Knowledge and Pre-trained Language Models Inside and Out: a deep-dive into datasets and external knowledge

Ph.D. Transfer Report

Chenyang Lyu
Supervisors: Dr. Jennifer Foster and Dr. Yvette Graham
ML-Labs
School of Computing
Dublin City University
chenyang.lyu2@mail.dcu.ie

April 26, 2022

Abstract

Pre-trained language models such as BERT and XLNet have greatly improved the performance of many NLP tasks. These models can capture rich semantic patterns from large-scale text corpora and learn high-quality representations of texts. However, such models have shortcomings – they underperform when faced with complicated noisy text or text that requires inference and/or external knowledge to be understood. Therefore the focus of this PhD project will be on the learning of knowledge for pre-trained language models. This project has two major goals. Firstly, I aim to explore how to inject extra knowledge into large-scale pre-trained models. Secondly, I aim to investigate how pre-trained language models acquire knowledge from fine-tuning data. I focus on two tasks: Sentiment Analysis and Question Answering. In the first part of this project, I focus on injecting knowledge into pre-trained models in sentiment analysis and question answering. I have made progress in both areas, developing a state-of-the-art sentiment classifier for product reviews and an unsupervised question answering system. For the second goal of this project, I have carried out detailed analysis of extractive question answering data in order to gain insight into how models learn from such data. My work to date has focused on data in the English language. For the remainder of this project, I will explore other languages, including cross-lingual representations for sentiment analysis and question answering, as well as understanding how multilingual data affects pre-trained language models.

1 Introduction

Recent years have witnessed the emergence of pre-trained language models (PLMs) such as ELMo, GPT, BERT, XLNet (Wang et al., 2018a; Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019a; Yang et al., 2019), which have been widely used in many NLP tasks and have shown superior performance compared to previous approaches (Devlin et al., 2019a; Qiu et al., 2020a). PLMs are firstly pre-trained on large-scale unlabeled text corpora using self-supervised objectives, followed by fine-tuning on downstream tasks with labeled data using supervised learning, resulting in a new paradigm for NLP research - pre-training on large-scale unlabeled data + fine-tuning on small-scale labeled data. This has been shown to surpass previous neural approaches trained only on labeled downstream task data (Devlin et al., 2019b). Different from early approaches producing static word embeddings where each word only has one embedding vector, PLMs produce *contextualized word representations* (Peters et al., 2018), where words

have different representation vectors within different contexts. This is in line with the commonsense assumption that the semantics of a word should not only depend on itself but also depend on its context. Such modifications, powered with large neural models (Vaswani et al., 2017) and large-scale corpora, give significant improvements on a wide range of NLP tasks including sentiment analysis, question answering and natural language inference. Probing tasks have shown that the representations learned by PLMs capture aspects of the semantics and syntax of language (Jawahar et al., 2019; Rogers et al., 2020).

Despite the success of pre-trained models in NLP, such models still lack the knowledge needed for tasks which require information beyond the text such as sentiment analysis, entity typing and question answering (Da and Kasai, 2019; Liu et al., 2020a). The incorporation of structured knowledge from knowledge graphs has been explored in Zhang et al. (2019); Yu et al. (2020); He et al. (2020); Colon-Hernandez et al. (2021); Wang et al. (2021), yielding improvements for various knowledge-intensive tasks including named-entity recognition, relation classification, entity typing and question answering especially for domains such as the medical domain. For example, in entity typing and relation classification, without external entity knowledge such as knowledge base triples of the form <*Entity1, Relation, Entity2*>, it is difficult for a pre-trained language model to produce the correct prediction even though it has captured rich information from pre-training on huge volumes of unstructured text. In Zhang et al. (2019), the use of entity information from knowledge graphs injected into the joint pre-training process in Devlin et al. (2019b) substantially improves a model's performance on entity-typing and relation extraction, where the token-entity alignment objective aims to inject the entity information into the representations learned by the transformer encoder.

Earlier work mainly focuses on incorporating structured knowledge from knowledge graphs (entity knowledge and linguistic knowledge). An exploration of methods for injecting other external information beyond text is lacking. Approaches for incorporating knowledge have been limited to learning joint representations of text and knowledge, requiring substantial modifications to model architectures. Furthermore, despite the success as well as the large volume of research conducted on PLMs (Qiu et al., 2020a; Zhang et al., 2020), less emphasis has been placed on the effects of the data used for fine-tuning. A better understanding of the data has the potential to improve the generalizability of models (Rogers, 2021; Gardner et al., 2021), as well as providing helpful information for constructing datasets (Bender and Friedman, 2018; Geva et al., 2019). Thus, I will explore two major research questions, 1) how to incorporate external knowledge beyond text into pre-trained language models and how to develop easier approaches for injecting knowledge without substantial modifications to model architectures, 2) what are the characteristics of fine-tuning data, how does fine-tuning data affect the performance of PLMs and how do PLMs learn knowledge from fine-tuning data?

1.1 Research Questions

I propose two high-level research questions and each high-level research question is followed by two specific research questions.

RQ1: How can we use external knowledge to improve the performance of pre-trained language models?

The primary goal of **RQ1** is to investigate how to use **external knowledge** beyond the normal fine-tuning data that is commonly employed to improve the performance of PLMs on downstream tasks. I focus on two tasks: document-level sentiment analysis and unsupervised question answering.

RQ1-1: How can we utilize the extra information in the metadata of product reviews to improve document-level sentiment analysis?

The goal of Sentiment Analysis is to predict the sentiment conveyed by a piece of opinionated text (often a review). In document-level sentiment analysis with user and product information, we also know the user who wrote the review and the product being evaluated by the review. User and product context can

be helpful for predicting the correct sentiment label: the same user may tend to use the same or a highly similar narrative style as well as similar word choices when writing reviews. For example, a user who has high expectations for the product being evaluated might use words like *good, nice* but only give a rating *medium positive* or even use such positive words sarcastically to give a negative rating; similarly, the reviews belonging to a particular product may have the same group of opinionated words and narrative style towards the product being evaluated. Earlier work (Tang et al., 2015; Chen et al., 2016b; Ma et al., 2017; Dou, 2017; Long et al., 2018; Amplayo, 2019; Amplayo et al., 2018) mainly focuses on modeling users and products as embedding vectors which are updated in the training process, with the expectation that such embedding vectors can implicitly learn the bias introduced by users and products. However, such approaches fail to fully make use of the textual information of historical reviews belonging to a user or a product, since it is difficult to learn meaningful representations of user and product if they are only updated and learned by back propagation, especially for users and products who only have small number of reviews. Therefore, RQ1-1 will focus on how to model the historical reviews of a user and product to learn more meaningful representations of user and product context for the purpose of improving the prediction of sentiment labels.

RQ1-2: How can we leverage linguistic knowledge and summarization data to improve Unsupervised Question Answering?

The goal of Question Generation (QG) is to generate plausible questions for given <*passage, answer*> pairs. QG can be applied in dialogue systems as well as education (Graesser et al., 2005) and as a data augumentation method for Question Answering (QA) (Puri et al., 2020). There are two classes of QG approaches: 1). *Template-based QG* (Heilman and Smith, 2009, 2010) which uses heuristics induced from linguistic knowledge to transform declarative sentences into questions. It has the shortcoming that the generated questions have high lexical overlap with the source text since *template-based* approaches only manipulate the existing constituents in the source text; 2). *supervised QG* (Du et al., 2017; Duan et al., 2017; Zhang and Bansal, 2019; Chen et al., 2019; Xie et al., 2020; Ma et al., 2020; Ji et al., 2021) which uses existing QA datasets to train a QG system. A disadvantage of the supervised approach is that it heavily relies on the availability of QA datasets which can cost a lot to obtain and are heavily tied to a certain domain/language. RQ1-2 will explore how to combine the advantages of the template-based and supervised methods as well as address the shortcomings in both approaches. The generated questions will then be used to train an unsupervised QA system.

RQ2: How does the fine-tuning data influence the performance of pre-trained language models?

I will analyse the effect of fine-tuning data on the performance of PLMs. I will firstly explore how PLMs learn from different parts of the fine-tuning data. Then I will focus on multilingual data and investigate how it influences models' performance and how to make the best use of it. This research question will focus more on data than modelling. Again, the focus will be on sentiment analysis and question answering.

RQ2-1: How does a machine learning model (typically a neural model) learn from sentiment analysis and QA data - which part of the data accounts for the model's performance on dev/test set?

After the emergence of PLMs, substantial improvements have been obtained on many NLP tasks including Sentiment Analysis and QA (Qiu et al., 2020b; Bommasani et al., 2021). However, we still cannot neglect the importance of the dataset, and indeed this has become a new focus of NLP research (Søgaard et al., 2021; Lewis et al., 2021; Liu et al., 2021). However, the following questions remain: How does a PLM learn from a dataset - which part of the dataset is essential to correctly predict the ground-truth outputs in dev/test set? Is there any biased, flawed or even falsely annotated data in the dataset and how can

such data affect model performance? What knowledge do models learn from the dataset? How can we measure model's robustness and generalizability by inspecting the training dataset? RQ2-1 will focus on analysing how a pre-trained model learns from a dataset using Sentiment Analysis and Question & Answering datasets as the testbed.

RQ2-2: Exploring multilingual representations and analysing their role in learning from multilingual corpora

Multilingual data has been widely used in NLP, especially in machine translation (Sennrich et al., 2016; Kondratyuk and Straka, 2019) and PLMs (Conneau and Lample, 2019; Rogers et al., 2020; Liu et al., 2020b). When training models using multilingual data, the data in resource-rich languages can be helpful for improving the performance on low-resource languages (Adams et al., 2017; Gu et al., 2018). In multilingual learning, I observed that different languages share a single backbone model, the only difference is their embedding matrix. Hence there is the possibility that using multilingual data is approximately a form of data augmentation. Since we feed data in different languages to the *same* backbone model, the weights of the model will be updated more compared to the model trained on monolingual data. However, questions like how multilingual data affects model performance and the difference between multilingual models and monolingual models still remain under-investigated. RQ2-2 will explore the learning mechanism when using multilingual data to train a PLM.

1.2 Outline

Background information including the development of pre-training techniques is provided in Section 2, progress so far in Section 3.

2 Background

In this section, I will give an overview of the material which is relevant for this Ph.D. project. In particular, I will first describe the development of pre-training techniques. There are two dimensions to categorizing pre-training techniques. The first is feature-based pre-training approaches (word2vec, GloVe,, ELMo) versus non feature-based approaches (GPT, BERT), the second is non-contextualized word embeddings (word2vec, GloVe) versus contextualized word representations (ELMo, GPT, BERT). Some representative approaches will be discussed briefly in the following sections. Lastly, I will discuss the research which is related to our main research questions concerning how to inject knowledge into pre-trained language models.

2.1 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Following the significant success of PLMs such as ELMo (Peters et al., 2018) and GPT (Radford et al., 2018), Devlin et al. (2019b) propose a new pre-trained model, BERT, which adopts new objectives in the pre-training stage and has been widely used in NLP research especially in natural language understanding tasks. BERT uses the *transformer* (Vaswani et al., 2017) as its building block which is same as the neural architecture used in GPT. To pre-train BERT model on large text corpora (Devlin et al., 2019b), two objectives - Masked Token Prediction and Next Sentence Prediction are used:

Masked Token Prediction (Devlin et al., 2019b) make use of large-scale text corpora to create a masked token prediction objective by masking a certain proportion of tokens in the original sequence and then training BERT model to recover the masked tokens based on the unmasked tokens. Specifically, supposing that for a given sequence $(w_1, w_2, ..., w_n)$, we randomly mask some tokens w in the original sequence by replacing masked token w with a special token *[MASK]*, the indices of masked tokens are denoted as $\hat{I}$ and the original indices of all tokens including masked tokens and unmasked tokens are

denoted as I, the indices for unmasked tokens are represented as $I - \hat{I}$. We input the edited sequence $(w_1, w_2, ..., w_n)$ in which some tokens are replaced with *[MASK]* to the BERT model and aim to predict the original replaced tokens. Therefore the objective of Masked Token Prediction can be formulated as:

$$J_1(\theta) = logP(\hat{W}|\tilde{W}) = \sum_{i \in \hat{I}} logP(w_i|w_{j_1}, w_{j_2}, ..., w_{j_n}; j_k \in \{I - \hat{I}\}) \quad (1)$$

where $\hat{W}$ and $\tilde{W}$ represent masked tokens and unmasked tokens respectively. Note that in BERT the predictions of masked tokens depend on the context on both directions, which differs from the causal language modeling in GPT where the prediction of the next token only depends on the historical context. This is also different from the language modeling objectives in ELMo. Although ELMo adopts a bidirectional LM, it only makes use of the context from a certain direction (either forward or backward) when predicting a word. The design of Masked Token Prediction allows BERT to model language dependencies bidirectionally by utilizing the information from bidirectional contexts.

Next Sentence Prediction In order to model the dependencies between units larger than words, Devlin et al. (2019b) propose Next Sentence Prediction working with Masked Token Prediction, which concatenates two sentences (sentence A and sentence B), inputs the sequence to BERT model, then predicts whether these sentence B is the sentence following sentence A in the original article. The *positive* examples {A,B} can be taken from articles in the corpus, *negative* examples {A, $\tilde{\text{B}}$} can be created by fixing sentence A and randomly drawing sentence $\tilde{\text{B}}$ from the corpus. The optimization objective of Next Sentence Prediction can be formulated as:

$$J_2(\theta) = \sum_{\{A,B\} \in D} logP(y|A, B) + \sum_{\{A,\tilde{B}\} \in \tilde{D}} logP(1 - y|A, \tilde{B}) \quad (2)$$

where D and $\tilde{D}$ represent the collections of *positive* and *negative* examples respectively, $y \in \{0, 1\}$ is the label for whether B is the next sentence of A. If y is the label for *positive* examples, the label for *negative* examples is $1 - y$.

The overall objective for the optimization of parameters θ of BERT model is $J(\theta) = J_1(\theta) + J_2(\theta)$. In experiments, BERT is firstly pre-trained on BookCorpus (Zhu et al., 2015) and English Wikipedia which contain 800 million and 2500 million words respectively. Then BERT is then transferred to downstream tasks with minor modifications to model architecture - only a few layers need to be added, according to the experimental results in Devlin et al. (2019b). BERT greatly improves the performance on many NLP tasks compared to state-of-the-art approaches, especially on the GLUE benchmark (Wang et al., 2018b) where the improvement is 7.7% absolute points. When employing BERT in downstream tasks, the whole model architecture including the word embedding matrix in the lower layer will be used. That is different from word2vec/GloVe which only transfers the learned static word embeddings to downstream tasks (Mikolov et al., 2013; Pennington et al., 2014).

2.2 Incorporating Structured Knowledge into Pre-trained Language Models

Although the contextualized word representations learned by large-scale pre-trained language models encode rich syntactic and semantic information (Jawahar et al., 2019; Clark et al., 2019a,b; Tenney et al., 2019), they still lack certain knowledge such as world knowledge from knowledge graphs, factual knowledge as well as commonsense knowledge that are crucial for certain tasks especially knowledge-intensive tasks. For example, although models like BERT can capture the co-occurrences among *Apple*, *Tim Cook*, *CEO*, they cannot establish explicit connections that *Tim Cook is the CEO of Apple*. Such knowledge needs to be explicitly injected into pre-trained models Zhang et al. (2019). Also pre-trained models lack factual knowledge. Taking BERT as an example, if we mask *CEO* and substitute *Apple* with *Microsoft* in *Tim Cook is the CEO of Apple*, the resulting sentence is *Tim Cook is the [MASK] of Microsoft*, the masked token predicted by BERT is *CEO* with a high probability. Moreover, pre-trained models lack

commonsense knowledge. For instance they cannot detect that *how many eyes does the Earth have?* is an nonsensical question. Such knowledge cannot be captured through self-supervised pre-training on text corpus, supervisory signals from an external knowledge base are needed for pre-trained language models.

Various approaches have been investigated and employed to incorporate knowledge into PLMs (Sun et al., 2019; Zhang et al., 2019; Peters et al., 2019; Yu et al., 2020; Qiu et al., 2020a; Roy and Pan, 2020; He et al., 2020; Colon-Hernandez et al., 2021; Lyu et al., 2020b; Wang et al., 2021; Wei et al., 2021). Most of them focus on injecting structured knowledge into pre-trained language models. I will present two examples: ERNIE – incorporating entity knowledge in pre-training stage – and K-BERT – injecting domain-specific knowledge information in fine-tuning and inference phase.

ERNIE: Incorporating entity knowledge into language models In order to enrich the text representations with informative entities for better language understanding, Zhang et al. (2019) propose to inject entity information from an external knowledge base into pre-trained language models. In Zhang et al. (2019) the text sequence is alligned with corresponding entities. The proposed model ERNIE comprises a text encoder and a knowledge encoder. The text encoder, which is adopted from BERT, is used to encode the text. The knowledge encoder, which is the proposed key component, is responsible for fusing entity representations and textual representations. The objective of ERNIE is to randomly mask the aligned <*word,entity*> pairs (e.g. by masking <*entity*> there will be no entity information fused into the representations of <*word*>) then train ERNIE model to predict the masked entity based on the fused representations.

In experiments, ERNIE is pre-trained on English Wikipedia containing 4500 million subwords (Johnson et al., 2017; Kudo and Richardson, 2018), the entity embeddings are obtained from Wikidata using TransE (Bordes et al., 2013). The experimental results show that ERNIE outperforms BERT on knowledge-rich tasks including relation classification and entity typing. ERNIE also obtains comparable performance with BERT on other common NLP tasks, demonstrating the efficacy of the knowledge fusion approach.

K-BERT: Injecting knowledge graph into BERT for enhanced language representations Different from (Zhang et al., 2019) where the entity knowledge is injected during the pre-training phase, Liu et al. (2020a) propose to incorporate knowledge in fine-tuning and inference phase by explicitly injecting knowledge graph information into text sequences. Their aim in doing this is to reduce the required computational resources for pre-training and knowledge graph embeddings. K-BERT firstly uses the entity information in sequence $(w_1, w_2, ..., w_n)$ to obtain the relevant $< entity_1, relation, entity_2 >$ triples from a knowledge graph, then these triples are injected into the original sequence directly by appending $< relation, entity_2 >$ to $< entity_1 >$ in the sequence. For examples, if a triple $< Bill\_Gates, CEO_{of}, Microsoft >$ is retrieved for sentence *Bill Gates calls for 'Green industrial revolution' to beat climate crisis*, then K-BERT will inject the triple into the sentence by modifying it to: *Bill Gates CEO of Microsoft calls for 'Green industrial revolution' to beat climate crisis*. It is worth noting that although *CEO of Microsoft* is inserted between *Bill Gates* and *calls for ...*, *CEO of Microsoft* still shares the same position embeddings Vaswani et al. (2017) with *calls for ....* In other words, the injection of *CEO of Microsoft* won't affect the original order of the sentence. After the injection of entity triples, the modified sequence can be fed into the transformer encoders.

Note that the design of K-BERT enables the incorporation of any domain knowledge graph for specific tasks without pre-training and knowledge embeddings since the entity and relation information can be directly injected into the text sequence. Therefore when employing K-BERT for specific tasks, one should use the same pre-training objectives as BERT (Devlin et al., 2019b) or directly initialize the transformer encoders using a public Google BERT model (Devlin et al., 2019b) then use appropriate knowledge graphs in the fine-tuning stage to inject entity and relation information into the text sequences and train K-BERT with task-specific objectives. In the experiments of (Liu et al., 2020a), K-BERT is pre-trained on Chinese corpora including WikiZh and WebtextZh, and the knowledge graphs used in downstream tasks include CN-DBpedia (Xu et al., 2017), HowNet (Dong et al., 2010) and MedicalKG. Experimental

results show K-BERT outperforms vanilla BERT on text classification tasks for the e-commerce domain, XMLI (Conneau et al., 2018) and domain-specific NER.

3 Progress

In this section, I will discuss my current progress and how it relates to the research questions listed in Section 1.

3.1 Improving Document-Level Sentiment Analysis with User and Product Context

RQ1-1 aims to explore how to encode extra information beyond the review texts to improve document-level sentiment analysis. Specifically, I focus on how to utilize the user and product context information (the user and product IDs are known for a review), to improve the accuracy of modeling the sentiment conveyed by a review. I propose an approach explicitly using the historical reviews for a certain user and a product as extra information to help the prediction of review sentiment. I will discuss the motivation for my approach and show the experimental results conducted on benchmark datasets of sentiment analysis including Yelp-2013, Yelp-2014 and IMDB (Tang et al., 2015). This section is based on our paper, *Improving Document-Level Sentiment Analysis with User and Product Context*, published in the COLING 2020 main conference (Lyu et al., 2020a).

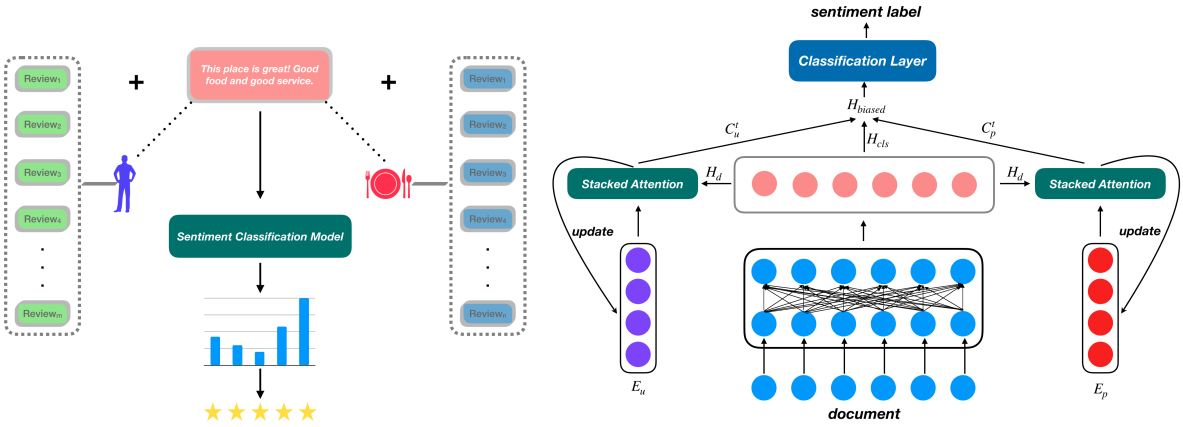

Figure 1: Utilizing all historical reviews of corresponding user and products (left); overall architecture of our model, where E_u and E_p are user and product representations (right).

Motivation Generally, there are two reasons to take user and product context into consideration when predicting the sentiment of a review. Firstly, the reviews from a given user tend to reflect their word uses when conveying sentiment. For example, a typical user might use words like *excellent service* with correspondingly high ratings but another user could use the same words sarcastically with a low rating. Secondly, a group of reviews of the same product could contain terms conveying sentiment that is specific to that product. For example, the historical reviews for a camera product could give hints for what a good/bad camera is and which aspects are important for a camera. These hints may not be useful when evaluating other products such as clothes because word patterns expressing positive sentiment towards cameras do not necessarily convey positive sentiment towards clothes. The idea of using historical reviews is depicted in Figure 1 (left). Earlier work mainly focuses on modeling user and product IDs as embedding vectors whose parameters are learned during training. Such approaches have the shortcoming that the user and product embeddings ,that are only updated by back propagation and learned implicitly, cannot fully capture the word use of a particular user and product, especially when a user or product only has a small number of reviews. Therefore, I propose to directly make use of the textual features

of historical reviews for a user and a product by explicitly *storing* the representations of all historical reviews belonging to a user and product. These historical reviews representations will be produced by a PLM as such representations contain rich syntactic and semantic information (Jawahar et al., 2019; Clark et al., 2019b). I will describe the details of my approach in the next section.

Methodology An overview of the model architecture is shown in Figure 1 (right). The model has two main components: a document (review) encoder and a user-product matrix. With each training sample, I firstly use the document encoder to obtain the textual representations of the review, which are then fed into two stacked attention layers where the corresponding user embedding and product embedding serve as *keys* to generate user-biased document representations and product-biased document representations. Then, I feed the biased document representations into a linear layer followed by a *softmax* layer to get the distribution of the sentiment label. Finally, these two biased document representations are incrementally added to the embeddings of the current user and product review. The updated user and product embeddings are subsequently stored in the user-product matrix as the new user and product representations.

The input to our model consists of $d = \{w_1, w_2, ..., w_{L'}\}, u, p$, which are the document, the user id and the product id respectively. u and p are both projected to embedding vectors, $E_u, E_p \in \mathbf{R^h}$ respectively, where E_u, E_p are transformed from the corresponding entries in the user embedding matrix and product embedding matrix. $d = \{w_1, w_2, ..., w_L\}$ is fed into the BERT encoder to generate a document representation $H_d = \{h_1, h_2, ..., h_L\} \in \mathbf{R^{L \times h}}$ where L is the length of the document after tokenization. We then inject E_u and E_p, to obtain the user-product biased document representation through stacked multi-head-attention (Vaswani et al., 2017)(Q, K, V), where $Q \in R^{L_Q \times h}$, $K \in R^{L_K \times h}$, $V \in R^{L_V \times h}$. Generally, $L_K = L_v$. In our approach, E_u and E_p are regarded as W, H_d as K and V. The user-product biased document representations are computed by:

$$C_u^t = \textbf{stacked-attention}(E_u, H_d, H_d) \quad C_p^t = \textbf{stacked-attention}(E_p, H_d, H_d) \tag{3}$$

where $C_u^t = attention(C_u^{t-1})$, $C_u^0 = E_u$ (similarly for C_p^t), and t is the number of layers of the attention function. In Equation (3), $C_u^t \in R^h, C_p^t \in R^h$.

We then fuse the general textual representation and user-product specific representation into biased document representation:

$$H_{biased} = H_{cls} + z_u \odot C_u^t + z_p \odot C_p^t \tag{4}$$

where $H_{cls} \in \mathbf{R^h}$ is the final hidden vector of the [CLS] token Devlin et al. (2019b) and $\odot$ is element-wise product, z_u and z_p are importance vectors controlling the contribution of user-biased and product-biased representations to sentiment label prediction: $z_u = \sigma(W_{zu}C_u^t + W_{zh}H_d + b_u)$ and $z_p = \sigma(W_{zp}C_p^t + W_{zh}H_d + b_p)$. By doing so $H_{biased} \in \mathbf{R^h}$ is able to capture user and product preferences.

Finally, we feed the biased document representation H_{biased} into a linear layer followed by a *softmax* layer to obtain the distribution of the sentiment label $(p(y|d_i, u_i, p_i)$ for a particular example (d_i, u_i, p_i).

We use *Cross-Entropy* function to calculate the loss between the predictions of our model and ground-truth labels:

$$J(\theta) = -\sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j} log(p(y_{i,j}|d_i, u_i, p_i)) \tag{5}$$

where n is the number of samples and m is the number of all classes, $y_{i,j}$ represents the actual probability of the i-th sample belonging to $class_j$, $y_{i,j}$ is 1 only if the i-th sample belongs to $class_j$ otherwise it's 0. $p(y_{i,j}|d_i, u_i, p_i)$ is the probability the i-th sample belongs to $class_j$ predicted by our model.

We implement the idea of utilizing all historical reviews written by u and all reviews about p by incrementally adding the current user/product-specific document representation C_u^t and C_p^t to the corresponding entries E_u and E_p in the embedding matrix at each step during training:

$$E_u^{'} = \sigma(E_u + \lambda_u C_u^t) \quad E_p^{'} = \sigma(E_p + \lambda_p C_p^t) \tag{6}$$

where $\lambda_u \in R$ and $\lambda_p \in R$ are both learnable parameters controlling the degree to which the representation of the current document should be employed.

After E_u has been updated at every step during the training process, it can *memorize* all reviews attached to the corresponding user, the same for E_p.

Datasets	Classes	Documents	Users	Products	Docs/User	Docs/Product	Words/Doc
IMDB	1–10	84,919	1,310	1,635	64.82	51.94	394.6
Yelp-2013	1–5	78,966	1,631	1,633	48.42	48.36	189.3
Yelp-2014	1–5	231,163	4,818	4,194	47.97	55.11	196.9

Table 1: Statistics of IMDB, Yelp-2013 and Yelp-2014.

Experiments The experiments are conducted on three benchmark datasets of document-level sentiment analysis: Yelp-2013, Yelp-2014 and IMDB (Tang et al., 2015). The statistics of the three datasets are shown in Table 1. The evaluation results are shown in Table 2. Our proposed model is named IUPC (**I**ncorporating **U**ser-**P**roduct **C**ontext). The first two rows are baseline models: BERT VANILLA which is the basic BERT model without user and product information, i.e. only review text, and IUPC W/O UPDATE, which is the same as our proposed model except that we do not update the user and product embedding matrix by incrementally adding the new review representations. We run BERT VANILLA, IUPC W/O UPDATE and IUPC five times and report the average Accuracy and RMSE. The subscripts represent standard deviation. The third row shows our proposed model, the baseline models included in comparison are: CHIM (Amplayo, 2019), CMA (Ma et al., 2017),DUPMN (Long et al., 2018),HCSC (Amplayo et al., 2018), HUAPA (Wu et al., 2018), NSC (Chen et al., 2016b), RRP-UPM (Yuan et al., 2019), UPDMN (Dou, 2017), UPNN (Tang et al., 2015).

	IMDB		Yelp-2013		Yelp-2014	
	Acc. (%)	RMSE	Acc. (%)	RMSE	Acc. (%)	RMSE
BERT VANILLA	$47.9_{0.46}$	$1.243_{0.019}$	$67.2_{0.46}$	$0.647_{0.011}$	$67.5_{0.71}$	$0.621_{0.012}$
IUPC W/O UPDATE	$52.1_{0.31}$	$1.194_{0.010}$	$69.7_{0.37}$	$0.605_{0.007}$	$70.0_{0.29}$	$0.601_{0.007}$
IUPC (our model)	$53.8_{0.57}$	$\mathbf{1.151_{0.013}}$	$\mathbf{70.5_{0.29}}$	$\mathbf{0.589_{0.004}}$	$\mathbf{71.2_{0.26}}$	$\mathbf{0.592_{0.008}}$
UPNN	43.5	1.602	59.6	0.784	60.8	0.764
UPDMN	46.5	1.351	63.9	0.662	61.3	0.720
NSC	53.3	1.281	65.0	0.692	66.7	0.654
CMA	54.0	1.191	66.3	0.677	67.6	0.637
DUPMN	53.9	1.279	66.2	0.667	67.6	0.639
HCSC	54.2	1.213	65.7	0.660	67.6	0.639
HUAPA	55.0	1.185	68.3	0.628	68.6	0.626
CHIM	**56.4**	1.161	67.8	0.641	69.2	0.622
RRP-UPM	56.2	1.174	69.0	0.629	69.1	0.621

Table 2: Experiment results on IMDB, Yelp-2013 and Yelp-2014. Following previous work, we use Accuracy (Acc.) and Root Mean Square Error (RMSE) for evaluation. We show the average results of five runs, the subscripts stand for variance of all runs.

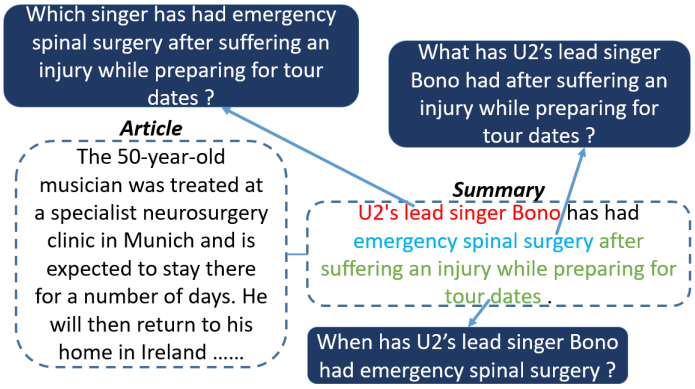

Figure 2: Example questions generated via heuristics informed by semantic role labeling of summary sentences using different candidate answer spans

As observed from Table 2, our model IUPC achieves the best classification accuracy and RMSE on Yelp-2013 and Yelp-2014, and the best RMSE on IMDB, Yelp-2013 and Yelp-2014. IUPC outperforms previous state-of-the-art results by 1.5 accuracy and 0.042 RMSE on Yelp-2013, by 2.1 accuracy and 0.029 RMSE on Yelp-2014, and by 0.01 RMSE on IMDB. Moreover, it outperforms the two baselines, BERT VANILLA and IUPC W/O UPDATE in both classification accuracy and RMSE on all three datasets. The classification accuracy of our model on IMDB is lower than many of the previous models. We suspect this is because the BERT model is not good at handling longer documents since the input length to BERT is fixed and the average length of documents in the IMDB dataset is much longer than the other two datasets, as shown in Table 1. However, it is worth noting that our model achieves the lowest RMSE which means the predictions of our model are *closer* to the gold labels. The empirical results prove the effectiveness of our proposed approach explicitly using historical reviews for a given user and product, addressing **RQ1-1**.

3.2 Improving Unsupervised Question Answering via Summarization-Informed Question Generation

For **RQ1-2**, I focus on incorporating syntactic and semantic knowledge as well as summarisation data to facilitate unsupervised question generation. I propose a summarisation-informed question generation approach utilizing dependency parsing, named-entity recognition and semantic role labeling, which is extrinsically evaluated on unsupervised QA. Experimental results show that my proposed approach outperforms previous unsupervised QA models with fewer QA examples. This section is based on our paper, *Improving Unsupervised Question Answering via Summarization-Informed Question Generation*, published in the EMNLP 2021 main conference (Lyu et al., 2021).

Motivation Question Generation (QG) is a task that aims to generate a plausible question for a given <*passage, answer*> pair. Previous work on QG can be categorized into:

1. *Template-based QG* that uses rules induced from syntactic knowledge to manipulate constituents in a declarative sentence to transform it to an interrogative sentence. Although template-based methods are capable of generating linguistically correct questions, the resulting questions often lack variety and incur high lexical overlap with corresponding declarative sentences. For example, the question generated from the sentence *Stephen Hawking announced the party in the morning*, with *Stephen Hawking* as the candidate answer span, could be *Who announced the party in the morning?*, with a high level of lexical overlap between the generated question and the declarative sentence. This is undesirable in a QA system Hong et al. (2020) since the strong lexical clues in the question would make it a poor test of real comprehension.

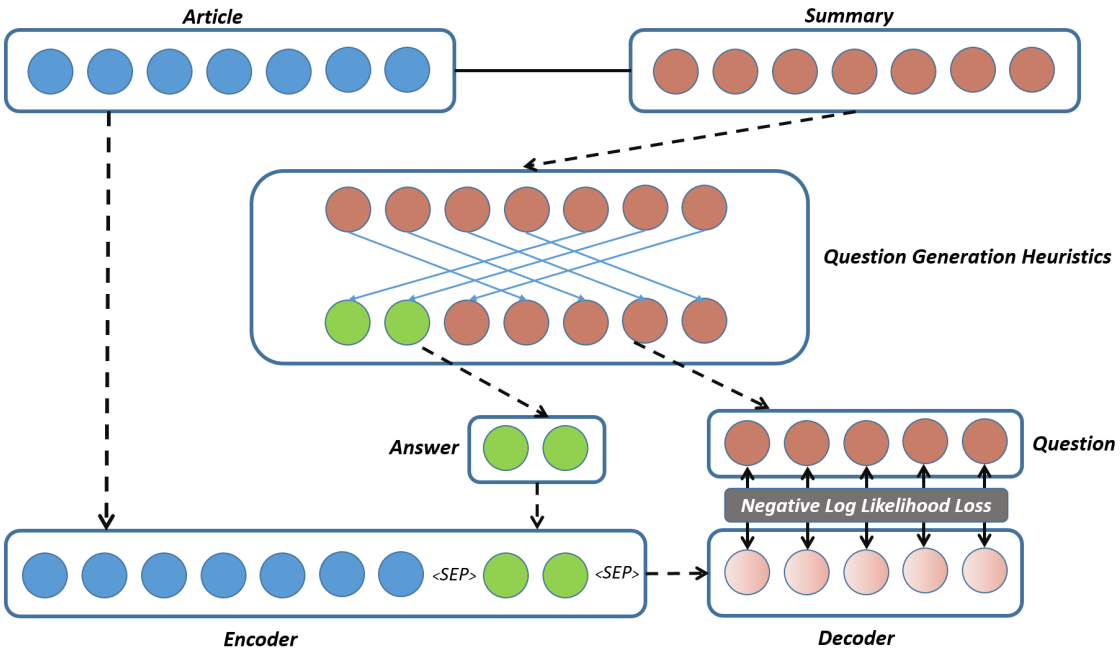

Figure 3: An overview of our approach where *Answer* and *Question* are generated based on *Summary* by the *Question Generation Heuristics*, the *Answer* is combined with the *Article* to form the input to the Encoder, the *Question* is employed as the ground-truth label for the outputs of the Decoder.

2. *Supervised QG* that uses the <*passage, answer, question*> triples in existing QA datasets to train a neural Seq2Seq generation model. This heavily relies on the availability of QA datasets that are costly to obtain, and QA datsets are heavily tied to a certain domain or language where there might not be sufficient QA data to train a QG model.

Methodology In order to overcome the shortcomings mentioned above, we propose an unsupervised QG method that frames question generation as a summarization-questioning process where we employ summarisation data to generate questions from summaries using heuristics based on syntactic and semantic parsing (similar to template-based approaches). Generating questions based on summaries results in questions sharing fewer words with the original passages as summaries are abstractly summarised from passages. An example is shown in Figure 2. The summary is used as a bridge between the questions and passages. Because the questions are generated from the summaries and not from the original passages, they have less of a lexical overlap with the passages. Crucially, however, they remain semantically close to the passages since the summaries by definition contain the most important information contained in the passages. A second advantage of this QG approach is that it does not rely on the existence of a QA dataset, and it is arguably easier to obtain summary data in a given language than equivalent QA data since summary data is created for many purposes (e.g. news, review and thesis summaries) whereas many QA datasets are created specifically for training a QA system. An overview of the proposed summarisation-informed QG is shown in Figure 3. By employing summarisation data, we have <*passage, summary*> pairs. We then parse the summaries. We then apply QG heuristics to the parsed summaries to obtain generated questions and answers. Finally, we pair the generated answers with the original passages to form the input sequence and use the generated questions as output targets for training a neural QG system using *Negative Log Likelihood Loss*.

To parse the summary sentences, we employ three syntactic and semantic analyzers: Dependency Parsing (DP), Named-Entity Recognition (NER) and Semantic Role Labeling (SRL). DP is used to identify the main verb (*root verb* in the dependency tree) as well as other constituents useful in QG. NER is responsible for tagging all entities in the summary sentence to facilitate the generation of the most appropriate question words. SRL is the key component of our QG heuristics. We use SRL to

parse summary sentences to a semantic frame format: *who did what to whom*, where the *did (verb)* is the pivotal component followed by a set of arguments in the sentence. The arguments in a summary sentence are answer candidates used to generate appropriate question words (wh-words). For example *ARG-0* is usually the *Agent* argument in a sentence that initiates the action described by the *verb* in the semantic frame. Its corresponding wh-words can be *what* or *who* depending on the entity information in it, *ARG-TMP* represents temporal arguments corresponding to wh-words *when* and locative argument *ARG-LOC* results in wh-words *where*.

For example, given the sentence *U2's lead singer Bono has had emergency spinal surgery after suffering an injury while preparing for tour dates* and given the semantic frame obtained from SRL: *[U2's lead singer Bono **ARG-0**] has [had **VERB**] [emergency spinal surgery **ARG-1**] [after suffering an injury while preparing for tour dates **ARG-TMP**].*, three questions can be generated based on arguments *ARG-0, ARG-1* and *ARG-TMP*: (i) ***Who** has had emergency spinal surgery after suffering an injury while preparing for tour dates?* (ii) ***What** has U2's lead singer Bono had after suffering an injury while preparing for tour dates?* (iii) ***When** has U2's lead singer Bono had emergency spinal surgery?*

The pseudocode for our algorithm to generate questions is shown in Algorithm 1. We first obtain

Algorithm 1: Question Generation Heuristics

$S = summary, srl\_frames = SRL(S), ners = NER(S), dps = DP(S)$
$examples = []$
for *frame in srl_frames* **do**
  $root\_verb = dps_{root}$
  $verb = frame_{verb}$
  if *root_verb* ***not equal to*** *verb* **then**
    $continue$
  end
  for *arg in frame* **do**
    $wh* = identify\_wh\_word(arg, ners)$
    $base\_verb, auxs = decomp\_verb(arg, dps, root\_verb)$
    $Q_{arg} = wh\_move(S, wh*, base\_verb, auxs)$
    $Q_{arg} = post\_edit(Q_{arg})$
    $examples.append(context, Q_{arg}, arg)$
  end
end

all dependency edges and labels (dps), NER tags ($ners$) and SRL frames (srl_frames) of a summary sentence S. Secondly, we skip the frames in which the $verb$ is not the $root\_verb$ (the verb whose dependency label is $root$). We then iterate through all arguments in the frame of the $root\_verb$ and identify appropriate wh-words ($wh*$) for each argument using the function $identify\_wh\_word$ according to its argument type and its NER tag. We follow Heilman and Smith (2009) in using the standard English wh-words associated with appropriate argument types and NER tags. We then decompose the current main verb to its base form ($base\_verb$) and appropriate auxiliary words ($auxs$) in the $decomp\_verb$ function, before finally inserting the wh-words and the auxiliary verbs in the appropriate positions. We use the wh_move function to move the $wh*$ and $auxs$ to appropriate positions in the question. Finally, we have a set of generated questions associated with the corresponding answer and context passage after some simple post-processing (such as appending a question mark *?*).

Experiments To answer **RQ1-2** (see Section 1.1) and evaluate the effectiveness of our proposed approach, we use extrinsic evaluation – unsupervised QA as the use of traditional metrics such as BLEU Papineni et al. (2002), ROUGE Lin (2004) and Meteor Banerjee and Lavie (2005) has been questioned Callison-Burch et al. (2006); **?**); Ji et al. (2022). We use the summarisation-informed QG system (Lewis et al., 2020) to generate synthetic QA data based on Wikipedia, then we apply the synthetic

	SQuAD1.1		NQ		TriviaQA	
Models	EM	F-1	EM	F-1	EM	F-1
SUPERVISED MODELS						
BERT-base	81.2	88.5	66.1	78.5	65.1	71.2
BERT-large	84.2	91.1	69.7	81.3	67.9	74.8
UNSUPERVISED MODELS						
Lewis et al. (2019)	44.2	54.7	27.5	35.1	19.1	23.8
Li et al. (2020)	62.5	72.6	31.3	48.8	27.4	38.4
Our Method	**65.6**	**74.5**	**46.0**	**53.5**	**36.7**	**43.0**

Table 3: In-domain experimental results of supervised and unsupervised methods on SQuAD1.1, NQ and TriviaQA. The highest scores of unsupervised methods are in bold.

	NewsQA		BioASQ		DuoRC	
	EM	F-1	EM	F-1	EM	F-1
Lewis et al. (2019)	19.6	28.5	18.9	27.0	26.0	32.6
Li et al. (2020)	33.6	46.3	30.3	38.7	32.7	41.1
Our Method	**37.5**	**50.1**	**32.0**	**43.2**	**38.8**	**46.5**

Table 4: Out-of-domain experimental results of unsupervised methods on NewsQA, BioASQ and DuoRC. The results of two baseline models on NewsQA are taken from Li et al. (2020) and their results on BioASQ and DuoRC are from fine-tuning a BERT-large model on their synthetic data.

QA data to unsupervised QA by using it for training a QA system[1] from scratch. We firstly conduct unsupervised QA experiments on three Wikipedia-based QA datasets (in-domain): SQuAD1.1 (Rajpurkar et al., 2016), Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017). The results are shown in Table 3. The baseline models used for comparison are Lewis et al. (2019) and Li et al. (2020). We also include supervised models BERT-base and BERT-large for comparison. The results show that our approach outperforms previous state-of-the-art unsupervised QA models on three in-domain datasets especially on NQ and TriviaQA (more than 10 points improvements on Exact Match score). We further apply our approach on three QA datasets from other domains (out-of-domain): NewsQA (Trischler et al., 2017) BioASQ (Tsatsaronis et al., 2015), DuoRC (Saha et al., 2018). Experimental results are shown in Table 4. Our approach also outperforms other unsupervised QA methods on out-of-domain datasets, suggesting that our approach has better transferability.

Moreover, we conduct additional experiments investigating the effects of the size of synthetic QA data. Results are shown in Figure 4 where we use an increased amount of synthetic QA data to train a QA system and evaluate its performance on NQ and SQuAD1.1. As observed from Figure 4, our method can achieve competitive performance with much less data (with only 20k QA examples), demonstrating the data-efficiency of our method. However with more synthetic QA examples, the performance of QA systems does not improve. We think the reason behind that is, with the increased size of synthetic QA data, models learn too much noise We also evaluate our approach in a few-shot setting where we use a few labeled QA examples. We conduct experiments on NQ and SQuAD1.1. Evaluation results in Figure 5 show that our approach performs better compared to Li et al. (2020) and BERT-large especially when there are only a few labeled QA examples available.

[1]The QA model we used is BERT-large-whole-word-masking. which is referred to as BERT-large in this project

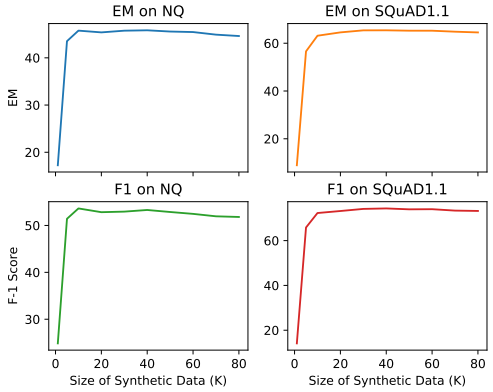

Figure 4: Experimental results on NQ and SQuAD1.1 of using different amount of synthetic data.

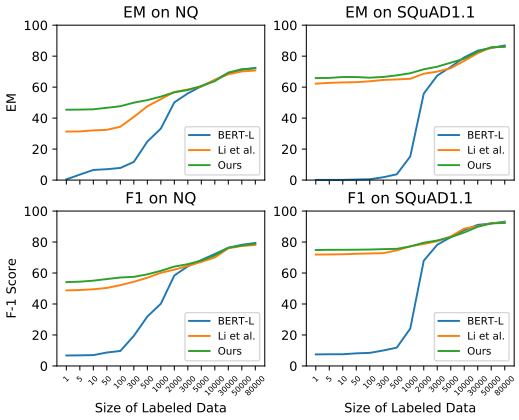

Figure 5: Experimental results of our method with comparison of Li et al. (2020) and BERT-large using different amount of labeled QA examples in the training set of NQ and SQuAD1.1.

3.3 Analysing Extractive Question Answering Data

The goal of **RQ2-1** is to investigate how pre-trained language models learn knowledge from QA datasets. Specifically, I explore how PLMs learn from QA examples of different question types and difficulty levels. Moreover, I examine whether QA models learn real comprehension by perturbing *answers* in QA examples. I will introduce more details of our methodology and experiments in the next sections. This section is partially based on our paper*Extending the Scope of Out-of-Domain: Examining QA models in multiple subdomains* published in ACL 2022 Workshop on Insights from Negative Results in NLP (Lyu et al., 2022).

Introduction There have been several studies analysing QA models and data (Chen et al., 2016a; Kaushik and Lipton, 2018; Weissenborn et al., 2017). Specifically, Jia and Liang (2017) explore the effect of adversarial examples on the performance of QA systems; Lewis et al. (2021); Liu et al. (2021) investigate the train-test data overlap and generalization problems in Open Domain QA and Al-Negheimish et al. (2021) use corrupted QA examples to examine the numerical reasoning ability of current QA systems. However, these studies only focus on a specific aspect of QA data. Following these earlier studies, we aim to explore QA data more broadly and provide deeper insights on how QA systems learn from QA data, especially for a pre-trained model. We conduct three experiments with two English extractive QA datasets:

1. We categorize QA examples by their question type (Li and Roth, 2002) and train a QA system using QA examples from each question type. We find that models learn relatively independently of examples from other question types. In other words, the performance on each question type mainly comes from the data of that same question type;

2. We divide QA examples into *easy* and *difficult* examples according to their context-question lexical overlap. Examples with low overlap are defined as *difficult* examples, examples with high overlap are defined as *easy* examples. We also modify the context in QA examples. Examples with the original full context are defined as *difficult*, examples with *single-sentence* context are defined as *easy*. We then train QA models using *difficult* and *easy* examples respectively and evaluate their performance. We find that more difficult QA examples result in better performance;

3. We examine QA systems on QA examples with perturbed answers and find that QA models do not make enough use of clues from the context, they instead simply learn how to match question and answer from the training QA examples.

Question type	Definition	Examples
HUM	people, individual, group, title	What contemptible scoundrel stole the cork from my lunch ? Which professor sent the first wireless message in the USA ? Who was sentenced to death in February ?
LOC	location, city, country, mountain, state	Where is the Kalahari desert ? Where is the theology library at Notre Dame ? Where was Cretan when he heard screams ?
ENTY	animal, body, color, creation, currency, disease/medical, event, food, instrument, language, plant, product, religion, sport, symbol, technique, term, vehicle	What relative of the racoon is sometimes known as the cat-bear ? What is the world's oldest monographic music competition ? What was the name of the film about Jack Kevorkian ?
DESC	definition, description, manner, reason	What is Eagle 's syndrome styloid process ? How did Beyonce describe herself as a feminist ? What are suspects blamed for ?
NUM	code, count, date, distance, money, order, other, percent, period, speed, temperature, size, weight	How many calories are there in a Big Mac ? What year did Nintendo announce a new Legend of Zelda was in the works for Gamecube ? How many tons of cereal did Kelloggs donate ?

Table 5: Definition of each question type (Zhang and Lee, 2003) and corresponding examples in SQuAD1.1 and NewsQA.

Experiments We employ two benchmark English extractive QA datasets SQuAD1.1 (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017) and question classification data (Li and Roth, 2002)[2], with the BERT-base-uncased model from Huggingface (Wolf et al., 2020)[3] for both question classification and QA.

Experiment 1: How QA models learn from different question types In Experiment 1, we aim to investigate how QA models learn from QA examples with different question types. We adopt question classification data (Li and Roth, 2002) to train a question classifier that categorizes questions into the following five classes: *HUM, LOC, ENTY, DESC, NUM* (Zhang and Lee, 2003). Definitions and examples of each question type are shown in Table 5. The QA examples in training data are then partitioned into five categories according to their question type. Question type proportions for SQuAD1.1 and NewsQA are shown in Table 6, with a high proportion of *ENTY* and *NUM* questions in SQuAD1.1, while NewsQA has more *HUM* and *DESC* questions. We use QA examples of each question type to train a QA system with increasing data size from 500 to 8000 with intervals of 500 and evaluate it on the test data, which is also divided into five categories according to question type. The results are shown in Figure 6. We find that a QA system learns to answer a certain type of questions mainly from the QA examples of the same question type – this is more obvious for *HUM* and *NUM* questions in SQuAD1.1 and *HUM*, *LOC* and *NUM* questions in NewsQA. For example, in Figure 6, with an increased amount of *NUM* training examples, the performance of a QA system on *NUM* test examples substantially improves, whereas the performance on the other question types (such as *LOC*) obtains only small improvements. In other words, the knowledge to answer *NUM* questions for a QA system almost exclusively comes from *NUM* training examples.

Experiment 2: How a QA model learns from *difficult* and *easy* examples? In order to analyse how a QA model learns from examples with different degrees of difficulty, we firstly define difficulty as the

[2]https://cogcomp.seas.upenn.edu/Data/QA/QC/

[3]https://huggingface.co/bert-base-uncased

		LOC	ENTY	HUM	NUM	DESC
SQuAD1.1	Train set	11.4	27.6	20.7	24.5	15.5
	Dev set	10.5	27.6	21.0	23.0	17.4
NewsQA	Train set	11.4	16.9	30.0	18.8	22.6
	Dev set	12.3	16.9	32.2	17.8	20.5

Table 6: The percentage of question types in the SQuAD1.1 and NewsQA train and dev sets.

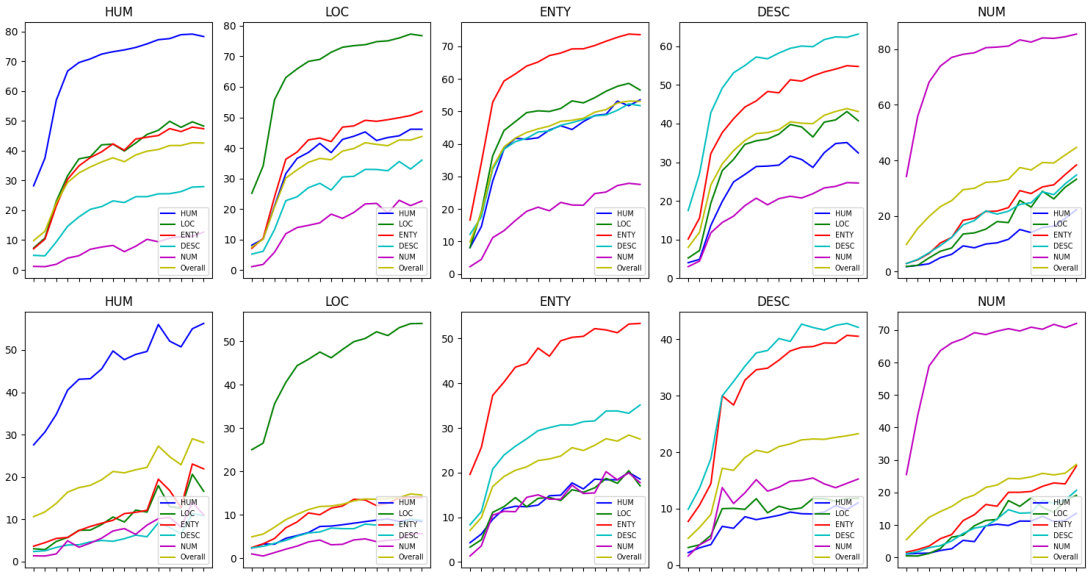

Figure 6: Visualization of F-1 learning curves for five QA systems trained on five question types (*HUM,LOC,ENTY,DESC,NUM*), tested on the dev sets for each question type and the original dev set. SQuAD1.1 (top) and NewsQA (bottom)

lexical overlap between the context and question in each QA example. The QA examples with high lexical overlap are defined as *easy* examples as too many lexical cues would make it easier for a QA system to learn (Hong et al., 2020). Secondly, we modify the *context* in QA examples to *single-sentence context* which means the context only keeps the sentence in the original context which contains the answer. Examples with *single-sentence context* are defined as *easy* examples as a shorter context makes it easier to locate the correct answer, whereas examples with the *original context* are defined as *difficult*.

Then we train QA models on *difficult* and *easy* examples separately and subsequently evaluate the trained models on *difficult* examples and *easy* examples respectively. The context-question lexical overlap results on SQuAD1.1 and NewsQA are shown in Figure 7. With the same amount of data, the QA system trained on QA examples with less context-question overlap (*less-overlap system*) can always yields better performance (F-1 score) compared to the QA system trained on QA examples with more context-question overlap (*more-overlap system*). Specifically, the *less-overlap system* is able to perform well on questions with more context-question overlap, whereas the *more-overlap system* can't achieve comparable performance on questions with less context-question overlap – this is even more apparent on the NewsQA dataset.

We show the experimental results of *single-sentence context* and *original context* on SQuAD1.1 and NewsQA in Table 7. The performance on easy (*single-sentence*) test data is always better than the performance on the difficult (*original*) test data.

Experiment 3: Question-answer match Generally speaking, an ideal QA system is expected to be able to find the correct answers using clues from the context via comprehension rather than using clues from the answers alone with shortcuts. Therefore, we propose to corrupt the answers in QA examples. By

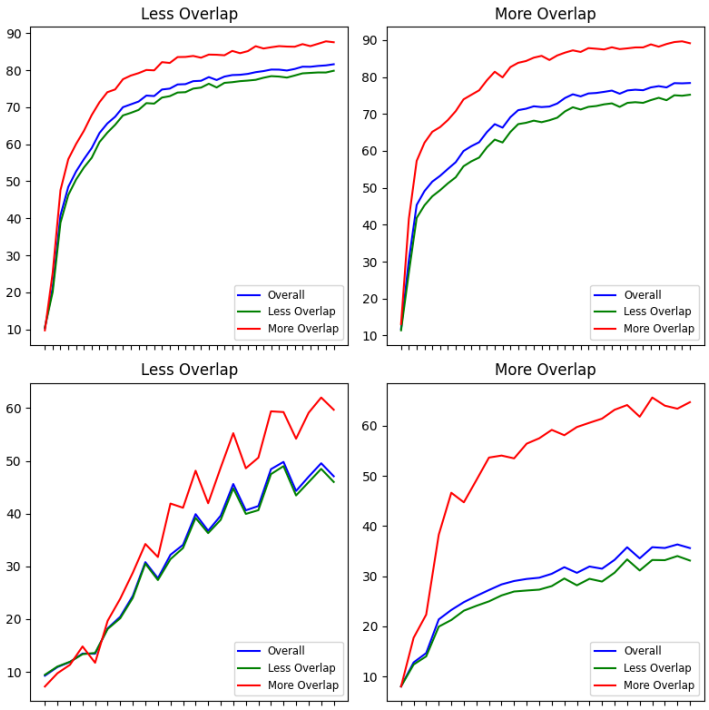

Figure 7: Visualization of F-1 score change over different lexical overlap levels and overall dev set with increased data size on *Less Overlap* and *More Overlap* SQuAD1.1 (top) and NewsQA (bottom)

		Dev-original	Dev-single-sent
SQuAD1.1	Train-original	80.61/88.25	81.75/89.50
	Train-single-sent	75.61/83.64	81.49/89.34
NewsQA	Train-original	49.55/64.53	60.51/79.18
	Train-single-sent	36.39/50.00	62.73/80.85

Table 7: Evaluation results (EM/F-1) of single-sentence context and original context QA examples on SQuAD1.1 and NewsQA.

doing so we can examine the degree to which QA models are able to make use of clues from the context. We propose a simple strategy to perturb/corrupt answers in training QA examples: *random tokens*, i.e. randomly generate meaningless tokens to replace the original answers.

Such perturbed QA examples are answerable for humans, for example below:

Context: *Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion* ***jysbdefziqvzbi*** *defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title.*

Question: *Which NFL team won Super Bowl 50?*

Original answer: *Denver Broncos*

Corrupted/correct answer: *jysbdefziqvzbi*

Humans can easily find the correct answer - ***jysbdefziqvzbi*** even if is a meaningless word. The goal of this experiment is to examine whether a QA system is capable of finding such corrupted correct answers.

We perturb the answers in the development sets of SQuAD1.1 and NewsQA, then we evaluate the QA models trained on the original training sets on these perturbed test examples. The average results of three runs on SQuAD1.1 and NewsQA are shown in Table 8. The results reveal that corrupting the semantic information of answer text causes a substantial performance drop (maximum ~25% F-1 score drop for SQuAD1.1 and ~50% F-1 score drop for NewsQA) – the drop is even larger for NewsQA (~30 F-1 score

		Overall
SQuAD1.1	Original	66.97/80.96
	Random tokens	55.99/61.40
NewsQA	Original	49.22/64.53
	Random tokens	31.72/35.91

Table 8: Evaluation results (EM/F-1) on dev sets of SQuAD1.1 and NewsQA with corrupted answers

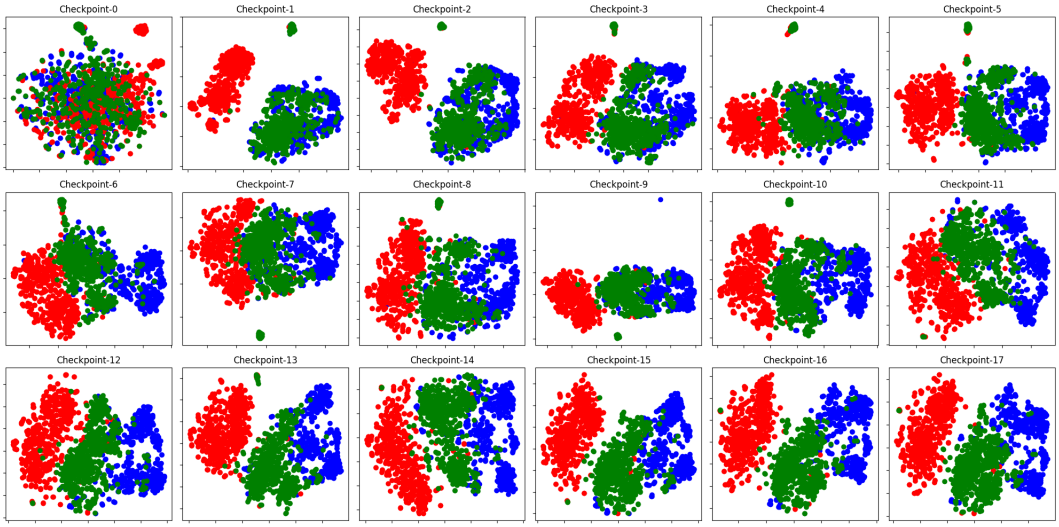

Figure 8: t-SNE visualization of randomly sampled *Answer, Question* and *Context* wordpiece representations from 18 checkpoints in training process starting from checkpoint-0 (the vanilla BERT) to checkpoint-17 (the BERT finetuned on SQuAD1.1), where *Answer* is in *blue*, *Question* is in *red*, *Context* is in *green*.

drop), demonstrating that QA models fail to make enough use of clues from the context, showing a lack of comprehension of the context and question.

Moreover, to further gain insights into the representations learned by the QA systems, we randomly sampled 500 context-question-answer Wordpiece Johnson et al. (2017) representations from 18 checkpoints of the BERT model during the fine-tuning process on SQuAD1.1 and use t-SNE van der Maaten and Hinton (2008) to visualize these representations in Figure 8. The visualization clearly shows the learning process of the QA system:

1. the representations of questions (red) are differentiated from the representations of context (green) and answers (blue).

2. as the fine-tuning process continues, the representations of context and answer are gradually separated.

References

Adams, O., Makarucha, A., Neubig, G., Bird, S., and Cohn, T. (2017). Cross-lingual word embeddings for low-resource language modeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 937–947.

Al-Negheimish, H., Madhyastha, P., and Russo, A. (2021). Numerical reasoning in machine reading comprehension tasks: are we there yet? In *Proceedings of the 2021 Conference on Empirical Methods*

in Natural Language Processing, pages 9643–9649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Amplayo, R. K. (2019). Rethinking attribute representation and injection for sentiment classification. 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 5602–5613, Hong Kong, China. Association for Computational Linguistics.

Amplayo, R. K., Kim, J., Sung, S., and Hwang, S.-w. (2018). Cold-start aware user and product attention for sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2535–2544, Melbourne, Australia. Association for Computational Linguistics.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Bender, E. M. and Friedman, B. (2018). Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. (2021). On the opportunities and risks of foundation models.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Chen, D., Bolton, J., and Manning, C. D. (2016a). A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.

Chen, H., Sun, M., Tu, C., Lin, Y., and Liu, Z. (2016b). Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659, Austin, Texas. Association for Computational Linguistics.

Chen, Y., Wu, L., and Zaki, M. J. (2019). Reinforcement learning based graph-to-sequence model for natural question generation. In *International Conference on Learning Representations*.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019a). What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019b). What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Colon-Hernandez, P., Havasi, C., Alonso, J., Huggins, M., and Breazeal, C. (2021). Combining pre-trained language models and structured knowledge. *arXiv preprint arXiv:2101.12294*.

Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 32:7059–7069.

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S., Schwenk, H., and Stoyanov, V. (2018). XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Da, J. and Kasai, J. (2019). Cracking the contextual commonsense code: Understanding commonsense reasoning aptitude of deep contextual representations. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 1–12.

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

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

Dong, Z., Dong, Q., and Hao, C. (2010). HowNet and its computation of meaning. In *Coling 2010: Demonstrations*, pages 53–56, Beijing, China. Coling 2010 Organizing Committee.

Dou, Z.-Y. (2017). Capturing user and product information for document level sentiment analysis with deep memory network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 521–526, Copenhagen, Denmark. Association for Computational Linguistics.

Du, X., Shao, J., and Cardie, C. (2017). Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Duan, N., Tang, D., Chen, P., and Zhou, M. (2017). Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.

Gardner, M., Merrill, W., Dodge, J., Peters, M., Ross, A., Singh, S., and Smith, N. A. (2021). Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Geva, M., Goldberg, Y., and Berant, J. (2019). Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Graesser, A. C., Chipman, P., Haynes, B. C., and Olney, A. (2005). Autotutor: an intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618.

Gu, J., Hassan, H., Devlin, J., and Li, V. O. (2018). Universal neural machine translation for extremely low resource languages. In *NAACL-HLT*.

He, B., Zhou, D., Xiao, J., Jiang, X., Liu, Q., Yuan, N. J., and Xu, T. (2020). BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2281–2290, Online. Association for Computational Linguistics.

Heilman, M. and Smith, N. A. (2009). Question generation via overgenerating transformations and ranking. Technical report, Carnegie-Mellon University.

Heilman, M. and Smith, N. A. (2010). Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, Los Angeles, California. Association for Computational Linguistics.

Hong, G., Kang, J., Lim, D., and Myaeng, S.-H. (2020). Handling anomalies of synthetic questions in unsupervised question answering. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3441–3448, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jawahar, G., Sagot, B., and Seddah, D. (2019). What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Ji, T., Graham, Y., Jones, G. J., Lyu, C., and Liu, Q. (2022). Achieving reliable human assessment of open-domain dialogue systems. *arXiv preprint arXiv:2203.05899*.

Ji, T., Lyu, C., Cao, Z., and Cheng, P. (2021). Multi-hop question generation using hierarchical encoding-decoding and context switch mechanism. *Entropy*, 23(11):1449.

Jia, R. and Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Joshi, M., Choi, E., Weld, D., and Zettlemoyer, L. (2017). TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Kaushik, D. and Lipton, Z. C. (2018). How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Kondratyuk, D. and Straka, M. (2019). 75 languages, 1 model: Parsing Universal Dependencies universally. 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 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., and Petrov, S. (2019). Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). 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. Association for Computational Linguistics.

Lewis, P., Denoyer, L., and Riedel, S. (2019). Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

Lewis, P., Stenetorp, P., and Riedel, S. (2021). Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Li, X. and Roth, D. (2002). Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Li, Z., Wang, W., Dong, L., Wei, F., and Xu, K. (2020). Harvesting and refining question-answer pairs for unsupervised QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6719–6728, Online. Association for Computational Linguistics.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Liu, L., Lewis, P., Riedel, S., and Stenetorp, P. (2021). Challenges in generalization in open domain question answering. *arXiv e-prints*, pages arXiv–2109.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., and Wang, P. (2020a). K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020b). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Long, Y., Ma, M., Lu, Q., Xiang, R., and Huang, C.-R. (2018). Dual memory network model for biased product review classification. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 140–148, Brussels, Belgium. Association for Computational Linguistics.

Lyu, C., Foster, J., and Graham, Y. (2020a). Improving document-level sentiment analysis with user and product context. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6724–6729, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lyu, C., Foster, J., and Graham, Y. (2022). Extending the scope of out-of-domain: Examining qa models in multiple subdomains. *arXiv preprint arXiv:2204.04534*.

Lyu, C., Ji, T., and Graham, Y. (2020b). Incorporating context and knowledge for better sentiment analysis of narrative text. In Campos, R., Jorge, A. M., Jatowt, A., and Bhatia, S., editors, *Proceedings of Text2Story - Third Workshop on Narrative Extraction From Texts co-located with 42nd European Conference on Information Retrieval, Text2Story@ECIR 2020, Lisbon, Portugal, April 14th, 2020 [online only]*, volume 2593 of *CEUR Workshop Proceedings*, pages 39–45. CEUR-WS.org.

Lyu, C., Shang, L., Graham, Y., Foster, J., Jiang, X., and Liu, Q. (2021). Improving unsupervised question answering via summarization-informed question generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4134–4148.

Ma, D., Li, S., Zhang, X., Wang, H., and Sun, X. (2017). Cascading multiway attentions for document-level sentiment classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 634–643, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Ma, X., Zhu, Q., Zhou, Y., and Li, X. (2020). Improving question generation with sentence-level semantic matching and answer position inferring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8464–8471.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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

Peters, M. E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., and Smith, N. A. (2019). Knowledge enhanced contextual word representations. 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 43–54.

Puri, R., Spring, R., Shoeybi, M., Patwary, M., and Catanzaro, B. (2020). Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. Association for Computational Linguistics.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. (2020a). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. (2020b). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Rogers, A. (2021). Changing the world by changing the data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2182–2194, Online. Association for Computational Linguistics.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Roy, A. and Pan, S. (2020). Incorporating extra knowledge to enhance word embedding. In *IJCAI*, pages 4929–4935.

Saha, A., Aralikatte, R., Khapra, M. M., and Sankaranarayanan, K. (2018). DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693, Melbourne, Australia. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Søgaard, A., Ebert, S., Bastings, J., and Filippova, K. (2021). We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832, Online. Association for Computational Linguistics.

Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., and Wu, H. (2019). Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Tang, D., Qin, B., and Liu, T. (2015). Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1014–1023, Beijing, China. Association for Computational Linguistics.

Tenney, I., Das, D., and Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., and Suleman, K. (2017). NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Tsatsaronis, G., Balikas, G., Malakasiotis, P., Partalas, I., Zschunke, M., Alvers, M. R., Weissenborn, D., Krithara, A., Petridis, S., Polychronopoulos, D., et al. (2015). An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):1–28.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018a). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018b). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., and Tang, J. (2021). Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Wei, X., Wang, S., Zhang, D., Bhatia, P., and Arnold, A. (2021). Knowledge enhanced pretrained language models: A compreshensive survey.

Weissenborn, D., Wiese, G., and Seiffe, L. (2017). Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, Vancouver, Canada. Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wu, Z., Dai, X.-Y., Yin, C., Huang, S., and Chen, J. (2018). Improving review representations with user attention and product attention for sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Xie, Y., Pan, L., Wang, D., Kan, M.-Y., and Feng, Y. (2020). Exploring question-specific rewards for generating deep questions. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2534–2546, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xu, B., Xu, Y., Liang, J., Xie, C., Liang, B., Cui, W., and Xiao, Y. (2017). Cn-dbpedia: A never-ending chinese knowledge extraction system. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 428–438. Springer.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Yu, D., Zhu, C., Yang, Y., and Zeng, M. (2020). Jaket: Joint pre-training of knowledge graph and language understanding. *arXiv preprint arXiv:2010.00796*.

Yuan, Z., Wu, F., Liu, J., Wu, C., Huang, Y., and Xie, X. (2019). Neural review rating prediction with user and product memory. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7,2019*, pages 2341–2344.

Zhang, D. and Lee, W. S. (2003). Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 26–32.

Zhang, S. and Bansal, M. (2019). Addressing semantic drift in question generation for semi-supervised question answering. 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 2495–2509, Hong Kong, China. Association for Computational Linguistics.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., and Liu, Q. (2019). ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Zhang, Z., Zhao, H., and Wang, R. (2020). Machine reading comprehension: The role of contextualized language models and beyond.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). 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.