Design of Transformer-Based QA Systems for Logistic Data*

Martin Bača[1,†], Šimon Horvát[1,*,†]

[1]Pavol Jozef Šafárik University in Košice, Jesenná 5, 040 01 Košice, Slovakia

Abstract

The surge in automation within logistics has led to an overwhelming growth in data, making it increasingly difficult to efficiently find relevant information. Traditional keyword-based search methods often fail to deliver precise and context-aware results. This paper presents the design and evaluation of advanced Question Answering (QA) systems tailored for logistic data, leveraging state-of-the-art Transformer-based language models. A core contribution of this work is the development of a novel, well-annotated QA dataset, automatically generated using large language models (LLMs). The dataset includes extractive and binary questions, enabling comprehensive evaluation across multiple QA tasks. The proposed pipeline covers the full methodology—from dataset generation to model training and evaluation—reducing manual effort while maintaining data quality. Additionally, a classification extension was introduced to improve the handling of different question types, particularly enhancing binary QA performance. Experimental results demonstrate improved QA accuracy and system adaptability, offering valuable insights for applying modern NLP in logistics.

Keywords

Dataset Generation, Logistics, Natural Language Processing, Question Answering, Transformers

1. Introduction

The landscape of logistics is undergoing a profound transformation driven by increasing automation, which in turn generates an unprecedented volume of operational data. This data explosion presents a critical challenge: efficiently extracting relevant and precise information from vast, complex datasets. Conventional keyword-based search techniques, limited by their inability to grasp context and nuances, frequently fall short in delivering the accurate and immediate answers required by dynamic logistical operations. Addressing this, Question Answering (QA) systems, a cutting-edge domain within Natural Language Processing (NLP), offer a robust solution by enabling machines to directly answer questions posed in natural language.

This paper focuses on the design and evaluation of advanced QA systems specifically tailored for logistic data. Leveraging state-of-the-art Transformer-based language models, which have revolutionized NLP by demonstrating superior capabilities in understanding and generating human language, we aim to overcome the limitations of traditional information retrieval methods. Our work highlights the efficacy and precision of these advanced models when applied to the unique context of logistics.

A central contribution of this research is the development of a novel, high-quality, and well-annotated QA dataset. This dataset is meticulously designed to support comprehensive evaluation across multiple QA tasks, featuring extractive and binary question types. Crucially, the entire dataset is automatically generated using large language models (LLMs), significantly reducing the manual effort typically associated with dataset creation while ensuring high data quality and consistency. This automated pipeline represents a scalable approach to building domain-specific QA resources.

Furthermore, we propose and implement a comprehensive methodology that spans the entire QA system development cycle: from the automated generation of the dataset to the training and rigorous

ITAT'25: Information Technologies – Applications and Theory, September 26–30, 2025, Telgárt, Slovakia

*Corresponding author.

†These authors contributed equally.

martin.baca@student.upjs.sk (M. Bača); simon.horvat@upjs.sk (Š. Horvát)

0009-0005-8120-0948 (M. Bača); 0000-0002-3191-8469 (Š. Horvát)

evaluation of the models. An innovative aspect of our approach includes the introduction of a classification extension to the Transformer models, specifically designed to improve the handling of different question types. This extension is particularly effective in enhancing the performance of binary QA, demonstrating a refined understanding of question intent.

Experimental results from our evaluation demonstrate notable improvements in QA accuracy and overall system adaptability within the logistics domain. These findings offer valuable insights into the practical application of modern NLP techniques for specialized information retrieval, showcasing the potential of Transformer-based QA systems to enhance efficiency and decision-making in complex logistical environments.

The remainder of this paper is structured as follows: Section 2 reviews related work in Question Answering, focusing on Transformer architectures and dataset generation. Section 3 details our proposed methodology, including the automated dataset generation pipeline, model architecture, and training procedures. Section 4 presents the experimental setup and discusses the performance results across various QA tasks. Finally, Section 5 concludes the paper, summarizing our contributions and outlining directions for future research.

2. Related Work

The field of Question Answering (QA) has evolved significantly, progressing from early rule-based and Information Retrieval (IR) methods to sophisticated machine learning-driven systems. Initial approaches, exemplified by systems participating in the TREC QA track, often relied on handcrafted rules and extensive knowledge bases to extract answers. The advent of statistical and machine learning methods marked a shift towards more flexible and robust systems, leading to hybrid architectures that combined the strengths of both symbolic and data-driven techniques [1, 2]. A notable milestone in this evolution was IBM Watson's DeepQA system, which showcased the potential of complex QA pipelines in open-domain scenarios [3].

The resurgence of deep learning has revolutionized Natural Language Processing (NLP), profoundly impacting QA research. Recurrent Neural Networks (RNNs) like LSTMs and GRUs, along with convolutional neural networks (CNNs), enabled models to capture sequential dependencies and local features in text, leading to improvements in machine comprehension tasks. The introduction of the attention mechanism significantly enhanced these models by allowing them to focus on relevant parts of the input sequence, overcoming the limitations of fixed-size context vectors [4].

The Transformer architecture, proposed by Vaswani et al. in 2017 [5], marked a paradigm shift in NLP. By entirely relying on attention mechanisms without recurrence or convolutions, Transformers enabled unprecedented parallelization and captured long-range dependencies more effectively. This innovation led to the development of powerful pre-trained language models such as BERT (Bidirectional Encoder Representations from Transformers) [6], which demonstrated superior performance across a wide array of downstream NLP tasks through its masked language modeling and next sentence prediction objectives. Subsequent advancements built upon BERT, including RoBERTa [7], ELECTRA [8], and efficient variants like ALBERT [9] and DistilBERT [10], further pushed the boundaries of what is achievable in QA.

The availability of large, high-quality datasets is crucial for training effective QA systems. Prominent datasets like SQuAD (Stanford Question Answering Dataset) [11], HotpotQA [12], and Natural Questions [13] have been instrumental in driving research in extractive, multi-hop, and open-domain QA, respectively. However, generating such datasets manually is a labor-intensive and time-consuming process, often limiting their domain specificity and scale. Recent progress in large language models (LLMs) has opened new avenues for automated dataset generation, enabling the creation of tailored datasets for specific domains like logistics, which is a focus of our work.

3. Methodology

In this section, we describe the methodology for constructing a QA dataset and fine-tuning transformer-based models to handle both extractive and binary QA tasks.

Our pipeline involves generating structured synthetic data, transforming it into natural language, creating question-answer pairs, and validating the output. Finally, we describe the training procedure for several selected models.

3.1. QA Dataset Creation Pipeline

To build a high-quality question answering (QA) dataset, we followed a four-stage pipeline:

1. Generation of structured data
2. Transformation into natural language
3. Automatic generation of question-answer pairs
4. Validation and formatting

Our goal was to produce rich, coherent contexts and relevant questions whose answers could be reliably inferred. The dataset was designed to support two major question types:

- **Extractive questions**: where the answer is a direct span from the context.
- **Binary (Yes/No) questions**: where the answer must be inferred as either true or false.

To promote model generalization and reduce overfitting, we introduced variability at the data generation stage. Structured fields—such as company names, cities, street names, and postal codes—were randomly sampled from realistic, diverse pools. This randomness acted as a form of regularization, encouraging the model to learn generalizable patterns rather than memorizing surface forms.

3.1.1. Generation of Structured Data

We began by generating a synthetic set of 500 structured logistics records in JSON format. Each record captures shipment details, including pickup and delivery addresses, time windows, package specifications, and vehicle requirements. Placeholder values were randomly populated before training using curated lists and realistic numerical ranges.

Example:


```
{
  "pickup": {
    "company_name": "COMPANY_PICKUP",
    "postal_code": "ZIP_PICKUP",
    "street_name": "STREET_PICKUP",
    "datetime": "2026-05-11 19:00:00"
  },
  "delivery": {
    "company_name": "COMPANY_DELIVERY",
    "postal_code": "ZIP_DELIVERY",
    "street_name": "STREET_DELIVERY",
    "datetime": "2026-05-13 23:15:00"
  },
  "goods": {
    "weight": "143kg",
    "dimensions": [
```

```
      {
        "length": "21cm",
        "width": "21cm",
        "height": "58cm",
        "weight": "88kg"
      },
      {
        "length": "49cm",
        "width": "14cm",
        "height": "41cm",
        "weight": "55kg"
      }
    ]
  },
  "required_vehicle": "Sprinter",
  "special_request": "ADR requested"
}
```


3.1.2. From Structure to Natural Language

Using large language models (LLMs), we converted each structured JSON record into a coherent natural language paragraph suitable for QA. This step ensured that the resulting contexts resembled real-world text and could serve as meaningful inputs for question answering.

Example:

> We need to transport a shipment of goods from **COMPANY_PICKUP** in **ZIP_PICKUP**, **STREET_PICKUP** on May 11th at 19:00 to **COMPANY_DELIVERY** in **ZIP_DELIVERY**, **STREET_DELIVERY** on May 13th at 23:15.
>
> The shipment consists of two items with the following dimensions:
>
> - Item 1: Length 21cm, Width 21cm, Height 58cm, Weight 88kg
> - Item 2: Length 49cm, Width 14cm, Height 41cm, Weight 55kg
>
> The total weight of the shipment is 143kg. We require a Sprinter vehicle with ADR (dangerous goods) certification. Please ensure the driver is available at the specified pickup and delivery times.

3.1.3. Generation of Question-Answer Pairs

We used LLaMA 3.3 [14] to generate question–answer pairs for each context, producing up to 20 items per record across various question types.

Categories:

1. **Extractive questions**: where the answer is a continuous span of text that appears explicitly in the context.
2. **Binary (Yes/No) questions**: where the answer must be inferred from the context and is not typically stated verbatim. These questions require a semantic understanding of the text to determine whether the answer is ”Yes” or ”No”.

Example QA Pairs:

- **Q:** Where do we need to pick up the goods?
  A: COMPANY_PICKUP, STREET_PICKUP, ZIP_PICKUP
- **Q:** What type of vehicle is needed for the transport?
  A: Sprinter
- **Q:** How many packages are being transported?
  A: 2

Binary Examples:

- **Q:** Are the pallets stackable?
  A: No
- **Q:** Is the transport of dangerous goods requested?
  A: Yes

Heuristic rules were applied to detect malformed, irrelevant, or ambiguous pairs. We manually reviewed a subset to ensure quality.

The final dataset includes 4,723 extractive and 4,913 binary QA pairs. All pairs were stored in an extended *SQuAD-v2* format with metadata such as question type, answer span, and label.

3.1.4. Validation

Validation was crucial to ensure high data integrity. For extractive questions, we confirmed that the answer was an exact substring of the context. Binary answers were verified through logical consistency checks.

If a generated context did not yield at least one consistent QA pair, the entire context was discarded from the dataset. This ensured that only contexts with valid supervision signals were included. To further improve data quality, we applied heuristic filtering rules. For example, we removed QA pairs with empty or overly short answers, pairs where the question text duplicated fragments of the answer, and pairs with ambiguous references. A random subset of the filtered dataset was then manually reviewed to verify correctness.

3.2. Model Selection and Training

To evaluate performance across different model sizes, we selected three Transformer-based models from the Hugging Face QA catalog:

1. **XtremeDistil l12 h384 Uncased** (33M parameters): A distilled multilingual QA model optimized for efficiency [15].
2. **XLM-RoBERTa Base** (279M parameters): A robust encoder trained on CommonCrawl in 100+ languages [16].
3. **XLM-RoBERTa Large** (561M parameters): A deeper version of the base model with enhanced multilingual performance [16].

3.2.1. Dual-Head Architecture

To effectively handle both extractive and binary QA within a single framework, we extended the baseline Transformer models with a dual-head architecture:

- **Span prediction head:** This component follows the standard extractive QA design. Two linear layers predict the start and end positions of the answer span, applied to the hidden states of the encoder. The outputs correspond to indices within the input sequence, trained with cross-entropy loss.

- **Binary classification head:** To handle yes/no questions, we added an additional classification layer on top of the [CLS] token embedding. The [CLS] representation, which encodes global context of the input, is passed through a fully connected feed-forward layer with dropout, followed by a softmax activation producing probabilities for the two classes (”Yes” or ”No”). Training uses binary cross-entropy loss.

The two heads share the same Transformer encoder, allowing the model to transfer knowledge between tasks. Each head is trained with its own loss function, described in Section 3.2.3.

This shared design reduces training cost while facilitating cross-task knowledge transfer between extractive and binary QA.

3.2.2. Input Handling and Tokenization

Input sequences were limited to 512 tokens, which corresponds to the maximum sequence length supported by all selected models. For contexts exceeding this limit, we applied a sliding window approach with a stride of 256 tokens. This technique ensures that long passages are split into overlapping chunks, reducing the risk of missing relevant answer spans while keeping memory requirements manageable. Although these settings are widely adopted defaults, we acknowledge that alternative configurations (e.g., longer sequence lengths with models supporting extended contexts, or optimized stride values) could be explored in future work to further improve coverage and efficiency.

3.2.3. Loss Function

The total training loss $\mathscr{L}$ is a weighted combination of the span prediction loss and the binary classification loss:

- $\mathscr{L}_{\text{span}}$: cross-entropy loss for start/end token prediction.
- $\mathscr{L}_{\text{binary}}$: binary cross-entropy loss for Yes/No classification.

The combined loss is defined as:

$$\mathscr{L} = \lambda \cdot \mathscr{L}_{\text{span}} + (1 - \lambda) \cdot \mathscr{L}_{\text{binary}},$$

where λ controls the relative contribution of each component. In our experiments, we set $\lambda = 0.5$. This choice assigns equal importance to both tasks and balances the impact of the two loss components.

We did not systematically explore alternative values of λ; investigating its sensitivity may provide further insights into the balance between extractive and binary objectives.

4. Experiments and Results

This section presents the experimental setup, model configurations, and results for both extractive and binary QA tasks. While the core methodology has been described earlier, selected implementation details — particularly parts of the dataset generation pipeline and training scripts — are available in the accompanying GitHub repository[1].

4.1. Experimental Environment

All models were trained and evaluated on a server equipped with 2× NVIDIA A100-PCIE-40GB GPUs. The training pipeline was implemented in Python using the Hugging Face Transformers [17] and Datasets [18] libraries, with PyTorch [19] as the backend framework.

[1] https://github.com/macomatom/logistics-qa-pipeline

4.2. Training Configuration

Models were fine-tuned using model-specific hyperparameters:

- **XtremeDistil:** learning rate 1e-5, batch size 8, gradient accumulation 2 (effective batch size 16), 3 epochs, dropout 0.15 in both hidden and attention layers, weight decay 0.01, 500 evaluation steps.
- **XLM-RoBERTa Base:** learning rate 1e-5, batch size 16, gradient accumulation 2 (effective batch size 32), 3 epochs, dropout 0.1, weight decay 0.01, 200 evaluation steps.
- **XLM-RoBERTa Large:** learning rate 2e-5, batch size 16, gradient accumulation 2 (effective batch size 32), 3 epochs, dropout 0.1, weight decay 0.01, 250 evaluation steps.

All models used the AdamW optimizer with cosine learning rate scheduling and a warmup ratio of 0.1. The training loss was computed with $\lambda = 0.5$, giving equal weight to span-based and binary objectives. Evaluation based on validation loss was performed every 200–500 steps, and early stopping was triggered after three consecutive evaluations without improvement. Effective batch sizes were adjusted using gradient accumulation (1–2 steps).

These values were adopted from Hugging Face baselines and prior literature. We did not perform systematic hyperparameter tuning, which remains an open direction for future work.

4.3. Training Dynamics

Training dynamics are visualized in Figure 1, which shows the progression of training and validation loss for each model. XtremeDistil exhibits a smooth and gradual convergence over more than two epochs, stabilizing around a validation loss of 0.97. XLM-RoBERTa Base achieves its lowest validation loss (0.61) after roughly two epochs, with well-aligned training and validation curves. XLM-RoBERTa Large converges the fastest, reaching a validation loss of 0.53 in under one epoch, but displays slight fluctuations thereafter—possibly due to its higher sensitivity to noise in smaller batches.

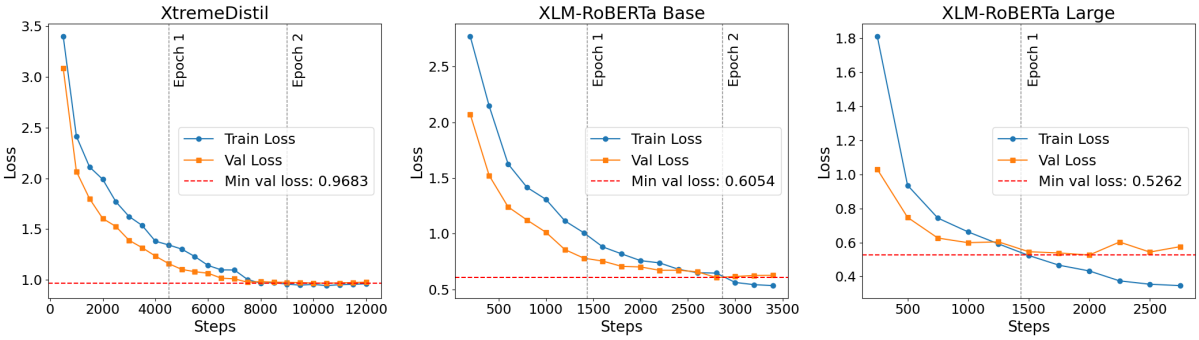

Figure 1: Training and validation loss progression across models.

4.4. Evaluation Metrics

To evaluate the performance of our models, we employed standard metrics tailored to the nature of each QA task. For extractive questions, we report Exact Match (EM), F1-score, precision, and recall, reflecting the model's ability to correctly identify and locate the answer span within the context. For binary (yes/no) questions, we use accuracy and F1-score, providing a balanced view of correctness and robustness in classification. These metrics are widely adopted in QA research and enable meaningful comparison across model sizes and task types. Additionally, confusion matrices were examined for the binary task to better understand the distribution of false positives and false negatives.

4.5. Extractive QA Performance

As shown in Table 1, all models demonstrate strong generalization in extractive QA, with F1 scores above 83%, and a clear accuracy gradient based on model size. The precision-recall balance remains consistent, with XLM-RoBERTa Large slightly favoring recall.

Model	EM	F1	Precision	Recall
XtremeDistil	62.4	83.4	81.1	85.8
XLM-RoBERTa Base	75.9	93.5	92.9	94.0
XLM-RoBERTa Large	82.3	95.5	95.4	95.6

Table 1
Results for extractive questions (in %).

4.6. Boolean QA Performance

In contrast, the binary QA task (see Table 2) reveals a sharper distinction between model capacities: the smaller XtremeDistil underperforms (F1 77.5%), while the two XLM-R models exceed 90% F1, demonstrating the difficulty of boolean classification for lighter architectures. This difference highlights the importance of model expressiveness for classification-type decisions in QA systems.

Model	Accuracy	F1	Precision	Recall
XtremeDistil	77.7	77.5	77.5	77.4
XLM-RoBERTa Base	90.2	90.1	90.1	90.3
XLM-RoBERTa Large	92.1	92.1	92.1	92.5

Table 2
Results for binary questions (in %).

4.7. Resource vs. Performance Trade-off

These results show that accuracy improves consistently with model capacity, with XLM-RoBERTa Large outperforming the other models on both extractive and boolean tasks. However, XLM-RoBERTa Base achieves nearly comparable F1-scores while requiring significantly fewer resources.

4.8. Impact of Question Length on Binary QA

As illustrated in Figure 3, binary QA performance improves with increasing question length. Longer questions appear to provide the model with richer semantic cues and more context for decision-making, resulting in higher F1-scores. This suggests that extended formulations help the model better capture the intent and relevant context of binary queries, ultimately enhancing classification accuracy.

4.9. Limitations and Error Analysis

While our results demonstrate that Transformer-based QA models can be successfully adapted for logistic data, several limitations must be acknowledged.

First, the dataset was generated synthetically using LLMs. Although heuristic filtering and partial manual validation were applied to reduce noise, occasional artifacts or hallucinated facts may persist. This limitation stems from the nature of LLM generation and causes. Consequently, the generated contexts are less diverse than naturally authored texts, which may reduce generalization to real-world scenarios.

Second, the structured records used to create synthetic contexts were generated by sampling fields independently from curated pools (e.g., company names, street names, vehicle types). While this

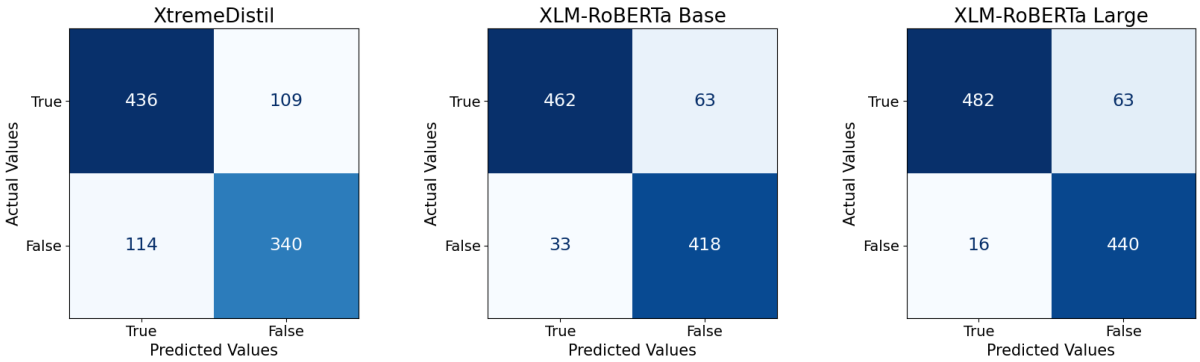

Figure 2: Confusion Matrices for binary questions classification.

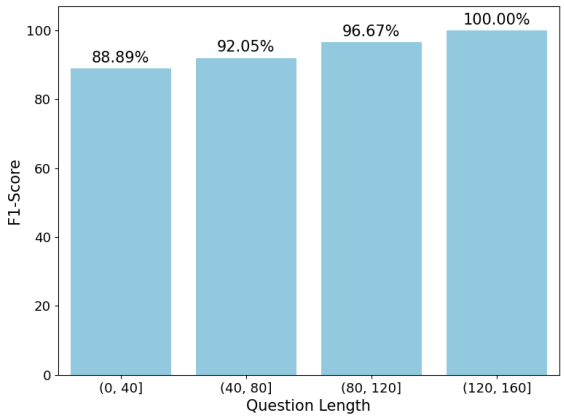

Figure 3: Performance of XLM-RoBERTa Large in binary questions by question length.

ensured diversity, it also limited realism, as attributes were not always internally consistent. The natural language conversions likewise reflect synthetic phrasing rather than the style of real customer communication.

Third, a limitation concerns the size of the dataset. Although it was generated from only 500 structured logistics records, the final corpus includes 4,723 extractive and 4,913 binary QA pairs. This apparent disproportion arises because each record was expanded into multiple questions covering different perspectives (e.g., locations, times, quantities, vehicle requirements). Although this increases diversity, it also means that the dataset is narrower in terms of unique contexts, which may limit the generalization of models to unseen real-world scenarios.

Beyond these dataset-related issues, we also analyzed the types of errors made by the models. The results showed that errors were not uniformly distributed but concentrated in a subset of particularly challenging instances. Many of these cases required reasoning over implicit relationships, such as comparing pickup and delivery dates or aggregating quantities across multiple items, which are not explicitly stated in the text. Interestingly, some errors recurred across all model sizes, suggesting that they stem from dataset-inherent complexities rather than limited model capacity. This indicates that current Transformer-based architectures struggle when questions demand logical inference beyond surface-level text matching, emphasizing the need for reasoning-augmented models and more diverse datasets.

5. Conclusion and Future Work

This paper addresses the challenge of the labor-intensive process of creating QA datasets for specialized domains. We present a comprehensive methodology for automatic dataset generation, demonstrated

on a logistics use case. By leveraging LLMs, we construct a high-quality QA dataset comprising both extractive and binary questions. This automated pipeline provides a scalable and efficient solution, significantly reducing manual effort while ensuring consistency and domain relevance.

Our proposed dual-head architecture enabled simultaneous handling of both QA tasks within a unified model. Through experimentation with models of varying capacities — from lightweight XtremeDistil to large-scale XLM-RoBERTa — we demonstrated that model size correlates strongly with performance, especially in binary QA. The results confirm that advanced multilingual Transformers can be effectively adapted for specialized domains like logistics, delivering high accuracy even with synthetic datasets.

Several directions offer promising opportunities for future work:

- **Multilingual QA:** Extend the dataset with new languages and compare cross-lingual performance.
- **New Question Types:** Incorporate additional QA formats to broaden task coverage, e.g. numbers, dates.
- **Data Variation:** Use augmentation to boost input diversity.
- **Span-Only Modeling:** Reformulate extractive QA task as span prediction task.

Overall, this work demonstrates the feasibility and scalability of combining LLM-powered data generation with fine-tuned Transformer models for domain-specific QA. Continued exploration of these extensions could further improve the adaptability and generalization of QA systems in real-world logistic applications.

Acknowledgments

This article was supported by the Scientific Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic under contract *VEGA 1/0539/25*.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] L. Hirschman, J. M. Prager, D. R. Radev, Question answering: Beyond the trec tracks, Natural Language Engineering 7 (2001) 291–302.

[2] D. I. Moldovan, M. Pasca, S. M. Harabagiu, R. Girju, The lcc d.o.e. trec-2003 question-answering system, Proceedings of the Text REtrieval Conference (TREC) (2003).

[3] D. A. Ferrucci, E. W. Brown, J. Chu-Carroll, H. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. M. Prager, et al., Building watson: an overview of the deepqa project, AI Magazine 31 (2010) 59–79.

[4] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, Advances in Neural Information Processing Systems 30 (2017).

[6] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[7] Y. Liu, M. Ouyang, D. Wu, Z. Ma, X. Duan, X. Chen, W. Chen, X. Shen, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[8] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, Electra: Pre-training text encoders as discriminators rather than generators, arXiv preprint arXiv:2003.10555 (2020).

[9] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, arXiv preprint arXiv:1909.11942 (2020).

[10] V. Sanh, T. Wolf, L. Debut, J. Chaumond, C. Delangue, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).

[11] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, Squad: 100,000+ questions for machine comprehension of text, arXiv preprint arXiv:1606.05250 (2016).

[12] Z. Yang, P. Qi, S. Zhang, S. Peng, X. Dai, X. Ma, Z. Cai, S. Zhou, J. Lin, J. Sun, Hotpotqa: A dataset for diverse question answering with multi-hop reasoning, arXiv preprint arXiv:1809.09600 (2018).

[13] T. Kwiatkowski, J. Palomaki, M. Reddi, M. Collins, A. Parikh, C. Albert, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, et al., Natural questions: a benchmark for question answering research, Transactions of the Association for Computational Linguistics 7 (2019) 450–466.

[14] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Rozière, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. M. Kloumann, I. Misra, I. Evtimov, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, et al., The llama 3 herd of models, CoRR abs/2407.21783 (2024). URL: https://doi.org/10.48550/arXiv.2407.21783.

[15] S. Mukherjee, A. Hassan Awadallah, Xtremedistil: Multi-stage distillation for massive multilingual models, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 2221–2234. URL: https://aclanthology.org/2020.acl-main.202/. doi:10.18653/v1/2020.acl-main.202.

[16] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8440–8451. URL: https://aclanthology.org/2020.acl-main.747/. doi:10.18653/v1/2020.acl-main.747.

[17] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2020, pp. 38–45. URL: https://aclanthology.org/2020.emnlp-demos.6. doi:10.18653/v1/2020.emnlp-demos.6.

[18] Q. Lhoest, A. V. del Moral, Y. Jernite, A. Thakur, P. von Platen, S. Patil, J. Chaumond, M. Drame, J. Plu, C. Xu, T. L. Scao, V. Sanh, L. Tunstall, L. Debut, T. Wolf, A. Rush, Datasets: A community library for natural language processing, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2021, pp. 175–184. URL: https://aclanthology.org/2021.emnlp-demo.21. doi:10.18653/v1/2021.emnlp-demo.21.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems 32 (2019).