Information Retrieval and Question Answering

Group:
Possakorn Kittipipatthanapong a1873765

The University of Adelaide

4333_COMP_SCI_7417 Applied Natural Language Processing

Lecturer: Dr. Alfred Krzywicki

Table of Contents

1. Abstract

In the digital age, the vast information landscape necessitates robust Information Retrieval-based Question Answering (IR QA) systems to efficiently navigate and extract precise knowledge. Our system, developed to address this need, employs innovative strategies to distill actionable answers from large datasets. The core methodology relies on selective preprocessing, mainly lowercase normalization, alongside Fastcoref for coreference resolution and TextBlob for sentence splitting, with the Semantic DistilBERT model excelling in similarity matching. These approaches resulted in an MRR of 1.00 and a MAP of 0.96, with an overall F1-score of 0.71, suggesting a high alignment with user queries. Challenges faced include managing complex queries and capturing longer sentences, areas identified for future enhancement. This project underscores the transformative impact of machine learning in information retrieval and sets the stage for further advancements in AI-driven question answering systems.

2. Introduction

In today's digital era, information is abundant, serving as a repository for invaluable data that makes our lives more convenient. The internet has emerged as a pivotal platform for the dissemination of knowledge. We, as humans, engage with this vast information through computers, seeking answers and exploring topics that serve our interests. While we have access to a large amount of data, finding the precise answers to our questions can resemble an attempt to search for a needle in a haystack. This is where an information retrieval system and question-answering system play a crucial role in providing rich and relevant answers based on statistics or scientific facts. A wide array of question-answering applications has been tailored to fulfil the informational demands of individuals, thereby saving considerable time that can be utilized for other pursuits. These applications are integral in various scenarios such as utilizing virtual assistants, navigating through databases, and employing search engines [1].

Furthermore, the pandemic has spurred a surge in the demand for artificial intelligence, particularly in the deployment of various pre-trained models. These models enhance performance across a multitude of tasks, ranging from complex large language models like GPT to fundamental operations such as coreference resolution and semantic matching. This advancement has significantly boosted efficiency and broadened the scope of applicability [2].

2.1. Information Retrieval-based Question Answering

Within the realm of question-answering systems, information retrieval-based QA stands out as a major paradigm, concentrating on extracting specific subsets of information to provide concise answers derived from brief text passages. Also known as open-domain QA, this approach involves parsing user queries to locate relevant information, retrieving pertinent passages, and pinpointing the answer within specific text

segments [1].

Information retrieval, often equated with search engines or ad-hoc retrieval systems, involves a process where a user submits a query into a system, which then retrieves and ranks a set of documents from a vast collection based on relevance to the query as shown in **Figure 01**.

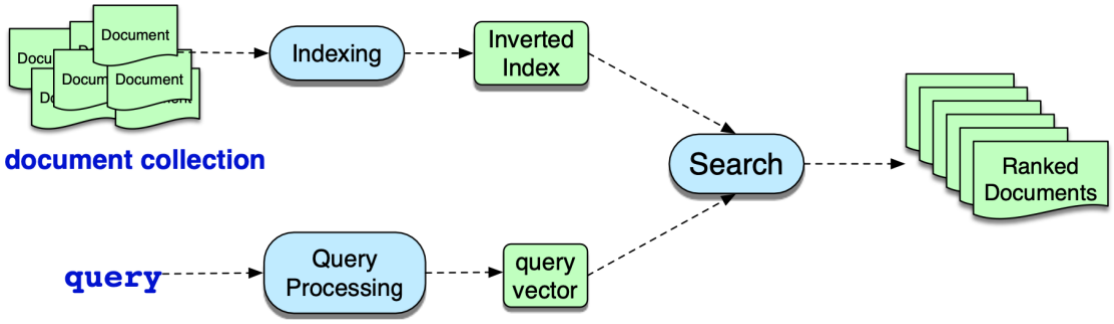


Figure 1: A **collection** denotes an assemblage of documents curated to fulfill user inquiries. A **term** denotes not only a single word within this collection but can also encompass phrases. Meanwhile, a **query** is the articulation of a user's information need formulated as an array of terms. To facilitate the retrieval process, both queries and documents are converted into vectors predicated on the frequency of individual words (unigram word counts). The ranking of relevant documents is then determined by computing the cosine similarity between these vectors, treating words independently from their sequential position in the text.

2.1.1. Weighting and Document scoring

Term Frequency-Inverse Document Frequency, or TF-IDF, is employed in information retrieval to assign a weight to each word in a document, signifying its importance. This measure is composed of two key metrics. First, Term Frequency (TF) gauges the prevalence of a word within a document, serving as an indicator of its significance. The frequency is adjusted to account for variations in document length and is typically logged to prevent a bias towards longer documents, as shown in **Equation 01**.

$$\mathrm{tf}_{t,d} = \log_{10}(\mathrm{count}(t,d)+1)$$

Equation 1: provides the mathematical formula for calculating the term frequency (TF). This measure reflects the number of times a term **t** appears within a given document **d**.

Second, Inverse Document Frequency (IDF) assesses the rarity of a word across all documents. Words that appear in a small number of documents tend to be more distinctive and are thus given a higher IDF score. Conversely, words that are common across the entire document collection are less useful for differentiation and receive a lower IDF score, as illustrated in **Equation 02.**

$$\mathrm{idf}_t = \log_{10}\frac{N}{\mathrm{df}_t}$$

Equation 2: Equation 02 delineates the mathematical formula for computing the **inverse document frequency (IDF)**. This metric quantifies the exclusivity of a **term t** by evaluating how infrequently it occurs across the entire set of **documents D.**

Collectively, these components of TF-IDF allow for a more nuanced understanding of a word's relevance

not only within a single document but also within the context of a broader collection of texts following **Equation 03.**

$$\text{tf-idf}(t,d) = \text{tf}_{t,d} \cdot \text{idf}_t$$

Equation 3: the dot-product of **Term Frequency (tf)** and **Inverse Document Frequency (idf)**

Document scoring is executed by computing the cosine similarity between the vector representation of documents and the query. This computation yields a result that falls within the range from one, indicating the most relevant documents, to zero, denoting documents that are not related at all. The calculated values consider the term counts, which are non-negative, as illustrated in **Equation 04.**

$$\text{score}(q,d) = \sum_{t \in \mathbf{q}} \frac{\text{tf-idf}(t,q)}{\sqrt{\sum_{q_i \in q} \text{tf-idf}^2(q_i,q)}} \cdot \frac{\text{tf-idf}(t,d)}{\sqrt{\sum_{d_i \in d} \text{tf-idf}^2(d_i,d)}}$$

Equation 4: show dot product of unit vectors between documents **vector d** and **query vector q**

2.2. Coreference resolution

Coreference resolution, a facet of the discourse model as introduced by Karttunen in 1969 [3], plays a crucial role in the incremental construction of meaning as a text is interpreted. This process involves mapping various words and phrases to the entities and relationships they denote within a text. As depicted in Figure 02, coreference resolution seeks to determine when different expressions in the text actually refer to the same entity or concept as shown in simple example as Figure 03.

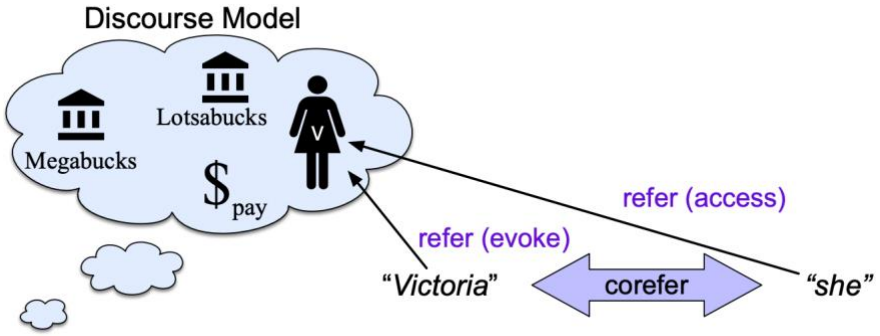


Figure 2: illustrates the mechanics of how mentions trigger and retrieve discourse entities within a discourse model. In this model, the name 'Victoria' serves as the referent, a specific entity in the discourse. The pronoun 'she' is an example of a mention, which is a linguistic expression that refers back to the referent. Both the referent and the mention are understood to denote the same entity within the discourse

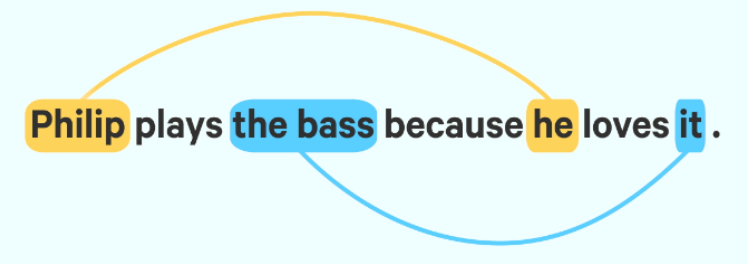


Figure 3: show the example corefer which mentions **Philip** as **he** and the **bass** as **it**.

Coreference is integral to NLP, aiding in several tasks including Information Extraction [4], Summarization [5], Machine Translation [6], and Question Answering [7], It's particularly crucial for systems to resolve coreferences accurately to comprehend the pertinent query terms and generate precise responses when questions are posed [8].

Nonetheless, coreference resolution comes with its set of challenges. One major limitation is the inherent ambiguity that occurs when multiple entities or concepts in a text possess similar attributes, especially if they span across several paragraphs. This increases the complexity of understanding the context over extended discourse. Moreover, correctly interpreting references often necessitates a degree of world knowledge which may be absent in the training data provided for the AI models [9].

2.3. Our system and related applications

Our system is developed for information retrieval in a Question Answering context, designed to pinpoint the answer, the most pertinent sentences, and a confidence score from a given article. Our suite of algorithms will assist the application in analyzing the provided article, delivering concise text snippets instead of comprehensive paragraph summaries. This application is predominantly geared towards general question answering uses, such as extracting answers from articles or books and supporting research requirements in educational institutions or fulfilling information retrieval needs within internal knowledge management systems.

Our system is optimized for handling factual questions that begin with interrogatives like who, what, where, when, and how, making it straightforward for the algorithm to operate. It is versatile enough to interact with various types of databases, including structured databases and extracting data directly from unstructured text.

However, there are limitations, particularly in processing complex and ambiguous queries. Questions that fall outside the domain expertise or the scope of the provided article may not yield any answers. Furthermore, the system may face challenges with a deep contextual understanding or handling inquiries that are excessively open-ended.

2.4. Purpose and scope

The objective and extent of our system's implementation involve several key steps. Initially, we commence by reading the dataset and conducting appropriate data preprocessing. Subsequently, we apply suitable machine learning and deep learning models for Coreference Resolution, Document Processing and Retrieval Engine, and Question Answering tasks. Post-implementation, we proceed to evaluate our model's performance in both information retrieval and question answering contexts. The final stages include discussing the outcomes and presenting conclusions drawn from our findings.

2.5. Related data sources

The dataset for this study is derived from real-world articles categorized into 10 groups, each containing 1000 entries. These entries are organized into 7 columns, which are: **'id'**, **'author'**, **'date'**, **'year'**, **'month'**, **'topic'**, and **'article'**. There are 7 null data related to author compared to completed data quality in other columns. The principal columns utilized for extracting information are 'article', which contains the content necessary to gather user-specific information, and **'article_id'**, which serves as the index reference. Owing to computational constraints, the experiment will focus on sampling from the top 3 topics, which collectively contribute to a dataset of 102 rows, or 34 rows per topic. The 'article' column also details the word count for each topic, with a comparative illustration provided in the corresponding figure against the sampled dataset.

3. Preprocessing

Preprocessing is a critical step that significantly improves the final output and readies the data for integration with advanced modeling techniques. In this particular scenario, the text preprocessing sequence will include the elimination of stop words, implementation of lemmatization, conversion of all letters to lowercase, execution of tokenization, and the application of an entity linkage through the use of a coreference resolution utility. Consequently, an ablation study will be conducted within the evaluation section to assess the impact of these omissions' context, thereby affecting the performance of NLP models.

3.1. Removing stop word

In numerous NLP and information retrieval endeavors, the preliminary step of excising stopwords from the text, which effectively reduces dimensionality, is known to enhance the efficiency and effectiveness of algorithms. Stopwords typically include articles, conjunctions, prepositions, pronouns, and auxiliary verbs, and their removal can influence the semantic integrity of the context, which affect to outcome of our models. In this experiment, it performs the preprocessing via **NLTK** [10] which the famous library to deal with many NLP tasks.

3.2. Lemmatization

Lemmatization is a linguistic technique that aims to normalize various inflected forms of a word by reducing them to their base or root form [1]. This process significantly improves the handling of morphologically complex languages. For this purpose, the **NLTK library**, a prominent toolkit in NLP, is employed to lemmatize the text in the articles.

3.3. lowering

A various of models may exhibit sensitivity to case variation, which can result in ambiguity when processing tasks, particularly when regular expressions are employed. During the dataset preprocessing phase, converting text to lowercase is a standard operation to achieve case insensitivity. This step simplifies the dataset, ensuring uniformity across the text. **The NLTK library** is frequently utilized for this purpose, offering straightforward methods to convert tokens to lowercase, thereby mitigating potential complications arising from case sensitivity.

3.4. Tokenization

During dataset preprocessing, text normalization is carried out to convert the data into a more usable or standard format. While English words are generally separated by white spaces, this method isn't always sufficient for clear differentiation. Hence, in this experiment, tokenization will be employed to systematically break down text into properly formed tokens. The **NLTK toolkit** will be used for this purpose, as it provides efficient methods for tokenizing words within the dataset.

4. System Architecture

The architecture of our Information Retrieval-based Question Answering (IR QA) system is conceptualized as a sequential, multi-component framework aimed at extracting meaningful answers from extensive text corpora. Utilizing machine learning, the system converts unstructured text into structured, actionable intelligence, thus streamlining the process for responding to user queries effectively. Once datasets are **handled** and **preprocessed**, they are channeled through the main operational pipeline, comprising these integrated modules:

- **Coreference Resolution Utility:** A critical component of our system is the coreference resolution utility as mentioned in **Section 2.2**, powered by a pre-trained machine learning model. This utility enhances the system's interpretative capabilities by resolving anaphoric references within the text, allowing for a more accurate representation of entities and their relationships.
- **Document Processing**: Central to our architecture is the document processing and retrieval engine, which

leverages the mathematical rigor of Cosine similarity and TF-IDF metrics as mentioned in **Section 2.1**. The engine indexes documents and evaluates their relevance to user queries, orchestrating a document ranking system that is continually refined to integrate the latest advancements in pre-trained models.

- **Question Answering**: At the purpose of the system is the question answering module, which directs the focus of the entire architecture towards extracting precise answers from the identified sentences. This module acts as the decision-making core, where the relevance of information is judged against a predefined confidence threshold to provide users with concise, accurate responses.

Each module interlinks to create an ecosystem that not only processes raw text data but also discerns and disseminates the most relevant information in response to specific queries posed by users.

5. Model Selection and Training

5.1. Choose of NLP and Machine learning model

Related to the NLP or machine learning model which integrated in each part, the model selection will be shown following the modules below:

5.1.1. Coreference Resolution

Our experiment's dataset preprocessing utilizes the Fastcoref library, renowned for its speed and efficiency in model development. Fastcoref [8] builds upon the Start-to-End (S2E) model, as shown architecture **on Figure 04**, to precisely identify text spans and leverages Knowledge Distillation to ensure swift processing without compromising on accuracy. Its 'leftovers batching' technique maximizes document processing parallelism, substantially reducing encoding durations commonly associated with coreference resolution.

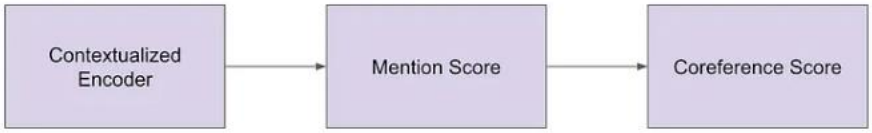
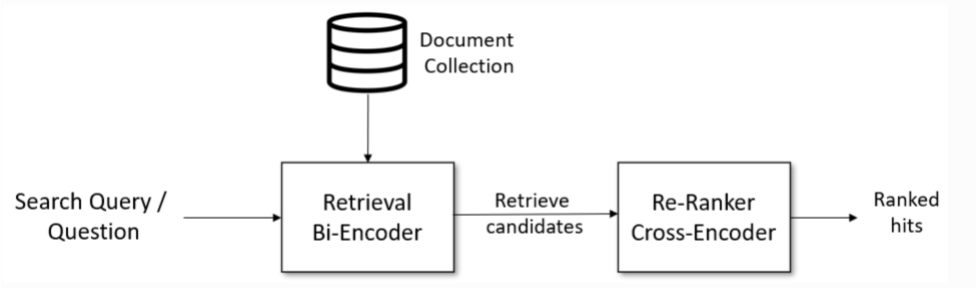


Figure 4: General F-coref algorithm pipeline included Longformer (a contextualized encoder), a parameterized mention scoring function (f_m) and a parameterized pairwise antecedent scoring function (f_a))[F-Coref].

For our study, we employ the F-coref model and a LINGMESS [11] model for comparative performance analysis across coreference resolution tasks. This streamlined system preserves accuracy and accelerates inference, essential qualities for our IR QA architecture.

5.1.2. Document Processing and Retrieval Engine

Our foundational system utilizes a retrieval engine based on Cosine similarity and TF-IDF metrics, as detailed in Section 2.1. In line with current advancements, we plan to enhance our system by incorporating pretrained models, particularly from the domain of NLP applications. We will integrate and evaluate Sentence-BERT, which uses Siamese BERT-Networks for sentence embeddings [12]. This model will be compared against our baseline to assess performance improvements, including variations of the model labeled X, Y, and Z from **sentence_transformers** library with retrieve & re-rank pipeline on Figure below.



5.1.3. Question Answering

Our question-answering system utilizes DistilBERT, a streamlined version of BERT that offers quicker processing times due to its smaller size. DistilBERT was pretrained on text without human-labeled data and further refined using knowledge distillation techniques, aligning its learning with the larger BERT model, which acts as a 'teacher'. This process aligns the direction of the hidden state vectors of the student model (DistilBERT) with those of the teacher. Using the AutoModels library, we specifically engage the 'distilbert-base-cased' model to efficiently extract answers from questions and articles presented to the system.

5.2. Evaluation Metrics

The evaluation metric used to measure in this system will be separated into parts. First, information retrieval evaluation and second part will be evaluated on generating results or snippet text which match on generated.

5.2.1. Information Retrieval Evaluation

For evaluating factoid question-answering performance, Mean Reciprocal Rank (MRR) is employed. MRR is particularly useful when a system provides a shortlist of ranked answers, and the metric focuses on the position of the first correct answer relative to the human-labeled gold standard. Meanwhile, Mean Average Precision (MAP) is another valuable metric, especially for information retrieval tasks, since it considers precision at every rank and the relevance of all retrieved documents. Although MRR is advantageous for systems where the priority is the highest-ranked answer, **both MRR and MAP will be considered to provide a comprehensive assessment from all available perspectives.**

5.2.2. Question answering Evaluation.

To evaluate the performance of our question-answering system, several metrics are available. While ROUGE is typically used to assess text similarity sequences, the F1-score, which is the harmonic mean of precision and recall, proves more effective for fact-based questions. The F1-score excels in scenarios where answers are discrete entities, as it precisely gauges the system's ability to identify correct answers within larger text snippets or documents. **Therefore, our system will utilize the F1-score as the primary metric for evaluation.**

5.3. Outline to fine-tuning model.

The model's fine-tuning process includes the adjustment of several parameters based on features like preprocessing, sentence splitting, coreference resolution, and similarity matching models**. For preprocessing**, options include whether to remove stopwords, lowercase text, and apply lemmatization. **Sentence splitting** can be done using a simple space, TextBlob, or a common splitting algorithm. **Coreference resolution** choices include 'fastcoref' or 'LingMessCoref'**. For similarity matching**, the options range from a 'Cosine-Tfidf model' to reranking models like 'msmarco-MiniLM-L6-cos-v5' or 'msmarco-distilbert-cos-v5', as well as semantic models like 'multi-qa-distilbert-cos-v1' or 'multi-qa-MiniLM-L6-cos-v1'.

5.4. Discussion on results

As the setting of experiment due the fine-tuneing model, there are features need to investigate including preprocessing model, sentence splitting type, coreference resolution model, and similarity model as outcome show below:

In our model's fine-tuning phase, preprocessing with just lowercase transformation achieved the best F1-score at 0.60 as shown **on Figure05-topleft**, outperforming sole lemmatization and stopword removal, which scored 0.57 and 0.51, respectively. Integrating all preprocessing features led to the lowest performance at 0.38, indicating that simple lowercasing was the most beneficial method.

For sentence splitting **on Figure05-bottomright**, spaCy and TextBlob outperformed common split methods, delivering superior performance in both information retrieval, with MAP and MRR scores of 0.67 and 0.71, and question answering, with an F1-score of 0.6. Common split methods struggled with special punctuation, leading to lower scores.

Coreference resolution models also impacted results illustrated **on Figure05-bottomleft**. LingMessCoref, a teacher model, slightly outperformed fastcoref in retrieval tasks, with MRR and MAP scores of 0.73 and 0.68 compared to fastcoref's 0.71 and 0.67. However, fastcoref led to better question answering performance

with an F1-score of 0.6 versus 0.57, suggesting it was more effective at locating correct answers despite the coreference resolution's interference.

Lastly, in terms of similarity matching **following Figure05-bottomleft**, the Semantic DistilBERT model excelled with the highest scores of 0.73 for MRR, 0.71 for MAP, and an F1-score of 0.62, surpassing MiniLM and the traditional Cosine-TFIDF method. Pretrained DistilBERT models showed better adaptability to new applications, indicating their ease of understanding and application.

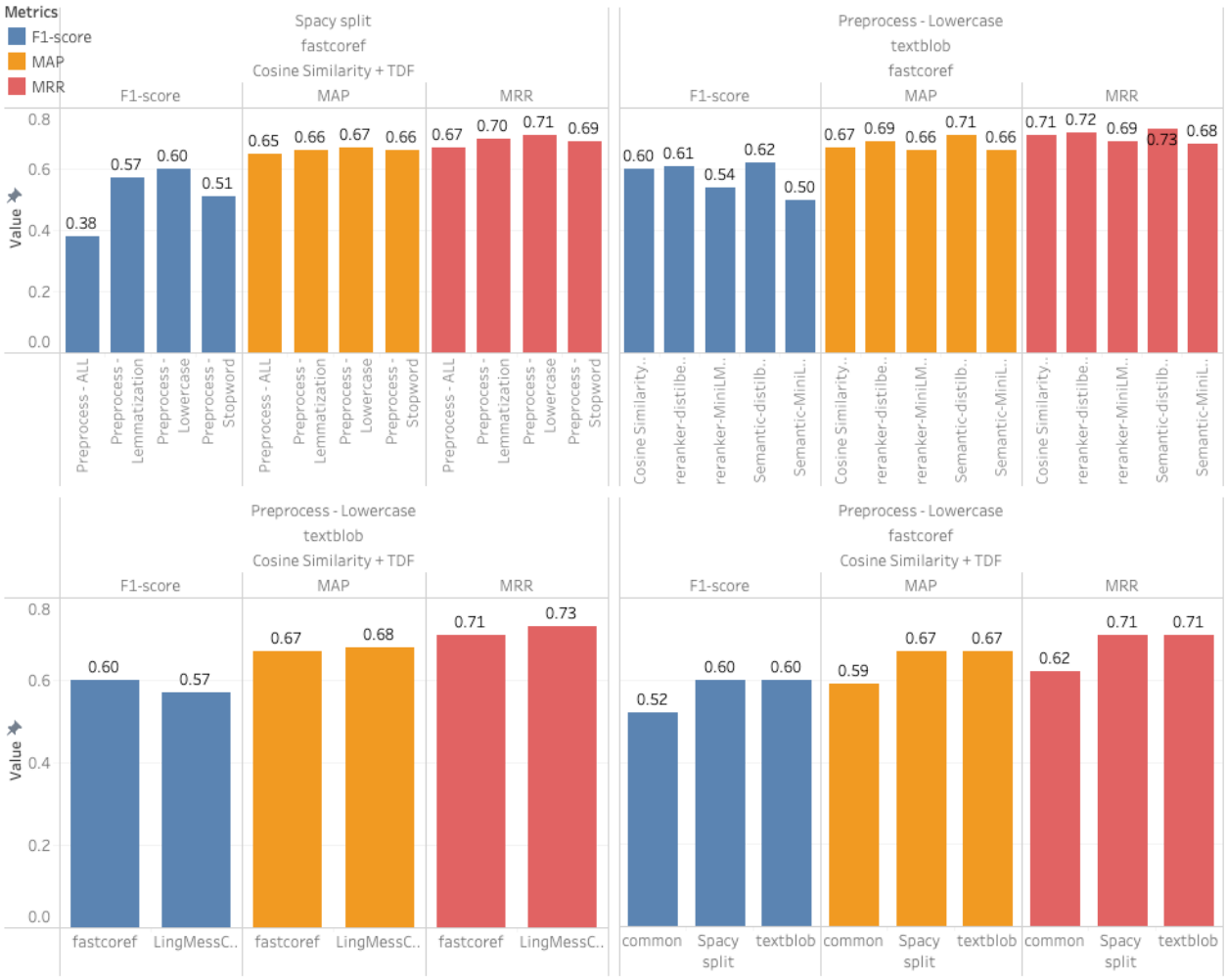


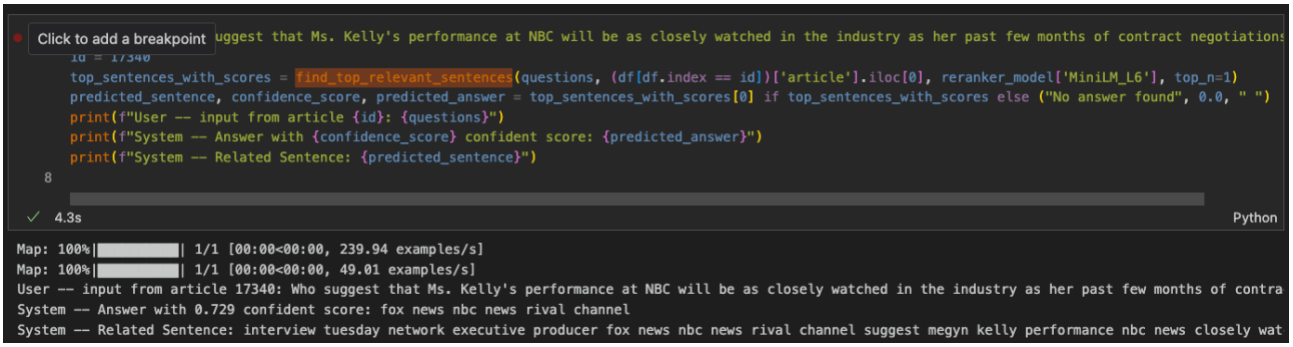
Figure 5: illustrates the impact of selected feature settings on the performance of the Question Answering model, with other variables held constant. The figure presents comparative results in terms of Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) for information retrieval effectiveness, as well as the F1-score for a broader evaluation. It showcases the relationship and performance influence of different features. Show the Relationship of (top-left) Preprocessing, (top-right) similarity model, (bottom-left) Coreference model, and (bottom-right) Splitting method.

6. User interaction with the system

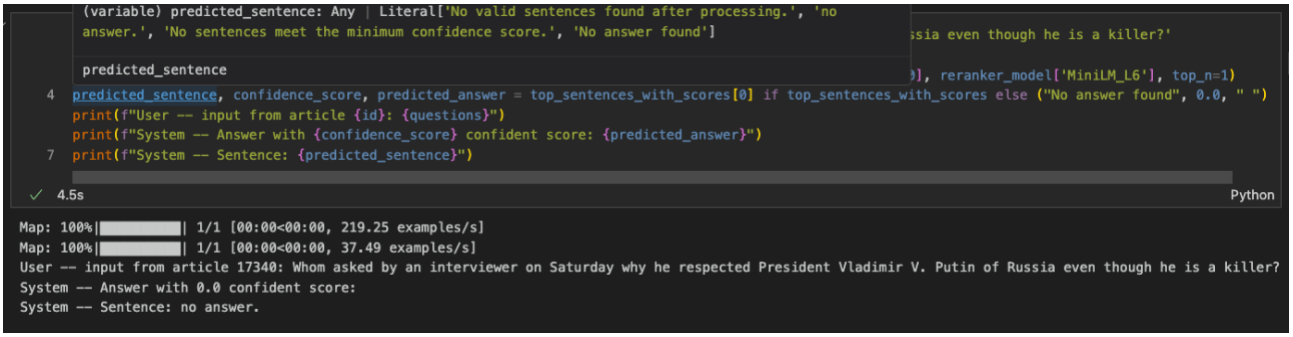
Users interact with our database by selecting an article and posing related questions. The system then retrieves text snippet answers from the article. If an answer doesn't achieve the required confidence threshold, the system will indicate 'no answer.' There are two scenarios for user queries: general and non-

related questions. General questions will yield answers pertinent to the content of the selected article. For non-related questions—those that don't pertain to the article—the system will return 'no answer.

General question's case



Non-related question's case



7. System Evaluation

In the discussion section, it was found that the most effective methodology combined preprocessing with lowercase only, the fastcoref coreference resolution model, TextBlob for sentence splitting, and the semantic distillbert_v1 model for similarity matching. Applying these optimal settings to a test set, which included 10 manually crafted questions and answers, yielded impressive results: an MRR of 1.00 and a MAP of 0.96. The overall F1-score was recorded at 0.71, indicating that most answers correctly matched the question snippets based on the provided article. Further analysis revealed that while the top related answers were correctly identified, the similarity matching model struggled with capturing longer sentences, as illustrated in the figure provided below.

Question: What happened at an Istanbul nightclub on New Year's Day?
Gold Standard Benchmark: The Turkish authorities are hunting for the gunman who opened fire at an Istanbul nig

	Rank	Score	Similarity	Sentence
0	1	0.70	0.97	here s what you need to know the turkish authorities are hunting for the gunman who opened fire at an istanbul nightclub on new year s day killing at least 39 people from no fewer than 12 countries .
1	2	0.66	0.86	the islamic state claimed the gunman who opened fire at an istanbul nightclub on new year s day killing at least 39 people from no fewer than 12 countries as a hero soldier of the caliphate and appeared to refer to turkey s role in the syrian war .
2	3	0.30	0.38	the new york times suicide bombers struck the international airport in mogadishu somalia killing at least three security officers .
3	4	0.24	0.09	good morning .
4	5	0.24	0.18	fans around the world plan to toast the professor at 9 p. m. local time .
5	6	0.24	0.28	most major markets reopen after new year s holiday .

8. Conclusion

In the creation of our IR QA system, strategic choices in model features have been pivotal. Lowercase preprocessing alone emerged as the most effective, contradicting the assumption that more complex methods are always better. Coreference resolution with Fastcoref and sentence splitting with TextBlob laid a solid foundation, while high number of amount parameter in pretrained model like Distillbert directly related to the performance of information retrieval model. These choices led to outstanding metrics: an MRR of 1.00 and a MAP of 0.96, reflecting the system's precision. Our exploration demonstrates that the judicious application of advanced pre-trained models is critical for answering queries accurately and efficiently.

Throughout development, challenges included handling complex and ambiguous queries and the difficulty of the similarity matching model to capture longer sentences effectively. Additionally, integrating multiple NLP tasks—from coreference resolution to similarity matching—posed complexities that necessitated fine-tuning and optimization.

For future improvements, we envisage exploring advanced techniques to better handle longer sentences and enhance the system’s ability to deal with open-ended questions. We also aim to reduce the performance gap seen when all preprocessing features are employed simultaneously. Further research could investigate the integration of more sophisticated models and the inclusion of domain-specific knowledge to address the noted limitations.

In conclusion, the success of our system reaffirms the potential of AI and machine learning in refining information retrieval and question answering processes. By continuing to innovate and address the identified challenges, we can significantly improve the user experience in navigating the vast expanse of digital information.

1. References

[1] D. Jurafsky and J. H. Martin, Speech and Language Processing : an Introduction to Natural Language processing, Computational linguistics, and Speech Recognition. India: Dorling Kindersley Pvt, Ltd, 2014. Available: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

[2] X. Han et al., "Pre-Trained Models: Past, Present and Future," arXiv.org, Aug. 11, 2021. https://arxiv.org/abs/2106.07139

[3] L. Karttunen, Discourse Referents. 1969.

[4] Yi Luan et al. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

[5] J. Christensen et al. 2013. Towards coherent multidocument summarization. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1163–1173, Atlanta, Georgia. Association for Computational Linguistics.

[6] D. Stojanovski et al. 2018. Coreference and coherence in neural machine translation: A study using oracle experiments. In Proceedings of the Third Conference on Machine Translation: Research Papers, pages 49–60, Brussels, Belgium. Association for Computational Linguistics.

[7] P. Dasigi et al.. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. 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 5925–5932, Hong Kong, China. Association for Computational Linguistics.

[8] S. Otmazgin, A. Cattan, and Y. Goldberg, "F-coref: Fast, Accurate and Easy to Use Coreference Resolution," arXiv (Cornell University), Jan. 2022, doi: https://doi.org/10.48550/arxiv.2209.04280.

[9] E. G. PhD, "Coreference Resolution in Natural Language Processing (NLP)," **AI monks.io**, Mar. 08, 2024. https://medium.com/aimonks/coreference-resolution-in-natural-language-processing-nlp-5ba4f570bffe (accessed Apr. 13, 2024).

[10] S. Bird, Natural language processing with python. O'reilly Media, 2016.

[11] S. Otmazgin, A. Cattan, and Y. Goldberg, "LingMess: Linguistically Informed Multi Expert Scorers for Coreference Resolution," arXiv (Cornell University), May 2022, doi: https://doi.org/10.48550/arxiv.2205.12644.

[12] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, doi: https://doi.org/10.18653/v1/d19-1410.