Information Retrieval & Question Answering in Aviation Safety Domain

Raj Gite and **Pushpak Bhattacharyya**
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
{rajgite, pb}@cse.iitb.ac.in

Abstract

In today's digital world, a tremendous amount of data is generated. Businesses use this data to model their operation strategies. If data is utilized effectively, then it is gold for these businesses. Information Retrieval and Question Answering provide correct information at the right time, helping businesses analyze a vast collection of data effectively. Thus Information Retrieval and Question Answering have significant demand all over the industry.

Like other industries, in the commercial aviation domain, there is massive data in the form of aircraft accident reports, aircraft maintenance manuals, and safety notifications. They serve the aviation industry's safety, maintenance, and compliance demands. A Question Answering System in the aviation domain would help serve these above demands more effectively. We present a Question Answering System using Deep Learning for the aviation domain. Two Question Answering paradigms, namely abstractive and extractive, are explored. A Question Answering system adopting both abstractive and extractive models is implemented to handle all four types of questions: factoid, confirmation, list and descriptive.

In this paper, we present the literature review done as a part to build the Question Answering System for aviation domain. We extensive try to cover Information Retrieval and Question Answering. In Information Retrieval, we concentrate on dense retrieval, which uses semantic features to retrieve passages relevant to a question. A wide range of techniques is covered, grouped into two categories: transformer models that perform re-ranking in multi-stage architectures and dense retrieval techniques that perform ranking directly. In Question Answering, we present two approaches: Deep Learning based QA (DLQA) and KG guided DL based QA. In DLQA, we discuss techniques falling into two categories: open-book and closed-book approaches. In open-book, we first access the relevant passage/document and then fetch an answer from the relevant passage/document. In close-book, we fetch an answer without retrieving the relevant passage/document. QA models lying under close-book approach store the corpus information in their parameters, making them very large. We can not use a close book model trained on corpus A directly on corpus B. Due to these two reasons, close-book approaches are not that popular. In KG guided DL based QA, we see several techniques and focus mainly on the recent work (KG based Text-enhanced QA). We also see two preliminary QA systems built for the aviation domain.

1 Problem Definition

In the aviation domain, there is a need to access an extensive collection of documents. These documents include accident reports, maintenance manuals, and safety notifications. Currently, these documents are not fully utilized due to poor access mechanisms like manual look-up or string-matching based search. Underutilization of these documents compromises safety in the aviation domain. A Question Answering system can solve this problem by providing correct information with very little latency. We call such a system an Aviation Question Answering system. Our goal is to develop a system that performs Question Answering over text using Deep Learning. We call this system Deep Learning based Question Answering (DLQA). Given a **question**, DLQA produces an **answer** and a **list of relevant passages** from where the answer is derived.

DLQA is a subsystem of the Aviation Question Answering system. Aviation QA also has another subsystem called Knowledge Graph based Question Answering (KGQA). These two subsystems operate in parallel to answer questions in the aviation domain. Unlike DLQA, KGQA performs QA over Knowledge Graph. Figure 1 shows the architecture of the Aviation QA system.

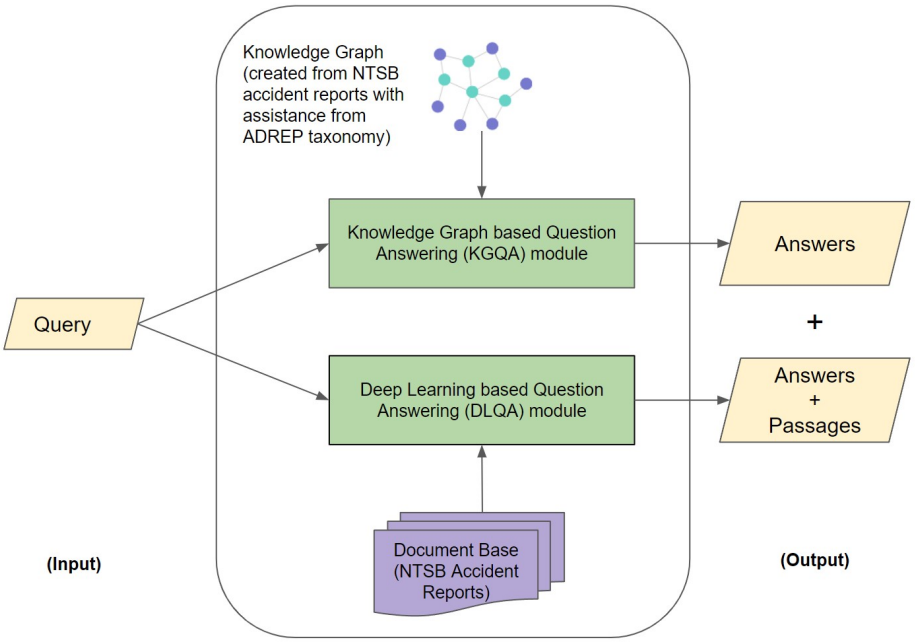

Figure 1: Architecture of Aviation Question Answering System. KGQA and DLQA are the subcomponents of the Aviation QA

2 Motivation

In the commercial aviation domain, safety is of prime importance. Time and efforts of many people are involved to ensure the safety of passengers and aircraft. These include aircraft accident investigators, aircraft maintenance staff and a safety compliance team. The nature of the job of the first two people is quite clear from their titles, but it is not for the safety compliance team. The safety compliance team regulates the aircraft to comply with the unsafe conditions notified by the regional airspace authority. We see how exactly these people contribute to safety in the aviation domain.

When an accident happens, investigators prepare an accident report to collect all the information about the accident. The details range from the pilot's flying experience to the surrounding environmental conditions during the accident. Mainly, the cause of the accident is recorded with minute details as it is a primary factor in deciding the preventive steps. The preventive steps may be changes in the design of an aircraft's component or changes in the operating procedures. Investigators formulate preventive measures by analyzing all these accident reports. One of the most straightforward analyses is to check if there are accident reports similar to a recent report and revisit them to formulate a measure that will prevent accidents of the same nature in the future. Usually, investigators decide on designing a preventive measure only if they find similar accident reports. A miss in finding an old report similar to the recent report will not provide enough confidence to investigators to act on the recent report, leaving chances of a similar accident in the future. Thus, aircraft accident investigators critically analyze the accident reports to prevent accidents of the same nature in the future and contribute to aviation safety.

During daily safety checks of an aircraft before its next take-off, aircraft maintenance staff need to resolve problems prompted by aircraft components or conveyed by the flight crew. They need to take desired actions to resolve the underlying problem, for which they must go through an extensive collection of aircraft maintenance manuals. These manuals enlist solutions to all kinds of problems for avionics, making them very technical and large in volume. Maintenance staff search these manuals and apply the enlisted solution. Searching for desired information in these manuals is challenging because of their large size and less time available. An undesired solution would compromise safety but also add unnecessary maintenance work. The simplest example is the swapping of a working part. The working part then goes to its Original Equipment Manufacturer (OEM). There it is thoroughly inspected and is found to be working correctly. Then it is again bought into function. All

this involves time, effort and money, and we do not get safety. Thus, it is crucial for aircraft maintenance staff to appropriately refer to the maintenance manuals to fetch the desired solution to the problem. In this way, the aircraft maintenance staff ensures an aircraft is safe to proceed with its next in-air operation.

Aircraft accident investigators and aircraft maintenance staff often encounter unsafe conditions in an aircraft during their daily job routine. They inform the regional airspace authority of this unsafe condition to make all aircraft operators aware of it. The airspace authorities regularly notify all the aircraft operators about such unsafe conditions along with required compliance actions to ensure the safety of the aircraft. Now, it is on the aircraft operators to keep track of such notifications to ensure the safety of their aircraft and avoid fines for missing on complying with these notifications. The aircraft operators have a group of people to perform this task, called the safety compliance team. This team needs to analyze the regulatory notifications, which are very large in numbers, to ensure all aircraft follow the safety standards. The aircraft operators are penalized with a hefty fine for missing a safety notification as it is a matter of aviation safety. Thus, the safety compliance team must be careful not to miss any aircraft from complying with any safety notification.

It is evident that to ensure safety in the aviation domain, there is a need to access an extensive collection of documents. The accident investigators analyze aircraft accident reports to formulate preventive measures (updations in operating procedures or improvements in the design of specific aircraft parts). The maintenance staff search aircraft maintenance manuals to resolve problems encountered during safety checks. Lastly, the safety compliance team tracks safety notifications to ensure all aircraft follow safety standards. There is a compromise in safety because, currently, aviation documents are not fully utilized due to poor access mechanisms like manual look-up or string-matching-based search. A QA system will: enhance aircraft safety by providing desired information with very little latency to the people involved in ensuring aircraft safety, reduce the time and efforts of accident investigators, aircraft maintenance staff, and safety compliance team to access documents and cut maintenance costs by preventing unnecessary repair which used to happen because of unavailability of time and information.

3 Literature Survey

In this section, we review and synthesize past work done in Information Retrieval and Question Answering. We begin with Information Retrieval. Information Retrieval techniques can be grouped into two categories: transformer models that perform re-ranking in multi-stage architectures and dense retrieval techniques that perform ranking directly. We will see a few techniques from each category. Question Answering techniques can also be categorized into two classes: open-book and close-book. Open-book techniques are based on retriever-reader design. We will see a wide range of design ideas for retrievers and readers. Close book techniques are retriever-free and we will see how large Seq2Seq language models are adopted for Question Answering. At last, we see two QA systems built for the aviation domain.

3.1 Information Retrieval

This section provides an overview of neural network architectures, specifically transformers developed for Information Retrieval. The combination of self-supervised pre-training and transformer architecture has been responsible for shifting the paradigm in IR. The transformer-based IR models produce promising results across many domains and are mainly categorized into two groups: re-ranking in multi-stage architecture and direct ranking (Yates et al., 2021). Relevance classification, document and query expansion, and evidence aggregation from multiple text segments are the approaches lying in the former category. The latter techniques use variants of the transformer to learn the dense representation of texts where the ranking is done by comparing query and document representation taking advantage of nearest neighbor search.

3.1.1 Re-ranking in Multistage Architectures

Text re-ranking can be formulated as a text classification problem, where the document text are classified into two categories: relevant and irrelevant. The document rank list can be computed by ordering the documents in descending order of their probability of lying in the relevant class. Nogueira et al. (2019) present mono-BERT, a simple and effective model for relevance classification.

3.1.2 Direct ranking using Dense Representations

The biggest revolution in IR is moving away from sparse features, mostly limited to exact matches, to continuous denser representations for query and text that can capture the semantics of the text. Text mapping into a semantic vector space solves the vocabulary mismatch problem faced by IR models using sparse features. The relevance scorer for each document is computed using similarity functions like dot product, euclidean distance, etc., and then documents are ordered in descending order of relevance. There are two approaches for direct dense retrieval:

- Single vector representations for text obtained from simple bi-encoders and ranking based on simple comparison operators such as dot product.
  Eg: Sentence-BERT (Reimers and Gurevych, 2019), DPR (Karpukhin et al., 2020), and ANCE (Xiong et al., 2020).

- Multiple vector representations for text obtained from enhanced bi-encoders and ranking based on complex comparison operators.
  Eg: ColBERT (Khattab and Zaharia, 2020), ME-BERT (Luan et al., 2020), and Poly-encoders (Humeau et al., 2019)

3.2 Deep Learning based Question Answering

Question Answering involves answering questions using an information source. The information source can be structured knowledge bases or unstructured text. In this section we see Question Answering over text. There are wide range of deep learning techniques used to tackle Question Answering over text. Thus, we can also call this approach of QA as Deep Learning based Question Answering. The textual information source can be a collection of passages or documents. Depending on how the textual information source is used, Question Answering techniques can be categorized into two classes: open-book and close-book. Consider the information source as a book. The pages of the book can be either passages or documents. In open-book, to answer a question, we find the relevant pages (passages/documents) and then fetch answers from those pages. The book remains open and for each question we are allowed to use it. So, the open-book approaches perform two steps: retrieval and reading. In close-book, the answer to the question is generated using whatever the model had remembered when it was given a chance to open the book. The book is opened just once and when questions are asked, the book is closed. So, the close-book approach directly answers the question without opening the book at inference time. This section discusses a wide variety of open-book and close-book techniques.

3.2.1 Open-book Approach

Retriever is usually regarded as an IR system. The job of Retriever is to perform Information Retrieval, which is to retrieve relevant passages/documents to a given query in natural language. It involves two steps: 1) mapping query and passages into a semantic vectors space such that the relevance score for relevant passages is high and that of irrelevant passages is low, and 2) computing top-k relevant passages by ordering the passages in the corpus in the descending order of relevance score. There are three types of Retrievers depending on the design used for encoding questions and documents: Representation-based Retriever, Interaction-based Retriever, and Representation-interaction Retriever, as illustrated in Figure 2.

Representation-based Retriever: A dual-encoder is used in Representation-based Retriever, which involves two independent encoders like BERT (Devlin et al., 2018) to encode question and document, respectively. A single similarity score is computed using the two representations for estimating relevance. ORQA (Lee et al., 2019) adopts a Representation-based Retriever with two independent BERT-based encoders to encode a question and a document, respectively. The dot product of query and document representations is used to compute the relevance score. They also pre-train the retriever using Inverse Cloze Task (given a sentence, predict its context). DPR (Karpukhin et al., 2020) also employs a dual-encoder design like ORQA. Instead of expensive pre-training, DPR uses an objective function specially designed for semantic search. Tuples containing a query, a positive passage and a few negative passages are required for training. The datasets only provide query and positive passage pair. Thus to get negative passages, three mechanisms are used: random selections from the corpus, top documents returned by BM25 that do not contain answer, and in-batch sampling where the positive passages of other training instances of the same batch are used to fetch negative passages using the above two mechanisms. It

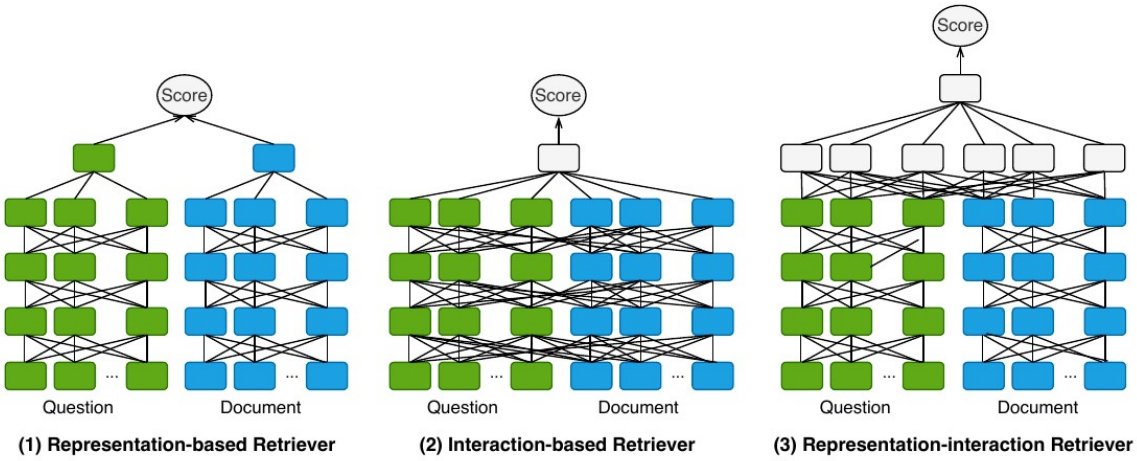

Figure 2: Three types of retrievers

is important to mention that their experiments show that the dot product function is optimal for calculating the relevance score for a dual-encoder retriever. Representation-based method (Karpukhin et al., 2020; Guu et al., 2020; Lee et al., 2019) can be very fast because the representation of documents can be pre-computed and indexed ahead of time. The effectiveness of Representation-based Retriever is sacrificed because the query and document representations are obtained independently, leading to only shallow interactions between them.

Interaction-based Retriever: Interaction-based Retriever gives query and document together as an input to the network to model rich token-wise interactions between the query and the document using a transformer-based encoder (Devlin et al., 2018; Vaswani et al., 2017), thus powerful. Nishida et al. (2018) propose jointly training Retriever and Reader using supervised multi-task learning (Seo et al., 2016). Nishida et al. (2018) add a retrieval layer to compute the relevance score between question and document based on BiDAF and use a comprehension layer to predict the answer span's start and end index. Nie et al. (2019) develop two dense retrievers: one for passage and the other for sentence, both based on BERT. They model dense retrieval as a binary classification problem where they take each pair of question and document as input and use the representation of [CLS] token to determine the relevance. They emphasize the requirement of both passage-level and sentence-level retrieval for good performance. Rich token-wise interactions make the Interaction-based method powerful but at the cost of less efficiency. Such a method requires heavy computation at the inference time, making it unsuitable for large document collection.

Representation-interaction Retriever: In order to get the best from both worlds, Representation-based Retrievers and Interaction-based Retrievers, some recent systems (Khattab et al., 2021; Zhao et al., 2020; Nie et al., 2020) combine these two methods and achieve both high accuracy and efficiency. Such kinds of retrievers are called Representation-interaction Retrievers. For example, ColBERT-QA (Khattab et al., 2021) uses a retriever based on ColBERT (Khattab and Zaharia, 2020), which includes a question-document token-level interaction step to the dual-encoder architecture for computing the relevance between question and document. Similar to any dual-encoder retriever like DPR (Karpukhin et al., 2020), ColBERT-QA first encodes the question and document independently using two BERT encoders. More formally, given a question q and a document d, having corresponding vector representations from the encoders denoted as $E_q(\,)$ and $E_d(\,)$, the relevance score is computed using the below MaxSim operator:

$$S_{q,d} = \sum_{i=1}^{n} max_{j=1}^{m} E_{q_i} \,.\, E_{d_j}^{T}$$

ColBERT then computes the score of each query token by considering all document tokens (maximum similarity matched document token is used) and then sums all these scores as the final relevance score between q and d. SPARTA (Zhao et al., 2020), another example adopting this retriever design, develops a neural ranker to compute token-level similarity score using dot product between a

non-contextualized encoded query and a contextualized encoded document. Precisely, given question and document representations, the weight of each query token is computed with max-pooling, ReLU and log sequentially, and the final relevance score is the sum of each question token weight. Due to a good trade-off between accuracy and efficiency, Representation-interaction Retrievers are promising but need further exploration.

Reader: The main component of any modern QA system is the Reader. It differentiates a QA system from an IR system. It is implemented as a modified Machine Reading Comprehension (MRC) model where an answer is inferred from a set of documents and not just one specific document (original MRC). There are two broad categories of Readers: Extractive Reader, which fetches an answer span from the documents, and Generative Reader, which generates an answer using Seq2Seq models.

Extractive Reader aims to predict an answer's start and end position span from the retrieved documents. It assumes that the answer to the question is present in the retrieved documents. There are two approaches depending on whether the retrieved documents are processed independently or jointly for answer extraction.

Many foundational systems rank the retrieved documents by the probability of answer inclusion and only consider the most probable document to extract the answer span. DS-QA (Lin et al., 2018) has a dedicated Passage Selector module to select the most probable document containing the answer by ranking all retrieved documents based on answer inclusion probability. As another example, DPR (Karpukhin et al., 2020) computes the answer inclusion probability of a passage and a token being the start and end position of an answer span using BERT as a reader and selects the answer with the highest probability after combining the two types of probabilities. Such models fail to take advantage of the evidence from multiple passages and fail to answer multi-hop questions.

Unlike previous models, some systems extract answer spans using all the retrieved documents jointly. DrQA (Chen et al., 2017) extracts various features like Part-of-Speech (POS), Named Entity (NE) and Term-Frequency (TF), etc. from the retrieved documents and then a multi-layer Bi-LSTM reader takes as input the question and documents and predicts an answer span. The answer scores are made comparable acorss documents by an unnormalized exponential function along with agrmax over all answer spans. As another example, BERTserini (Yang et al., 2019) models a reader based on BERT by removing the softmax layer to enable answer scores to be compared and combined among different documents. Clark and Gardner (2017) propose a Shared-Normalization mechanism by modifying the objective function to normalize the start and end scores across all documents as using un-normalized scores (eg. exponential scores or logits scores) for all answer spans is suboptimal. Clark and Gardner (2017) achieved a gain in performance from such normalization. After that, many OpenQA systems develop their readers by applying this mechanism based on original MRC models like BiDAF (Seo et al., 2016), BERT (Devlin et al., 2018) and SpanBERT (Joshi et al., 2020).

Generative Reader is designed to produce natural answers instead of text spans extracted from documents. This is done using Seq2Seq models. As an example, S-Net (Tan et al., 2018) combines extraction and generation models to complement each other. It uses an extraction model to collect evidence by predicting the text-span boundary that can be a potential answer. Then, the text span is fed to the Seq2Seq model for final answer generation. Recently, some QA systems use pre-trained Seq2Seq language models, like BART and T5, to develop their Readers. RAG (Lewis et al., 2020) adopts a pre-trained BART model as its reader and uses DPR for retrieval. FID (Izacard and Grave, 2020) first encodes each retrieved document independently using T5 or BART encoder and then performs attention over all the output representations using the decoder to generate the final answer. It also uses DPR for retrieval. However, Generative Readers need to be futher explored because they suffer from syntax error and incoherency.

3.2.2 Close-book Approach

Large Seq2Seq Language Models based on Transformer architecture have greatly improved downstream NLG tasks. They are pre-trained on large data in an unsupervised setting. Such models include GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), BART (Lewis et al., 2019) and T5 (Raffel et al., 2020). GPT adopts a left-to-right decoder of Transformer while BART and T5 use both encoder and decoder of Transformer. These models have the capability to store knowledge from large-scale text data in their underlying parame-

ters and can directly be used for question answering without access to external knowledge (corpus). For example, GPT-2 is able to generate the answer given only a question without fine-tuning correctly. GPT-3 achieves competitive performance with few-shot learning compared to prior state-of-the-art approaches. Recently, Roberts et al. (2020) performed experiments to evaluate the capability of these language models for question answering without access to external knowledge and found that these language models gain impressive performance on various benchmarks inventing a new approach of Retrieval-free methods to QA.

3.3 Knowledge Graph guided Deep Learning based Question Answering

Question Answering involves answering questions using an information source. The information source can be structured knowledge bases (like knowledge graphs) or unstructured text. In this section, we see Question Answering over knowledge graphs and unstructured text. We call such QA Knowledge Graph guided Deep Learning based Question Answering. Before moving forward, let us first understand various approaches for Question Answering over Knowledge Graph (KGQA). These methods can be roughly categorized into two main groups: semantic parsing based (SP-based approaches) and information retrieval based approaches(IR-based approaches). SP-based approaches aim to construct a semantic parser to convert natural language queries to KG-friendly queries, which can be used to query the KG to find answers. These approaches require supervised training and thus are limited to specific domains. Several efforts are made to overcome these limitations (Abujabal et al., 2017; Hu et al., 2017; Krishnamurthy and Mitchell, 2012; Liang et al., 2013).

In IR-based approaches, a set of candidate answers from the knowledge graph are constructed and then the question and candidate answers are mapped into vector space for calculating the similarity scores between them. The foundation and significant part of IR-based approaches is transforming questions and passages into semantic vector space. Several pioneer works (Bordes et al., 2014; Dong et al., 2015; Hao et al., 2017; Xu et al., 2016a,b) use neural networks to learn the representation of question and candidate answers. Entity description text significantly enhances knowledge graph embedding model performance in the area of knowledge graph representation as they contain rich entity information. This gives space to improve the context representation of candidate answers by rational utilization of unstructured text from the corpus, paving the way towards the combination of knowledge graph and text for Question Answering.

Tian et al. (2021) propose a novel model to improve the representation of candidate answers by associating entities with external text. A co-occurrence network is used to associate entities in knowledge graph with external description text (external information) along with a novel approach to describe candidate answer in knowledge graph (internal information) and apply an attention model to fuse these internal and external information as shown in Figure 3. The candidate answers are produced by fetching the topic entities of a question from Freebase and using their two-hop nodes as candidate answers. The embedding representation for questions is obtained from Transformer's encoder. A co-occurrence network is used to get entity description text for each candidate answer. The entity description text constructs external information for the candidate answers through word embedding matrix. The internal information of candidate answers comprises of entity itself, entity type, entity relation and entity context. An attention mechanism is applied to jointly combine the internal and external information to get a promising representation for candidate answers. Lastly, a similarity score for each candidate answer is computed using the generated vector representations.

The internal information about candidate answers is encoded using the TransE (Bordes et al., 2013) knowledge graph embedding model. Three aspects of the answer, the embedding of the candidate answer itself, the average of embeddings of relations that appear on the answer path and the average of embeddings of entities and relations that directly connect to the answer entity from the knowledge graph are used to describe the candidate answer.

The external information of candidate answers is represented as follows. Given a knowledge graph G and textual corpus $C = c1, c2, .., cn$, we annotate the text corpus with knowledge graph's entity labels using entity linking tool to get entity-annotated text corpus $A = a1, a2, ..., am$, where $m \leq n$, as multiple adjacent words could be labeled as one entity. Next, we construct a co-

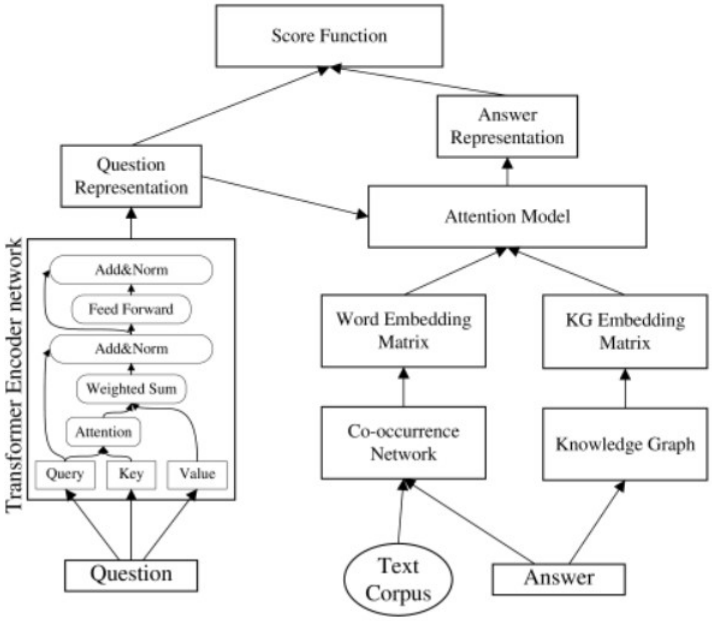

Figure 3: Knowledge Graph based Text-enhanced Question Answering

occurrence network to bridge the candidate answer entity and the entity-annotated text corpus A. Using the co-occurrence frequency, each candidate answer gets its external neighboring nodes. A weighted average of these external nodes is computed to get the external information representation embedding of the candidate answer.

An attention model is created to dynamically aggregate vectors in order to fully integrate internal information from the knowledge graph and external information from the text corpus.. The degree of attention is assessed for each candidate answer based on how closely the representation of the input question and various answer aspect embeddings relate to one another. In this way text information outside the knowledge graph is utilized to enhance the representation ability of candidate answers for the task of QA.

Tian et al. (2021) also carry out in-depth tests to assess the parts of their model. Figure 4 displays the model's performance for various component combinations. Transformer refers to the fact that just Transformer Encoder network is utilized to obtain a representation of the query and that the many aspects of the response are not employed. External information refers to context information obtained from the text corpus outside of the knowledge graph. Internal information indicates that the knowledge graph's answer components are used, while the text corpus is disregarded.

Methods	Macro F1
Transformer	39.8
Transformer + external information	41.8
Transformer + internal information	42.5
Transformer + internal and external information	43.7

Figure 4: Result

The conclusions derived from the results in Figure 4 are as follows. First, employing internal information can result in a higher F1 score when utilised alone, as opposed to using external information. This indicates that the knowledge graph's information is more comprehensive and more suited to the representation of candidate answers. Then, the external description data filtered by the co-occurrence network can effectively amplify the candidate answers' contextual data. The model's performance can be further enhanced and the best grade can be attained by making full use of both internal and external data.

3.4 Question Answering Systems in Aviation Safety Domain

Question Answering has been a widely explored area in general. However, not much progress has been made in the aviation domain due to the frequently occurring in-domain technical jar-

gon. A few existing works use large pre-trained transformer-based language models for Question Answering. Kierszbaum and Lapasset (2020) use Distilled BERT for Question Answering on ASRS reports for a small set of documents and limited test data. Arnold et al. (2020) employs a BM25-based retriever, followed by BERT fine-tuned for QA on a general domain data set.

3.4.1 Distilled BERT for Question Answering

Kierszbaum and Lapasset (2020) performed QA on Aviation Safety Reporting System's (ASRS) incident reports. These reports have many sections. For QA, only the narrative sections are used. There is no use of a document/passage retriever. The report on which question answering has to be performed is given to Distilled BERT. Distilled BERT takes the report's narrative as context along with the question and fetches the text span from the narrative as an answer to the question. Distilled BERT is the same as BERT but with less number of parameters. It does inference faster compared to BERT but also preserves the performance. Distilled BERT is fine-tuned on SQuAD dataset for Machine Reading Comprehension task. Distilled BERT was evaluated on a very small set of documents and limited types of questions. This QA system was developed to assist experts in using ASRS reports in a precise and restrictive setting.

3.4.2 BM25-BERT pipeline for Question Answering

Arnold et al. (2020) performed QA on Flight Crew Operating Manuals (FCOM). BM25 is used to retrieve passages relevant to the question. These passages are then passed to the BERT model to get answer spans. BERT large was fine-tuned using a multi-task approach. In the multi-task approach, the model is fine-tuned on multiple tasks simultaneously to improve all tasks' generalization performance. Two tasks were used. The first one was the standard Machine Reading Comprehension task. The second was a classification task, where the model had to predict whether the answer to the question was present in the context passage. SQuAD 2.0 dataset was used for fine-tuning as it contains all the information for both the tasks. The objective functions of each task were added together and then minimized. The results conveyed that the multi-task approach improves the performance of QA compared to single-task fine-tuning.

4 Summary

In this paper, we summarize the most relevant and recent literature of Information Retrieval and Question Answering that was referred to develop a Question Answering system over text using Deep Learning for the aviation safety domain. Information Retrieval techniques are grouped into two categories: transformer models that perform re-ranking in multi-stage architectures and dense retrieval techniques that perform ranking directly. The former set of methods performs rich interactions between query and passage terms and thus has better accuracy than the latter set of methods. The accuracy comes with a compromise in efficiency; thus, these methods are not scalable for large corpus. The direct ranking techniques are not as accurate as the re-ranking techniques but are very efficient, making them popular. We present two approaches towards QA: DLQA and KG guide DL based QA. DLQA techniques are categorized into two classes: open-book and close-book. The open-book methods perform two steps: retrieval and reading. Retriever retrieves relevant passages and Reader uses those passages to find an answer. Retrievers are of three types: Representation-based, Interaction-based, and Representation-interaction Retrievers. The first are efficient, the second are accurate, and the third are efficient + accurate. Readers are also of two types: Extractive readers and Generative readers. The former extracts text spans and the latter generates text from the relevant passages. The close-book methods directly answer questions by storing corpus information in learnable weights. Thus they demand a lot of training and are also large in size, making them not so attractive and exciting. In KG guided DL based QA, we discuss a novel method that utilizes text information outside the knowledge graph to enhance the representation ability of candidate answers for the task of QA. At last, we see two QA systems in the aviation domain built using BM25 and BERT.

References

Abdalghani Abujabal, Mohamed Yahya, Mirek Riedewald, and Gerhard Weikum. 2017. Automated template generation for question answering over knowledge graphs. In *Proceedings of the 26th international conference on world wide web*, pages 1191–1200.

Alexandre Arnold, Gérard Dupont, Félix Furger, Catherine Kobus, and François Lancelot. 2020. A question-

answering system for aircraft pilots' documentation. *arXiv preprint arXiv:2011.13284*.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Christopher Clark and Matt Gardner. 2017. Simple and effective multi-paragraph reading comprehension. *arXiv preprint arXiv:1710.10723*.

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

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over freebase with multi-column convolutional neural networks. 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 260–269.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 221–231.

Sen Hu, Lei Zou, Jeffrey Xu Yu, Haixun Wang, and Dongyan Zhao. 2017. Answering natural language questions by subgraph matching over knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 30(5):824–837.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for openqa with colbert. *Transactions of the Association for Computational Linguistics*, 9:929–944.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Samuel Kierszbaum and Laurent Lapasset. 2020. Applying Distilled BERT for Question Answering on ASRS Reports. In *NTCA 2020 New Trends in Civil Aviation*, NTCA 2020 New Trends in Civil Aviation, pages 33–38, Prague, Czech Republic. IEEE.

Jayant Krishnamurthy and Tom Mitchell. 2012. Weakly supervised training of semantic parsers. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 754–765.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Percy Liang, Michael I Jordan, and Dan Klein. 2013. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1736–1745.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. *arXiv preprint arXiv:2005.00181*.

Ping Nie, Yuyu Zhang, Xiubo Geng, Arun Ramamurthy, Le Song, and Daxin Jiang. 2020. Dc-bert: Decoupling question and document for efficient contextual encoding. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1829–1832.

Yixin Nie, Songhe Wang, and Mohit Bansal. 2019. Revealing the importance of semantic retrieval for machine reading at scale. *arXiv preprint arXiv:1909.08041*.

Kyosuke Nishida, Itsumi Saito, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2018. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 647–656.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Chuanqi Tan, Furu Wei, Nan Yang, Bowen Du, Weifeng Lv, and Ming Zhou. 2018. S-net: From answer extraction to answer synthesis for machine reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jiaying Tian, Bohan Li, Ye Ji, and Jiajun Wu. 2021. Text-enhanced question answering over knowledge graph. In *The 10th International Joint Conference on Knowledge Graphs*, pages 135–139.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016a. Hybrid question answering over knowledge base and free text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2397–2407.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016b. Question answering on freebase via relation extraction and textual evidence. *arXiv preprint arXiv:1603.00957*.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: Bert and beyond. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1154–1156.

Tiancheng Zhao, Xiaopeng Lu, and Kyusong Lee. 2020. Sparta: efficient open-domain question answering via sparse transformer matching retrieval. *arXiv preprint arXiv:2009.13013*.