of papers, the grade schemes of most of the datasets were provided almost without proper standardization.

As Table 2.2 shows, there are seven popular datasets used in ASAG. These datasets are categorized based on student's level and most of them are in the beginner category except Dt-grade [3]. Among these datasets, Kaggle[8] (provided by the Hewlett Foundation), Biology Dataset (provided by University of Auckland and University of Otago, New Zealand [2]), SemEval (provided by the Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge [1]) and FOSS (provided by the Lawrence Hall of Science, University of California, Berkeley and prepared by University of Colorado, Boulder [43]) are the biggest in terms of size (with more than 10000 answers). In comparison, CSD [32], XCSD [55] (both provided by University of North Texas) and Dt-grade [3] are smaller.

[8] https://www.kaggle.com/c/asap-sas

The ASAG datasets shown in Table 2.2 usually contain either nominal or ratio (numerical) labels [87]. In the nominal scale, grades are in the format of labels like correct, incorrect, incomplete, etc. The Biology, SemEval, FOSS and Dt-grade datasets include such types of labels. Kaggle, CSD and XCSD contain numerical grades like 1, 1.5, etc.

2.3.4 ASAG Evaluation Measures

In this section, we categorize the evaluation measures (inspired by [87]) into classification and regression types.

- *Classification metrics:* In these metrics, there are four building blocks: *true positive, true negative, false positive and false negative.* True means that a predicted label (for a specific example) is equal to the label (for the same specific example) in the dataset. Otherwise, the predicted label is considered false. Positive (such as correct which is assigned to a student answer that is semantically similar to a reference answer) and negative (such as incorrect which is assigned to a student answer that is semantically similar to a reference answer) are the names of the ASAG labels. Some of these metrics are as follows:
  - Accuracy: The number of true positive (such as truly predicted correct answer) and negative examples (truly predicted incorrect answers) among all the predicted examples:

$$accuracy = \frac{true\ positive + true\ negative}{true\ positive + false\ positive + true\ negative + false\ negative} \tag{2.5}$$

  - Precision: Measurement of the number of the true positive examples (such as predicted correct student answers) among all the predicted positive examples (such as predicted correct and incorrect student answers):

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \tag{2.6}$$

  - Recall: Measurement of the number of the truly predicted positive examples (such as truly predicted correct student answers) among all the truly predicted positive examples and falsely predicted negative examples (such as falsely predicted incorrect answers)

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{2.7}$$

- F1 score (micro, macro and weighted): The harmonic mean of precision and recall. Micro version counts the total number of true positives, false negatives and false positives. The formula is as follows:

$$f1\_score = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.8}$$

  In the macro version, we calculate f1-score for each class and then average the total f-score values among the total number of classes (we call it N).

$$macro\_f1\_score = \frac{1}{N}\sum_{i=1}^{N} f1\_score_i \tag{2.9}$$

  Finally, weighted version considers the number of true instances for each grade while calculating the macro version. In the following formula, w stands for the number of true instances for each class.

$$weighted\_f1\_score = \frac{1}{N}\sum_{i=1}^{N} w_i \times f1\_score_i \tag{2.10}$$

- Area Under the Curve (AUC) and Receiver Operating Characteristic (ROC): They are represented as a curve where TPR (true positive rate) represents the y-axis and FPR (false positive rate) is the x-axis.

$$TPR = \frac{true\ positive}{true\ positive + false\ negative} \tag{2.11}$$

$$FPR = \frac{false\ positive}{false\ positive + true\ negative} \tag{2.12}$$

- *Agreement metrics in regression task:* These metrics are specifically used in regression type of reference-based ASAG. Overall, we define two distributions d1 and d2 which belong to grades given by the teacher in the data and the generated grades by the ASAG model successively. Some of these metrics are as follows, where n designates the number of answers in the following formulas:

  - Pearson Correlation: statistical relationship between two distributions d1 and d2.

$$PC = \frac{\sum_{i=1}^{n}(d1_i - mean(d1_i))(d2_i - mean(d2_i))}{standard\ deviation(d1) - standard\ deviation(d2)))} \tag{2.13}$$

  - Kappa value and its variations (such as Quadratic weighted kappa): Measurement of agreement (reliability) between the elements of two distributions d1 and d2

$$\frac{observed\ agreement - chance\ agreement}{1 - chance\ agreement} \tag{2.14}$$

- *Error analysis metrics in regression task:* These metrics are used for example in [55] to analyze the grades across each individual question. We provide some of the metrics as follows:
  - Mean absolute error: Mean absolute differences between the elements of two distributions (d1 and d2)

    $$MAE = \frac{1}{n}\sum_{i=1}^{n}|d1_i - d2_i| \tag{2.15}$$

  - Root-mean-Square Error (RMSE): Similar to MAE but imposes stricter penalties on the large errors

    $$RMSE = (\frac{1}{n}\sum_{i=1}^{n}(d1_i - d2_i)^2)^{\frac{1}{2}} \tag{2.16}$$

2.4 Previous ASAG Top-performing Approaches

The findings from our papers set indicate that reference-based ASAG methods are rather robust. Among them, deep neural networks have been shown to have performed well. This is mostly due to the neural network architecture and the introduction of embeddings which fall into corpus-based features in section 2.3.1.

In this section, we summarize the closely related works that are based on supervised classification and related to the current thesis. As we will report our result for each dataset separately in chapter 4, we compare several top-performing systems in each of the datasets chosen to evaluate our work in this thesis. The top-performing systems in SOTA are as follows:

- ETS [88]: In this system, Heilamn et al. [88] used the statistical features (introduced in section 2.3.1) such as n-gram features (based on lower-cased words and characters) and similarity features such as corpus-based similarity in a reference-based framework. Also, the authors used Daume domain adaptation technique [89]. Through this technique, different weights are assigned to several copies of a single feature based on the characteristics of the test data. These copies are called a generic copy, a domain-specific copy, and an item-specific copy. For example, Heilamn et al. [88] considered only generic features for an answer to a question which is in different domain than the one in the train data. On the other hand, these authors considered all types of copies (the generic, domain-specific, and item-specific copies) for the items of the training data.

- CoMet [90]: Ott et al. [90] used syntactic features such as POS, dependency relations and constituent structures. Their ASAG method was the meta-classifier with logistic regression on the top. This meta-classifier was built upon the outputs of the subsystems, namely CoMiC (Comparing Meaning in Context), CoSeC (Comparing Semantics in Context), and three shallower bag approaches. Through CoMiC, linguistic units such as POS were mapped from a student answer to those in a model answer. The meaning by CoSeC (Comparing Semantics in Context) approach was compared based on LRS within reference-based framework. Finally, three bags of words, lemmas and Soundex hashes of the student answer were extracted and fed into support vector-based machine learner.

- SOFTCARDINALITY [72]: Jimenez et al. [72] calculated the overlap between student and reference responses by means of soft cardinality [71]. In this regard, they measured the number of common character n-grams (called q-grams) using Dice coefficient. Then, the authors assigned weights to the tokens based on the cardinality of these commonalities. In fact, these weights represent the informativeness of the characters/character n-grams in a token. Overall, they used soft cardinality method mainly to enhance the accuracy in calculating the similarity.

- Ramachandran et al. [91]: Ramachandran system generated the summary of the top students' answers using summarization techniques such as graph-based cohesion and MEAD [92]. The features used were mostly similarity measures between their generated reference answer, the student answer and the question. The similarities include word overlap, cosine similarity and Lesk [74] similarity and f-measure of the overlaps between the compared texts. It should be noted that f-measure as feature is different from the one introduced in section 2.3.4. Ramachandran et al. [91] defined it as the harmonic mean of the precision and recall of the overlaps between student and reference answer. In this sense, precision is the number of overlaps (between student and reference answer) over the number of tokens in student response. Recall is the overlap (between student and reference answer) over the number of tokens in the reference text.

- Sultan et al. [81]: The system combined several features such as word embeddings-based semantic similarity, text alignment, question demoting, term weighting and length ratios in a reference-based framework. The system was shown to be robust both in classification task (using Random Forest) and regression task (Ridge Linear Regression). Due to less runtime complexity and less complex features, the performance of the system was described as *fast and easy ASAG* by the authors.

- Saha et al. [44]: The authors suggested the combination of token features (word overlap and similarities between student answer and reference answer) with sentence embeddings features (all question, reference answer and student answer are combined) based on InferSent [93] and POS tags. They made considerable improvements in SOTA (until 2018). Their reference-based ASAG approach used these features in Multinomial Logistic Regression (regression type introduced in section 2.3.2) and Random Forest.

- Mantecon [8]: He used some of the features word overlap and sentence embeddings features applied by Saha et al. [44] in an ensemble-based ASAG. He indicated the approach of Saha et al. [44] seems still useful even without considering features such as POS tags.

- Sung et al. [94]: The authors trained BERT model on 3way classification task of SciEntBank dataset and two other psychology domain datasets (publicly not available). Their results indicate an improvement in SOTA.

Thanks to the introduction of transformers [60], Sung et al. [94] and Ghavidel et al. [61] showed the outstanding performance of BERT [12] in the reference-based ASAG task. We [61] further indicated that XLNET [13] is quite robust or competing with SOTA, especially in SciEntBank dataset. Overall, we [61] obtained SOTA results with BERT [12] and XLNET [13] over traditional feature-based algorithms.

2.5 Language Models

Goldberg [95] defined language modeling as a task of analyzing the sequence of a text in order to find its probability for it. He categorized language modeling into traditional and neural models. In this section, we explain these models in detail.

2.5.1 Traditional Language Models

Generally, traditional language models represent probability distributions over any sequence of words. This definition can be expressed in the following chain-rule probability formula:

$$P(w_{1:n}) = P(w_1)P(w_2|w_1)...P(w_n|w_{1:n-1}) \tag{2.17}$$

In Formula 2.17, w stands for word and n for the length of the sequence (the index of the last word in a sequence). This formula suggests that any word such as w_n can be predicted based on the words in $w_{n\text{-}1}$ context.

According to Goldberg [95], traditional language models are built based on a k-order Markov property. In other words, the prediction of the next word in a sequence relies on the previous k words. In this regard, maximum likelihood estimates are calculated for n-grams. Traditional language models are used in several NLP tasks such as machine translation, spell correction, speech recognition, etc. For example in speech recognition, the language model is used in combination with an acoustic model. In this case, the output of acoustic model is evaluated by the language model to determine how probable the sequence is in a natural language.

2.5.2 Neural Models

While traditional language models are quite simple to train and can be extended to corpora of any size, they suffer from the following drawbacks:

- Sparsity: Larger ngrams such as quadrigrams are rare in natural language and thus makes the statistics sparse.
- Memory efficiency: The number of ngram types can be calculated by $|V|^n$ where V is the size of vocabulary and n the bigram type. Larger ngrams are filling lots of memory.
- Dependency on the exact pattern: traditional language models lack a generalization capability across various contexts. For example, *red pencil* and *yellow pencil* don't help calculating the estimates for *purple pencil*.
- Smoothing limitations: Some of the techniques such as Katz backoff [96] are to be manually tuned through adjusting the backing-up order. Also, the backoff technique doesn't consider long-range dependencies (for example the relation of words in position 1 and position 10). As a result, it is not straightforward to be adjusted for different contexts.

To overcome these limitations, neural language models were introduced. To the best of our knowledge, there are three types of neural language models introduced in SOTA: feed-forward neural models [97], recurrent neural networks (RNN) models and transformers. What differentiates these models from the traditional models is that they scale more easily to wider and various contexts. Also, they do not require manually adjusting smoothing parameters.

Feed-forward Models

In general, a feed-forward neural network includes the input data (feature vectors), the hidden nodes and the output nodes. In terms of language modeling, these models were introduced by Bengio et al. [97]. These models use the distributed representations of words to learn a language. Overall, this model has vector representations for each word. Besides the distributed representations of each word, this model also learns the probability function of a sequence. This function serves the needs of calculating a probability for any sequence such as phrase or sentence.

Overall, these models have some limitations:

- The number of words in the context is predefined and we cannot go above this number in real applications.
- The word representations are not context-sensitive. For example, we have only one vector for the word *address* no matter if it means[9] *speak to someone directly* or *details of the place.*

RNN Models

In contrast with feed-forward models, the output of each input of RNN models depends on the past or future outputs in the time steps. In other words, the information for each word is cycled throughout the whole network. These models were mainly proposed in [97, 98]. Mikolov et al. [98] used these models by the goal of reducing perplexity which exists in traditional language models.

Recent RNN language models such as ELMo (Embeddings from Language Models) [99] were based on long short-term memory networks (LSTMs and BiLSTM). The general training process is depicted in Figure 2.2. ELMO representations are built upon the following characteristics:

- They are bound to the whole context in which they occur.
- They are produced based on all the layers of the network
- They are based on characters.

[9] https://www.ldoceonline.com/dictionary/address

[10] https://www.analyticsvidhya.com/blog/2019/03/learn-to-use-elmo-to-extract-features-from-text/

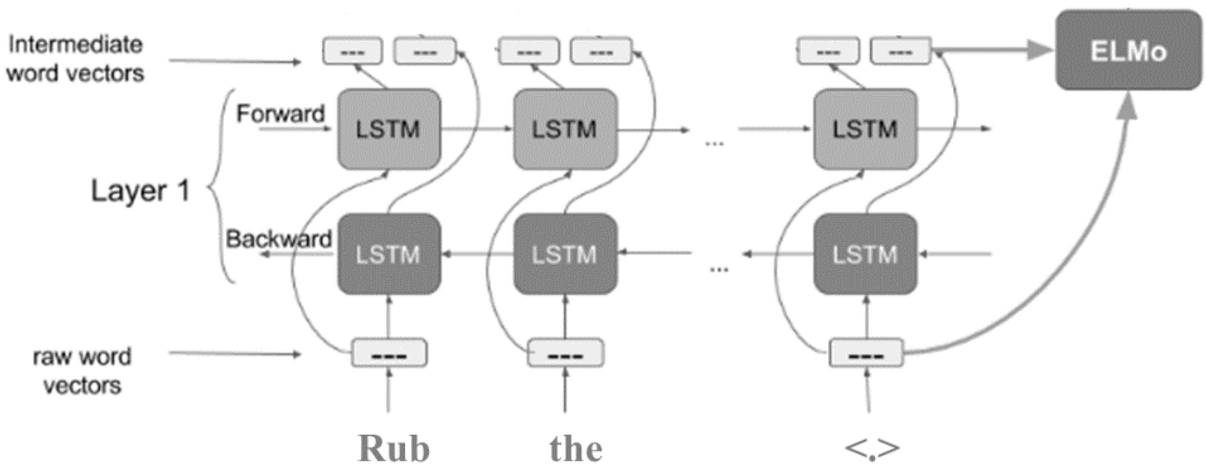

Figure 2.2 ELMO training process[10]

Transformers

The recent language models are based on the transformer architecture [60]. The overall architecture of transformers is depicted in Figure 2.3. These models rely largely on a mechanism called attention. This mechanism brings the focus into all parts of the sentence (e.g. student and model answers) at each time step. As shown in Figure 2.3, a typical transformer architecture consists of a stack of encoders and decoders.

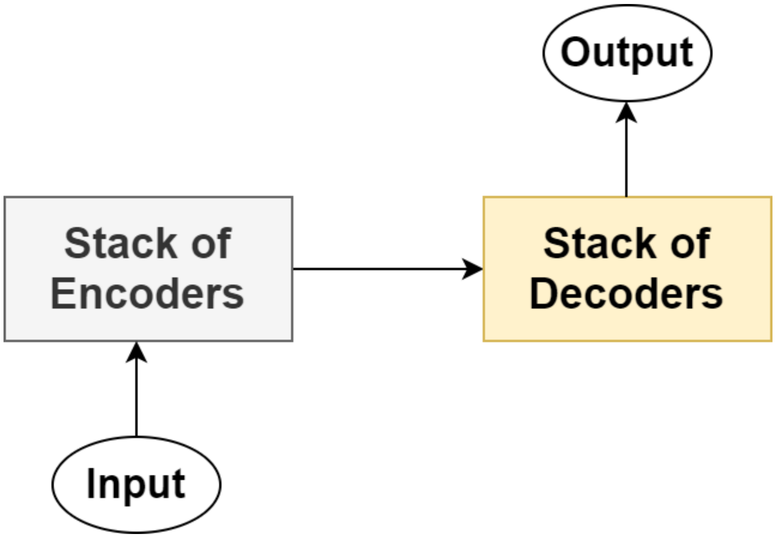

Figure 2.3 Transformers

As Figure 2.4 shows, the encoder's input is fed firstly to a self-attention layer. This layer draws the attention of encoder to the other words in the input sentence. Self-attention is particularly useful for learning long-range dependencies in the network. Catching these dependencies is considered as one of the main challenges of natural language understanding. Overall, the outputs of this layer are flows to a feed-forward neural network in which a new representation is generated. This representation is called "contextualized embeddings".

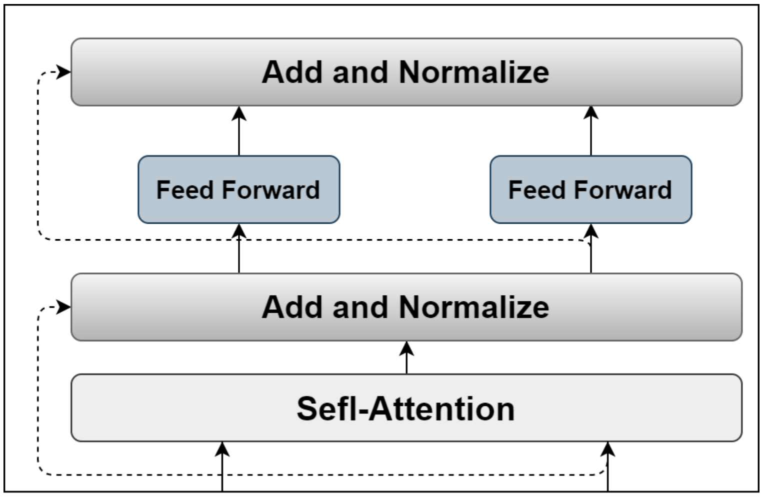

Figure 2.4 Encoder

Figure 2.5 shows the decoder architecture. Based on this figure. the decoder takes the state generated by the encoder in the encode-decoder-attention layer and use it to produce an output. For example in a machine translation task, the encoder generates a semantic representation from a sentence in English and then the decoder generates, using this representation, the sentence in French.

The popular transformer-based models are OpenAI[11], BERT (Bidirectional Encoder Representations from Transformers) [12] and XLNET (Extra Long Network) [100]. A distinctive feature of these models is that their embeddings are context sensitive. Compared to ELMo [99] which concatenates separately trained left-to-right and right-to-left LSTM in order to generate features, BERT [12] (as an example) representations are learned on the left and right context simultaneously. What makes BERT and XLNET different from OpenAI is they are conditioned on both left and right context. Also, the learned representations of these models are based on subwords instead of words. We are interested in BERT and XLNET in this thesis. We explain both of these models in chapter 3.

[11] https://openai.com/blog/language-unsupervised/

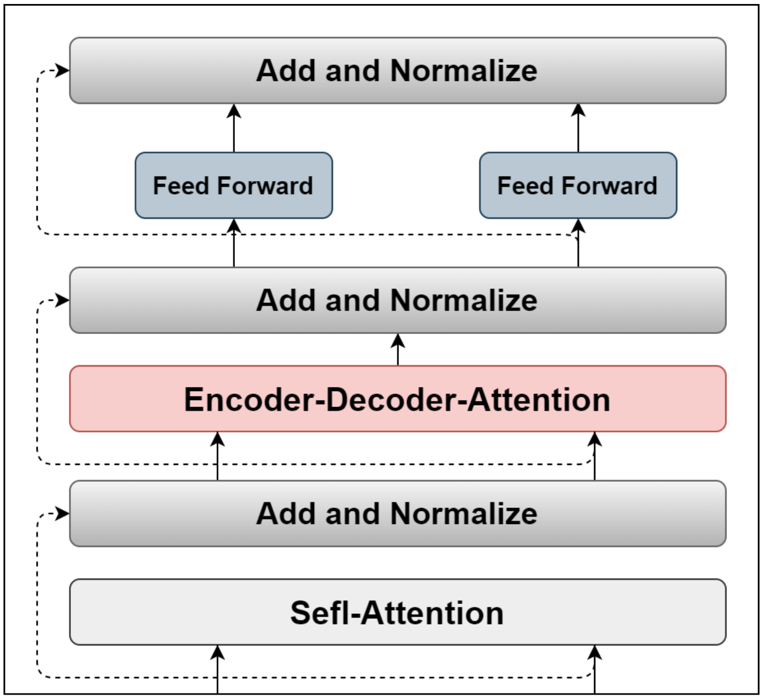

Figure 2.5 Decoder

2.6 Transfer Learning

Several machine learning or deep learning models are built by assuming that they inherit the characteristics of the task, domain and data. When these characteristics change, the models need to be trained all over again. According to [101], training models from the beginning is quite costly. To bring down these costs and also possibly improve the performance, it would be useful to employ transfer learning. Goodfellow et al. [20] defines it as the utilization of what is learned in one task in another task to enhance the generalization capability. In this thesis, we are interested in understanding how transfer learning can benefit the ASAG task through fine-tuning the pretrained models for the ASAG task.

Pan and Yang [101] categorized the techniques of transfer learning into the following categories based on the sameness of source and target tasks and domains:

- *Inductive transfer learning*: The source and target tasks are different. The sameness of source and target domains is not considered important. It is mostly used in areas such as multi-task learning.

- *Transductive transfer learning*: Only the source and target domains are different. It is mostly used in areas such as domain adaptation.

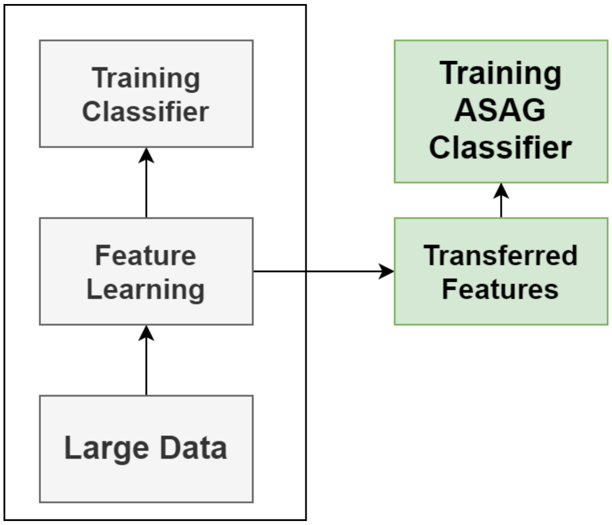

Figure 2.6 Transfer learning

- *Unsupervised transfer learning*: The source and target tasks are different, but they are related.

In deep learning, the inductive transfer learning is usually used. In this thesis, we conduct the inductive transfer learning process using the following approaches:

- ***One-level fine-tuning***: In this approach, we use the off-the-shelf pre-trained BERT [12] and XLNET [13] models as the starting point and fine-tune them on the ASAG task.

- ***Two-level fine-tuning***: In this two-level approach, we start with an off-the-shelf pre-trained BERT model and then we train it on a GLUE task (source task). Then, we fine-tune the obtained model on the ASAG task (target task). For example, the source and target tasks can be textual entailment and ASAG respectively,

2.7 Summary

In this chapter, we introduced the types of MOOCs assessments and we provided a definition for short answers and the ASAG task. We also reported our synthesis of selected ASAG papers in terms of datasets, features, grading methods and finally measures for the evaluation of ASAG models. After that, we reviewed SOTA in ASAG. Then, we explained language models and transfer learning.

In the following chapter, we describe our methodology in detail. We mainly focus on the specifications about the datasets, models and evaluation measures.

CHAPTER 3 METHODOLOGY

In this chapter, we describe our general research methodology and describe the language model architectures BERT [12] and XLNET [100] in more details. It should be noted that parts of the results of the models presented in this chapter were published in CSEDU (Computer Supported Education) 2020 conference [61].

3.1 Datasets

To analyze the performance of our models in various contexts, we select all the available ASAG datasets for classification (with labels) introduced in section 2.3.3: SemEval [1] (SciEntBank and BEETLE), Dt-grade [3] and Biology dataset [2]. We summarize the characteristics of these datasets in table 3.1. The characteristics are number of questions, model answers and student answers. The labels for each dataset are as follows:

- SciEntBank and BEETLE:
    - 2 way: correct and incorrect
    - 3 way: correct, contradictory and incorrect
    - 5 way: correct, incorrect, contradictory, partially-correct-incomplete, irrelevant and non-domain
- Biology: correct and incorrect (the version used by [8, 102])
- Dt-grade: correct, correct-but-incomplete, contradictory and incorrect

We provide examples for each dataset in table 3.2, Table 3.3, table 3.4 and table 3.5. These tables show a sample question with the model answer, student answer and the gold label. In Dt-grade [3], the dataset providers separate problem descriptions from the questions.

3.2 Data Preparation

In this section, we elaborate on how we prepare our datasets. Our preparation stage consists of two sub-stages namely data splitting and data preprocessing.

Table 3.1 Characteristics of our ASAG datasets

Dataset	No of Questions	No of Model Answers	No of Student Answers
SciEntBank train	135	135	4969
SciEntBank TUA	135	135	540
SciEntBank TUQ	15	15	733
SciEntBank TUD	45	45	4562
BEETLE train	36	176	3668
BEETLE TUA	36	176	435
BEETLE TUQ	8	43	804
Biology	6	6	1658
Dt-grade	38	38	4929

Table 3.2 Example from SciEntBank [1] dataset

Question	Model Answer	Student Answer	Grade
You used several methods to separate and identify the substances in mock rocks. How did you separate the salt from the water?	The water was evaporated, leaving the salt.	Let the water evaporate and the salt is left behind	Correct
		You put the water, a little bit, in a cup and leave it overnight.	Incorrect

3.2.1 Data Splitting

SemEval [1] already provided three test sets for SciEntBank dataset and two test sets for BEETLE dataset. The test sets are TUA (test of unseen-answers), TUQ (test of unseen-questions) and TUD (test of unseen-domains). In TUA, the questions are similar to those in the train data. In TUQ, the questions are different from those in train data. Finally, TUD contains totally different questions within different domain compared to the questions in train data. SciEntBank test datasets are TUA, TUQ, and TUD scenarios. BEETLE only includes TUA and TUQ. The statistics about these test sets are provided in Table 2.2. To split a validation set from train set in both SciEntBank and BEETLE dataset, we randomly take 20 percent of answers from each question. This helps us have full coverage on all the questions and prevents us from a question-blind division of the dataset.

For the rest of the datasets, we do not have a pre-separated test and validation (development) set. As a result, we take 70 percent for train, 15 percent for validation and 15 percent

Table 3.3 Example from BEETLE [1] dataset

Question	Model Answer	Student Answer	Grade
Explain why you got a voltage reading of 1.5 for terminal 1 and the positive terminal.	Terminal 1 and the positive terminal are separated by the gap	Positive battery terminal is separated by a gap from terminal 1	Incorrect
		Because terminal 1 is connected to the positive battery terminal	Correct

Table 3.4 Example from Biology [2] dataset

Question	Model Answer	Student Answer	Grade
Inotropic state is a term that is sometimes used to describe the contractility of the heart. Can you describe what is meant by contractility?	Contractility is the force or pressure generated by the heart muscle during contraction.	Pressure generated	Correct
		Ability to contract	Incorrect

for the test set. The same procedure of train/validation split for SemEval dataset is done for the Biology dataset and the Dt-grade dataset.

3.2.2 Data Preprocessing

In this subsection, we explain the data preprocessing stages for our baseline models and final models. For our baseline models, we remove the punctuation marks, tokenize the answers using NLTK word_tokenize[1], convert the words to lowercase and remove the stopwords. Then, we lemmatize the words using NLTK WordNetLemmatizer[2]. In conclusion, we have lemmatized answers associated with their grades (labels).

In our final models, the input is the student answer, model answer and the grades (labels) associated with each student answer. As our models have their specific tokenizers, the maximum number of tokens (max sequence lengths) in an input sequence differs for each dataset. This maximum sequence length is needed by models based on transformers, namely BERT

[1] https://www.nltk.org/api/nltk.tokenize.html

[2] https://www.nltk.org/$_{m}odules/nltk/stem/wordnet.html$

Table 3.5 Example from Dt-grade [3] dataset

Problem	**Question**	**Model Answer**	**Student Answer**	**Grade**
A car windshield collides with a mosquito, squashing it.	A car windshield collides with a mosquito, squashing it. How does Newton's third law apply to this situation?	The action is the windshield squashing the mosquito, and the equal and opposite reaction is the mosquito hitting the windshield.	The windshield will apply a force to the mosquito equal the force applied by the mosquito to the windshield	Correct but incomplete
			The force exerted on the windshield by the mosquito was an action while the force exerted on the mosquito from the windshield was an equal and opposite reaction	Correct
			It applies to the forces of the bug and windshield being exerted on one another, and why they react in such a way	Incorrect

and XLNET. Therefore, we calculate the max sequence lengths (considering the [CLS] and [SEP[3]] tokens) of student and model answers in each trainset. For both BERT and XLNET, we provide a tokenized example of the input in section 3.3.

[3] separator

3.3 Graders

3.3.1 Baselines

For our baseline models, we selected five different algorithms. The algorithms are Naive Bayes, support-vector machines, decision tree algorithms, random forests (random decision forests) and logistic regression. For all our algorithms, we use a TFIDF (term frequency–inverse document frequency) bag of words model considering unigrams, bigrams and trigrams. For carrying out the whole model development process, we used the scikit-learn[4] library in Python programming language.

3.3.2 BERT

BERT [12] is a bidirectional model based on a Transformer encoder [60] illustrated in Figure 2.4 in chapter 2. In other words, it is based on a stack of encoders which include multi-headed attention (several self-attentions within a sequence). BERT is available with the following two types of models:

- **BERT base**: There are 12 layers, 12 attention heads and 768 neurons. The number of total parameters is 110 millions. This model is provided in cased and uncased versions.
- **BERT Large**: This model includes 24 layers, 16 attention heads and 1024 neurons. Similar to BERT Base, there are also cased and uncased versions for this model.

As discussed in section 3.2.2, BERT tokenizer has max sequence lengths. The tokenizer is based on WordPiece embeddings [103] with a vocabulary of 30,000 tokens. In the cased version of the BERT model, the capital letters are considered in the tokenization process. Note a BERT-based tokenized example [61] from model answers in SciEntBank [1] train set:

Rub the minerals together and see which one scratches the other. [61]

- **BERT Base uncased example [61]:** [<rub>, <the>, <minerals>, <together>, <and>, <see>, <which>, <one>, <scratches>, <the>, <other>, <.>] [61]
- **BERT Base cased example [61]**: [<R>, <ub>, <the>, <minerals>, <together>, <and>, <see>, <which>, <one>, <scratch>, <es>, <the>, <other>, <.>]

[4] https://scikit-learn.org

BERT framework consists of two stages: pretraining and fine-tuning. In the pretraining stage, BERT is trained on unannotated data. Two corpora were used for this stage: BooksCorpus (800M words) and Wikipedia (2,500M words). The pretrained stage consists of two tasks, namely masked language model (MLM) task and next sentence prediction (NSP) task. Overall, MLM and NSP are trained together in order to reduce the combined loss function.

MLM [104] is used for masking some of the tokens at random (15% of the tokens within a sequence) and predicting these tokens using softmax classifier. The alternative term for MLM is *Cloze* [104] task. It should be noted that BERT is based on *denoising autoencoders [20]*. Alternatively stated, it is trained through corrupting the data i.e. changing a number of the input values at random and trying to predict them. An example of masking is provided in figure 3.1.

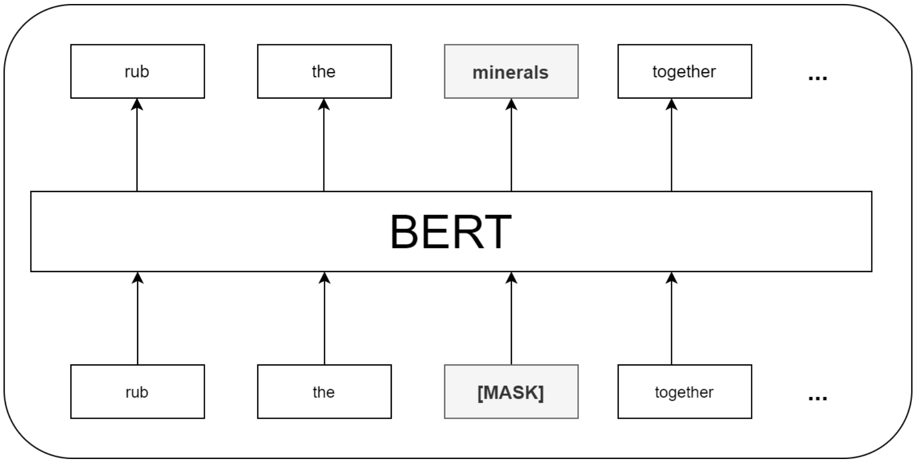

Figure 3.1 A toy example for masking a token (by BERT uncased tokenizer)

In NSP, a couple of sentences are used in succession to obtain discourse information. In fact, BERT describes the relationship between a couple of sentences by means of a softmax classifier. Accordingly, 50% of the time the second sentence is the real next sentence which follows the first sentence (IsNext). In the rest of the cases, the second sentence randomly comes after the first sentence (NotNext).

In the fine-tuning stage, we determine the input and output for a down-stream task and fine-tune the parameters. We use two symbols, [CLS[5]] and [SEP[6]], to prepare the input. Overall, the input for BERT in our task looks like as follows:

[5] classification

[6] separator

[CLS] + tokenized (student answer) + [SEP] + tokenized (model answer) + [SEP]

The input embeddings for BERT are the results of summing up (element-wise) the embeddings of token, segmentation and token position. The segmentation includes the embeddings as a means of distinguishing between the first sentence (student answer) and second sentence (model answer). The embeddings for the position of the token is initialized randomly and trained during pretraining. The position vector includes the sines and cosines of the position of each token (within the sequence and the embedding vector), which is obtained through the following formula [60]:

$$\vec{p_t}^{(i)} := \begin{cases} \sin(\omega_i \times t), & \text{if } i \textit{ is even} \\ \cos(\omega_i \times t), & \text{if } i \textit{ is odd} \end{cases} \tag{3.1}$$

In the above formula, i is the dimension index (total number of dimensions are indicated by d) of the vector and t is the index of the token in the sentence. In formula 3.1, ω_i is calculated as follows:

$$\omega_i = \frac{1}{1000^{\frac{2i}{d}}} \tag{3.2}$$

Overall, we end up with the following vector for the position embeddings:

$$\vec{p_t}^{(i)} = \begin{bmatrix} sin(w_0 \times t) \\ cos(w_0 \times t) \\ \ldots \\ sin(w_{\frac{d}{2}} \times t) \\ cos(w_{\frac{d}{2}} \times t) \end{bmatrix} \tag{3.3}$$

3.3.3 XLNET

Similar to BERT, XLNET [13] is built upon the transformers. The available models for XLNET are cased XLNET base and cased XLNET large. The number of layers, attention heads and neurons is similar to those of cased BERT base and cased BERT large.

XLNET has its own specific tokenizer. The XLNET-based tokenized version of the example [61] we provided for BERT in section 3.3.2 is as follows:

- **XLNET Base cased example [61]**: [< >, <Rub>, <the>, <minerals>, <together>, <and>, <see>, <which>, <one>, <scratches>, <the>, <other>, <.>]

As the above-mentioned tokenized example shows, we have three different lengths for each tokenized BERT Base uncased, Base cased and XLNET Base cased example. They are 12, 14 and 13 respectively. We observe that the tokenization of words *Rub* and *scratches* result in different tokens and thus different lengths.

Principally, XLNET calculates the probability of a token within all permutations of the token in a sentence instead of considering only the left or right context. An example of permutation is provided in table 3.6. As the table shows, there are 24 (N!) permutations for a four token sentence [< >,"Rub","the","minerals"]. We note that permutation makes optimization a rather challenging problem, which leads to slow convergence. XLNET solves this problem by predicting the last tokens in a factorization order, which is called partial prediction by the authors.

Table 3.6 A toy example for permutation of "Rub the minerals" by XLNET tokenizer

(< >,"Rub","the","minerals"),(< >,"Rub","minerals","the"),(< >,"the","Rub","minerals"),
(< >,"the","minerals","Rub"),(< >,"minerals","Rub","the"),(< >,"minerals","the","Rub"),
("Rub",< >,"the","minerals"),("Rub",< >,"minerals","the"),("Rub","the",< >,"minerals"),
("Rub","the","minerals",< >),("Rub","minerals",< >,"the"),("Rub","minerals","the",< >),
("the",< >,"Rub","minerals"),("the",< >,"minerals","Rub"),("the","Rub",< >,"minerals"),
("the","Rub","minerals",< >),("the","minerals",< >,"Rub"),("the","minerals","Rub",< >),
("minerals",< >,"Rub","the"),("minerals",< >,"the","Rub"),("minerals","Rub",< >,"the"),
("minerals","Rub","the",< >),("minerals","the",< >,"Rub"),("minerals","the","Rub",< >)

After permutations are built, XLNET uses the formula 3.4 to maximize the probability of the tokens in a given sequence of length L with permutation z [105].

$$\hat{\Theta} = \underset{\Theta}{argmax}[E_{z\sim Z}[\sum_{l=1}^{L} log[Pr(x_{z[l]}|x_{z[<l]})]]] \tag{3.4}$$

Overall, XLNET architecture is inspired mainly by BERT. However, there are differences. It is built upon a bidirectional autoregressive language model, which means it predicts future tokens based on past tokens. On the contrary, BERT model is a bidirectional autoencoder. From this perspective, XLNET relies on the potential permutations of context words nearby a target word. This helps XLNET avoid the [MASK] token, which creates an artificial situation in the pre-training and fine-tuning stage that is not retrieved in a test situation in BERT. Also, permutation helps XLNET cover the dependencies between words as well. This is ignored by BERT as it considers independence of a predicted token from the rest of the sequence. We provide the following example to make the dependency problem clear.

$([MASK] \mapsto$ *Rub) the minerals together and see which one* $([MASK] \mapsto$ *scratches) the other.*

In this example, both words "Rub" and "scratches" are masked to be predicted by BERT. When the prediction of "Rub" influences the prediction of "scratches", BERT falls short. This is where XLNET is successful through using an attention mask. In other words, the XLNET model considers the context using the same token order and masks the tokens which are not in the context.

During pretrainng, XLNET borrows two features of Transformer-XL [106] in permutation language modeling: relative positional embeddings and the segment recurrence mechanism. For the latter, XLNET allows the model to use hidden states which belong to the previous segments. Therefore, the representation for each new segment benefits the larger context. Moreover, Transformer-XL [106] helps XLNET perform well in terms of perplexity and avoid context fragmentation problem. In other words, XLNET can handle long sequence dependencies properly.

Finally, XLNET model consists of two types of self-attentions:

- **Content stream representation:** In this type of self-attention, the content stream vectors are initialized with token embeddings added to positional embeddings.
- **Query representation:** This type of self-attention replaces the [MASK] from BERT only with position information but not its content.

3.4 Summary of our Architecture for our Final Models

According to what we explained in the previous sections of this chapter, we prepare the data based on an entailment structure. To summarize, we hypothesize that correct student answer entails reference answer. Then, we conduct a classification task which includes two types of experiments: one-level and two-level fine-tuning.

As figure 3.2 shows, we take training examples from the dataset and feed them as input into the pretrained BERT and XLNET models. After that, we take the output of the CLS token and pass it to the softmax function in order to produce the label. Figure 3.3 shows that we use the embeddings fine-tuned on GLUE datasets in our general ASAG pipeline.

3.5 GLUE Tasks

GLUE (General Language Understanding Evaluation)[7] [107] includes several natural language understanding tasks which are trained, evaluated and analyzed. Based on GLUE

[7] https://gluebenchmark.com/tasks

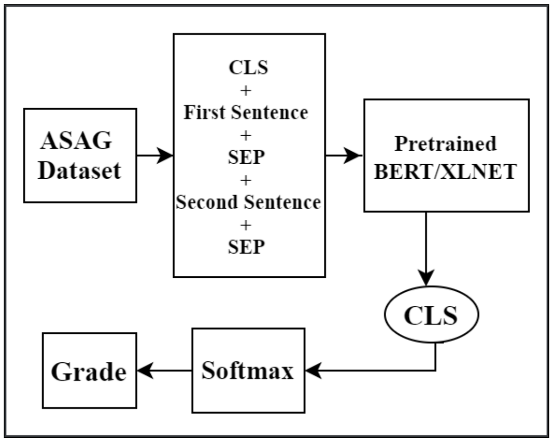

Figure 3.2 Our general pipeline for one-level fine-tuning

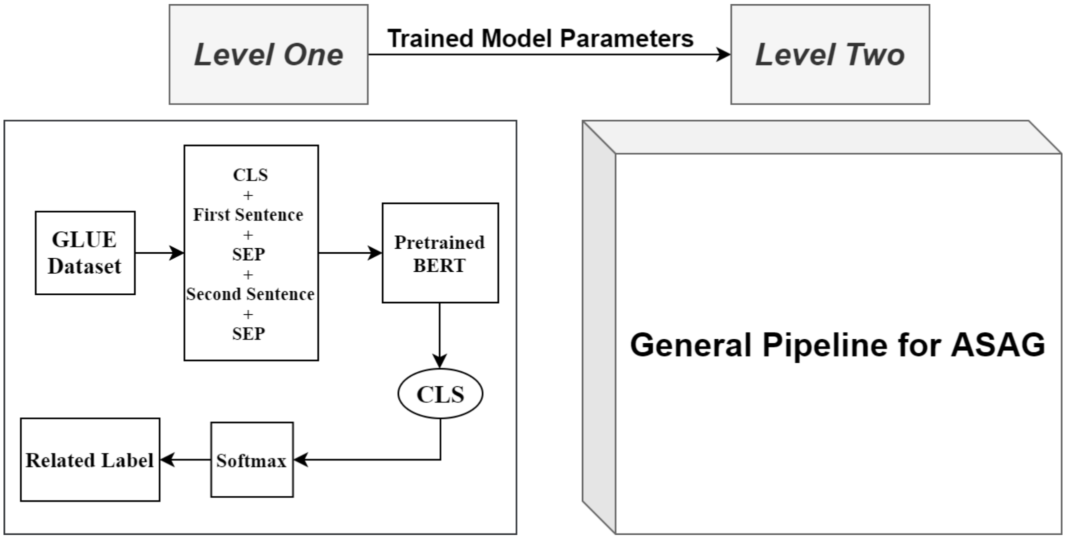

Figure 3.3 Our pipeline for two-level fine-tuning

relevant tasks to ASAG, we select the following tasks to conduct two-level fine-tuning based on a set of hypotheses that are described below. These tasks fall within the entailment, question answering, paraphrase identification and semantic similarity groups.

- The entailment group of tasks is selected based on our main assumption of ASAG in the current thesis.
- As ASAG includes questions and answers, the question answering group of tasks bares resemblance to ASAG.
- Sultan et al. [81] considers ASAG as a paraphrase identification task in which student answer is the paraphrase of the model answer. As a result, we select this task in our

two-level fine-tuning phase.

- As we try to find similarity between student and model answer, we also select a semantic similarity task among the GLUE tasks.

Each GLUE task includes a relevant dataset. Table 3.7 provides the approximate size for the datasets relevant to each task.

Table 3.7 GLUE datasets and their approximate size

Dataset	Size
MNLI (Multi-Genre Natural Language Inference)	433k
SQuAD (The Stanford Question Answering Dataset)	100k
QNLI (Question Natural Language Inference)	116k
STS-B (Semantic Textual Similarity Benchmark)	8500
MRPC (Microsoft Research Paraphrase Corpus)	5500
Textual Entailment Recognition using RTE	5500

3.5.1 Entailment

- *MNLI (Multi-Genre Natural Language Inference*)

In this task [108], a pair of sentences is classified as *entailment, contradiction, or neutral.* The data provided for the task is a crowd-sourced gathering of human-annotated sentence pairs. These pairs include premise and hypothesis. The dataset is provided in several genres such as transcriptions, reports, speeches, letters, and press releases, short posts and fiction.

The examples for each class of sentence pairs in MNLI are as follows. Note the premise entails the hypothesis:

- **label:** *entailment*
  - *Premise:* At the other end of Pennsylvania Avenue, people began to line up for a White House tour.
  - *Hypothesis:* People formed a line at the end of Pennsylvania Avenue.

  label: *neutral*
  - *Premise:* The Old One always comforted Ca"daan, except today.
  - *Hypothesis:* Ca"daan knew the Old One very well.

label: *contradiction*

  - *Premise:* Yes now you know if if everybody like in August when everybody"s on vacation or something we can dress a little more casual or
  - *Hypothesis:* August is a black out month for vacations in the company.

- *Textual Entailment Recognition using RTE*

The task [107] is similar to MNLI task. However, the number of examples is less in RTE. The dataset is the combination of RTE1 [10], RTE2 [109], RTE3 [110] and RTE5 [111]. The type of data is news and Wikipedia. In the following example, the text entails the hypothesis:

- **label:** *not entailment*
  - *Sentence1:* No Weapons of Mass Destruction Found in Iraq Yet
  - *Sentence2:* Weapons of Mass Destruction Found in Iraq.

  label: *entailment*
  - *Sentence1:* The currency used in China is the Renminbi Yuan.
  - *Sentence2:* The Renminbi Yuan is the currency used in China.

3.5.2 Question Answering

- *Question Answering using SQuAD*

In this task, the goal is to predict the span of the Wikipedia passage that represents an answer for a question. We use SQuAD (The Stanford Question Answering Dataset) v1.1 [112]. The following example is provided from this dataset.

- *Question:* Which NFL team represented AFC at Super Bowl 50?
- *Context:* Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily

suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50.

- *Answer:* [answer start: 177] "Denver Broncos"

- *Question-Answer Classification using QNLI*

QNLI [107] (Question Natural Language Inference) is a binary classification version of Stanford Question Answering Dataset [112]. There are positive and negative examples in this dataset. The positive examples contain question and correct answer pairs (entailed). On the contrary, the negative examples include an unrelated answer from the same paragraph (not entailed). The following positive and negative examples are extracted from QNLI [107] dataset:

- **label:** *entailment*
  - *Question:* How many alumni does Olin Business School have worldwide?
  - *Sentence:* Olin has a network of more than 16,000 alumni worldwide.

  label: *not entailment*
  - *Question:* The environmental intervention was linked to the conceptualization of what process?
  - *Sentence:* Between 1791 and 1833, Saint Helena became the site of a series of experiments in conservation, reforestation and attempts to boost rainfall artificially.

3.5.3 Paraphrase Identification

Being collected from news sources, MRPC (Microsoft Research Paraphrase Corpus) [113] provides a benchmark for automatic paraphrase identification. In fact, the goal is to see if the second sentence is the paraphrase of the first or not. An example of for each class is provided as follows. The label 1 means the second sentence is the paraphrase of the first one. Otherwise, label 0 is given for the pair of sentences.

- **label:** *1 (paraphrase)*
  - *Sentence1:* The DVD-CCA then appealed to the state Supreme Court.
  - *Sentence2:* The DVD CCA appealed that decision to the U.S. Supreme Court.

label: *0 (not paraphrase)*

  - *Sentence1:* America Online last quarter lost 846,000 dial-up subscribers
  - *Sentence2:* No wonder AOL lost 846,000 subscribers last quarter.

3.5.4 Semantic Textual Similarity

STS-B[8] [114] (Semantic Textual Similarity Benchmark) is a regression task wherein a sentence pair is scored from 0-5 based on similarity. The dataset was obtained from news headlines, video and image descriptions, glosses from lexical resources including WordNet [45], FrameNet [51], OntoNotes, web discussion forum, plagiarism, machine translation post-editing and QA datasets. In what follows, we provide three examples with the lowest, middle and highest similarity.

- **score:** *0*
  - *Sentence1:* Two red buses driving in front of a garden.
  - *Sentence2:* Train pulling into a station.
- **score:** *2.5*
  - *Sentence1:* The black bird is sitting on the ground.
  - *Sentence2:* The bird is sitting on the branch.
- **score:** *5*
  - *Sentence1:* A plane is taking off.
  - *Sentence2:* An airplane is taking off.

3.6 Experimental Setup

In this section, we explain the hyperparameters we used for our baseline models (Naive Bayes, support-vector machines, decision tree algorithms, random forests and logistic regression) and final models (BERT and XLNET). These hyperparameters are particularly useful for reproducing our results.

For the baseline models, we use the scikit-learn default hyperparameters. For our final models, we have hyperparameters in BERT [12] and XLNET [13] to tune for our task. Our hyperparameters are as follows:

[8] http://ixa2.si.ehu.es/stswiki/index.php/Main$_{Page}$

- Epochs = 10
- Dropout probability for all the layers = 0.1
- Warmup Proportion = 0.1
- Mini Batch size = 16
- Learning rate = 5e-6 for BERT and 5e-5 for XLNET
- Random seed = 123

We set all the above-mentioned hyperparameters experimentally. Other than using the dropout probability, we monitor the loss of train and validation set in each epoch in order to have a control on the fitting process. In this regard, we stop the training process before the losses increase. This helps removing the risks of overfitting and underfitting.

For the infrastructure, we used the GPUs with following specifications:

- Google Colab (NVIDIA Tesla T4) with the memory of 15109 Mebibyte (Mib)
- NVIDIA Quadro P6000 with the memory of 24449 Mebibyte (Mib)

Overall, it took approximately 1 hour for one-level fine-tuning tasks and 6-10 hours (it depends on the dataset size) for the two-level fine-tuning tasks to be trained. Also, it took 1-3 seconds (it depends on the length of the sentence) on average for classifying an example student answer using our trained models.

3.7 Evaluation Measures

We evaluate all our grading models using the SemEval-2013 [1] challenge measures: Accuracy (ACC), Macro average F1 score (M-F1) and Weighted average F1 score (W-F1). These measures are explained in formula 2.5, formula 2.9 and formula 2.10 in chapter 2.

3.8 Summary

In this chapter, we described our 4 datasets and the types of classification on each of them. Our datasets are SemEval [1] (SciEntBank and BEETLE), Dt-grade [3] and Biology dataset [2]. Our classification types are 2, 3, 4 and 5way.

In the next part of this chapter, we detailed our baselines and final models. For our baseline models, we selected Naive Bayes, support-vector machines, decision tree algorithms, random forests and logistic regression. For these algorithms, we use a TFIDF-based bag of words model with unigrams, bigrams and trigrams. For the final models, we conduct ASAG using BERT [12] and XLNET [100]. After that, we provided details about the GLUE MNLI, SQuAD, STS-B, QNLI, MRPC and RTE tasks, which were tested to evaluate a potential impact on ASAG.

Finally, we defined our hyperparameters and evaluation measures. In the next chapter, we present our results for each of our experiments in detail.

CHAPTER 4 RESULTS

In this chapter, we present the evaluation results of our models on several datasets and compare them with the results achieved in SOTA. We group datasets based on the expected number of output labels: 2way, 3way, 4way and 5way. On each type of classification, we compare our final models (BERT and XLNET models) with baseline systems through bar charts as the differences are visually noticeable. Then, we compare our models with the SOTA through tables because we want to illustrate clearly the numerical differences. In addition, we report the performance of transfer learning experiments using a table.

4.1 SciEntBank

SciEntBank is one of the SemEval-2013 [1] datasets introduced in chapter 3. In this section, we report on the results achieved through our experiments of 2way, 3way and 5way ASAG tasks on TUA, TUQ and TUD datasets (described in chapter 3 section 3.2.1). We note that we focus on our one-level fine-tuned models in this section.

4.1.1 Results for 2way Task

As we mentioned in chapter 3, 2way classification includes these labels: correct and incorrect.

We illustrate the results from our 5 baseline algorithms using TFIDF-based BOW of unigrams, bigrams and trigrams in Figure 4.1. For all the baseline models, we achieved the best results on TUA which includes unseen answers. Overall, the results are similar for all 5 algorithms. As the figure shows, all our BERT and XLNET models produced better results than those of the baseline models.

We also provide the results of the 2way experiments on the SciEntBank dataset in Table 4.1. As Table 4.1 shows, we compare the results of 8 top-performing SOTA systems to the results of our three experiments.

In terms of unseen answers, the results we obtained with our BERT models are the best in SOTA despite not considering the questions and without hand-crafted features. In terms of unseen questions and domains, Saha et al. [44] system was the top system. As explained in chapter 2, this system used token features (TF) and sentence features (SF). Followed by this system, the Softcardinality [72] model and our XLNET Base cased model are competing with Sultan et al. [81] system.

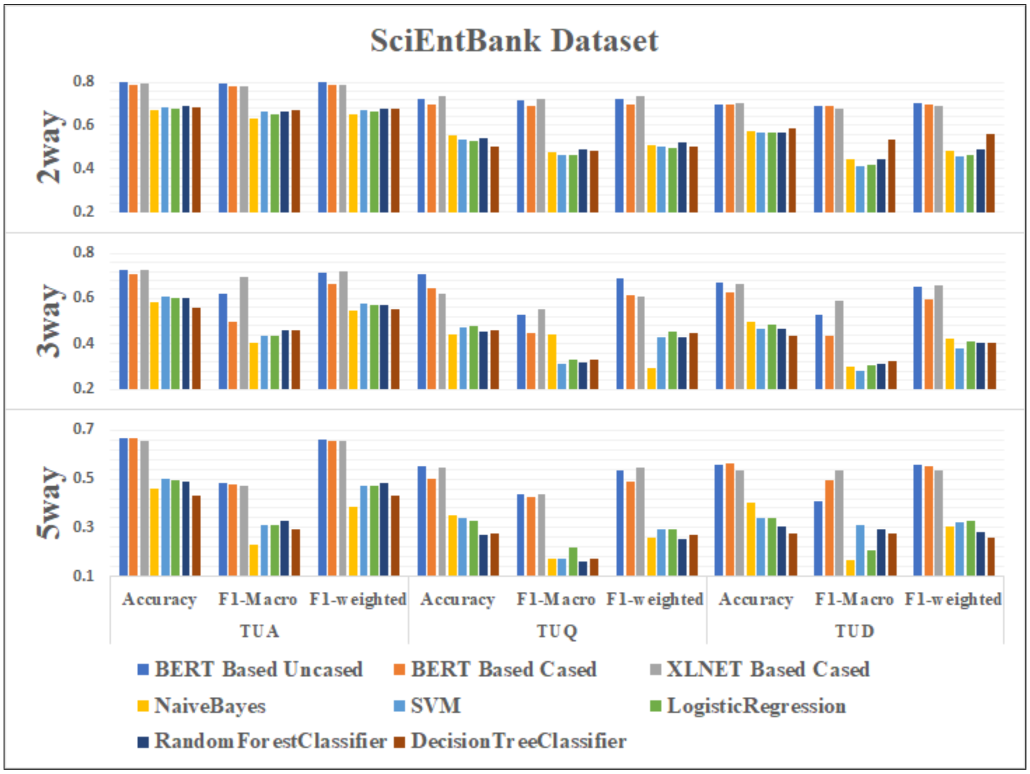

Figure 4.1 Comparison of the results of our baseline models with our final models (2way, 3way and 5way tasks on TUA, TUQ and TUD test sets)

Table 4.1 Comparison of BERT and XLNET models with SOTA on 2way SciEntBank dataset: The highlighted numbers are the best in the SOTA

	TUA			**TUQ**			**TUD**		
	ACC	**M-F1**	**W-F1**	**ACC**	**M-F1**	**W-F1**	**ACC**	**M-F1**	**W-F1**
COMET [90]	0.774	0.768	0.773	0.603	0.579	0.597	0.676	0.67	0.677
ETS [88]	0.776	0.762	0.77	0.633	0.602	0.622	0.627	0.543	0.574
SOFTCARDINALITY [72]	0.724	0.715	0.722	0.745	0.737	0.745	0.711	0.705	0.712
Sultan [81]	0.708	0.676	0.69	0.705	0.678	0.695	0.712	0.703	0.712
Graph [115]	-	0.644	0.658	-	-	-	-	-	-
MEAD [115]	-	0.631	0.645	-	-	-	-	-	-
TF+SF [-question] [44]	0.779	0.771	0.777	**0.749**	**0.738**	**0.747**	0.708	0.690	0.702
TF+SF [+question] [44]	0.792	0.785	0.791	0.702	0.685	0.698	**0.719**	**0.708**	**0.717**
Mavarniya [116]	-	0.773	0.781	-	-	-	-	-	-
BERT Base uncased	**0.798**	**0.792**	**0.797**	0.723	0.718	0.724	0.699	0.693	0.7
BERT Base cased	0.79	0.783	0.788	0.697	0.690	0.698	0.698	0.689	0.697
XLNET Base cased	0.792	0.781	0.788	0.736	0.724	0.734	0.702	0.679	0.693

4.1.2 Results for 3way Task

As we mentioned in chapter 3, 3way classification includes three labels: correct, contradictory and incorrect.

As we display in Figure 4.1, we achieved the best results (both by our baseline and final models) on TUA followed by TUD and TUQ respectively in terms of all the measures. As a whole, the results are roughly equal for all 5 algorithms using TFIDF-based BOW of unigrams, bigrams and trigrams. As the figure shows, all our final models yielded better results than those of the baseline models.

We show the results of the 3way experiments on the SciEntBank dataset in Table 4.2. Our Table 4.2 demonstrates the results of 9 top-performing systems along the results of our 3 experiments.

Overall, the results of Sung et al. [94] who also use a BERT model are better than our results on SOTA with regard to unseen answers. However, we couldn't reproduce their results as we have no access to their experimental setup. In terms of unseen questions, BERT Base uncased model outperformed the SOTA considering the ACC and W-F1 scores. Finally, our XLNET Base cased model performed better than all other systems on the unseen domain test set. Followed by this model, the results on the BERT Base uncased model are the best, especially in terms of ACC. As Table 4.2 shows, BERT models are robust on unseen answers and unseen questions while XLNET model works better for unseen domains.

Table 4.2 Comparison of the BERT and XLNET models with SOTA on 3-way SciEntBank dataset: The highlighted numbers are the best in the SOTA

	TUA			**TUQ**			**TUD**		
	ACC	**M-F1**	**W-F1**	**ACC**	**M-F1**	**W-F1**	**ACC**	**M-F1**	**W-F1**
COMET [90]	0.713	0.64	0.707	0.546	0.38	0.522	0.579	0.404	0.55
ETS [88]	0.72	0.647	0.708	0.583	0.393	0.537	0.543	0.333	0.461
SOFTCARDINALITY [72]	0.659	0.555	0.647	0.652	0.469	0.634	0.637	0.486	0.62
Sultan [81]	0.604	0.443	0.569	0.642	0.455	0.615	0.626	0.451	0.603
Graph [115]	-	0.438	0.567	-	-	-	-	-	-
MEAD [115]	-	0.429	0.554	-	-	-	-	-	-
TF+SF [-question] [44]	0.718	0.666	0.714	0.613	0.491	0.628	0.632	0.479	0.611
TF+SF [+question] [44]	0.718	0.657	0.711	0.653	0.489	0.636	0.640	0.452	0.61
Marvaniya [116]	-	0.636	0.719	-	-	-	-	-	-
Sung et. al. [94]	**0.759**	**0.72**	**0.758**	0.653	**0.575**	0.648	0.638	0.579	0.634
BERT Base uncased	0.726	0.622	0.714	**0.708**	0.528	**0.686**	**0.672**	0.528	0.6514
BERT Base cased	0.707	0.5	0.667	0.645	0.45	0.616	0.63	0.438	0.6
XLNET Base cased	0.726	0.7	0.723	0.622	0.55	0.61	0.665	**0.6**	**0.657**

4.1.3 Results for 5way Task

As we introduced in chapter 3, 5way classification includes five labels: correct, incorrect, contradictory, partially-correct-incomplete, irrelevant and non-domain.

The 5way results using our BERT and XLNET models, as illustrated in Figure 4.1, are better than those of the baseline models. We achieved slightly better results on TUD than on TUQ. On each test set, the results are similar for all 5 algorithms using TFIDF-based BOW of unigrams, bigrams and trigrams.

We present the results of the 5way experiments on the SciEntBank dataset in Table 4.3. Similar to 2way and 3way which are all based on SemEval-2013 [1], we explain our results in terms of accuracy, macro F1-score and weighted F1 score. Table 4.2 demonstrates the results of 8 top-performing systems along the results of our 3 experiments.

In general, we outperformed the SOTA with all our models. For unseen answers, BERT Base uncased model dominates all our models for all the evaluation measures. Within unseen questions, BERT Base uncased model outperformed the SOTA in terms of ACC (around 3 percent) and M-F1 (around 6 percent) score. We achieved strong results using BERT Base cased model in ACC within unseen domains. The results of BERT Base uncased and XLNET Base cased models are the best in terms of W-F1 and M-F1 respectively for these unseen domains.

Table 4.3 Comparison of BERT and XLNET models with SOTA on 5-way SciEntBank dataset: The highlighted numbers are the best in the SOTA

	TUA			**TUQ**			**TUD**		
	ACC	**M-F1**	**W-F1**	**ACC**	**M-F1**	**W-F1**	**ACC**	**M-F1**	**W-F1**
COMET [90]	0.6	0.441	0.598	0.437	0.161	0.299	0.421	0.121	0.252
ETS [88]	0.643	0.478	0.64	0.432	0.263	0.411	0.441	0.38	0.414
SOFTCARDINALITY [72]	0.544	0.38	0.537	0.525	0.307	0.492	0.512	0.3	0.471
Sultan [81]	0.489	0.3298	0.487	0.480	0.302	0.467	0.506	0.344	0.484
Graph [115]	-	0.372	0.458	-	-	-	-	-	-
MEAD [115]	-	0.379	0.461	-	-	-	-	-	-
TF+SF [-question] [44]	0.644	0.480	0.642	0.5	0.316	0.488	0.508	0.357	0.492
TF+SF [+question] [44]	0.629	0.472	0.630	0.506	0.376	0.471	0.51	0.342	0.486
Mavarniya [116]	-	0.579	0.61	-	-	-	-	-	-
BERT Base uncased	**0.66**	**0.484**	**0.662**	**0.552**	**0.437**	0.533	0.557	0.41	**0.558**
BERT Base cased	**0.66**	0.478	0.658	0.5	0.424	0.487	**0.562**	0.5	0.552
XLNET Base cased	0.658	0.47	0.655	0.544	0.435	**0.545**	0.532	**0.535**	0.535

4.2 BEETLE

As we explained in chapter 3, BEETLE is the second dataset introduced in SemEval-2013 competition [1]. In this section, we describe the results we obtained via our experiments of 2way, 3way and 5way ASAG on this dataset.

4.2.1 Results for 2way Task

Similar to SciEntBank labels, 2way classification includes two labels: correct and incorrect. However, the only available test sets for this dataset are TUA and TUQ datasets.

As Figure 4.2 shows, we achieved roughly similar results for all our baseline models on all the test sets in terms of all the measures. Again, our final models produced better results than those of the baseline models (using TFIDF-based BOW of unigrams, bigrams and trigrams) in TUA. However, the baseline results closely compete with those of our final models in TUQ.

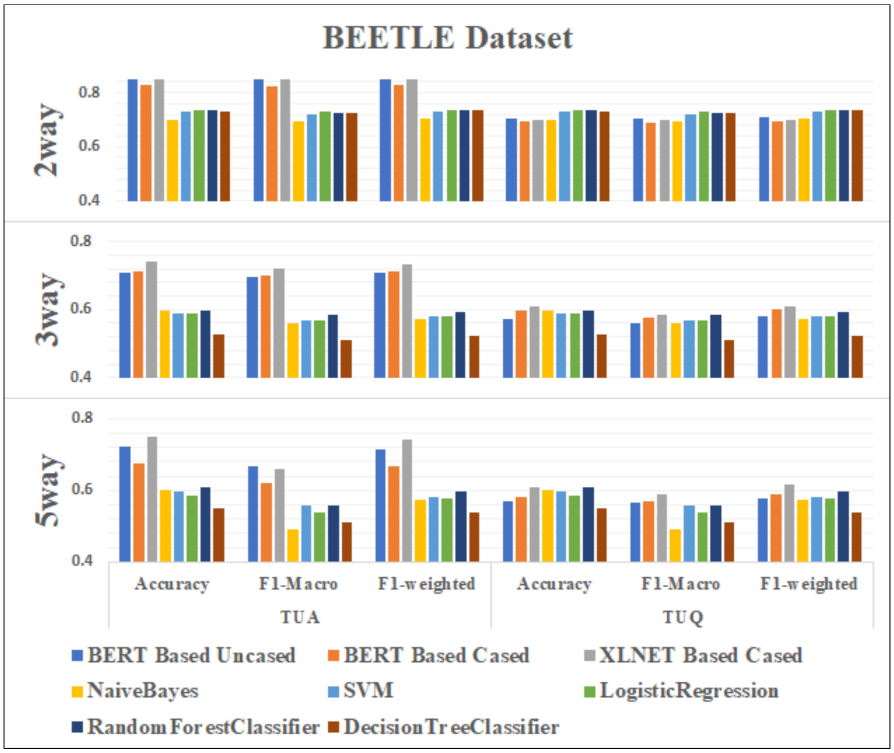

Figure 4.2 Comparison of the results of our baseline models with our final models (2way, 3way and 5way tasks on TUA and TUQ test sets)

We provide the results of the 2way experiments on the BEETLE dataset in Table 4.4 together with the results of 2 top-performing systems. It should be noted that not all the results for each evaluation measure are provided in the papers. As a result, some of the cells are empty in the table.

In terms of unseen answers and questions, our BERT Base uncased model outperformed the SOTA with all our measurements. By the XLNET model, we increased the SOTA by 5 percent (in terms of all the measurements) for unseen answers. For unseen questions, the accuracy of ETS [88] is the best. The M-F1 score of BERT Base uncased model is a little better than COMET [90]. Overall, all our 3 models are better than or compete with the SOTA.

Table 4.4 Comparison of the proposed system with SOTA on 2-way BEETLE dataset: The highlighted numbers are the best in the SOTA

	TUA			**TUQ**		
	ACC	**M-F1**	**W-F1**	**ACC**	**M-F1**	**W-F1**
COMET [90]	0.838	0.833	-	0.702	0.695	-
ETS [88]	-	0.802	-	**0.72**	-	-
BERT Base uncased	0.854	0.851	0.855	0.707	**0.702**	**0.707**
BERT Base cased	0.828	0.823	0.829	0.692	0.692	0.693
XLNET Base cased	**0.882**	**0.8791**	**0.883**	0.700	0.698	0.699

4.2.2 Results for 3way Task

Similar to SciEntBank labels, 3way classification includes three labels: correct, contradictory and incorrect. As we mentioned for 2way ASAG, TUA and TUQ datasets are the available test sets for evaluating the trained models.

As Figure 4.2 shows, we achieved roughly similar results on all the test sets in terms of all the measures though the decision tree classifier performed rather weakly. The figure shows that our final models performed better than the baseline models (using TFIDF-based BOW of unigrams, bigrams and trigrams) in TUA. However, the baseline models compete tightly with our final models in TUQ.

The results of the 3way experiments on the BEETLE dataset are shown in Table 4.5. We note that some of the values for each evaluation measure are not available in the papers. In terms of unseen answers and questions, our XLNET Base cased model outperformed the SOTA. Both our BERT models behave in an identical manner. In terms of ACC, the results of our XLNET model is different with the SOTA by 9 percent.

Table 4.5 Comparison of the proposed system with SOTA on 3-way BEETLE dataset: The highlighted numbers are the best in the SOTA

	TUA			**TUQ**		
	ACC	**M-F1**	**W-F1**	**ACC**	**M-F1**	**W-F1**
COMET [90]	0.731	0.715	0.728	0.518	0.466	0.488
ETS [88]	-	0.71	0.723	-	0.585	0.597
BERT Base uncased	0.710	0.697	0.708	0.571	0.558	0.579
BERT Base cased	0.714	0.699	0.712	0.595	0.574	0.6
XLNET Base cased	**0.74**	**0.722**	**0.733**	**0.6081**	**0.586**	**0.611**

4.2.3 Results for 5way Task

The 5way classification task includes five labels: correct, incorrect, contradictory, partially-correct-incomplete, irrelevant and non-domain. As stated in section above, the results for BEETLE are only available for TUA and TUQ test sets.

Figure 4.2 indicates better results on all the test sets in terms of all the measures. While being compared on TUQ, the results of baseline models (using TFIDF-based BOW of unigrams, bigrams and trigrams) and final models are not significantly different.

As we pointed out in section 4.2.1 and section 4.2.2, we show the results on TUA and TUQ test sets in Table 4.6. We compare our results with 3 top-performing systems. As a number of values for ACC are not available in the papers, we use dash in the place of their values.

Our XLNET Base cased model performed the best both in SOTA (with increase of 7-10 percent in all the evaluation metrics) and among all our models on unseen answers. Followed by XLNET Base cased model, BERT Base uncased model predicted the grades strongly. As Table 4.6 shows, BERT Base cased model performed better than BERT Base uncased model in the case of ACC and W-F1 scores in terms of unseen questions.

Table 4.6 Comparison of the proposed system with SOTA on 5-way BEETLE dataset: The highlighted numbers are the best in the SOTA

	TUA			**TUQ**		
	ACC	**M-F1**	**W-F1**	**ACC**	**M-F1**	**W-F1**
COMET [90]	0.688	0.569	0.675	0.488	0.3	0.445
ETS [88]	-	0.619	0.705	-	0.552	0.614
Archand [80]	-	0.597	0.709	-	**0.592**	**0.625**
BERT Base uncased	0.721	**0.666**	0.717	0.568	0.566	0.576
BERT Base cased	0.675	0.622	0.668	0.581	0.568	0.591
XLNET Base cased	**0.7513**	0.66	**0.743**	**0.608**	0.590	0.618

4.2.4 Biology

We used the same version of the Biology dataset used by Mantecon et al. [8,102]. It includes correct and incorrect labels. We show the results from our 5 baseline algorithms in Figure 4.3. Overall, the results are similar for all 5 algorithms (using TFIDF-based BOW of unigrams, bigrams and trigrams) though decision tree performed slightly better. However, all our final models produced better results than those of the baseline models. Among our final models, the results of the XLNET Base cased model are on the top.

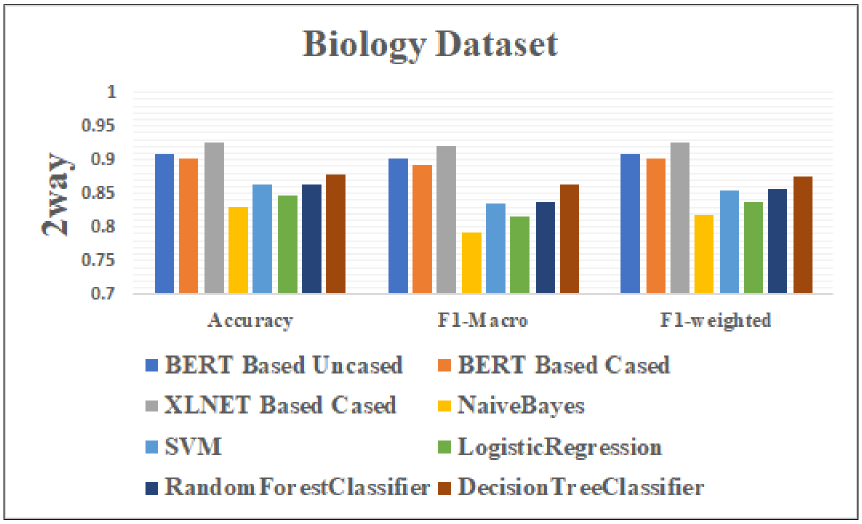

Figure 4.3 Comparison of Baseline and Our Proposed Models

According to Table 4.7, all our models achieved results above 0.9. XLNET base cased model achieved roughly similar results (with less than 1 percent difference) to those obtained by Mantecon [8] model in which he stacked the response-based and reference-based models and used FastText embeddings [117] among other features. His evaluation procedure is based on 10 fold cross-validation. On the contrary, we separated train, validation and test sets based on the question-aware separation explained in chapter 3.

Table 4.7 Results achieved on the Biology dataset

	ACC	**M-F1**
Best Results of Mantecon [8]	0.934	0.928
BERT Base uncased	0.91	0.901
BERT Base cased	0.901	0.891
XLNET Base cased	0.925	0.92

4.3 Dt-grade

The Dt-grade dataset contains correct, correct-but-incomplete, contradictory and incorrect labels. Figure 4.3 shows the results obtained using our baseline and final models. Overall, the weaker results belong to Naive Bayes among 5 algorithms (using TFIDF-based BOW of unigrams, bigrams and trigrams). All our final models produced better results than those of the baseline models.

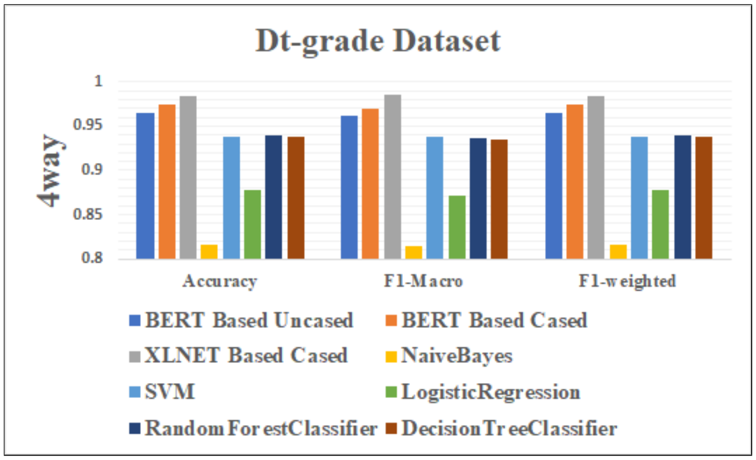

Figure 4.4 Comparison of baseline and our proposed models

In this section, we compare the results of our 3 models with the SOTA systems in Table 4.8. As previous works only analyzed ACC and W-F1, we only provide the values of these evaluation measures for our models.

The results of the two SOTA systems [42, 85] seem to be almost similar. Maharjan et al. [85] used several hand-crafted features such as content word counts, word overlap and cosine similarity in Gaussian mixture model (GMM). Rus [42] used same types of features similar to [85] considering anaphora resolution and negation. All our 3 models outperformed the results reported in these papers, among which XLNET model is above 0.98 both in ACC and W-F1. BERT Base cased model worked better than BERT Base uncased model though not considerably. Overall, the difference between our results and SOTA is around 40 percent. This is significant in SOTA as both previous works used hand-crafted features.

4.4 Results of Transfer Learning on SciEntBank

In this section, we explain the results obtained using two-level fine-tuning of BERT on only SciEntBank dataset. As we explained in chapter 1, we explore how transfer learning from GLUE tasks to ASAG help us obtain stronger results.

Table 4.8 Comparison of the proposed system with SOTA on 4-way Dt-grade dataset: The highlighted numbers are the best in the SOTA

	ACC	**W-F1**
GMM [85]	0.58	0.582
Rus [42]	0.5881	0.5739
BERT Base uncased	0.965	0.965
BERT Base cased	0.974	0.974
XLNET Base cased	**0.984**	**0.984**

4.4.1 2way SciEntBank

As Table 4.9 shows, the majority of the GLUE-based models improve the performance of our one-level fine-tuning of BERT model.

For unseen answers, all the results obtained by SQuAD, MRPC and RTE are better than those of the one-level fine-tuning of BERT model. Among these models, RTE performed best in terms of all the evaluation metrics.

For unseen questions, the results achieved by all the GLUE-based models except RTE surpass the performance of the one-level fine-tuning of BERT model. MNLI and SQuAD brought off the best results among our GLUE-based models.

Similar for evaluation of unseen questions test set, the results of the GLUE-based models except RTE and SQuAD improved the performance of the one-level fine-tuning of BERT model in unseen domains. As Table 4.9 depicts, MNLI produced the best results.

4.4.2 3way SciEntBank

As Table 4.9 shows, a large proportion of the GLUE-based models improved the results achieved by our one-level fine-tuning of BERT model.

For unseen answers, the results obtained by MNLI and QNLI are better than those of the one-level fine-tuning of BERT model. Between these two models, MNLI performed better especially in terms of M-F1 score.

For unseen questions, MRPC and SQuAD brought off the best results among our GLUE-based models in terms of M-F1 score. For the unseen domains, the results of the GLUE-based models did not improve much the performance of the one-level fine-tuning of BERT model. According to Table 4.9, SQuAD, MRPC and QNLI performed well in terms of M-F1 score. Finally, the best increase in terms of W-F1 score belongs to QNLI.

Table 4.9 Comparison of BERT and GLUE-based BERT on 2way, 3way and 5way SciEntBank dataset: The highlighted numbers indicate highest increase in the performance

Classification Type	**Models**	**TUA**			**TUQ**			**TUD**		
		ACC	**M-F1**	**W-F1**	**ACC**	**M-F1**	**W-F1**	**ACC**	**M-F1**	**W-F1**
2way	**BERT**	0.798	0.793	0.797	0.723	0.719	0.725	0.700	0.694	0.701
	MNLI	0.787	0.780	0.785	**0.756**	**0.750**	**0.757**	**0.735**	**0.724**	**0.732**
	SQuAD	0.809	0.804	0.809	0.753	0.748	0.754	0.694	0.685	0.693
	MRPC	0.802	0.797	0.801	0.738	0.734	0.740	0.704	0.698	0.705
	QNLI	0.791	0.793	0.798	0.744	0.741	0.746	0.710	0.705	0.711
	STS-B	0.789	0.784	0.789	0.734	0.733	0.736	0.699	0.698	0.700
	RTE	**0.815**	**0.810**	**0.814**	0.711	0.708	0.713	0.678	0.674	0.680
3way	**BERT**	0.726	0.622	0.714	0.708	0.528	0.686	0.672	0.528	0.651
	MNLI	0.741	**0.684**	**0.735**	0.675	0.507	0.651	0.624	0.484	0.610
	SQuAD	0.720	0.642	0.713	0.668	**0.528**	0.652	0.652	**0.589**	0.643
	MRPC	0.715	0.642	0.711	0.696	0.520	0.673	0.655	0.529	0.640
	QNLI	**0.744**	0.646	0.734	0.682	0.503	0.661	0.665	0.579	**0.656**
	STS-B	0.709	0.626	0.707	0.679	0.509	0.659	0.653	0.523	0.638
	RTE	0.715	0.565	0.693	0.662	0.489	0.638	0.639	0.454	0.610
5way	**BERT**	0.667	0.485	0.662	0.553	0.438	0.534	0.557	0.409	0.558
	MNLI	**0.681**	**0.523**	**0.681**	**0.595**	**0.552**	**0.591**	0.557	**0.429**	**0.562**
	SQuAD	**0.681**	0.501	0.676	0.568	0.454	0.558	0.554	0.396	0.550
	MRPC	0.667	0.487	0.664	0.581	0.467	0.565	0.532	0.369	0.526
	QNLI	0.680	0.500	0.660	0.530	0.450	0.550	0.510	0.400	0.510
	STS-B	0.646	0.472	0.644	0.588	0.444	0.557	0.523	0.355	0.518
	RTE	0.674	0.504	0.674	0.558	0.424	0.525	0.538	0.371	0.534

4.4.3 5way SciEntBank

As Table 4.9 shows, every GLUE-based model performed better than or similar to our one-level fine-tuning of BERT model except STS-B.

For unseen answers, questions and domain, the results obtained by MNLI are the best among the GLUE-based models. Following MNLI, the best results belong to SQuAD in terms of unseen answers and questions. For unseen questions, the results achieved by all the GLUE-based models except QNLI in terms of ACC are better than those of the one-level fine-tuning of BERT model. According to Table 4.9, MRPC, QNLI, STS-B and RTE impacted negatively the performance of our BERT and XLNET models on the unseen domain test set.

4.5 Summary

In this chapter, we showed how the results of our final models are better than those of the baseline models. In addition, we indicated how our final models yield better or competitive results than those in the SOTA. Among the tasks, all our models performed considerably stronger as we moved from 2way to 5way type of classification when compared to the SOTA, Among our models in 5way tasks for example, XLNET model was the best in terms of all the measures. Conversely, BERT models were stronger in 2way classification especially in unseen answers.

To attempt to improve the BERT model performance, we also showed how transfer learning from GLUE benchmark is effective. Almost all the GLUE tasks made positive impact on the ASAG performance among which entailment/inference tasks such as MNLI were most effective in all types of classification.

In the next chapter, we discuss how BERT and XLNET models classify student answers. Furthermore, we investigate the limitations and future works.

CHAPTER 5 DISCUSSION AND CONCLUSION

In this chapter, we describe how our ASAG models classify the student answers in our datasets. Firstly, we discuss thoroughly the performance of BERT and XLNET models in terms of strong and weak points. Afterwards, we examine the predictive strength of the GLUE-based BERT models on the SciEntBank 2way dataset. Then, we discuss our qualitative analysis of the difficulties our models encountered while classifying the answers. Finally, we answer our research questions and make a conclusion about our findings. We also explain our limitations and potential directions for this work.

5.1 BERT

We trained BERT Base uncased and cased on SemEval [1] (SciEntBank and BEETLE), Dt-grade [3] and Biology dataset [2]. The ASAG tasks for SciEntBank and BEETLE datasets are 2 way, 3 way and 5 way. Dt-grade [3] dataset is 4 way and finally Biology dataset [2] is 2 way. According to the results presented in Chapter 4, our findings indicated that BERT yields better or competitive results when compared with the systems in SOTA.

According to results we obtained on all our datasets, we lose performance as we move from TUA to TUD test set. In fact, TUQ and TUD test sets seem to be difficult for both BERT models in terms of all our evaluation measures. In this regard, the results from Biology dataset [2] further proves that BERT models work well for the test sets which contain the same set of questions from the same domain. Otherwise, all the values for ACC, M-F1 and W-F1 decrease. Exceptionally in 5way task, the results for TUD are sometimes (visible in the values of ACC and W-F1) better than those of TUQ. On the other hand, the performance of BERT models for 3 way grading changes across datasets. For SciEntBank, the best results belong to BERT Base uncased model. Conversely, BERT Base cased model performed better for BEETLE dataset.

While comparing our results to SOTA, we observed that the BERT model seems to carry out 2way ASAG rather solidly in all the datasets. It competes with feature-based systems such as [44] and [81]. Overall, our best results belong to Biology [2] and Dt-grade [3] datasets. Furthermore, the best ASAG models belong to 2way classification of student answers except for Dt-grade [3].

5.2 XLNET

Similar to the experiments with BERT, we trained XLNET models for ASAG on SemEval [1] (SciEntBank and BEETLE), Dt-grade [3] and Biology datasets [2]. In general, XLNET seems to work better than BERT as the number of labels increase. However, we lose performance as we move from TUA to TUD. Our findings show that XLNET model depended highly on the questions and domain already seen.

As for Dt-grade [3] and Biology datasets [2], our XLNET model outperformed the SOTA and our BERT models in terms of all our evaluation measures. As the values for ACC and W-F1 are more than 0.98, we observe a conspicuous success of XLNET trained on Dt-grade [3].

5.3 GLUE-based Models

Our GLUE-based models include only BERT model on SciEntBank dataset. We trained a 2way, 3way and 5way ASAG model out of the embeddings obtained from the entailment, question answering, paraphrase identification and semantic similarity tasks.

According to the results presented in section 4.4, we found out that MNLI mostly improved the results we obtained by one-level fine-tuning of BERT model. For the 2way task, the results for unseen questions and unseen domains were better on the MNLI fine-tuned model. This seems to indicate that MNLI adds some generalizability to the one-level fine-tuning of BERT Base uncased model. This was also true for SQuAD and RTE in unseen answers though not significantly much better.

When we compare the results of 3way GLUE-based models to those of one-level fine-tuning of BERT Base uncased, MNLI was better only for unseen answers. In this regard, it competes with QNLI. On the other hand, MRPC showed close results to those of one-level fine-tuning of BERT Base uncased model on unseen questions. In terms of unseen domains, one-level fine-tuning of BERT Base uncased model performed better than all the GLUE-based models except for SQuAD on only M-F1.

For the 5 way task, MNLI improved the results of one-level fine-tuning of BERT Base uncased model on all TUA, TUQ and TUD in terms of all the evaluation measures. Following MNLI, MRPC and RTE yielded better or competitive results than the BERT Base models. The results obtained by QNLI on unseen answers illustrate at least 2 percent improvement over the results we achieved by one-level fine-tuning of BERT Base uncased model.

5.4 Difficulty of Datasets for BERT and XLNET

In this section, we discuss the difficulty of classification observed in our models. We used 4 datasets and trained separate models for each. As we showed in section 3.1, the number of questions and answers was different in each dataset. For example, BEETLE and Dt-grade datasets contain more than one model answer to be compared with the individual student answers. We might have had better results if this was true for SciEntBank dataset.

As our results show SciEntBank dataset was the most difficult dataset. In what follows, we discuss our observations on this dataset. Among all the classification tasks, 5way ASAG was the most challenging task. One possible reason is that we found it difficult to make a distinction between labels such as *correct* and *partially correct*, or *irrelevant* and *not in the domain.* For example, it is rather sophisticated to classify the following sample (extracted from SciEntBank [118]) as *partially correct* through concentrating on ”scratch”:

- *Question*: "Georgia found one brown mineral and one black mineral. How will she know which one is harder?"
- *Model Answer:* "The harder mineral will leave a scratch on the less hard mineral. If the black mineral is harder, the brown mineral will have a scratch."
- *Student Answer:* "The one with a scratch."

We also found that the characteristics of the test sets add to the difficulty of the classification. The SOTA and our evaluation results show that SciEntBank was more complicated than our other selected datastes as it included several domains in the train set and unseen domains in the test sets, which adds to the complexity of the dataset. The other datasets were of single domain and the test sets were based on unseen answers and unseen questions. Consequently, we conclude the challenge seems to increase when the domain of the test sets change.

To conduct further analysis of difficulty of SciEntBank dataset for our models, we provide two sets of examples of true positive, true negative, false positive and false negative for a model answer in 2way SciEntBank TUD test set through a case study. In both sets, we show what our classifiers consider most while confronted the test examples of unseen domain:

- *Question*: "Alice planted one radish seed in each of 5 separate pots (Pot one, Pot 2, Pot 3, Pot 4 and Pot 5). She used the same amount and the same kind of soil in each pot. She put them by the same window. The first graph shows the amount of water she put into each pot every day (Pot one got 5 milliliters, Pot 2 got 10 milliliters, Pot 3 got 15 milliliters, Pot 4 got 20 milliliters, and Pot 5 got 25 milliliters). The second graph shows the length of each root after 7 days (0 centimeters for Pot one, 4 centimeters for Pot 2, 8 centimeters for Pot 3, 5 centimeters for Pot 4, and 0 centimeters for Pot 5). What is the range of tolerance for water for these radish seeds? Explain how you decided the range of tolerance."

- *Model Answer:* "10 milliliters to 20 milliliters. The radishes did not grow at all in pots one and 5 so the range of tolerance is for pots 2, 3 and 4, which got 10 to 20 milliliters of water."
  - *Student Answer:* [true positive] "10 to 20 milliliters of water. Pot 2, 3, and 4 had been watered between 10 and 20 milliliters of water and they were the only ones that grew."
  - *Student Answer:* [false positive] "0 to 25 milliliters, 0 to 25 milliliters because the plants all grew from 0 to 25 milliliters of water."
  - *Student Answer:* [true negative] "15 milliliters of water, looking at the graphs."
  - *Student Answer:* [false negative] "10 to 20, I decided those numbers because those are the ones that actually had roots and were growing."

As the above-mentioned example shows, our models were sensitive towards wording of the model answers. For example, the classifier failed to classify "10 to 20, I decided those numbers because those are the ones that actually had roots and were growing." as there is a small word overlap similarity between this answer and the model answer when the domain changes. In this regard, "0 to 25 milliliters, 0 to 25 milliliters because the plants all grew from 0 to 25 milliliters of water." was incorrectly classified as positive though the meaning of it is not similar to the meaning of model answer.

- *Question*: "What is the main job of muscles in the body?"
- *Model Answer:* "The main job of muscles is to move bones."
  - *Student Answer:* [true positive] "To move bones."
  - *Student Answer:* [false positive] "The main job for the muscle is to pull the tendon."
  - *Student Answer:* [true negative] "To help you move."
  - *Student Answer:* [false negative] "To make the bones move that make us move."

We observed that our models were sensitive towards the exact words and phrases appeared in the model answers when the domain of test sets are different. In the above-mentioned example, it seems "The main job for the muscle is to pull the tendon." was predicted positive because "The main job for the muscle is" appeared in the model answer. The other observation is it seems when the model answer is worded differently as in the example "To make the bones move that make us move.", the classifier can hardly determine the class truly.

As a conclusion of our qualitative analysis in this section, we found that the number of teacher-provided reference answers, the accuracy of human labeling (grading the student answers in the train datasets), domain variety and word overlap similarity (especially when the domain changes) made classification difficult for our models.

5.5 Answers to our Research Questions

In this section, we answer our two research questions that we mentioned in chapter 1. Our first research question was *How do modern language models perform on the ASAG task?*. Through several experiments on various datasets, we showed that transformer-based models are robust in ASAG when compared to using baseline TFIDF-based bag of word models and other models in SOTA. Our results suggest that we achieved better or competitive performance with BERT Base (cased and uncased) and XLNET Base (cased) models. Overall, we showed that BERT and XLNET classifiers seem to be strong models for the ASAG task.

Our second research question was *How do NLP tasks and transfer learning impact the ASAG task?*. We conducted two-level fine-tuning in the third phase of our experimental analysis. We fine-tuned BERT Base uncased model firstly on GLUE tasks and then carried out the second fine-tuning on ASAG task. We showed that two-level fine-tuning with MNLI dataset yielded better results while we compared it with previous final models on SciEntBank dataset.

5.6 Conclusion

In this thesis, we compared the performance of BERT [12] and XLNET [13] with top-performing traditional machine learning systems on the same datasets. We included all the available classification datasets in our experiments: SemEval [1] (SciEntBank and BEETLE), Dt-grade [3] and Biology dataset [2]. We experimented 2way (correct and incorrect), 3way (correct, contradictory and incorrect), 4way (correct, correct-but-incomplete, contradictory and incorrect) and 5way (correct, incorrect, contradictory, partially-correct-incomplete, irrelevant and non-domain) classification. Additionally, we indicated how GLUE tasks namely

MNLI , STS-B, MRPC and RTE improve the pretraining phase of an ASAG model. For the evaluation of the models, we reported the performance in terms of accuracy, f1-score and f1-weighted measures. We showed that BERT and XLNET models outperform or equal the SOTA feature-based approaches. Also, we indicated that our BERT model performed much satisfactorily when we trained an ASAG model using GLUE entailment MNLI task.

5.7 Limitations

There are several limitations to our work. Our datasets were of different subjects and the objective of the questions differed. For example, the questions in Biology dataset mostly prompt the students to provide the answers only based on the question. This was not present in other datasets such as SemEval-2013 where some questions were about graphs and images. As a result, understanding questions and providing answers in natural language were of different types in different datasets and might lead to writing different natural language answers. In Dt-grade dataset, there was a problem description next to the questions too. This provided the students with more context to answer the questions.

We used the available test sets for SemEval [1] (SciEntBank and BEETLE) dataset in terms of different answers, different questions and different domains. However, we didn't consider question and sub-domain difference while dividing test sets for Dt-grade [3] and Biology dataset [2] due to the limitation in terms of the number of questions and our domain expertise. Addressing this limitation might enhance our conclusion about the generalizability of transformer-based models in ASAG.

Our experiments were carried out on only supervised classification ASAG. Therefore, we used only label-based datasets, namely SemEval [1] (SciEntBank and BEETLE), Dt-grade [3] and Biology dataset [2]. Regression task trained on datasets such as CSD [32] and XCSD [55] (both provided by University of North Texas) could also be used to confirm the desirable characteristics of BERT and XLNET for ASAG.

There are larger versions of BERT and XLNET models. However, we restricted the experiments to the base models. Using the largest models might be useful to enhance the generalizability strength of our models in our future experiments. In this thesis, we fixed a random seed to make our experiments reproducible. The alternative way is to repeat the experiments a certain number of times and calculate the standard deviation.

Finally, we only examined the impact of GLUE tasks on BERT Base uncased models trained on SciEntBank dataset.

5.8 Future Research

We suggest a few further points to be investigated as a future research. We presume that these practicable avenues might enhance the strength of our current models.

Training a transformer-based classifier without using hand-crafted features has already been proved to produce better results in SOTA (for example [12]) for NLP tasks such as inference. We also noticed the robustness of these types of classifiers in ASAG. Nonetheless, there is an avenue to merge transformer-based models with manually-engineered features such as cosine similarity in ASAG and this might provide effective results.

Using stacked classifiers (for example in [8]) proved to produce good results in ASAG. Therefore, stacking transformer-based classifiers with MLP, CNN or LSTM might yield promising results in ASAG task. Also, the effectiveness of these types of experiments might open research avenues for the extension of transformer-based architectures in other NLP tasks such as question answering, machine translation, etc.

Finally, transformer-based models might be rather difficult to explain. This is one of the most significant limitations in educational settings where teachers are facing challenges to fully understand the decision of the ASAG systems. As a result, an explanation mechanism together with transformer-based models is definitely an avenue worth exploring.

REFERENCES

[1] M. Dzikovska *et al.*, “SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge,” in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013).* Atlanta, Georgia, USA: Association for Computational Linguistics, Jun. 2013, pp. 263–274.

[2] J. McDonald *et al.*, “Short answers to deep questions: supporting teachers in large-class settings,” *Journal of Computer Assisted Learning*, vol. 33, no. 4, pp. 306–319, 2017.

[3] R. Banjade *et al.*, “Evaluation Dataset (DT-Grade) and Word Weighting Approach towards Constructed Short Answers Assessment in Tutorial Dialogue Context,” in *the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, vol. 11, 2016, pp. 182–187.

[4] L. Yuan and S. J. Powell, “Moocs and open education: Implications for higher education,” 2013. [Online]. Available: https://publications.cetis.org.uk/wp-content/uploads/2013/03/MOOCs-and-Open-Education.pdf

[5] S. Burrows, I. Gurevych, and B. Stein, “The eras and trends of automatic short answer grading,” *International Journal of Artificial Intelligence in Education*, vol. 25, no. 1, pp. 60–117, 2015.

[6] E. Badger and B. Thomas, “Open-ended questions in reading,” *Practical assessment, research, and evaluation*, vol. 3, no. 4, p. 03, 1992.

[7] B. Riordan *et al.*, “Investigating neural architectures for short answer scoring,” in *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 2017, pp. 159–168.

[8] J. G. Alvarado Mantecon, “Towards the automatic classification of student answers to open-ended questions,” mémoire de maîtrise, Université d’Ottawa/University of Ottawa, 75 Laurier Ave. E, Ottawa, ON, 2019. [Online]. Available: https://ruor.uottawa.ca/handle/10393/39093

[9] K. Sakaguchi, M. Heilman, and N. Madnani, “Effective feature integration for automated short answer scoring,” in *Proceedings of the 2015 conference of the North*

American Chapter of the association for computational linguistics: Human language technologies, 2015, pp. 1049–1054.

[10] I. Dagan, O. Glickman, and B. Magnini, "The pascal recognising textual entailment challenge," in *Machine Learning Challenges Workshop.* Springer, 2005, pp. 177–190.

[11] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2018.

[12] J. Devlin *et al.*, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[13] Z. Yang *et al.*, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[14] M. Scriven, "The methodology of evaluation," *In Stake, R. E. (ed.). Curriculum evaluation. Chicago: Rand McNally. American Educational Research Association (monograph series on evaluation, no. 1.*, pp. 39–83, 1967.

[15] T. Bloom, Benjamin S.and Hasting and G. Madaus, *Handbook of formative and summative evaluation of student learning.* New York, USA: McGraw-Hill., 1971.

[16] E. Reilly *et al.*, "Evaluating the validity and applicability of automated essay scoring in two massive open online courses," *The International Review of Research in Open and Distributed Learning*, vol. 15, no. 5, 2014.

[17] E. B. Page, "The imminence of grading essays by computer," *The Phi Delta Kappan*, vol. 47, no. 5, pp. 238–243, 1966.

[18] D. R. Krathwohl, "A revision of bloom's taxonomy: An overview." *Theory into practice*, vol. 41, no. 4, pp. 212–218, 2002.

[19] S. Burrows, I. Gurevych, and B. Stein, "The eras and trends of automatic short answer grading," *International Journal of Artificial Intelligence in Education*, vol. 25, pp. 60–117, 2015.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[21] J. G. Alvarado *et al.*, "A Comparison of Features for the Automatic Labeling of Student Answers to Open-ended Questions," in *EDM*, 2018, pp. 55–65.

[22] C. Leacock and M. Chodorow, "C-rater: Automated scoring of short-answer questions," *Computers and the Humanities*, vol. 37, no. 4, pp. 389–405, 2003.

[23] C. Siddiqi, R., & Harrison, “A systematic approach to the automated marking of short-answer questions,” in *Proceedings of the 12th international multitopic conference. Karachi: IEEE.*, 2008, p. 329–332.

[24] U. Hasanah *et al.*, “A Review of an Information Extraction Technique Approach for Automatic Short Answer Grading,” 2016, pp. 192–196.

[25] T. Mitchell *et al.*, “Towards robust computerised marking of free-text responses,” in *6th CAA Conference*, 2002.

[26] L. F. Bachman *et al.*, “A reliable approach to automatic assessment of short answer free responses,” in *COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes*, 2002, pp. 1–4.

[27] P. Thomas, “The evaluation of electronic marking of examinations,” in *ITiCSE 03: Proceedings of the 8th annual conference on Innovation and technology in computer science education*, 2003, pp. 50–54.

[28] J. Z. Sukkarieh, S. G. Pulman, and N. Raikes, “The evaluation of electronic marking of examinations,” in *the 29th Annual Conference of the International Association for Educational Assessment*, 2003, pp. 1–15.

[29] S. Jordan and T. Mitchell, “e-Assessment for learning? The potential of short-answer free-text questions with tailored feedback,” *British Journal of Educational Technology*, vol. 40, no. 2, pp. 371–385, 2009.

[30] L. Cutrone and M. Chang, “Auto-assessor: computerized assessment system for marking student’s short-answers automatically,” in *2011 IEEE International Conference on Technology for Education*, 2011, pp. 81–88.

[31] M. Hahn and D. Meurers, “Evaluating the meaning of answers to reading comprehension questions: A semantics-based approach,” in *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, 2012, pp. 326–336.

[32] M. Mohler and R. Mihalcea, “Text-to-text semantic similarity for automatic short answer grading,” in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on - EACL ’09*, no. April, 2009, pp. 567–575.

[33] E. Alfonseca, D. Pérez, and P. Diana, “Automatic Assessment of Open Ended Questions with a BLEU -Inspired Algorithm and Shallow NLP,” in *Advances in Natural Language Processing*, 2004, pp. 25–35.

[34] J. Nielsen, R.D., Ward, W., Martin, “Learning to assess low-level conceptual understanding,” in *21st International Florida Artificial Intelligence Research Society Conference*, 2008, pp. 427–432.

[35] V. Gonzalez-Barbone and M. Llamas-Nistal, “eassessment of open questions: An educator’s perspective,” in *38th Annual Frontiers in Education Conference*, 2008, pp. F2B–1.

[36] Y. N. Shourya Roy and O. D. Deshmukh, “A Perspective on Computer Assisted Assessment Techniques for Short Free-text Answers,” in *International Computer Assisted Assessment*, vol. 571, no. July, 2015, pp. 96–109.

[37] L. B. Galhardi and J. Brancher, “Machine learning approach for automatic short answer grading : A for automatic short answer grading,” in *16th Ibero-American Conference on AI, Trujillo, Peru*, no. November, 2018, pp. 380–391.

[38] C. Salazar, “Fast and Easy Short Answer Grading with High Accuracy,” in *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1070–1075.

[39] O. Levy *et al.*, “UKP-BIU: Similarity and Entailment Metrics for Student Response Analysis,” *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, vol. 2, no. SemEval, pp. 285–289, 2013.

[40] M. Heilman and N. Madnani, “ETS: Domain Adaptation and Stacking for Short Answer Scoring *,” in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, vol. 2, no. SemEval, 2013, pp. 275–279.

[41] N. Ott *et al.*, “CoMeT: Integrating different levels of linguistic modeling for meaning assessment,” in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, vol. 2, no. SemEval, 2013, pp. 608–616.

[42] V. Rus, “Explanation-based Automated Assessment of Open Ended Learner Responses,” in *the 14 th International Scientific Conference eLearning and Software for Education Bucharest,*, 2018, pp. 120–128.

[43] R. D. Nielsen *et al.*, “Annotating students’ understanding of science concepts,” in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*, 2008, pp. 3441–3448.

[44] S. Saha *et al.*, “Sentence level or token level features for automatic short answer grading? use both,” in *International Conference on Artificial Intelligence in Education.* Springer, 2018, pp. 503–517.

[45] G. A. Miller, “WordNet : A Lexical Database for English,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[46] C. Bizer and R. Cyganiak, “A Nucleus for a Web of Open Data,” in *The Semantic Web (ASW)*, 2007, pp. 722–735.

[47] P. Ferragina *et al.*, “TAGME : On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities),” in *CIKM ’10 Proceedings of the 19th ACM international conference on Information and knowledge management*, 2010, pp. 1625–1628.

[48] P. N. Mendes *et al.*, “DBpedia Spotlight : Shedding Light on the Web of Documents,” in *I-Semantics ’11 Proceedings of the 7th International Conference on Semantic Systems*, pp. 1–8.

[49] U. Pad, “Get Semantic With Me ! The Usefulness of Different Feature Types for Short-Answer Grading,” in *COLING 2016, the 26th International Conference on Computational Linguistics*, 2016, pp. 2186–2195.

[50] F. Richter and M. Sailer, “Basic concepts of lexical resource semantics.” in *Proceedings of the 15th european summer school in logic language and information volume 5 of collegium logicum*, 2015, pp. 87–143.

[51] C. F. Baker, C. J. Fillmore, and J. B. Lowe, “The Berkeley FrameNet Project,” in *the 17th international conference on Computational linguistics-Volume 1*, 1997, pp. 86–90.

[52] K. K. Schuler, “VerbNet: A Broad-coverage, Comprehensive Verb Lexicon,” Ph.D. dissertation, University of Pennsylvania, 2005.

[53] P. Kingsbury and M. Palmer, “From TreeBank to PropBank,” in *Language Resources and Evaluation Conference (LREC)*, 1993, pp. 1989–1993.

[54] I. Aldabe, M. Maritxalar, and O. Lopez De Lacalle, “EHU-ALM: Similarity-Feature Based Approach for Student Response Analysis,” in *Second Joint Conference on Lexical*

*and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, vol. 2, no. SemEval, 2013, pp. 580–584.

[55] M. Mohler, R. Bunescu, and R. Mihalcea, “Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments,” in *the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 752–762.

[56] S. Dumais, “Latent Semantic Analysis,” *Annual Review of Information Science and Technology*, vol. 38, pp. 188–230.

[57] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet Allocation,” in *Machine Learning Research*, vol. 3, 2003, pp. 993–1022.

[58] T. Mikolov *et al.*, “Distributed Representations of Words and Phrases and their Compositionality,” in *Advances in neural information processing systems*, 2013, pp. 1–9.

[59] J. Pennington, R. Socher, and C. D. Manning, “GloVe : Global Vectors for Word Representation,” in *Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[60] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[61] H. A. Ghavidel, A. Zouaq, and M. Desmarais, “Using BERT and XLNET for the Automatic Short Answer Grading Task,” in *Proceedings of the 12th International Conference on Computer Supported Education (CSEDU)*, 2020, pp. 58–67.

[62] R. Mihalcea, C. Corley, and C. Strapparava, “Corpus-based and Knowledge-based Measures of Text Semantic Similarity,” in *the American Association for Artificial Intelligence (AAAI 2006)*, Boston, 2006, p. 775–780.

[63] W. Gomaa and A. Fahmy, “Short Answer Grading Using String Similarity And Corpus-Based Similarity,” *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 11, pp. 115–121, 2012.

[64] P. James .L., “Computer programs for detecting and correcting spelling errors,” vol. 23(12), pp. 676–687, 1980.

[65] M. A. Jaro, “Advances in record linkage methodology as applied to the 1985 census of tampa florida,” *American Statistical Association*, vol. 84(406), pp. 414–420, 1989.

[66] ——, “Probabilistic linkage of large public health data file,” *Statistics in Medicine*, vol. 14, pp. 491–498, 1995.

[67] W. E. Winkler, “String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage,” in *the Section on Survey Research Methods(American Statistical Association)*, 1990, pp. 354–359.

[68] S. Needleman and C. Wunsch., “A General Method Applicable to the Search for Similarities in the Amino Acid Sequences of Two Proteins,” *Molecular Biology: A Selection of Papers*, vol. 48, pp. 443–453, 2012.

[69] T. F. Smith and M. S. Waterman., “Identification of Common Molecular Subsequences,” *Molecular Biology*, vol. 147, pp. 195–197, 1981.

[70] P. R. Barrón-Cedeno Alberto, E. Agirre, and G. Labaka, “Plagiarism detection across distant language pairs,” in *23rd International Conference on Computational Linguistics (Coling 2010)*, 2010, pp. 37–45.

[71] S. Jimenez, F. Gonzalez, and A. Gelbukh, “Text comparison using soft cardinality,” in *International symposium on string processing and information retrieval.* Springer, 2010, pp. 297–302.

[72] S. Jimenez, C. Becerra, and A. Gelbukh, “Softcardinality: Hierarchical text overlap for student response analysis,” in *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, vol. 2, 2013, pp. 280–284.

[73] C. Leacock and M. Chodorow, “Combining local context and WordNet sense similarity for word sense identification,” *MIT Press*, vol. 49(2), pp. 265–283, 1998.

[74] M. Lesk, “Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone.” in *SIGDOC 86: Proceedings of the 5th annual international conference on Systems documentation*, 1986, pp. 24–26.

[75] Z. Wu and M. Palmer, “Verbs semantics and lexical selection,” in *The 32nd annual meeting on Association for Computational Linguistics*, 1994, pp. 133–138.

[76] P. Resnik, “Using information content to evaluate semantic similarity,” in *14th International Joint Conference on Artificial Intelligence.*, 1995, p. 448–453.

[77] C. Lin, “An information-theoretic definition of similarity,” in *International Conf. on Machine Learning*, 1998, pp. 296–304.

[78] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *The international Conference on Research in Computational Linguistics.*, 1997, pp. 19–33.

[79] Hirst and D. St-Onge, "Lexical chains as representations of contexts for the detection and correction of malaproprisms," *MIT Press*, pp. 305–332, 1998.

[80] S. Archana and Kumar Plaban Bhowmick, "Feature Engineering and Ensemble-based Approach for Improving Automatic Short-answer Grading Performance," *IEEE Transactions on Learning Technologies*, pp. 77–90, 2015.

[81] M. A. Sultan, C. Salazar, and T. Sumner, "Fast and easy short answer grading with high accuracy," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1070–1075.

[82] M. Baroni and G. Dinu, "Don ' t count , predict ! A systematic comparison of context-counting vs . context-predicting semantic vectors," in *52nd Annual Meeting of the Association for Computational Linguistics*, 2014, pp. 238–247.

[83] S. J. Miller, "The method of least squares." pp. 1–7, 2006.

[84] "The automated student assessment prize: Short answer scoring," The Hewlett Foundation, 2012.

[85] N. Maharjan, R. Banjade, and V. Rus, "Automated assessment of open-ended student answers in tutorial dialogues using Gaussian Mixture Models," in *FLAIRS 2017 - Proceedings of the 30th International Florida Artificial Intelligence Research Society Conference*, 2017, pp. 98–103.

[86] S. Kumar and S. Roy, "Earth Mover ' s Distance Pooling over Siamese LSTMs for Automatic Short Answer Grading," in *International Joint Conference on Artificial Intelligence*, 2015, pp. 2046–2052.

[87] S. Roy, A. Rajkumar, and Y. Narahari, "Selection of automatic short answer grading techniques using contextual bandits for different evaluation measures," *International Journal of Advances in Engineering Sciences and Applied Mathematics*, vol. 10, no. 1, pp. 105–113, Mar 2018.

[88] M. Heilman and N. Madnani, "Ets: Domain adaptation and stacking for short answer scoring," in *Second Joint Conference on Lexical and Computational Semantics*

(SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, vol. 2, 2013, pp. 275–279.

[89] H. D. III, "Frustratingly easy domain adaptation," *arXiv preprint arXiv:0907.1815*, 2009.

[90] N. Ott *et al.*, "Comet: Integrating different levels of linguistic modeling for meaning assessment," in *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, vol. 2, 2013, pp. 608–616.

[91] L. Ramachandran, J. Cheng, and P. Foltz, "Identifying Patterns For Short Answer Scoring Using Graph-based Lexico-Semantic Text Matching," in *the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2015, pp. 97–106.

[92] D. R. Radev, M. S. Hongyan Jing, and D. Tam., "Centroid-based summarization of multiple documents," *Information Processing Management*, vol. 40, no. 6, pp. 919–938, 2004.

[93] A. Conneau *et al.*, "Supervised learning of universal sentence representations from natural language inference data," *arXiv preprint arXiv:1705.02364*, 2017.

[94] C. Sung, T. I. Dhamecha, and N. Mukhi, "Improving short answer grading using transformer-based pre-training," in *International Conference on Artificial Intelligence in Education*. Springer, 2019, pp. 469–481.

[95] Y. Goldberg and G. Hirst, *Neural Network Methods in Natural Language Processing*, 2017.

[96] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer." *IEEE transactions on acoustics, speech, and signal processing*, vol. 35, no. 4, pp. 400–401, 1987.

[97] Y. Bengio *et al.*, "A Neural Probabilistic Language Model," *Machine Learning Research*, vol. 3, pp. 1137–1157, 2003.

[98] T. Mikolov *et al.*, "Recurrent neural network based language model," in *INTERSPEECH*, 2010, pp. 1045–1048.

[99] M. E. Peters *et al.*, "Deep contextualized word representations," *arXiv preprint 1802.05365*, 2018.

[100] Q. Yang *et al.*, “Grounding interactive machine learning tool design in how non-experts actually build models,” *DIS 2018 - Proceedings of the 2018 Designing Interactive Systems Conference*, pp. 573–584, 2018.

[101] . Y. Q. Pan, S. J., “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 20, no. 10, pp. 1345–1259, 2009.

[102] J. G. A. Mantecon *et al.*, “A comparison of features for the automatic labeling of student answers to open-ended questions,” in *Proceedings of 11th International Educational Data Mining (EDM) Conference*, 2018, pp. 55–65.

[103] Y. Wu *et al.*, “Google’s neural machine translation system: Bridging the gap between human and machine translation,” *arXiv preprint arXiv:1609.08144.*, 2016.

[104] W. L. Taylor, ““cloze procedure”: A new tool for measuring readability,” *Journalism Bulletin*, vol. 30, no. 4, pp. 415–433, 1953.

[105] Z. Yang *et al.*, “XLNet : Generalized Autoregressive Pretraining for Language Understanding,” no. NeurIPS, 2019, pp. 1–18.

[106] Z. Dai *et al.*, “Transformer-xl: Attentive language models beyond a fixed-length context,” *arXiv preprint arXiv:1901.02860*, 2019.

[107] A. Wang *et al.*, “Glue: A multi-task benchmark and analysis platform for natural language understanding,” *arXiv preprint arXiv:1804.07461*, 2018.

[108] A. Williams, N. Nangia, and S. R. Bowman., “A broad-coverage challenge corpus for sentence understanding through inference,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, p. 1112–1122.

[109] R. B. Haim *et al.*, “The second pascal recognising textual entailment challenge,” in *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005, pp. 1–9.

[110] D. Giampiccolo *et al.*, “The third pascal recognizing textual entailment challenge,” in *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing.* Association for Computational Linguistics, 2007, pp. 1–9.

[111] L. Bentivogli *et al.*, “The fifth pascal recognizing textual entailment challenge,” in *Textual Analysis Conference (TAC)*, 2009, pp. 1–9.

[112] P. Rajpurkar *et al.*, “Squad: 100,000+ questions for machine comprehension of text,” *arXiv preprint arXiv:1606.05250*, 2016.

[113] W. B. Dolan and C. Brockett, “Automatically constructing a corpus of sentential paraphrases,” in *Proceedings of the Third International Workshop on Paraphrasing (IWP2005).* Asia Federation of Natural Language Processing, 2005, pp. 9–16.

[114] D. Cer *et al.*, “Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation.” pp. 1–14, 2018.

[115] L. Ramachandran and P. Foltz, “Generating reference texts for short answer scoring using graph-based summarization,” pp. 207–212, 2015.

[116] S. Marvaniya *et al.*, “Creating scoring rubric from representative student answers for improved short answer grading,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* ACM, 2018, pp. 993–1002.

[117] P. Bojanowski *et al.*, “Enriching word vectors with subword information,” *Transactions of the Association for Computational Linguistics*, p. 135–146, 2017.

[118] R. D. Nielsen *et al.*, “Annotating students’ understanding of science concepts.” in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*, 2008, pp. 3441–3448.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346 USA