Universidad del País Vasco Euskal Herriko Unibertsitatea

Exploration of aunnotation strategies for entailment-based Automatic Short Answer Grading

Author: Aner Egaña Azpiazu

Advisors: Oier Lopez de Lacalle & Itziar Aldabe

hap/lap

Hizkuntzaren Azterketa eta Prozesamendua
Language Analysis and Processing

Final Thesis

September 2022

Departments: Computer Systems and Languages, Computational Architectures and Technologies, Computational Science and Artificial Intelligence, Basque Language and Communication, Communications Engineer.

Laburpena

Erantzun labur automatikoen sailkapenaren inguruan azken urteetan egindako ikerketek atazaren birformulazio eraginkorra eraikitzea posible dela erakutsi dute, inferentzia testualaren atazarako birformulazioa, bereziki. Gure lan honetan, birformulazioaren eraginkortasuna erakusten da adibide gutxitako eszenarioetan (few-shot) eta adibide gabeko eszenarioetan (zero-shot) ere bai. Are eta garrantzitsuago, atazarako adibideak anotatzeko estrategiak modeloaren erredimenduan eragin nabarmena duela erakusten da. Adibide gutxi batzuk idaztean, emaitza enpirikoek erakusten dute hobe dela galderaren aldeko aldagarritasuna handitzea, galdera bakoitzeko idatzitako erantzun-kopurua murriztearen kostuari dagokionez, galdera gutxiagorekin eta erantzun gehiagorekin idatzitako adibide-kopuru bera izatea baino. Idazteko estrategia honi jarraituz, entrenamendu osoko datu-basearen %10a erabiliz artearen egoerako sistemen errendimenduaren parekoa da, SciEntsBank domeinuko datu-basean. Azkenik, Beetle eta SciEntsBank domeinuen gainean aurrera eramandako esperimentuek domeinuz kanpoko galdera-erantzun adibide bikoteek errendimendurako mingarriak izan daitezkeela erakutsi dute, beste domeinu batetik ataza ezagutzen duten sistemek ataza ezagutzen ez dutenak baino emaitza apalagoak emateko joera dutela ondorioztatuz, aztertutako domeinuetan behintzat.

Abstract

Recent work has shown that Automatic Short Answer Grading can effectively be reformulated as a Textual Entailment problem. In this work we show that this reformulation is also effective in zero-shot and few-shot settings, where we report competent results close to state-of-the-art performance with the few-shot setting. More importantly, we show that the annotation strategy can have significant impact on performance. When annotating few examples, empirical results show that increasing the variability on the question side, at cost of decreasing the amount of annotated answers per question, is preferable than having the same number of annotated examples with less questions and more answers. With this annotation strategy, using only the 10% of the full training set our model levels with state-of-the-art systems in the SciEntsBank dataset. Finally, experiments over SciEntsBank and Beetle domains show that the use of out-of-domain annotated question-answer examples can be harmful, concluding that task-aware fine-tuned models obtain significantly lower results compared to task-agnostic general purpose inference models, at least with the domains employed for this work.

Keywords: Automatic Short Answer Grading - Fine-tuning - Transfer learning - Task reformulation - Zero-shot - Few-shot - Cross-domain learning

Contents

1 Introduction **1**

2 Literature Review **4**
2.1 State of the Art 4
2.1.1 Classical Machine Learning (ML) 5
2.1.2 Deep Learning (DL) 8
2.1.3 Textual Entailment as pivoting task 16
2.2 Benchmark Datasets 17
2.3 Evaluation Metrics 19

3 SemEval-2013 SRA Dataset **22**

4 Entailment-based Answer Grading **26**
4.1 Model Description 26
4.2 Fine-tuning of ASAG Model 27

5 Experimental Design **28**
5.1 Annotation Strategies 28
5.2 Experimental Setting 29

6 Experimental Results **32**
6.1 Few-shot Results 32
6.2 Cross-domain Results 33
6.3 Comparison to the State of the Art 35

7 Conclusions and Future Work **38**

List of Figures

1 Taxonomy of Automatic Short Answer Grading methods. Source: https://www.researchgate.net/figure/Taxonomy-of-Automated-Short-Answer-Grading-methods-The-categorization-of-methods-is_fig1_359813357 5
2 Linear relation among the vector representations of the words king, man, woman and queen. Source: https://kawine.github.io/blog/nlp/2019/06/21/word-analogies.html 9
3 Basic RNN architecture. Source: https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg 11
4 Comparison among the three main types of RNNs: simple RNN, LSTM and GRU. Source: http://dprogrammer.org/rnn-lstm-gru 11
5 Schema of the NLI based ASAG model where the input of question, reference answer and student answer are reformulated as an entailment model. Concatenation of the question and student answer form the *premise* of the NLI model, whereas the *hypothesis* is generated with the reference answer. Prediction of the entailment model is then mapped to ASAG 3-way label . 27

List of Tables

1 Main statistics of the SemEval-2013 SRA dataset with regard to the SciEntsBank and Beetle subsets. . . . 23
2 A question, reference answer and correct student answer triple example extracted from SciEntsBank and Beetle . . . 24
3 Available reference answers to a certain question with different degrees of correctness. Extracted from the SemEval-2013 SRA dataset, belongs to the Beetle subset. . . . 25
4 Number of questions, reference answers and student answers per each train and test split for SciEntsBank and Beetle subsets. . . . 25
5 Number of questions (#Q) and student answers (#A) for each few-shot scenario according to the specific annotation strategy (Ann.) as well as the number of training examples (Total) for each of the few-shot setting and dataset (DS). SB stands for SciEntsBank, and BT for Beetle. . . . 30
6 Number of validation examples for unseen answers and unseen question test scenarios. . . . 31
7 Results for the few-shot experiments in which we fine-tune an entailment model (NLI-roberta) using %1, %2, %5 and %10 of training data and evaluated in unseen answers (UA), unseen questions (UQ) and unseen domains (UD). Q2S annotation correspond to training data where we annotate one question per student, and Q2A correspond to one question to all student annotation procedure. . . . 32
8 Average weighted macro f1-score results of annotation procedures in few-shot experiments. . . . 33
9 Macro f1-score results of zero-shot cross-domain evaluation. Task-aware column shows the results for entailment model fine-tuned in one domain and evaluated in another domain. BT stands for Beetle dataset and SB for SciEntsBank. . . . 34
10 Results of cross-domain few-shot evaluation. . . . 34
11 Comparison to state-of-the-art f1-macro score results. Underlined figures denote that current results outperform previous state-of-the-art models. * for results not directly comparable with ours. Bold for best among comparable results. . . . 36

1 Introduction

In the educational field, large attention is given to the learning process in order to measure the success and efficiency of new learners gaining knowledge. These measurements are often carried out by assessing and quantifying the knowledge gained by means of tests and examinations. As a consequence of this time-consuming process, building efficient rendering systems such as grading automators is crucial nowadays. In the context of automatic educational assessment, the evaluation of short answers authored by students is referred to as Automatic Short Answer Grading (ASAG), which nowadays keeps being a challenging task despite all the research and enhancements made in the field of Natural Language Processing (NLP). The task of ASAG requires both deep textual understanding and a detailed analysis. These automatic graders need to deal with a huge variability of student answers as well as different question formats in terms of text length, question or answer type, etc., and usually grade them against a reference answer. In this way, the current datasets used to test ASAG systems' performance consist of a question, the student's answer to the given question and the reference answer, the latter considered as a good or best answer to the question given.

One of the main objectives of current NLP researchers is to build highly performing systems in real-world conditions, conditions that most of the time are limited due to the lack of annotated data, considered as one of the main drawbacks when developing real-world NLP models. In particular, annotating data for the task of ASAG is quite demanding and time-consuming, making the data for ASAG really sparse in comparison with the datasets for other NLP downstream tasks. Besides, benchmark datasets for ASAG are often complex to compare as they belong to different domains, are built following different evaluation criteria, and, as a consequence, the task is nowhere near having a single benchmark dataset that serves as a mean of comparison of all the approaches the task has had through the years. This causes an unability to take advantage of external datasets to enhance perfomance over another dataset. To counteract this problem, recent NLP investigations propose the concept of **task reformulation** (Du and Cardie, 2020; Sainz et al., 2021; Levy et al., 2017; Schick and Schütze, 2020). Basically, task reformulation relies on transforming a certain NLP task into, for instance, well-known Natural Language Inference or Question Answering task. The fact that considerable knowledge has been gained over these conventional tasks make them suitable to transfer knowledge across tasks and improve results, especially low-resource tasks can be highly benefited. With respect to ASAG, it can be reformulated as an Textual Entailment task (Dzikovska et al., 2013), and it has been demonstrated as an effective method to obtain great results (Camus and Filighera, 2020).

In addition to the task reformulation concept, one of the main trends of current NLP research is fine-tuning unsupervisedly pretrained Language Models (LM) with a little set of labeled (relative to the huge amount of unlabeled data used for building the LMs) for the target task, also known as **transfer learning** pretrained LMs across different NLP tasks. Nevertheless, due to the demanding conditions for annotating ASAG data such as topic of

questions, grading scale and the cost of human annotation, there is typically a very small number of labeled examples in real-world ASAG scenarios, and models built by fine-tuning a pretrained LM over this datasets perform poorly (see Table 11). As a result, the reality is otherwise concerning the likes of ASAG. Influenced by the aforementioned annotation sparsity, this thesis carries out the investigation of building competent ASAG systems in low-resource scenarios: in particular, both zero-shot and few-shot scenarios. These two scenarios refer to situations where annotated examples for the task are nonexistent and situations where annotated examples are scarce, respectively, and thanks to the possibility of task reformulation and transfer learning, models that deal with these scenarios have emerged. Still, the strategy to choose for the labeling of examples is an open question in ASAG, since its examples do not consist of conventional Textual Entailment pairs[1] and distinct annotation strategies can be taken into account.

Therefore, in this thesis the focus is put on using entailment models to explore zero and few-shot learning in student automatic short answer grading. We define different scenarios where it is assumed there is no sufficient training examples for fine-tuning the model, and pose the following research questions in order to devise better strategies of data annotation:

- **Research Question 1 (RQ1):** Having a task-agnostic[2] generic entailment model, what would be the best way to annotate data, and how much data would be needed to obtain state-of-the-art results?
- **Research Question 2 (RQ2):** Having a task-agnostic generic entailment model, how much of the model can be transferred in the case we only have out-of-domain examples to fine-tune? That is, is it better to fine-tune in a related task but in a different domain or, on the contrary, is it better to just apply a zero-shot model on the new dataset? With this question, it is of interest to analyse the performance of a task-aware but out-of-domain system against a task-agnostic system.
- **Research Question 3 (RQ3):** Having a task-aware entailment model (fine-tuned in ASAG) trained in an out-of-domain dataset and having a few in-domain annotated examples, how much in-domain data would be required to obtain state-of-the-art results?

With the goal of answering this three research questions supported by empirical results, this thesis describes the process where we conduct experiments over the SemEval-2013 Student Response Analysis (SRA) dataset (Dzikovska et al., 2013), and the following contributions are made to the field of ASAG and NLP in general:

- Incorporating question information in entailment-based short answer grading. Previous work made on the fine-tuning of pretrained LMs with ASAG examples only

[1] Usually, Textual Entailment examples are (PREMISE, HYPOTHESIS) pairs whereas in ASAG examples consist of (QUESTION, REFERENCE ANSWER, STUDENT ANSWER) triplets.

[2] A system which does not recognize the given task, ASAG in this case.

considered to use student and reference answer, setting aside the information that the initial question offers (Sung et al., 2019b; Camus and Filighera, 2020).

- Showing that the annotation strategy can have a significant impact in the system's final performance, as the annotation that increases the variability on the question side, at cost of decreasing the amount of annotated answers per question, is preferable than having the same number of annotated examples with less questions and more answers.
- Reformulating ASAG as an entailment problem and fine-tuning a pretrained entailment model allows to obtain state-of-the-art results.
- Related to the previous contribution, we show that zero-shot entailment models can perform close to state-of-the-art results.
- Finally, we show that the impact of the domain can be larger than the knowledge that can be acquired from the task. That is, using a generic entailment model is more effective than fine-tuning it with out-of-domain examples, at least when working with the two subsets of the SemEval-2013 SRA dataset, SciEntsBank and Beetle.

The written thesis is structured as follows: in Section 2 we take a deep look to the recent past of ASAG research, referring to the state of the art, available benchmark datasets and employed evaluation metrics for ASAG. In Section 3, the benchmark dataset used for the research is described, mentioning the way it was built and taking a look into the main characteristics statistic-wise to the two subsets of the dataset. Section 4 focuses on the creation of our entailment-based ASAG model and the main aspects of the eventual model in terms of reformulating ASAG as Textual Entailment and the fine-tuning of the base pretrained inference system. Afterwards, Section 5 describes the practical part of the experiment where the focus is given to the actions taken in order to try to solve our research questions, explaining the different annotation strategies and the experimental setting (i.e. dataset, creation of the validation set, fine-tuning setting, etc.). Empirical results are displayed in Section 6, where based on the outcome of the experiments it is tried to give an answer to the research questions posed, concerning zero-shot, few-shot, cross-domain and state of the art comparison. The conclusion over the work done is written in Section 7, comparing the initial thoughts had with the eventual reality, accompanied as well by a reflection about which direction could this research take in the future and which challenges remain uncontested.

2 Literature Review

The development of Automatic Short Answer Grading dates back to 1996 when the first thoughts of automatically evaluating student answers to a given question against an optimal reference answer were introduced. Since then, the task of ASAG has gone through various milestones and has been experimented with several methodologies. The historical research presented in Burrows et al. (2015) mentions the following eras during the evolution of ASAG: **concept mapping**, a technique where student and reference answers are decomposed into concepts and the sentence-level comparison between answers is done considering the absence or presence of concepts in both answers; **information extraction**, another technique concerned with fact finding in student answers, the idea is to extract structured data from unstructured source, generally modeled by regular expression or constituency parsers; **corpus-based** techniques, which try to exploit the statistical property of large document corpora with metrics such as BLEU or Latent Semantic Analysis (LSA); lastly, the era of **Machine Learning** (ML), which typically uses some NLP measurements and techniques for extracting features to the input sources, and the information of those features is fed to classification or regression models.

In addition to these eras, the latest review of Haller et al. (2022) gives a more specific view to the most recent methodologies for obtaining the most prominent results in ASAG. According to them, the conventional Machine Learning models consisted in modeling **hand-engineered** features representing the lexical, syntactic and semantic information, extracted from the input texts, using conventional ML algorithms, whereas the continuous evolution of the NLP technologies brought the process of learning new features from the input text by means of the **Deep Learning**. An example of this features are the so-called **word embeddings**. Deep Learning also favoured the development of **sequence-based** models, capable of learning dependencies in sequence of words as well as of the latest **attention-based** models, widely known as Transformers, able to compute long-range text dependencies and its discovery resulted in a real breakthrough for the world of NLP.

2.1 State of the Art

The progress made on the field of the ASAG task has caused a constant enhancement in the performance of built systems for automatic student assessment. The most recent eras of Machine Learning and Deep Learning are worth standing out over the mentioned rest when referring to state-of-the-art ASAG models. Figure 1 displays the categorization of the recent past with respect to ASAG methods proposed in Haller et al. (2022).

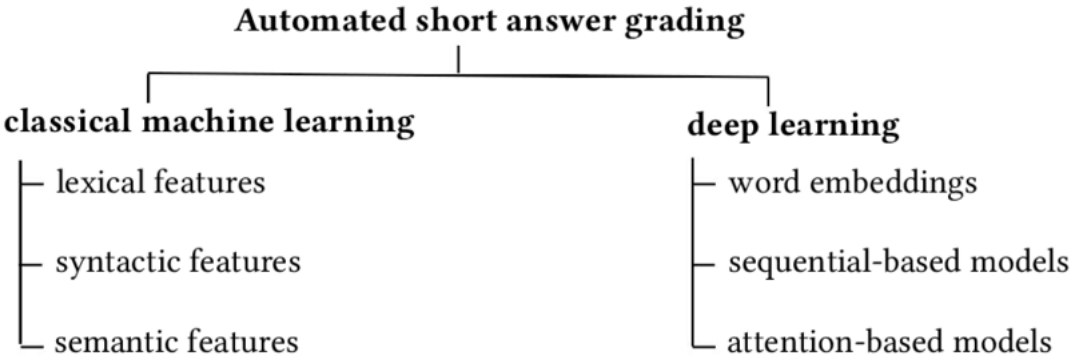

Figure 1: Taxonomy of Automatic Short Answer Grading methods. Source: https://www.researchgate.net/figure/Taxonomy-of-Automated-Short-Answer-Grading-methods-The-categorization-of-methods-is_fig1_359813357

2.1.1 Classical Machine Learning (ML)

These conventional approaches involve the processing of raw text into features that are able to detect different patterns in the input sources by, for instance, extracting lexical, syntactic and semantic characteristics. The main goal of these features is to describe key components (e.g. specific terms, concepts) of good answers by detecting specific patterns. Each feature type represents a certain observation of the unstructured text:

- **Lexical features:** Represent the textual characteristics of the input (e.g. number of single words, lemmatized or stemmed words). Besides, the automatization of algorithms to compute the degree of overlap between answers and n-gram representation has made quite an impact in ASAG, providing strong results over the benchmark datasets.

- **Syntactic features:** They give an insight of the processed text's meaning, analysing its structure via dependency parsers or Part of Speech (POS) tagging. Extracting features from the degree of overlap of n-grams that consist in concrete POS combinations between answers showed to be useful when comparing the meaning of different answers.

- **Semantic features:** Capable of representing the meaning in a more robust way than syntactic features, extracted with the crucial help of knowledge bases (e.g. WordNet (Miller et al., 1990)) and semantic vector spaces such as Latent Semantic Analysis (LSA) and Explicit Semantic Analysis (ESA) (Gabrilovich et al., 2007).

The features obtained are represented by a feature vector. These vectors are then employed as essential input information so as to model them into classification or regression systems. The modeling can be done by means of diverse Machine Learning algorithms, which depending on the kind of task and type of data some can result more suitable than

others. Examples of these are classification algorithms such as Random Forest, Support Vector Machine (SVM), Naive Bayes, and regression algorithms like Linear, Polynomial and Lasso Regression. Regarding the best performing systems of this era, most of them have proved their performance on the benchmark dataset that was made available in the SemEval-2013 SRA challenge.

SOFTCARDINALITY (Jimenez et al., 2013) The system was one of the best performing: 1st in the 2-way granularity task and 2nd in the other two tasks, 3-way and 5-way granularity, overall. They proposed a methodology to extract lexical features from the given ASAG datasets, relying mostly on character and word overlap. Similarity between pair of words/sentences was computed by Dice's coefficient. The classification models for the three tasks were learned using a J48 graft tree, and boosted with 15 bagging iterations.

COMET (Ott et al., 2013) Another system that also stood out is the meta-classifier COMET, a result of the combination of three subsystems: COMIC, COSEC and bag approaches. They worked in the concept that systems with different targets are complementary to each other and can be combined to build an all-around model. COMIC subsystem focuses on the alignment of linguistic units found on the learner's answer to those extracted from the reference answer. The features are computed via the Unstructured Information Management Architecture (UIMA) and range from very basic linguistic units such as sentences and tokens with POS and lemmas up to full dependency parses of the input. COSEC subsystem gives more focus to semantics, as it performs meaning comparison on the basis of an unspecified semantic representation robustly derived from the learner and the reference answers. Based on Lexical Resource Semantics (LRS) its representations can be directly derived following a two-step approach consisting POS tagging and dependency parse trees. The LRS representation of student and reference answers and also questions are aligned by COSEC and the overall semantic similarity scores are obtained by the combination of the computed using knowledge bases like WordNet or metrics as Minimum Edit Distance, among others. The third and last subsystem consists in bag approaches, influenced by the bag-of-words concept. Student answers' information is stored in three bags: words, lemmas and Soundex hashes[3]. Each answers' bag representation is then classified against a model trained with all the other known answers, a support-based machine learner (SVM). Finally, the scores or label probabilities calculated by each subsystem are used to feed the COMET meta-classifier.

UKP-BIU (Levy et al., 2013) The system follows the conventional way of the other systems explained. Their system is based on the combination of different feature categories extracted from the learner and reference answers. They make use of bag-of-words features (BOW) in order to identify words that tend to appear in correct answers, also extended with the top 10 basic and dependent n-grams, the latter representing syntactic characteristics, which are binary features. Basic similarity features, Semantic Similarity as well as spelling

[3]Encodes english words according to how they sound. Similar sounding words will have similar codes.

features are added to the combination. The second type is based on ESA measures, in order to counteract the possible vocabulary gap between answers. Last but not least, Textual Entailment features are computed using BIUTEE entailment recognition (Stern and Dagan, 2012). The classifier is trained with different data, depending on the test scenario, with Naive Bayes method via Weka software.

ETS (Heilman and Madnani, 2013) The work proposes the Logistic Regression classifier modeling a diverse set of features. They take advantage of the baseline features provided by the SemEval task organizers (mostly, lexical similarity features), the explicit intercept feature with always the value of 1 for allowing modeling the *a priori* class distribution for each domain and item, word and character n-grams and Text Similarity features computed using the BLEU and PERP metrics (the latter is an edit-based metric developed by the authors) with student responses against reference and correct student answers. They incorporate the concept of domain adaptation in the built systems, trying to shape the modeled classifier according to the target test scenario. They made several copies of some features and gave them different weights, resulting in generic, domain-specific and item-specific copies. This was the most outstanding system overall of the SemEval-2013 SRA task.

Galhardi et al. (2018) The research aims at exploring distinct feature categories for the task of ASAG, similar to the predecessorial systems seen in the SemEval SRA challenge. Text statistical, Lexical Similarity and Semantic Similarity features are the main feature sets they tried to exploit. For the text statistics, the features were extracted from each individual student answer and at some ratio with the reference answer alongside the question (e.g. Spelling Error, Length Ratio, Word Length Average and Words per Sentence Average). Moreover, the features based on the lexical similarity between student and reference answers and questions are computed using four different metrics (Token-based, Edit-based, Sequence-based and Compression-based). The third of set of features focuses on word-to-word semantic similarity between the student and reference answer. Six algorithms implemented for computing similarity based on the synsets of WordNet were used. Then the aforementioned features are mapped into a single feature vector in order to be modeled by best perfoming classifiers such as Random Forest and Extreme Gradient Boosting. Additionally, for the Unseen Answers test scenario, the set of features extracted from the bag of n-grams was modeled individually by a classifier and the class probabilities were fed alongside the probabilities obtained from the modeling of the other 3 feature sets to a eventual meta-classifier. This work reported great improvements in the performance for ASAG, especially over the SciEntsBank subset of the SemEval-2013 dataset.

2.1.2 Deep Learning (DL)

The latest era in which the task of ASAG and every other NLP research is immersed is the era of Deep Learning. The constant technological advancements gave the possibility of representing text with more sophisticated features that capture the semantic relation, not just between words but also between sentences and bigger pieces of text, in a more precise way. These innovating sentence and word representations were obtained after the automatic processing of large corpora containing plenty of text as well as textual knowledge, which counteract the necessary and time-consuming human effort to extract effective characteristics from text, where the methods seen in the previous era come from, although the existence of automatic algorithms such as the aforementioned BLEU or PERP did not make everything manual. Still, this hand-engineered features have not been set aside, as it has been shown that their impact in modern ASAG systems has been crucial (e.g. in combination with word embeddings). Deep Learning has had its own evolution throughout the years, and we look over the principal milestones of it, following the research of Haller et al. (2022).

Word Embeddings

This concept was firstly introduced in the paper of Mikolov et al. (2013) where they took advantage of the power of Neural Networks to build linear representations of words in a n-dimensional space, popularly known as WORD2VEC. This is a similar idea from the conventional Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA), the continuous representation of words groups word vectors with similar meaning or form. The interesting fact about word embeddings is that the words in the vector space can be grouped according to multiple degrees of similarity, as each vectorial dimension can be considered as a certain property or characteristic of the word. Besides, the linear relation among word representations makes it possible to compute word and sentence representations from the linear combination of other word vectors. The traditional example displayed in Figure 2 shows that the vector of the word *queen* can be obtained by subtracting the vector of the word *man* to the representation of *king* and adding the vector of the word *woman*. In the last years, more research has been carried out in the field of word embeddings, and for example, the work carried out by Pennington et al. (2014) attempted to combine the idea of statistical representation of words of LSA with the predictive representations trained in word2vec, resulting int the global vector representation called GLOVE. This pretrained word representations were based on the advantage that count-based global statistics bring to the linear prediction methods. GloVe was described as a global log-bilinear regression model for the unsupervised learning of word representations. In the field of short answer grading the following works made use of the word embeddings.

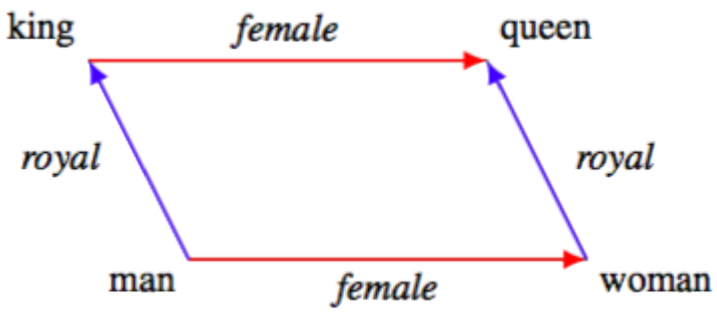

Figure 2: Linear relation among the vector representations of the words king, man, woman and queen. Source:
https://kawine.github.io/blog/nlp/2019/06/21/word-analogies.html

Sultan et al. (2016) The work describes a fast, simple and high-performing (at the time) short answer grading system. Based on the methodolgy of classical ML systems for ASAG, their feature extraction focuses mainly in the semantic similarity between the student and reference answers. Features are obtained via alignment, where the proportion of content words in one answer having a semantically similar word in the other answer is computed. The similar word pairs were identified using and aligner and computed the semantic similarity of pairs as a weighted sum of lexical and contextual similarities. In addition to these semantic features, they enriched the feature space with semantic vector similarity. Vector representation of each answer was calculated summing the word vector representations of the lemmas in the sentence and similarity between the answers was measured using the cosine similarity metric between vectors. The authors of this paper reported improvements on the SemEval-2013 5-way granularity subtask, outperforming aforementioned systems such as ETS and SoftCardinality. Nevertheless, the need for further developments of sentence representations was highlighted as well.

Magooda et al. (2016) They proposed a system which combines various types of similarity with main dependency on word vector representation. They used three pretrained word embeddings: word2vec, GloVe and Sense Aware Vectors (Neelakantan et al., 2015). The third vector representation of word is built on the word2vec vectors by giving each word multiple vectorized representations, one for each sense. Sentence representations were computed using two methods: the traditional adding of word representation vectors and weighted IDF summation. This second method is based on multiplying to each vector its IDF value extracted from the corpora and then the normalization over the IDF summation is done. This method takes word importance into account as it gives different weights to the words appearing in the sentence. Following the previous work, the similarity between student and reference answers is computed by cosine similarity and this worked as a feature as well. Using Support Vector Regression to model the sentence vector similarity features combined with various others the authors of this paper reported strong results in some

generic datasets but fall short in the benchmark SemEval-2013 5-way task where systems such as ETS and SoftCardinality turned out to perform better in some scenarios.

Roy et al. (2016) An iterative transfer learning based ensemble technique for ASAG was described in this work. Their novel research was based on a gradual transfer of knowledge from a source to a target question while accounting for question specific variations. In order to do this, two independent classifiers were built, a question specific and question agnostic. The first one was trained using bag of words technique corresponding to question specific student answers. Second one consisted in real-valued features capturing similarity of question agnostic student answers with respect to the model answer by means of extracting lexical, semantic and vector-space measures. The weighted esemble of these two classifiers was used to predict the final label. Transfer learning source knowledge to target questions was obtained by an iterative approach where the projection of the features from the second classifier were used (classical canonical correlation analysis) to predict the pseudo labels of target questions. Confident pseudo labels where then used to train the first classifier and to build the ensemble model in order to obtain new confident pseudo labels to use in the next iteration. Process ended either when reaching a certain number of iterations or when every target question had a confident predicted pseudo label. Although it is not a comparable system by its methodology, results reported outperformed by a margin the best perfomance of the SemEval-2013 SRA task, on the SciEntsBank subset.

Sequence-based Models

Researches carried out in the word embeddings era made it clear that the learning of sentence representations had a lot of room for improvement. The usage of pretrained word embeddings were not enough, since ASAG systems solely depending on those vector representations were not beating their predecessors. The field not only of ASAG but also of the whole NLP had its focus on being able to obtain embeddings that represent larger chunks of texts, such as sentences. Regarding ASAG, measuring the semantics of long-range dependencies in text is key, as a student is capable of answering within a range of few words and multiple sentences. As a consequence, Recurrent Neural Network (RNN) architecture began to take more part in the world of NLP. These networks are based on the concept of introducing the notion of time in basic NNs. Basically, this kind of NN does not process each unit (word) of a sequence (sentence) independently, it stores the information gathered at a certain time step for the unit which is about to be processed in the next time step. Figure 3 displays the basic structure of a RNN, where the sequence to be processed is x, represented by its time steps $(x^{(1)}, ..., x^{(t)}, x^{(t+1)})$ and in each time step, the historical state of the previous step is denoted by the vector v. Systems that derived from this architecture incorporated novel sentence representations and provided unseen results to the task of ASAG. The most prominent types of RNN are the simple RNN, Gated Recurrent Unit (GRU) (Gulcehre et al., 2014) and Long Short-Term Memory (LSTM) (Sak et al., 2014), which mainly differ on the hidden layer memory cell, employed

for storing information of the previous time steps of the sequence, displayed in Figure 4. The following works made use of sequential-based methods for ASAG.

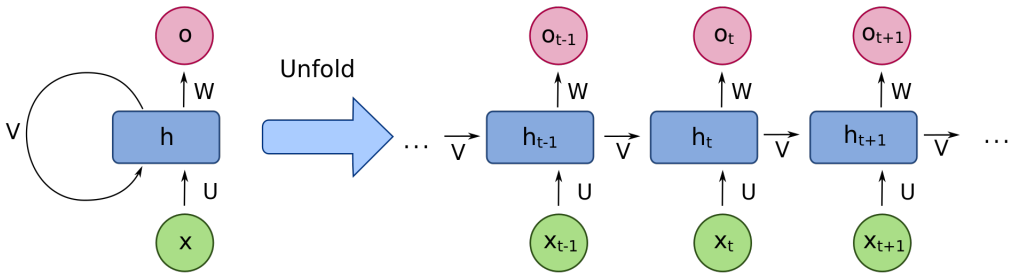

Figure 3: Basic RNN architecture. Source: https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg

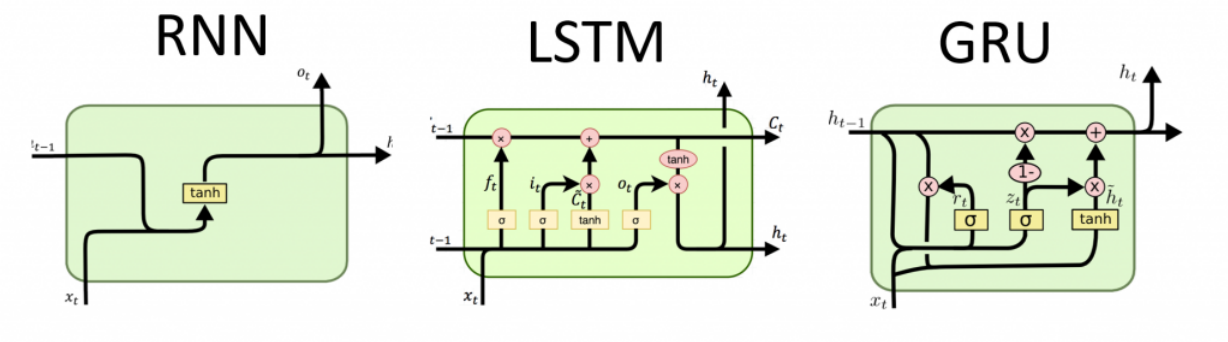

Figure 4: Comparison among the three main types of RNNs: simple RNN, LSTM and GRU. Source: http://dprogrammer.org/rnn-lstm-gru

Ans2Vec (Gomaa and Fahmy, 2019) The authors proposed a skip-thought[4] vector approach to convert both the student and model or reference answer into meaningful vectors to measure similarity between them. They took advantage of pretrained sentence embeddings and did not require NLP pre and postprocessing such as stopword removal, POS tagging, lemmatizing, etc. Human-build knowledge bases such as WordNet were not required neither. The semantic vector of the student and reference answer were obtained via combine-skip model and the concatenation of the position-wise product and absolute difference between the two answers, which were used as features. A logistic linear classifier was trained fed with these features. They reported improvements in some of the benchmark

[4]A Neural Networks model for learning fixed length representations of sentences in any Natural Language without any labeled data or supervised learning.

datasets such as Texas and Cairo. Results for SemEval SciEntsBank dataset were not enough to beat Sultan et al. (2016)'s system.

Kumar et al. (2017) The research focused on obtaining long-range semantic dependencies between the student and reference answers to the given question in an ASAG dataset. A part of their sequence-based model consisted of a bidirectional siamese LSTM to obtain the sentence representations of the student and reference answers independently. The Long Shor-Term Memory is one of the aforementioned derivations of RNNs and it was implemented to counteract the vanishing gradient problem[5] basic RNNs had. In addition, answer representations were computed with a bidirectional structure, taking into account forward and backward representations of the answers. Answers were independently encoded as previously said, having as input for the biLSTM unsupervised word embeddings for each word in the answers. Then a pooling layer of Earth Mover's Distance is used to capture long distance semantics between the answers. The output of this layer was fed to a Support Vector Ordinal Regression (SVOR) layer to compute the final label. The training of this system was done backpropagating the EMD pooling errors to the LSTM weights. Results were reported over datasets such as Mohler CS and SemEval SciEntsBank.

Saha et al. (2018) The work parts from the comparison of sentence-level and token-level features for the task of ASAG. They stated that the enhancements towards getting proper sentence representations still had its drawbacks such as non-sentential answers of students and out-of-domain student answers. Token-level features where included in the mix to try counteracting this problem. Sentence representations of questions, student and reference answer were computed using InferSent, which is a unsupervisedly pretrained model to calculate universal sentence representations (Conneau et al., 2017). This universal system was built through training a biLSTM encoder over the SNLI dataset, a benchmark dataset for inference mean to focus on learning semantic relations between pair of sentences. Token-level features consisted of a combination of different feature representations: Word Overlap, Histogram of Partial Similarities (HoPS) and HoPS of POS tags. Finally, these combined features were used to train a multinomial Logistic Regression system and results showed that the impact of combining sentence-level and token-level features was really positive for ASAG, showing that sentence- and token-level features encode non-overlapping aspects of information, achieving better or competitive results compared to state of the art over benchmark datasets such as Mohler's, SemEval-2013 SciEntsBank and a Large Scale Industry dataset.

Joint Multi-Domain (Saha et al., 2019) The authors proposed a novel method for ASAG based on Joint Multi-Domain learning. Inspired by the fact that para-phrasal constructs can have similar meaning on different domains and that not strictly similar words can have the same meaning in certain domains, they built a model with both domain-generic and domain-specific similarity scorers. The methodology consisted on having a

[5]Losing the stored information of previous time steps when learning large data sequences.

single generic scorer trained with data from every domain and having a domain-specific scorer for each domain in the training data. Therefore, having k domains available in the training data, the overall system would consist of k domain-specific scorers and 1 domain-generic scorer. The sentence encodings of both student and reference answer were the input for every scorer and the data was modeled using a multinomial Logistic Regression classifier. They carried out experiments over the datasets of Large-Scale Industry and SemEval-2013 SciEntsBank. Throughout the experiments, the comparison between Transfer Learning and Task-specific learning was also investigated. The first consisted on computing answer representation using the pretrained representation (Conneau et al., 2017) and the second on including the training of an answer biLSTM encoder. Overall results showed joint-domain and task-specific learning outperformed the rest of models and on the dataset of SemEval-2013 SciEntsBank state-of-the-art results were reported.

Attention-based Models

The constant evolution of Deep Learning brought another milestone in the field of NLP, the breakthrough of architectures known as Transformers. Systems where the attention mechanism was used alongside the recurrent and convolutional systems had been seen at the time. However, Vaswani et al. (2017) proposed a novel architecture only relying on the attention mechanism. The attention between pieces of text is capable of distributing the meaning text throughout the words that compound it, quantitatively. In this way, the words that affect semantics the most can be identified and thus a more sophisticated feature representation is obtained from source text. Besides, it was shown that the operations computed on attention layers were more efficient and less computationally complex than the ones computed in recurrent and convolutional layers and it showed a better capability of paralleling, computing attention with fast matrix operations. As a result, this architecture could be modeled significantly faster and larger corpora could be used as input, increasing the potential knowledge for the system to gain, and thus replaced the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention layers. This less complex architecture was also able to capture long-range dependencies better than the previous architectures, as the path length between forward and backward signals withing the network was shorter as well.

Based on the concept of self-attention, Devlin et al. (2018) published a pretrained LM for Language Understanding, concretely Bidirectional Encoder Representations from Transformers (BERT). BERT was designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. Thanks to its attention mechanism, BERT is able to be fine-tuned for most of NLP downstream tasks, without mattering whether they involve single text or text pairs. In addition, the core of the system barely varies depending on the target task and thus BERT can be modeled for tasks such as Question Answering, Natural Language Inference or Sequence Labeling, among others. During pretraining, the model is trained unsupervisedly on unlabeled data over different pretraining tasks. The aforementioned capability of attention

mechanisms to parallel computation enabled to feed BERT with a huge amount of text, where of course, being an unsupervised training there is no need to label the input data. As a result, the pretrained Language Model gets to understand natural language without almost any human effort. For fine-tuning, the BERT model is first initialized with the pretrained parameters, and these are fine-tuned using labeled data from the downstream task datasets. In this way, transfer learning the knowledge gained with unlabeled data was proven to enhance significantly the performance in the given NLP task.

In the last years, new pretrained LMs have appeared, inspired by the success of BERT. Examples of this are the likes of RoBERTa (Liu et al., 2019), where focus was given to the impact of different hyperparameters and training data, XLNet (Yang et al., 2019), where an autoregressive formulation was proposed to counteract BERT model's weaknesses, XLM (Lample and Conneau, 2019), XLMRoberta (Conneau et al., 2019), DistilBERT (Sanh et al., 2019), which used the technique of distilation to approximate BERT with a smaller one, AlBERT (Lan et al., 2019) which lowers memory consumption and training time by applying parameter reduction to BERT, and T5 (Raffel et al., 2020). The works described below took advantage of pretrained LMs to boost ASAG performance.

Sung et al. (2019b) The capability of measuring transfer learning to data-starved ASAG task using transformer-based pretrained systems was firstly shown in the work described in this article, where BERT was fine-tuned with ASAG datasets as a sentence pair classification task, given the supervised student and reference answer as input. In their experiments, it was shown that task-specific supervised fine-tuning is possible with small number of samples. For the SemEval-2013 SciEntsBank 3-way dataset, results that outperformed the state of the art (Saha et al., 2018) were reported and thus stated that unsupervised pre-training of BERT helps to leverage a huge amount of existing natural language material. Besides, experiments carried out over their two psychological domain datasets showed that limited task-specific fine-tuning data can obtain competent results as increasing the training data from 20% to 80% gave an improvement of 10% on macro f1-score metric. Besides, the ability of fine-tuned models to generalize unseen domains was shown. Finally, empirical results suggested that domain-specific supervised data is indeed required for efficient fine-tuning, but joint-domain fine-tuning yielded results relatively similar to domain-specific tuning.

Sung et al. (2019a) Rather than focusing on the fine-tuning of pretrained BERT, the work described incorporated ASAG-related data to the pretraining of BERT. In order to do this, experiments with two different approaches were analysed. The first approach relied on usage of textbooks from specific domains of short answer grading, feeding each paragraph of the textbooks for pretraining objectives. The second approach consisted in adding supervised ASAG (question, student answer, reference answer) triples to pretraining data, in form of (question, student answer) and (studen answer, reference answer) pairs. Just correct answers were considered, since non-correct ones

are often grammatically incorrect and could harm language model learning. The data to be incorporated for pretraining was extracted from the large-scale industry dataset, consisting of three domains. Textbook data from pretraining was generated from two of the domains, in order to test perfomance of BERT in unseen domains with the third. For the second approach, data of all three domains were added. Empirical results showed that including domain resource in the pretraining of BERT improves in-domain perfomance, but the model tended to lose capability of domain generalization as the out-domain performance dropped.

Camus and Filighera (2020) The authors of the article experienced with the fine-tuning of several transformer-based pretrained systems for the task of ASAG, on the SemEval SciEntsBank 3-way granularity subtask. The fine-tuning was done in a similar way to Sung et al. (2019b), a classification layer was added to the transformer architecture and training was done as a sentence pair classification task, having the student and reference answer as input pairs. They also trained multilingual models by adding other translated examples of the data. Among the fine-tuned pretrained systems were the likes of BERT, RoBERTa, AlBERT, XLM and XLMRoBERTa. Their experiments showed some interesting findings. They of course showed that transformed-based pretrained systems obtain state-of-the-art results when they are fine-tuned in ASAG. In fact, the fine-tuning of a pretrained system already fine-tuned in the GLUE benchmark MNLI dataset was able to ideally transfer knowledge into ASAG and obtained the best results by a margin. This finding indicated that the capability for textual understanding acquired from fine-tuning on a Textual Entailment dataset like MNLI can be helpful for ASAG, as the task was reformulated to an inference one. Lastly, the capàbility of multilingual models to generalize across languages not seen in training in contrast to the monolingual counterparts was also pointed out by the authors.

Chen and Li (2021) The work describes a transformer-based approach although pretrained language models from prior knowledge where employed just as auto-encoders. The authors of the article claimed that most of previous ASAG systems mainly concentrated on exploiting feature extraction from the textual information between the student answer and the model answer. Their novel system incorporates question information to obtain a better feature representation by doing a two stage training. In the first stage, an auto-encoder layer extracts features independently from question-student answer and student-model answer pairs. The second stage consisted of feature fusion layer based on pooling and the outputs of the first stage forward propagation are used to feed it. Finally, prediction is done in the output layer, using softmax for classification tasks, and cross-entropy loss is computed against ground-truth labels. This system was tested over the regression CS dataset and the SemEval-2013 5-way granularity subtask. Prior state-of-the-art systems were outperformed and their novel system combining feature fusion layer with conventional transformer-encoder layer demonstrated its effectivity.

Khayi et al. (2021) Inspired by the success of transfer learning pretraining-fine-tuning paradigm in low-resource NLP tasks, the authors exhaustively experimented with fine-tuning several transformer-based pretrained models for the task of Open Student Answer Assessment. In particular, the experiments were done over the DT-Grade dataset and the following pretrained systems were fine-tuned: BERT, RoBERTa, XLNet, DistilBERT, AlBERT and T5. 500 experiments were conducted in total, repeating 100 times each experiment, where the input for the fine-tuning consisted of student and reference answer pairs. Empirical results shown that T5 performed the best in the test scenario and every fine-tuned system surpassed all the previous methods with significant margin. The performance of distilled versions of BERT, DistilBERT and AlBERT, were demonstrated to be feasible for open student answer assessment as well.

2.1.3 Textual Entailment as pivoting task

Textual Entailment was firstly introduced in Dagan et al. (2005) in the first Recognizing Textual Entailment (RTE) challenge and was further developed by Bowman et al. (2015), when they introduced the first large-scale dataset for RTE as the task was considered fundamental for obtaining proper semantic representations for the understanding of natural language. The task of RTE, currently better known as Natural Language Inference (NLI), consists in learning the semantic concepts of entailment and contradiction between pairs of sentences. Given a premise P and a hypothesis H, it is said that P entails H if the information that P gathers implies that H is also true. Nevertheless, if the information given by P implies that H is not true, it is said that P contradicts H. Therefore, NLI can be seen as a binary classification task between sentences, but the labeling granularity can be expanded. Most of NLI tasks either predict that P entails H or P contradicts H or that P and H have no relation between them, resulting in a 3-class prediction. As an insight to the importance of NLI, several state-of-the-art systems consist of large pretrained LMs fine-tuned over NLI datasets (Lan et al., 2020; He et al., 2020; Conneau et al., 2019; Liu et al., 2019; Lewis et al., 2020).

Textual Entailment has been shown to be useful as pivot task for zero/few-shot learning. As inference classification requires semantic understanding at different degrees, reformulating low-resource NLP tasks into Textual Entailment has had remarkable impact. For instance, several text classification systems consist of a pretrained entailment system where no or few data of the task at hand is fed to it. This kind of systems have been shown to be capable of generalizing across domains and thus do not require large datasets to learn a specific task. For example, Sainz and Rigau (2021) reported state-of-the-art results on zero-shot domain labeling task reformulating the given task into an textual inference problem. Examples of other contributions for having Textual Entailment as pivoting task are the works of White et al. (2017), which proposed to unify a variety of existing semantic classification tasks under a single Textual Entailment task, Poliak et al. (2018), where 13 datasets where from 7 semantic phenomena into a common NLI structure, and Yin et al. (2019), which proposed a Textual Entailment framework which can work with or without

the annotated data of seen labels for text classification.

Moreover, Relation Extraction systems have had such a growth in performance following the same idea of reformulating the task to Textual Entailment: Levy et al. (2017) showed that Relation Extraction tasks can be reduced to answering simple reading comprehension questions, and Sainz et al. (2021) reformulated the task by doing simple verbalization of relations and showed results close to state of the art in the few-shot scenario with 20 times less data than other fully supervised systems. Similarly Sainz et al. (2022a) recasted the Event Argument Extraction as inference and reported strong results on few-shot scenario pointing out the importance of having a pretrained inference system with multiple domains. These works give an insight that task reformulation can be of real help in scenarios where the effort of annotation is so expensive that the task becomes low-resource. Recasting as an Textual Entailment task requires a transformation of original task samples, but this effort cannot be compared to annotating a large quantity of samples. This reformulation gives the opportunity to measure the knowledge of the pretrained system in both zero and few-shot scenarios, which has turned to be more close to real-world scenarios, in contrast to fully supervised systems.

Task reformulation to Textual Entailment has been already seen in ASAG (Camus and Filighera, 2020), where the authors made use of a pretrained transformer-based system fine-tuned on the MNLI dataset and demonstrated that transferring the knowledge gathered in the fine-tuning resulted to be really positive, reaching state-of-the-art performance. Nevertheless, the ASAG fine-tuning was done without question information, which could still improve performance, and neither they tested the system in low data regimes such as zero and few-shot scenarios.

2.2 Benchmark Datasets

The perfomance of ASAG systems has been and is measured over diverse public datasets and this has made it almost impossible to compare different approaches to the task, as the variability of ASAG datasets often make them incomparable and the lack of having a clear dataset for benchmarking the different systems that emerge is clear. They show a lot of variation in terms of the language, the topic of the question, grading scale, number of questions, reference answers, domain, answer length and so on. Following the recent survey (Haller et al., 2022), the characteristics and the motive for being designed of the most widely-used datasets are described in the following paragraphs. The public availability of these datasets and diversity of the answer domains allow to evaluate different aspects of the performance and capabilities of automated grading algorithms.

SciEntsBank and Beetle Both SciEntsBank and Beetle belong to the SemEval-2013 (Semantic Evaluation) Student Response Analysis (SRA) challenge, where they were released in order to benchmark ASAG systems' performance. The challenge was released

aiming to bring together researchers in educational NLP technology and the Textual Entailment task. The corpus contains manually labeled student responses to explanation and definition questions typically seen in practice exercises, tests, or tutorial dialogue. Each datasets sample consists of a question, at least one reference answer and 1- or 2-sentence student answer labeled with a degree of correctness. The challenge offers three types of label granularity resulting in 2-way, 3-way (these two concerning Textual Entailment) and 5-way datasets. For the 2-way granularity, student answers are labeled either by CORRECT or INCORRECT; for the 3-way granularity, available labels are the CORRECT, INCORRECT and CONTRADICTORY; finally, for the 5-way task CORRECT, PARTIALLY CORRECT BUT INCOMPLETE, CONTRADICTORY, IRRELEVANT and NON-DOMAIN are the possible labels. It can be seen that this datasets benchmark a classifier not a regressor as labels are capable of grading answers, but not scoring them.

SciEntsBank (SB) This dataset is a subset of the SemEval-2013 SRA corpus and is based on the corpus of student answers to assessment questions collected by Nielsen et al. (2008). It contains a total of 10,804 student answers distributed in answers to 181 questions (each question with a reference answer) from 12 domains.

Beetle (BT) BEETLE is based on transcripts of students interacting with BEETLE II tutorial dialogue system (Dzikovska et al., 2010). The dataset is mainly comprised of 56 questions in the domain of basic electricity and electronics requiring one or two sentence answers and it has 5,199 student answers in total to the 56 questions. In contrast to SCIENTSBANK, question-answer pairs in BEETLE may have one or more reference answer which can be labeled as BEST, GOOD or MINIMAL. For 56 questions, there is a total of 238 reference answers. Questions are either factual questions, or explanation and definition questions.

Texas The Texas dataset was developen by Mohler et al. (2011) and it consists of a dataset of questions from introductory computer science assignments with answers provided by a class of undergraduate students. The assignments were administered as part of Data Structures course at the University of North Texas. 31 students responded to a total of 80 questions, having a total of 2,273, less than the predicted queantity ($31 \times 80 = 2480$) since some students did not answer some assignments. The answers were not graded but scored by 2 teaches according to a given correct answer within an integer range of 0 and 5 and the ground truth score for each question-answer pair was computed by the average of the two human scores. The samples belonging to the dataset are scored with continuous values; hence, a regression approach would be needed to model this dataset.

ASAP-SAS The Automated Student Assessment Prize Short Answer Scoring (ASAP-SAS) data set was released as part of a Kaggle competition[6] in 2013, sponsored by the

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

Hewlett Foundation. It consists of a set of 10 questions from domains such as Science, Biology and English. In total, 22,431 student responses are collected and the scoring rubrics can be within the range between 0 and 2 or 0 and 3. For each degree of score, the requirements for the answers to calculate the score are presented to the evaluators. The average length of student answer is around 50 words although a small portion exceeds 100 words. This dataset requires a classification system since the the scoring range is not continuous.

2.3 Evaluation Metrics

Evaluation of ASAG systems can be done in two ways, according to whether the range of labels is discrete or continuous. For the first case, the task would be considered as a classification one and for the second a regression one. Depending on the type of task evaluation metrics can be classified into two groups.

Classification Metrics

Accuracy It is probably the simplest classification metric and it is computed by dividing the number of correct predictions by the total number of predictions the system at hand has made. Having a confusion matrix extracted from the comparison between predictions and ground truth labels, accuracy is the division of the sum of True Positives (TP) and True Negatives (TN) divided by the total sum of TPs, TNs, False Positives (FP) and False Negatives (FN).

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Recall Measures the proportions of positive classes that have been correctly predicted which is obtained by the division the TPs and the sum of TPs and FNs.

$$recall = \frac{TP}{TP + FN}$$

Precision Measures the proportion of predicted positives being truly positives. It is computed by dividing TPs by the sum of TPs and FPs.

$$precision = \frac{TP}{TP + FP}$$

F1-score It is the harmonic mean between the precision and recall measures since these two metrics are in continuous trade-off and it can happen to have a system with high precision but low recall and vice versa. F1-score is obtained with the following formula:

$$F1 - score = \frac{2 * precision * recall}{precison + recall}$$

F1-score has two variants that are mainly used depending on the class balance of a dataset. It can happen to obtain biased results provoked by class imbalance. The two variants are macro (macro-average) f1-score and weighted f1-score. The first variant gives equal weight to every class in the dataset whereas the second one gives different weight to the classes according to the occurrence of samples of the dataset with that label.

All the aforementioned metrics give a score that lies within the $[0, 1]$ interval. A value close to 1 in these metrics is ideal and it would be a clear sign of a reliable system, in contrast to having a value close to 0, which is an indicative of unrelible system.

Cohen's Kappa Its origins are in the field of psychology: it is used for measuring the agreement between two human evaluators or raters (e.g. psychologists) when rating subjects (patients). For ASAG, it is developed to account for the possibility that answer graders guess on at least some variable due to uncertainty. The metric is computed using the following formula:

$$K = \frac{p_o - p_e}{1 - p_e}$$

where p_o is the observed agreement and p_e is the expected agreement between raters. This metric ranges within the $[-1, 1]$ interval and a value close to 1 denotes a reliable system while a value down 0 or close to -1 is useless.

Quadratic Weighted Kappa QWK measures the agreement between two raters as the previous metric and it also takes into account the by chance probability of the two raters agreeing. Its value normally lies between 0 and 1, but it can also be negative. Similarly to the previous metric, a value close to 1 denotes a reliable predictor and a value close to or under 0 is not trustworthy. To compute QWK, first the weight matrix $W_{i,j}$ needs to be obtained:

$$W_{i,j} = \frac{(i - j)^2}{N - 1}$$

where i and j are two raters predictions, respectively, and N is the number of labels. $0_{i,j}$ and $E_{i,j}$ matrixes are also computed. The first corresponds to the adoption records that have a rating of i and predicted a rating of j. The second is the histogram matrix of expected agreements. Thus, QWK is computed by the following formula:

$$K = \frac{\sum_{i,j} W_{i,j} O_{i,j}}{\sum_{i,j} W_{i,j} E_{i,j}}$$

Regression metrics

Root Mean Squared Error Is a standard evaluation metric to assess the perfomance of regression systems. Having a set of predictions $\widehat{y} = (\widehat{y}_1 ... \widehat{y}_n)$ and set of ground truth

scores $y = (y_1...y_n)$ RMSE is obtained as follows:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n} n\frac{(\widehat{y}_i - y_i)^2}{n}}$$

RMSE is always non-negative, and a value of 0 (almost never achieved in practice) would indicate a perfect fit to the data. In general, a lower RMSE is better than a higher one.

Correlation coefficients These coefficients measure the strength of association between two variables. This correlation values lie within the $[-1, 1]$ interval and value bigger than 0 indicates a positive correlation between the variables, which is an indicative of reliability. For ASAG, the variables for which the correlation is measured are the prediction of the system at hand and the ground truth predictions. The most used correlation coefficients are Spearman's correlation and Pearson's correlation. Spearman's correlation is computed by the following formula:

$$r = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2(y_i - \overline{y})^2}}$$

where $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$ are the system's scores and ground truth scores respectively, and $\overline{x}$ and $\overline{y}$ are their respective averages. Pearson's correlation is computed by the following formula:

$$p = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

where x and y denote the same as in the previous correlation coefficient formula and n is the number of observations.

3 SemEval-2013 SRA Dataset

This dataset has been previously described in Section 2.2 where an insight of the creation of the SciEntsBank and Beetle subsets, the subtasks according to label granularity and some basic statistics about number of questions, student answers and reference answers has been provided. In this section, we take a deeper look to it.

Data Statistics Table 1 summarizes some of the main characteristics of the two subsets that belong to the entire dataset. The numbers shown in the table have been extracted gathering both the training and test splits available in the dataset[7], statistics regarding training and test splits separately will be shown afterwards. At first view, looking at Table 1, SciEntsBank dataset is shown to be larger as well as richer, as the differences in number of questions, student answers and number of domains are an indicative of this. Note that SciEntsBank contains five times more domains, almost four times more questions and more than the double of student answers. However, each question sample of Beetle seems to be more complete in terms of reference and student answers, since there is a significant difference in average number of the two answers types per question compared to SciEntsBank. Note that SciEntsBank data samples contain a single reference answer per question.

As for textual statistics, the table suggests that the textual information of SciEntsBank tends to be longer in words than the information offered by Beetle as the length averages of questions and both answer types is longer. In particular, it seems that questions in SciEntsBank are significantly more dense than in Beetle, something logical if the fact that Beetle is created from dialogues with human interaction whereas SciEntsBank was created by human annotation of explanation and definition questions is taken into account. To illustrate this, a question, student answer and reference answer triple is shown in Table 2a and Table 2b, corresponding to the SciEntsBank and Beetle subsets, respectively. The significant difference in length of the questions is easily perceived, as the SciEntsBank question example contains first an explanation of the situation to be analyzed and then two questions regarding the facts given. If this was to be the tendency in SciEntsBank, it would be logical to think that the reference and student answers would be more explanatory and thus longer in words.

Finally, regarding label distribution, the principal difference between the two subsets lies on the sparsity of contradictory answers (below 10%) in SciEntsBank in contrast to Beetle, where although this label keeps being the minority class its presence is close to a third of the entire set. The Incorrect label takes almost half of the examples in SciEntsBank and in Beetle the Correct label takes more than 40% of the entire set. For this last label, both subsets show a similar proportion. The label distribution is further decomposed on Table 4.

[7] We already had the official dataset in the machine where the experiments were carried out.

Characteristic	SciEntsBank	Beetle
Number of questions	181	56
Number of domains	15	3
Number of ref, answers	181	248
Average ref. answers per question	1	4.4
Number of std. answers	10,804	5,199
Average std. answers per question	59.7	92.8
Minimum question length	5	1
Maximum question length	186	55
Average question length	46.7	12
Minimum reference answer length	2	1
Maximum reference answer length	53	40
Average reference answer length	18.2	10.3
Minimum std. answer length	1	1
Maximum std. answer length	110	103
Average std. answer length	12	10
Correct answers	4,459 (41.2%)	2,185 (42.1%)
Incorrect answers	5,307 (49.2%)	1,610 (30.9%)
Contradictory answers	1,038 (9.6%)	1,404 (27%)

Table 1: Main statistics of the SemEval-2013 SRA dataset with regard to the SciEntsBank and Beetle subsets.

Problem Formulation ASAG can be defined as follows. Given a triplet of question, reference answers and student answers as input of our system, the system has to assess the student answer classifying it with a label that denotes the degree of correctness. According to label granularity, the SemEval-2013 SRA dataset includes three sets of labels that correspond to 2-, 3- and 5-way task problems, respectively. As mentioned earlier, both the 2-way and 3-way subtasks are derived from the 5-way task in order to approach ASAG as a Recognizing Textual Entailment (RTE) task. The 2-way task consists of classifying the answers either as Correct or Incorrect, whereas for the 3-way task each answer is labeled as either Correct, Contradictory or Incorrect. In this work, focus is given to the 3-way subtask and its examples are shown in tables 2a and 2b. Exceptionally for Beetle, the subset contains different kinds of reference answers denoting a certain degree of correctness. Table 3 shows different reference answers to a single question where these answers are categorized as Minimal, Good or Best.

Question	Jim used a solid and water to make Mixtures one (one spoon of solid in 100 milliliters water was clear with nothing on the bottom), 3 (3 spoons of solid in 100 milliliters water was clear with nothing on the bottom), 4 (4 spoons of solid in 100 milliliters water was clear with material on the bottom), and 5 (5 spoons of solid in 100 milliliters water was clear with material on the bottom) as shown below. He stirred each one and observed the results. If Jim made Mixture 2 with 2 spoons of solid in 100 milliliters of water, what would he observe? What evidence do you have to support this?
Ref. Answer	Jim would see a clear solution. 3 spoons of solid dissolves, so 2 spoons will dissolve.
Correct	It is going to be clear at the bottom and you will not see anything and I know this because if it does not dissolve in mixture it will not dissolve.
Incorrect	There would be a clear liquid with no solid on the bottom.
Contradictory	If mixture 3 is dissolved, and mixture one is not dissolved. Mixture 2 will not be dissolved because it is less concentrated than mixture 3.

(a) SciEntsBank

Question	Why was bulb A on when switch Y was open and switch Z was closed?
Ref. Answer	Bulb A is still contained in a closed path with the battery.
Correct	It has a closed path with the battery.
Incorrect	There was a closed path not effected by the switch.
Contradictory	The circuit was complete.

(b) Beetle

Table 2: A question, reference answer and correct student answer triple example extracted from SciEntsBank and Beetle

Evaluation Scenarios The SemEval-2013 challenge gives three different test scenarios in order to evaluate model generalization capabilities across problems and domains:

- **Unseen Answers** (UA): A set containing held-out student answers from questions which are available for training the system and contain some other student answers.
- **Unseen Questions** (UQ): A set containing held-out questions in order to assess the system in non seen questions but still lying in the same domain than the ones used for training.
- **Unseen Domains** (UD): Available only for SciEntsBank, a domain-independent test set of responses to topics not seen in the training data. From the 15 domains

Quesstion	Describe the paths in this diagram and explain how those paths account for the results.
Minimal	There is a path containing A and a different path containing B and C.
Good	Bulb A is in a path which does not contain B and C and isn't affected by B or C. B and C are in the same path and affect each other.
Best	Bulb A is in a path which does not contain B and C, so bulbs B and C don't affect it. Bulbs B and C are in the same path. They affect each other, but Bulb A doesn't affect them.

Table 3: Available reference answers to a certain question with different degrees of correctness. Extracted from the SemEval-2013 SRA dataset, belongs to the Beetle subset.

available for SciEntsBank, 3 are taken to this test split.

Once the evaluation scenarios have been defined, it is convenient to decompose the global statistics shown in Table 1 and take a look to Table 4. The significant difference in samples between SciEntsBank and Beetle seen previously is due to the Unseen Domains test scenario split, which takes almost half of the total student answers and one quarter of the total questions and reference answers of the subset. Without taking this independent scenario, the comparison between the number of samples between the two subsets is leveled a bit although SciEntsBank keeps being a larger subset. Main differences appear regarding the quantity of the questions and reference answers. The tendency shown in the global label distribution of Table 1 is mostly maintained in every split where the main aspect to consider is the imbalance of the Correct and Incorrect classes with regard to the Contradictory class in SciEntsBank in contrast to the more balanced distribution provided in Beetle.

	SciEntsBank				Beetle		
	Train	Test UA	Test UQ	Test UD	Train	Test UA	Test UQ
Question	135	135	15	46	47	47	9
Ref. answer	135	135	15	46	205	205	43
Std. answer	4,969	540	733	4,562	3,941	439	819
Correct	2,008 (40.4%)	233 (43.1%)	301 (41.1%)	1,917 (42.0%)	1,665 (42.2%)	176 (40.1%)	344 (42%)
Incorrect	2,462 (49.6%)	249 (46.1%)	368 (50.2%)	2,228 (48.9%)	1,227 (31.2%)	152 (34.6%)	231 (28.2%)
Contradictory	499 (10%)	58 (10.8%)	64 (8.7%)	417 (9.1%)	1,049 (26.6%)	111 (25.3%)	244 (29.8%)

Table 4: Number of questions, reference answers and student answers per each train and test split for SciEntsBank and Beetle subsets.

4 Entailment-based Answer Grading

In this section, our approach to model the ASAG task as Textual Entailment is described in a similar way done in Camus and Filighera (2020), although question information is included in our model. ASAG is regarded as a low-resource task and our main goal has been to research the capability of a NLI system understanding the textual language and transferring its knowledge to another NLP task by reformulating the task at hand to the conventional NLI format. First, our model is explained and a afterwards we give insight of the fine-tuning for the ASAG task.

4.1 Model Description

According to the standard definition of Textual Entailment, given two text fragments called Premise (P) and Hypothesis (H), it is said that P entails H if, typically, a human reading P would infer that H is most likely true (Dagan et al., 2005). On the contrary, it is said that P contradicts H if given the former it can be deduced that the latter is false. It is possible, as well, for P not to have any relation with H. In a typical answer assessment scenario, we expect that a correct student answer would entail the reference answer, while an incorrect answer would not. However, students often skip details that are mentioned in the question or may be inferred from it, while reference answers often repeat or make explicit information that appears in or is implied from the question (Dzikovska et al., 2013). Hence, a more precise formulation of the task in this context considers entailing text P as consisting of both the original question and the student answer, while H is the reference answer.

Figure 5 shows the schema of our entailment-based ASAG model where the input consisting of a question, reference answer and student answer is reformulated as a Textual Entailment problem. As ASAG examples contain $(question_i, stud.answer_i, ref.aswer_i)_{i\in 1...n}$ triplets, reformulation is done by concatenating each *question* and *stud.answer* in a single premise (P) while each $ref.answer$ is regarded as the hypothesis (H). As a result, each ASAG entailment example is defined as $(P_i, H_i)_{i\in 1...n}$ where $P_i = \{question_i, stud.answer_i\}$ and $H_i = ref.answer_i$.

In our experiments we focus on the 3-way classification task so the predictions of the entailment model are mapped to the 3-way set of labels in the Semeval-2013 SRA dataset. That is, the predictions of *entailment*, *contradiction* and *neutral* of the NLI model are mapped into CORRECT, CONTRADICTORY and INCORRECT, respectively. The 2-way granularity could also be reformulated to NLI, however, gathering both the INCORRECT and CONTRADICTORY in a single INCORRECT label is too generic and it is of little use for assessing students into finding the reason for their answer to be non-correct.

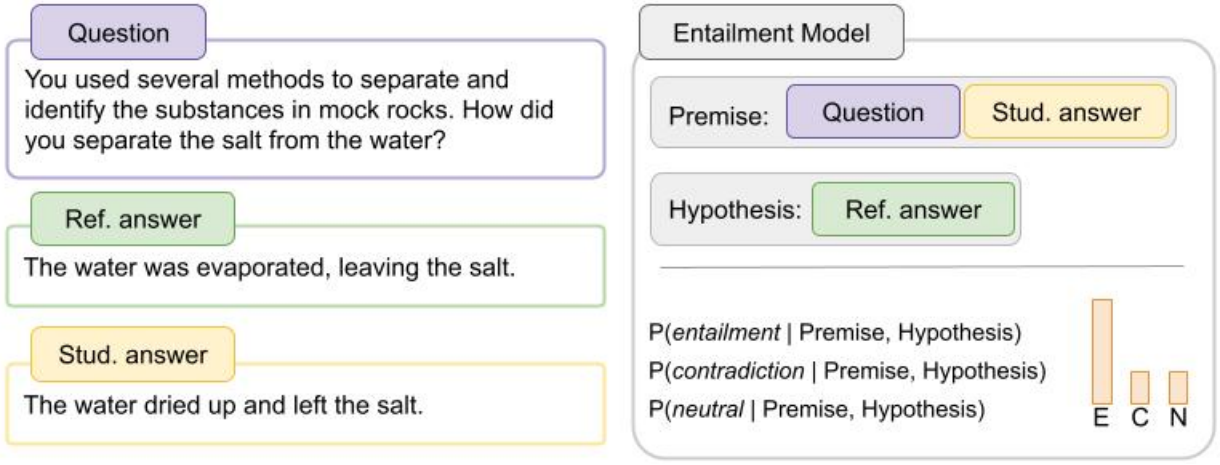

Figure 5: Schema of the NLI based ASAG model where the input of question, reference answer and student answer are reformulated as an entailment model. Concatenation of the question and student answer form the *premise* of the NLI model, whereas the *hypothesis* is generated with the reference answer. Prediction of the entailment model is then mapped to ASAG 3-way label

4.2 Fine-tuning of ASAG Model

We take advantage of the ability that NLI has to represent other NLP downstream tasks, ASAG in this case. We make use of RoBERTa (Liu et al., 2019), a variation of the BERT pretrained LM that stated that BERT was undertrained during in its pretraining. They researched the impact of some parameters during pretraining and, for instance, they added more data to pretraining extracted from corpora such as BookCorpus (Zhu et al., 2015), CC-News[8], OpenWebText[9] and Stories (Trinh and Le, 2018). Moreover, the removal of Next Sentence Prediction (NSP) during pretraining combined with larger BPE vocabulary for the pretraining of RoBERTa gave state-of-the-art results over the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018), among other datasets.

In particular, in our research, we select a large LM (RoBERTa (Liu et al., 2019) in this case) fine-tuned on the MNLI dataset (RoBERTa-MNLI) as base for the ASAG Entailment system. The fine-tuning of the ASAG model is built by reformulating the triplets (question, reference answer, student answer) provided in both SciEntsBank and Beetle datasets as traditional inference pairs (premise, hypothesis) as displayed in Figure 5.

[8] http://web.archive.org/save/http://commoncrawl.org/2016/10/newsdataset-available
[9] http://web.archive.org/save/http://Skylion007.github.io/OpenWebTextCorpus.

5 Experimental Design

For the experimental design let's recall first the research questions that were introduced in Section 1. In the beginning of our investigation, the capability of transfer learning the knowledge of a pretrained entailment system to ASAG was wondered and thus these main research points were established.

- Referring to RQ1, we explore effective ways to select new annotated examples in order to save time and effort. We experiment with different annotation strategies as well as different annotation quantities in order to see whether state-of-the-art performance can be reached without exploiting the whole training data.

- Referring to RQ2, we compare the performance between a task-agnostic system and a task-aware but out-of-domain system. It is known that one of the main weaknesses of ASAG is the capability to generalize across domains and, as a result, systems based on training an ensemble of diverse domains such (Saha et al., 2019) have been shown to be essential in order to build a domain-independent ASAG system. Our investigation in this aspect is based on testing a pretrained entailment model in the SRA SemEval-2013 dataset with no further training data (i.e. zero-shot) and make use of the two subsets (i.e. SciEntsBank and Beetle) to test the performance of a task-aware model trained with out-of-domain data. For instance, having a ASAG entailment model trained in SciEntsBank and test its quality over Beetle and vice versa. It is of interest to see whether out-of-domain data is helpful to learn the task for another domain or in the contrary, if it is harmful and zero-shooting a pretrained entailment system is more trustworthy.

- Referring to RQ3, which is related to the previous research question, we focus on the impact of adding out-of-domain examples and investigate whether training an out-of-domain task-aware system can lead to in-domain state-of-the-art results with a smaller quantity of training data than in the case of fine-tuned task-agnostic system with in-domain examples. It is of interest, as in the previous question, to investigate whether having learned the task from a different domains can be of any help towards reaching state-of-the-art performance with a smaller subset of in-domain data.

5.1 Annotation Strategies

The main contribution of the thesis is to explore the effectiveness of different annotation strategies when there is a need of having new annotated examples. RQ1 not only deals with data quantity, but also on the way to select new samples to effectively save time and effort. Similarly, RQ2 and RQ3 take into account the importance of selecting unseen data wisely in order to take advantage of the annotation as much as possible.

In order to answer these research questions, we explore two strategies of data annotation using the SemEval-2013 SRA dataset. As the dataset has multiple student answers

for a given question, the sampling of labeled data can be done answer- or question-wise. Specifically, we define two ways for sampling the training set of our experiments:

One Question per Student (Q2S) This scenario ideally annotates a unique question and student answer pair. That is, in the case we had 10 students we would create 10 different questions and would have 10 different answers, thus having a one-to-one relationship between questions and answers. The goal of this strategy is to increase the variability of the questions, losing the capacity of generalization over the answers to the same question. Note that, having very few examples for a given question might necessarily be a better strategy. Note as well, that in some cases it is not possible to exactly sample the defined dataset as there are not enough amount of questions in the dataset compared to number of samples required. In those cases, we tried to generate an approximated dataset by having the less possible answers to each selected question.

One Question to All Students (Q2A) This scenario annotates multiple answers for a single question with the goal of having larger variability on the answers side, that is, having diverse ways to answer the same question. For instance, in the case we had 10 students, we would create and ask a single question to all the students in order to get many answers for the question, thus resulting in a one-to-many relationship between a certain question and a set of student answers responding to it. Note that, in most of the cases there is not enough number of answers for a single question, so we tried to sample a dataset that approximate it as much as possible in a similar way to the previous annotation strategy.

Table 5 displays the few-shot settings designed for the experiments. It can be seen that for each few-shot percentage (1%, 2%, 5%, 10%), the two aforementioned annotation strategies (Q2S, Q2A) are defined as well, and the table also shows the number of questions and student answers that each few-shot setting contains. As stated previously, the ideal Q2S and Q2A annotations are not always collected since there are limited questions and student answers per question. Even so, each annotation strategy aims at adding more variability to either the questions or the student answers at hand. Finally, the last column shows the entire training set and cannot be sampled following different annotation strategies since all the data available is employed.

5.2 Experimental Setting

Dataset We use the data provided in the SemEval-2013 SRA shared task in our experiments. As explained above, the dataset consists of two distinct subsets: SciEntsBank and Beetle. The former is based on a corpus of student answers to assessment questions collected, whereas the latter is based on transcripts of students interacting with the Beetle II dialogue system. Although both subsets show similar structure, Beetle contains more than one reference answer for each question[10] while in SciEntsBank a single refer-

[10]We use one reference answer in our experiments chosen by random selection.

DS	Ann.	1%			2%			5%			10%			100%
		#Q	#A	Total	#Q	#A	Total	#Q	#A	Total	#Q	#A	Total	Total
SB	Q2S	40	1	40	80	1	80	100	2	200	100	4	400	3966
	Q2A	4	10		8	10		20	10		40	10		
BT	Q2S	28	1	28	28	2	56	35	4	140	35	8	280	2833
	Q2A	1	28		1	56		2	70		5	56		

Table 5: Number of questions (#Q) and student answers (#A) for each few-shot scenario according to the specific annotation strategy (Ann.) as well as the number of training examples (Total) for each of the few-shot setting and dataset (DS). SB stands for SciEntsBank, and BT for Beetle.

ence answer is given. SciEntsBank includes 150 assessment questions with 150 reference answers and 6,242 student answers in total, setting aside the UD test scenario. Beetle is a smaller subset as it has been shown in Section 3, which includes 56 questions, 283 reference answers and 5,199 student answers in total.

Validation Set As there is no validation set in the SemEval-2013 SRA dataset, we created it separating some examples from the original training set. We obtained a specific validation set for each test scenario. For the UA scenario the selection of validation examples was done answer-wise, and we held out a set of student answers for questions existing in the training part. For UQ, the selection was carried out question-wise, and we selected the same number of questions that were extracted for the test set. We sample 15 and 9 questions from the SciEntsBank and Beetle training datasets, respectively. For SciEntsBank, a validation set for the UD scenario was not created in order to have the same number of experiments for both subsets. In all the cases, we select the validation examples in a way to keep the class distribution as similar as possible to the test dataset in order to evaluate our models during training against a set of samples with similar characteristics to the test splits so as to try estimating the potential performance of the model in those. Table 6 shows the sizes of the validation sets in terms of number of questions, reference answers and student answers for each subset and validation scenario, accompanied by the label distribution of each created validation split. Comparing the validation splits to those of test, overall the validation splits contain fewer examples, but label proportion is maintained accordingly except for the SciEntsBank validation UA split which collects almost the same number of Correct and Incorrect student answers.

Few-shot Scenarios In order to measure the effectiveness of the annotation strategies in different few-shot scenarios, we generate the same training sizes for Q2S and Q2A. We created samples of 1%, 2%, 5%, 10% of the remaining training set, and we reduced the validation set in the same ratio. Table 5 shows the number of training examples for each few-shot scenario in the *Total* columns. Although 100% denotes full training, as a consequence of using a certain amount of training examples as validation set, it actually

	SciEntsBank		Beetle	
	UA	UQ	UA	UQ
#Questions	120	15	38	9
#Ref. ans	120	15	170	35
#Stu. ans	472	531	351	757
Correct	213 (45.1%)	206 (38.8%)	143 (40.7%)	308 (40.7%)
Incorrect	215 (45.6%)	281 (52.9%)	121 (34.5%)	234 (30.9%)
Contradictory	44 (9.3%)	44 (8.3%)	87 (24.8%)	215 (28.4%)

Table 6: Number of validation examples for unseen answers and unseen question test scenarios.

contains 1,003 and 1,108 less examples than the original training set for SciEntsBank and Beetle subsets, respectively.

Pretrained NLI Base Model As in Camus and Filighera (2020), we used the RoBERTa large (Liu et al., 2019) fine-tuned on the MNLI dataset as the base model for our zero and few-shot experiments. The model is publicly available in Huggingface[11]. We performed the following hyperparameter exploration for each few-shot scenario: we ran our model for 25 epochs with a batch size of 4 and select the best learning rate among 1e-5, 5e-5 and 4e-6, and the best gradient accumulation between 8 and 32. For the model selection we took the checkpoint with the lowest validation loss (cross-entropy) value from those 25 epochs. Taking the diverse annotation strategies, the hyperparameter search and the evaluation scenarios into account, a total of 24 (3 learning rate values × 2 gradient accumuluation values × 2 evaluation scenarios × 2 annotation strategies) different trainings were done for each few-shot percentage, resulting in a total of 96 tranining executions for each subset of the SemEval-2013 SRA dataset. After analysing the best performing model over each annotation strategy and validation scenario, 16 models were selected to be tested and to report the results for the few-shot experiments. Note that, model selection for UD could not be done as the validation split for this scenario was not created, and the few-shot results for this scenario were reported by taking the maximum score from the models trained for UA and UQ scenarios.

[11] https://huggingface.co/roberta-large-mnli

6 Experimental Results

6.1 Few-shot Results

Table 7 shows the results of the effect of the annotation strategies in the few-shot scenario and tries to answer the question that having a pretrained NLI model what would be the best strategy to annotate new data (**RQ1**). The mentioned table shows the macro f1-score for the few-shot experiments in which we fine-tune an entailment model using 1%, 2%, 5% and 10% of training data and evaluated in unseen answers (UA), unseen questions (UQ) for both subsets of the SemEval-2013 SRA dataset, and unseen domains (UD) for just the SciEntsBank subset.

Domain	Scenario	Annotation	0%	1%	2%	5%	10%	100%
SciEntsBank	UA	Q2S	56.2	**59.5**	**63.2**	63.9	**67.0**	71.0
		Q2A		58.3	60.0	**64**.1	59.6	
	UQ	Q2S	65.8	**67.0**	**66.7**	64.4	64.2	68.6
		Q2A		62.7	65.6	**65.9**	**66.8**	
	UD	Q2S	59.0	57.9	58.7	**58.8**	**61.2**	67.6
		Q2A		**58.2**	**59.2**	56.0	58.2	
Beetle	UA	Q2S	51.0	50.0	**52.3**	**52.7**	**56.6**	73.8
		Q2A		**50.1**	50.8	52.5	51.5	
	UQ	O2S	36.1	**37.0**	36.8	**38.0**	**43.1**	61.8
		Q2A		34.8	**37.8**	36.5	37.1	

Table 7: Results for the few-shot experiments in which we fine-tune an entailment model (NLI-roberta) using %1, %2, %5 and %10 of training data and evaluated in unseen answers (UA), unseen questions (UQ) and unseen domains (UD). Q2S annotation correspond to training data where we annotate one question per student, and Q2A correspond to one question to all student annotation procedure.

The results show that, overall, increasing the number of annotated questions at cost of reducing the number of different answers to the same question (Q2S) seems to be the best strategy compared to increasing the variability of answers (at cost of reducing the variability of seen questions) when annotating new question-answer pairs. This trend is confirmed in Table 8, where we report the macro-average of each few-shot setting. In addition, results suggest that Q2S annotation strategy yields better generalization properties as we increase the number of examples. For instance, when we annotate 400 examples in SciEntsBank and 280 examples in Beetle (10% few-shot setting), Q2S outperforms Q2A in almost 4

points, and the score increases steadily compared to the rest of few-shot settings. Nevertherless, this overall tendency is not reflected in the results reported for SciEntsBank UQ test scenario where the f1-macro score decreases monotonically as training size increases while the evolution for SciEntsBank UA scenario is totally the opposite. It is an outlier with respect to the general behavior, but it probably requires further research.

Annotation	1%	2%	5%	10%
Q2S	**55.7**	56.5	**56.6**	**59.2**
Q2A	55.2	56.5	54.6	55.8

Table 8: Average weighted macro f1-score results of annotation procedures in few-shot experiments.

As a final observation to the table, it is worth mentioning the capability of the RoBERTa MNLI system of transfer learning to SciEntsBank, but not much to Beetle. Zero-shooting is already quite competitive for the first subset and with a tenth portion of the data, results are close to full training. As for the second subset, Beetle, zero-shot is not capable of generalizing across the subset and even with 10% of the data the performance is far from being close to full training. As an hypothetical cause for this, the fact that Beetle samples are shorter in words, as seen in Table 1 may have affected the results to be poorer as the system would not be able to resolve non-sentential answers or few-word questions.

6.2 Cross-domain Results

Zero-shot Cross-domain Scenario Table 9 shows the results of the zero-shot cross domain evaluation, in which we try to answer the question posed in **RQ2**. In this setting, we have an entailment-based ASAG model fine-tuned in an out-of-domain dataset (e.g. Beetle) and evaluated in the target domain dataset (i.e ScientsBank). We compare the fine-tuned (task-aware) model to zero-shot entailment-based model (task-agnostic) in order to measure the effect of using out-of-domain task-related examples in learning.

Contrary to our expectation, task-aware fine-tuned models obtain significantly lower results compared to the task-agnostic model that is only pretrained in the MNLI dataset and not fine-tuned in the specific task.

Results suggest that the impact of the domain is bigger than the knowledge that can be acquired from the task. The drop is larger in unseen questions scenario (UQ) in both Bettle and SciEntsBank subsets. It can be explained assuming that unseen question scenarios require a higher capacity of generalization in order to perform better. In that

Test	Scenario	task-agnostic	task-aware
SB	UA	**56.2**	55.8
	UQ	**65.8**	59.7
	UD	**59.0**	53.9
BT	UA	**51.0**	50.4
	UQ	**36.0**	33.8

Table 9: Macro f1-score results of zero-shot cross-domain evaluation. Task-aware column shows the results for entailment model fine-tuned in one domain and evaluated in another domain. BT stands for Beetle dataset and SB for SciEntsBank.

sense, results suggest that generalization can not only be achieved using related tasks for transfer learning. In order to effectively transfer task related nuances, domain needs to be related as well. Therefore, it seems necessary to assess the degree of similarity between two domains before attempting cross-domain transfer learning. For instance, after checking the few-shot results of the previous section, having seen the total opposite behavior of the pretrained entailment system over two ASAG subsets, this unexpected empirical results are more understandable.

Few-shot Cross-domain Scenario Results of the top rows in Table 10 tries to answer question posed in **RQ3**. In this scenario we assume that we already have an entailment-based ASAG model fine-tuned in an out-of-domain dataset (e.g SciEntBank) and we get some annotated examples of our target domain (i.e Beetle). We evaluate the performance of adding target domain examples into the out-of-domain task-aware model.

Train → test	Scenario	0%	5%	10%
BT+%SB → SB	UA	55.8 (↓56.2)	58.9 (↓63.9)	**63.3** (↓67.0)
	UQ	59.7 (↓65.8)	62.5 (↓65.9)	**62.8** (↓66.8)
	UD	53.9 (↓59.0)	56.0 (↓58.8)	**59.3** (↓61.2)
SB+%BT → BT	UA	50.4 (↓51.0)	51.0 (↓52.7)	**54.0**(↓56.6)
	UQ	33.8(↓36.1)	34.6(↓38.0)	**37.9**(↓43.1)
SB+%BT → SB	UA	**71.0**	70.6	68.5
	UQ	68.6	69.8	**74.3**
	UD	**67.6**	66.7	64.0
BT+%SB → BT	UA	**73.8**	72.7	71.0
	UQ	**61.8**	59.9	55.7

Table 10: Results of cross-domain few-shot evaluation.

As expected, results show that adding few in-domain examples improve the results

compared to the model trained only in the out-of-domain scenario. On the contrary, the results are significantly below compared to in-domain few-shot models (figures in parenthesis), extracted from the few-shot enpirical results of Table 7. The results are in accordance with the ones obtained in the mentioned table, and suggest that the domain differences can affect negatively even if we are modeling the same task (which is something unexpected according to some recent work (Sainz et al., 2022b). That is, we can conclude that having an entailment model it is better to start from scratch rather than learning a out-of-domain ASAG model and retraining with a few in-domain examples (we draw similar conclusions in RQ2).

We defined a new setting where we do have an in-domain ASAG model (NLI model fine-tuned with target domain examples) and we add some out-of-domain examples, we can see a similar behavior of the model as in the previous settings. Results are shown in the bottom rows of Table 10. In general, we can conclude that mixing in-domain examples with out-of-domain examples is not helpful (only unseen questions scenario in SciEntsBank obtains any improvement), at least without measuring the degree of similarity between the domains to be combined, as mentioned earlier. In this last setting, the concept of **catastrophical forgetting** can be mentioned although the drop on the in-domain perfomance is not huge. Still the model suffers from being trained with out-of-domain data when being tested on in-domain test samples.

6.3 Comparison to the State of the Art

Table 11 shows the comparison of our model with state-of-the-art systems in SciEntsBank and Beetle datasets and the corresponding evaluation scenarios: unseen answers (UA), unseen questions (UQ) and unseen domains (UD). The table is organized in three groups: 1) top rows include the best systems that took part in the SemEval-2013 shared task, which correspond to a hand-engineered feature Machine Learning (ML) systems. In particular, the two best performing systems are shown, COMET and ETS. The system of Galhardi et al. (2018) is also included in this group although it is a more recently developed system, but follows a similar methodology; 2) middle rows include system that rely on Deep Learning (DL) methodologies as a main building block of their systems. Part of this group are the work of Saha et al. (2018) where pretrained sentence embeddings were employed as well as the likes of Sung et al. (2019b) and Camus and Filighera (2020), taking advantage of transformer-based unsupervisedly pretrained LMs; 3) bottom rows include our models fine-tuned with ASAG data. Firstly, a non-entailment RoBERTa fine-tuned with the whole training data (validation set for model selection), and secondly, the entailment RoBERTa fine-tuned using 10% of the data annotated with Q2S strategy, and fine-tuned using the whole set of the original training examples (validation set included for training). It is necessary to address the fact that FT does not contain the same meaning as 100%. FT denotes full training, but the training was done in two steps. Firstly, the pretrained inference system was fine-tuned with 100% of training data following the setting described in Section 5, being evaluated over the custom validation splits, in order to set a fixed number

Model	SciEntsBank			Beetle	
	UA	UQ	UD	UA	UQ
CoMeT Ott et al. (2013)	64.0	38.0	40.4	71.5	46.6
ETS Heilman and Madnani (2013)	64.7	45.9	43.9	71.0	58.5
Galhardi et al. (2018)	70.2	49.3	53.7	67.7	58.8
Saha et al. (2018)	66.6	49.1	47.9	-	-
Sung et al. (2019b)	72.0*	57.5*	57.9*	-	-
Camus and Filighera (2020)	78.3*	65.7*	70.9*	-	-
Our RoBERTa base 100%	47.8	47.4	46.8	51.0	38.7
Ours 10% (Q2S)	67.1	67.3	62.5	58.9	48.2
Ours FT	**76.5**	**72.3**	**69.1**	**76.7**	**70.0**

Table 11: Comparison to state-of-the-art f1-macro score results. Underlined figures denote that current results outperform previous state-of-the-art models. * for results not directly comparable with ours. Bold for best among comparable results.

of epochs. After setting the number of epochs, the pretrained inference system was reset and fine-tuned with the whole training set (100% training data + validation split) with the same parameters as the previous training with the difference that this time there was no evaluation during the training, just training loss optimization through the set number of epochs (i.e. there was no model selection).

It is worth noting that best performing systems in SciEntsBank (Sung et al., 2019b; Camus and Filighera, 2020) are not directly comparable with the rest of the models as it is not clear how the model selection was carried out[12] after researching the documentation of the work submitted by the authors.

Regarding our few-shot model (10%-Q2S), results show that annotating only 400 examples for training following the Q2S strategy is effective to outperform state-of-the-art systems in SciEntsBank dataset. Using the 10% of the training dataset (280 examples) in Beetle is not sufficient to attain state-of-the-art results and these results suggest we still need more annotated data for this domain. It is worth noting that Beetle seems more demanding as recent state-of-the-art models Galhardi et al. (2018)) are not able to surpass systems that participate in the SemEval-2013 shared task. More recent works such as the likes of Sung et al. (2019b); Camus and Filighera (2020) did not report results for this dataset neither. The fact that SciEntsBank data is quite well understood by the pretrained entailment system needs to be taken into account as well, since zero-shooting the model without further learning already yields strong results. This can be a sign to

[12]We suspect that for the development of models described in Sung et al. (2019b); Camus and Filighera (2020) the test set was used for fine-tuning the supervised system.

attempt transfer learning another pretrained inference system that better suits BEETLE to reach results close to state of the art with lower quantity of training data.

When we fine-tune our model using all the data available (FT) in the training set, the model yields state-of-the-art results in both datasets, and shows impressive generalization capabilities in those scenarios that presumably are more challenging. For example, our few-shot model improves in 18.0 macro f1-score points in SCIENTSBANK compared to the best comparable model in the unseen questions (UQ) scenario (49.3 vs 67.3), and we increase the margin up to 23.0 points when we use the whole training set for fine-tuning the model (ours FT). In BEETLE, the improvement in UQ scenario goes up to 11.2 macro f1-score points with the fully trained model.

As a last observation, the importance of having a pretrained entailment system is clear in order to reformulate ASAG to Textual Entailment. The base RoBERTa fine-tuned with just ASAG examples underperfomrs significantly and with 10% of the data, the entailment system is notoriously a more reliable system. Besides, referring to the zero-shot results for SCIENTSBANK of Table 7, the task-agnostic entailment system outperforms the fully trained RoBERTa. As for BEETLE, both systems yield similar results.

7 Conclusions and Future Work

In this work we reformulated the task of Automatic Short Answer Grading (ASAG) as an entailment problem, and explored to what extent the annotation strategies are effective in few-shot scenarios. The task reformulation to ASAG was carried out applying the currently leading methodology in the field of NLP: taking advantage of the huge LMs that are provided freely to the community for research purposes, and fine-tune this unsupervised language model to the task at hand. We transferred the knowledge of the pretained RoBERTa fine-tuned on the MNLI inference dataset to ASAG, incorporating the question information, in contrast to Camus and Filighera (2020). We also fine-tuned a RoBERTa base system with ASAG examples without entailment-based previous fine-tuning.

Among the various experiments we did, empirical findings showed that pretrained inference systems can be capable of yielding positive perfomances over ASAG datasets, as the zero-shot experiments over SciEntsBank supports the fact that the reformulation of ASAG into an entailment problem can be naturally done. In this aspect, zero-shooting a general purpose entailment system already surpasses a non-entailment-based pretrained system fully trained over ASAG examples. With regard to few-shot experiments, it was also shown that increasing the variety of questions in the annotation is more effective than annotating more answers of the same question. Our method makes effective use of available labeled examples, and using only 400 annotated examples is able to perform on par of state-of-the-art approaches in SciEntBank. Besides, when we use full training data, our model outperforms the comparable state-of-the-art systems across the two subsets. Finally, our analysis indicates that using cross-domain annotated examples is not beneficial and it is more effective to use a task-agnostic general purpose entailment model, at least when the degree of similarity between two domains of ASAG is unknown, as happened with SciEntsBank and Beetle.

As future work, we would like to explore ways to extend our method to a 5-way task, as well as to refine the selection of good examples in combination with active learning techniques. Besides, the sparsity of annotated data requires more cross-domain experiments to further analyse the capability of combining diverse ASAG datasets to both counteract the lack of data as well as produce a single system which is capable of grading student answers on different domains and variable question structures.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075.

Steven Burrows, Iryna Gurevych, and Benno Stein. The eras and trends of automatic short answer grading. International Journal of Artificial Intelligence in Education, 25 (1):60–117, 2015.

Leon Camus and Anna Filighera. Investigating transformers for automatic short answer grading. In International Conference on Artificial Intelligence in Education, pages 43–48. Springer, 2020.

Shuang Chen and Li Li. Incorporating question information to enhance the performance of automatic short answer grading. In International Conference on Knowledge Science, Engineering and Management, pages 124–136. Springer, 2021.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364, 2017.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116, 2019.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, 2005. URL http://www.cs.biu.ac.il/~glikmao/rte05/.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

Xinya Du and Claire Cardie. Event extraction by answering (almost) natural questions. arXiv preprint arXiv:2004.13625, 2020.

Myroslava Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. 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), pages 263–274, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/S13-2045.

Myroslava O Dzikovska, Johanna D Moore, Natalie Steinhauser, Gwendolyn Campbell, Elaine Farrow, and Charles B Callaway. Beetle ii: a system for tutoring and computational linguistics experimentation. In Proceedings of the ACL 2010 System Demonstrations, pages 13–18, 2010.

Evgeniy Gabrilovich, Shaul Markovitch, et al. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In IJcAI, volume 7, pages 1606–1611, 2007.

Lucas B Galhardi, Helen C de Mattos Senefonte, Rodrigo C Thom de Souza, and Jacques D Brancher. Exploring distinct features for automatic short answer grading. In Anais do XV Encontro Nacional de Inteligência Artificial e Computacional, pages 1–12. SBC, 2018.

Wael Hassan Gomaa and Aly Aly Fahmy. Ans2vec: A scoring system for short answers. In International Conference on Advanced Machine Learning Technologies and Applications, pages 586–595. Springer, 2019.

Caglar Gulcehre, Kyunghyun Cho, Razvan Pascanu, and Yoshua Bengio. Learned-norm pooling for deep feedforward and recurrent neural networks. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 530–546. Springer, 2014.

Stefan Haller, Adina Aldea, Christin Seifert, and Nicola Strisciuglio. Survey on automated short answer grading with deep learning: from word embeddings to transformers, 2022. URL https://arxiv.org/abs/2204.03503.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654, 2020.

Michael Heilman and Nitin 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), pages 275–279, 2013.

Sergio Jimenez, Claudia Becerra, and Alexander 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), pages 280–284, 2013.

Nisrine Ait Khayi, Vasile Rus, and Lasang Tamang. Towards improving open student answer assessment using pretrained transformers. In The International FLAIRS Conference Proceedings, volume 34, 2021.

Sachin Kumar, Soumen Chakrabarti, and Shourya Roy. Earth mover's distance pooling over siamese lstms for automatic short answer grading. In IJCAI, pages 2046–2052, 2017.

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291, 2019.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942, 2019.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=H1eA7AEtvS.

Omer Levy, Torsten Zesch, Ido Dagan, and Iryna Gurevych. Ukp-biu: Similarity and entailment metrics 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), pages 285–289, 2013.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. arXiv preprint arXiv:1706.04115, 2017.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

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

Ahmed Ezzat Magooda, Mohamed Zahran, Mohsen Rashwan, Hazem Raafat, and Magda Fayek. Vector based techniques for short answer grading. In The twenty-ninth international flairs conference, 2016.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. International journal of lexicography, 3(4):235–244, 1990.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pages 752–762, 2011.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space. arXiv preprint arXiv:1504.06654, 2015.

Rodney Nielsen, Wayne Ward, James H Martin, and Martha Palmer. Annotating students' understanding of science concepts. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), 2008.

Niels Ott, Ramon Ziai, Michael Hahn, and Detmar Meurers. 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), pages 608–616, 2013.

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

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. Towards a unified natural language inference framework to evaluate sentence representations. 2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21(140):1–67, 2020.

Shourya Roy, Himanshu S Bhatt, and Y Narahari. An iterative transfer learning based ensemble technique for automatic short answer grading. arXiv preprint arXiv:1609.04909, 2016.

Swarnadeep Saha, Tejas I Dhamecha, Smit Marvaniya, Renuka Sindhgatta, and Bikram Sengupta. Sentence level or token level features for automatic short answer grading?: Use both. In International conference on artificial intelligence in education, pages 503–517. Springer, 2018.

Swarnadeep Saha, Tejas I Dhamecha, Smit Marvaniya, Peter Foltz, Renuka Sindhgatta, and Bikram Sengupta. Joint multi-domain learning for automatic short answer grading. arXiv preprint arXiv:1902.09183, 2019.

Oscar Sainz and German Rigau. Ask2transformers: Zero-shot domain labelling with pretrained language models. arXiv preprint arXiv:2101.02661, 2021.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. Label verbalization and entailment for effective zero-and few-shot relation extraction. arXiv preprint arXiv:2109.03659, 2021.

Oscar Sainz, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Eneko Agirre. Textual entailment for event argument extraction: Zero-and few-shot with multi-source learning. arXiv preprint arXiv:2205.01376, 2022a.

Oscar Sainz, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Agirre Eneko. Textual entailment for event argument extraction: Zero- and few-shot with multi-source learning. In Findings of the Association for Computational Linguistics: NAACL-HLT 2022, Online and Seattle, Washington, July 2022b. Association for Computational Linguistics.

Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014.

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

Timo Schick and Hinrich Schütze. Exploiting cloze questions for few shot text classification and natural language inference. arXiv preprint arXiv:2001.07676, 2020.

Asher Stern and Ido Dagan. Biutee: A modular open-source system for recognizing textual entailment. In Proceedings of the ACL 2012 System Demonstrations, pages 73–78, 2012.

Md Arafat Sultan, Cristobal Salazar, and Tamara 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, pages 1070–1075, 2016.

Chul Sung, Tejas Dhamecha, Swarnadeep Saha, Tengfei Ma, Vinay Reddy, and Rishi Arora. Pre-training bert on domain resources for short answer grading. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 6071–6075, 2019a.

Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. Improving short answer grading using transformer-based pre-training. In International Conference on Artificial Intelligence in Education, pages 469–481. Springer, 2019b.

Trieu H Trinh and Quoc V Le. A simple method for commonsense reasoning. arXiv preprint arXiv:1806.02847, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461, 2018.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. Inference is everything: Recasting semantic resources into a unified evaluation framework. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 996–1005, 2017.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 32, 2019.

Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. arXiv preprint arXiv:1909.00161, 2019.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision, pages 19–27, 2015.