Article

When to Use Large Language Model: Upper Bound Analysis of BM25 Algorithms in Reading Comprehension Task

Tingzhen Liu[1]*, Qianqian Xiong[2] and Shengxi Zhang[3]

[1] CROS, Tencent IEG
[2] College of Electromechanical and Information Engineering, Shandong Univerisity
[3] ANU Joint Science College, Shandong Univerisity
*Corresponding author: firstsg@outlook.com

Abstract: Large language model (LLM) is a representation of a major advancement in AI, and has been used in multiple natural language processing tasks. Nevertheless, in different business scenarios, LLM requires fine-tuning by engineers to achieve satisfactory performance, and the cost of achieving target performance and fine-tuning may not match. Based on the Baidu STI dataset, we study the upper bound of the performance that classical information retrieval methods can achieve under a specific business, and compare it with the cost and performance of the participating team based on LLM. This paper gives an insight into the potential of classical computational linguistics algorithms, and which can help decision-makers make reasonable choices for LLM and low-cost methods in business R&D.

Keywords: Large Language Model; Natural Language Processing; Reading Comprehension; Computational linguistics; Information Retrieval; BM25

1. Introduction

At present, large language model (LLM) is widely used in various tasks of natural language processing, and has achieved the performance of state-of-the-art on many mainstream lists [1,2,3]. However, this does not mean that LLM can achieve good results in various specific businesses. Engineers need to fine tune the business data to determine the performance that LLM can achieve. Whether the final effect matches the cost of manpower and calculation required for fine-tuning is a problem that technical decision-makers need to consider.

Compared with LLM, classical computational linguistic features are considered to be unable to describe semantics well [4]. But its advantage is that the development and calculation costs are very low. Therefore, decision makers need to make a choice between LLM and classical methods according to business requirements. For example, Zhu et al.[5,6] compiled a series of benchmarks for CTR tasks to help engineers determine whether certain methods are competitive in business. Weimao et al.[7] sorted out multiple benchmarks of text classification tasks. They conducted classification experiments on this basis, so as to compare and analyze the advantages and disadvantages of various methods. Lai T M et al.[8] proposed a baseline for coreference resolution, which provides evidence for the necessity of justifying the complexity of existing or newly proposed models. Huliyah et al.[9] compared the benchmark of random forest and naive bayes algorithm to know which modeling process has the best value of accuracy for sentiment classification in texts. Salemi et al.[10,11] introduced RTAnews benchmark dataset and conducted extensive benchmarking tests of most of the well-known multi-label learning algorithms for Arabic text classification, so as to compare the effectiveness of these algorithms. Naseem et al.[12] benchmarked the performance of different state-of-the-art ML text classification mechanisms, which can assist

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 January 2023 doi:10.20944/preprints202301.0219.v1

governments worldwide in analyzing public sentiment and its dynamic during the pandemic, so as to plan effective public health responses.

In this paper, we study the performance upper bound of classical computational linguistic metrics for specific tasks. We conduct research based on the contest model of Baidu Search Technology Innovation Challenge 2022 (STI) [13]. This competition suggested that participating teams make fine-tuning based on LLM ERNIE [14] proposed by Baidu to complete the reading comprehension task under specific business data. But we study the performance upper bound of classical information retrieval methods under this business task, and compare with the cost and performance of participating teams based on LLM. Our work attempts to provide insight into the potential of classical computational linguistics algorithms to help decision-makers make reasonable choices in business development.

2. Reading Comprehension Task of STI

Baidu Search Technology Innovation Challenge 2022 requires the participating teams to complete several reading comprehension tasks based on ERNIE, and the dataset provided was built on the basis of Baidu search desensitization business data.

ERNIE is the NLP pre-training model proposed by Baidu. It has been proven to have superior performance over BERT in various Chinese NLP tasks such as named entity recognition and natural language inference. The application of ERNIE has significantly improved the performance of Baidu's decimated intelligent question answering system. However, in the open domain search scenario, there are problems such as the length of web documents is different, the quality level is uneven, the length of questions and answers is long, and the distribution is scattered. This brings challenges to the extraction of answers and the calculation of answer confidence.

The dataset provided by the competition contains the training set, the verification set and the test set. Among them, the training set contains about 900 queries and 30,000 query-document pairs; the verification set and test set each contains about 100 queries and 3.000 query-document pairs. The main characteristics of the data are:

- The length of documents is generally long, the quality is uneven, and there is often a lot of noise inside the document
- Answer fragments are of sentence-level. An answer usually consists of several sentences that contain the complete context
- The annotated data only guarantees the relevance between answer fragments and searched questions. It does not guarantee correctness, and there exist documents that do not contain any answer

The competition task requires that the participating team should take the query and the document in the test set as model input, find reasonable answer fragments in the corresponding document according to the query, and output them as answer sets after integration. More specifically, the test set gives a set of searched questions. Based on each searched question Q, a set of web documents D retrieved by a search engine is given, including up to 40 web documents. For each (D, Q) pair, the contest model is required to extract answer fragments A that can answer the query Q from D. If the model forecasts that documents do not contain any answer, "NoAnswer" is returned. The contest model should achieve good and robust answer extraction effect in the data environment with variable document lengths and long answer lengths.

Evaluation Metrics

Each contest model predicts the answers of each (D, Q) pair, and submits them to the evaluation system after integration. The evaluation system calculates precision, recall and F1 value according to the character granularity of the answer corresponding to each (D, Q) pair and the character granularity of groundtruth. Ultimately it takes the average F1 valve of test data as final score of a contest model.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 January 2023 doi:10.20944/preprints202301.0219.v1

When both the standard answer and predicted answer are "NoAnswer", P, R and $F1$ are all 1; when only one of them is not "NoAnswer", P, R and $F1$ are all 0; when they are not "NoAnswer", the evaluation system calculates the character-granularity similarity of the two texts:

Firstly remove the punctuation marks in the two texts, and then calculate precision P and recall R:

$$P = L_c/L_1 \quad (1)$$

$$R = L_c/L_2 \quad (2)$$

Among which L_1 and L_2 are the number of characters in the standard answer and predicted answer, and L_c is the number of characters of the same kind in both. On the basis of getting P and R, F1 is further calculated as a comprehensive index to measure the performance of the model:

$$F1 = 2 \times P \times R/(P + R) \quad (3)$$

The baseline provided by STI is based on ERNIE with full fine-tuning. Based on the evaluation method above, the evaluation results corresponding to baseline are as follows:

Table 1. Evaluation Result of Baseline.

F1	Precision	Recall
0.35335	0.36416	0.48317

Upper Bound Analysis of BM25

In this paper, we analyze the upper bound that the classical information retrieval method BM25 [16] can reach on the reading comprehension task. It is generally believed that such classical metrics are only calculated based on the statistics of word frequency. They can only calculate the relevance of text without understanding semantics, so they are not suitable for reading comprehension tasks. However, the raw results (as shown in Table 1) show that the end-to-end generation of ERNIE is not good without meticulous fine-tuning.

BM25 calculates the correlation between text paragraphs and queries. Because we only filter text based on BM25 metrics, and do not use other metrics. Therefore, the algorithm has no ability to exclude texts that are highly relevant but cannot answer questions. We can only exclude the text that cannot answer the question because of low relevance based on the minimum BM25 score related to the query. The steps of this process are:

1) Split the document D into a set of paragraphs
2) Calculate the BM25 score of the corresponding query Q for this set of paragraphs (each paragraph will have a BM25 score)
3) Filter out the paragraph whose BM25 score is greater than t as an answer to the query

The key point of this process is that for each (D, Q) pair, what value should threshold t take to obtain the best result. In the training data, we have groundtruth answer A for each (D, Q) pair. For document A, we can map each paragraph to one of the most similar paragraphs in D. This gives an BM25 estimate for each paragraph in the document A. Let:

$$A = v_1, v_2, \dots\dots, v_n$$

Among which v_i is the BM25 estimate of the i^{th} paragraph in document A. Obviously, the minimum is the correlation boundary of "whether the paragraph is in the answer". Namely:

$$t = min(A)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 January 2023 doi:10.20944/preprints202301.0219.v1

To implement this method, we also need to consider how to calculate the most similar paragraph of A in D. Same as the evaluation indicators of the task, for each paragraph s_i in D, we calculate the character-granularity similarity with it and each paragraph in A. If the similarity is less than threshold t', the paragraphs are regarded as similar paragraphs:

1. For s_i in D
2. For s_i' in A
3. If $CharSimilarity(s_i, s_i') < t'$
4. Add s_i' to $SimilarParagraphs$

However, the problem with the algorithm is that some groundtruth answers in the reading comprehension task are derived directly from candidate document D, while some groundtruth answers reorganize the language. So for each (D, A) pair, the threshold t' is different. To solve the problem, we first set an initial value for t', and if no similar paragraphs are found, t' is be automatically adjusted downward by step τ. Therefore, the final algorithm is:

1. For s_i in D
2. For s_i' in A
3. If $CharSimilarity(s_i, s_i') < t'$
4. Add s_i' to $SimilarParagraphs$
5. If $SimilarParagraphs$ is empty
6. $t' \leftarrow t' - \tau$
7. Repeat this algorithm

t' and τ are hyperparameters of this algorithm. Due to $CharSimilarity(s_i, s_i') \in [0,1]$, we can try all combinations of $t' \in [0.5,0.99]$ and $\tau \in [0.01,0.1]$. The algorithm will search hyperparameter that can make the evaluation result best. The effect achieved at this time is the upper bound of our estimation. The final result is:

Table 2. Final Evaluation Result of BM25 Method's Upper Bound.

$\mathbf{t'}$.	τ	F1	Precision	Recall
0.9	0.1	0.61666	0.64045	0.68939

3. Experiment Result

Performance Comparison

Table 3 is the evaluation list of STI's 168 participating teams, where the red row is the upper bound evaluation score of BM25 method:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 January 2023 doi:10.20944/preprints202301.0219.v1

Table 3. Part of The STI 2022 Ranking List[17].

Rank	F1	Precision	Recall
1	0.70154	0.737	0.72145
2	0.68987	0.7012	0.73348
⋮	⋮	⋮	⋮
35	0.61999	0.70506	0.63367
36	0.61878	0.70452	0.6308
37	0.61705	0.6828	0.62577
	0.61666	0.64045	0.68939
38	0.61647	0.70245	0.62423
39	0.61595	0.7079	0.61761
40	0.61532	0.70212	0.6185
⋮	⋮	⋮	⋮
108	0.55916	0.63066	0.56449
109	0.55916	0.63066	0.56449
110	0.52056	0.63029	0.49799
111	0.44279	0.49423	0.44157
112	0.44069	0.49664	0.43897
113	0.41693	0.54303	0.39978
⋮	⋮	⋮	⋮
147	0.35335	0.36416	0.48317
148	0.35335	0.36416	0.48317
⋮	⋮	⋮	⋮

It can be seen that an obvious gap has been formed between the F1 scores of the 110th and 111th ranked team. We take the gap position and the upper bound F1 score as the boundaries to roughly divide the ranking scores into three grades. They are the first grade: 1st-37th, the second grade: 38th-110th and the third grade: 111th-168th. Here we use the average value of F1 scores of multiple contest models to represent their overall performance. The average value of all F1 scores in the ranking list is 0.52851, which is 0.08815 lower than the upper bound result; the average F1 score of the first grade is 0.64910, which is 0.03244 higher than the upper bound result; the average F1 score of the third grade is 0.36200, which is 0.25466 lower than the upper bound result. The F1 score corresponding to the baseline is 0.00865 lower than the average value of the third grade F1 scores, and about 11.9% of contest models fail to surpass it.

According to the requirements of the competition, most of the contest models are fine tuned based on ERNIE, but their effects cannot exceed the BM25 upper bound results. This means that most teams have paid excess costs on LLM but failed to achieve better results. Only 22% of the contest models achieved better evaluation scores than BM25 upper bound results. This result reflects the uncertainty of fine-tuning LLM for specific tasks.

Calculation Cost Comparison

In order for the BM25 algorithm to reach this upper bound, it is necessary to accurately select the threshold t according to the groundtruth data. If this problem is regarded as the problem of stopping point estimation of autocorrelation random process, by using the method proposed by Liu et al.[18], the sequence pattern of the BM25 score can be learned to infer the threshold t for each (D, Q) pair.

We compare the calculation cost based on this method with that of fine-tuning ERNIE. According to statistics, participating teams train 118 epochs on average, and the calculation duration on V100 GPU is 2.5h (only the last training-inferencing duration is counted, excluding the time for hyperparameter adjustment). Since the method in [18] only models the BM25 score sequence, its network size is far smaller than ERNIE. The network only needs 0.884h to train 118 epochs on RTX3060 GPU. Considering that the performance gap between the V100 and the RTX3060 is more than 1.23 times (FP32

 doi:10.20944/preprints202301.0219.v1

floating-point performance estimation is used here [19]), this means that the cost of fine-tuning LLM is at least 3.48 times of the model in [18].

However, it was also pointed out in [18] that only using BM25 sequence on STI dataset maybe could not provide enough information for inferring stopping point. This means that the upper bound cannot be reached directly by the model in [18], and text is still needed to provide auxiliary information. Therefore, the network structure that reaches the upper bound is more complex than this one. The performance cost estimate made here does not represent the final result.

4. Conclusion

According to experimental results, the effect of fine-tuning LLM in specific tasks is uncertain. For instance, out of 168 participating teams, most teams have not got satisfactory fine-tuning results, and only 37 teams are able to make their results beyond the upper bound of the BM25 method. Whereas the average computility they use far exceeds the encoder architecture of the estimated BM25 sequence stopping point, which means that paying excessive costs on LLM does not necessarily lead to better effects. Basing on this observation, we recommend that technical decision-maker reasonably determine the expected indicators firstly that the algorithm needs to achieve according to business requirements. If fairly precise results are not required, whether costs should be spent on LLM needs to be carefully considered.

References

[1] Chen W . Large Language Models are few(1)-shot Table Reasoners[J]. arXiv e-prints, 2022.

[2] Xu R, Luo F, Zhang Z , et al. Raise a Child in Large Language Model: Towards Effective and Generalizable Fine-tuning[J]. Association for Computational Linguistics, 2021.

[3] Wei J , Tay Y , et al. Emergent Abilities of Large Language Models[J]. arXiv e-prints, 2022.

[4] Manshadi M H . Towards a Robust Deep Language Understanding System[C]// Twenty-fourth Aaai Conference on Artificial Intelligence. DBLP, 2010.

[5] Zhu J, Liu J, et al. Open Benchmarking for Click-Through Rate Prediction, in Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM), 2021.

[6] Zhu J, Mao K, et al. BARS: Towards Open Benchmarking for Recommender Systems, in Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2022.

[7] Weimao, Ke. Least information document representation for automated text classification[J]. Proceedings of the American Society for Information Science & Technology, 2013.

[8] Lai T M, Bui T, Kim D S . End-to-end Neural Coreference Resolution Revisited: A Simple yet Effective Baseline[J]. 2021.

[9] Hulliyah K, Bakar N, Ismail A R , et al. A Benchmark of Modeling for Sentiment Analysis of The Indonesian Presidential Election in 2019[C]// 2019 7th International Conference on Cyber and IT Service Management (CITSM). 2019.

[10] Al-Salemi B, Ayob M, Kendall G , et al. RTAnews: A Benchmark for Multi-label Arabic Text Categorization. 2018.

[11] Al-Salemi B, Ayob M, Kendall G , et al. Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms[J]. Information Processing & Management, 2019, 56(1):212-227.

[12] Naseem U, Razzak I, Khushi M , et al. COVIDSenti: A Large-Scale Benchmark Twitter Data Set for COVID-19 Sentiment Analysis[J]. IEEE Transactions on Computational Social Systems, 8(4):1003-1015.

[13] Baidu AI Studio. Baidu Search Technology Innovation Challenge 2022[EB/OL]. December 6, 2022. http://sti.baidu.com/

[14] Sun Y, Wang S, Feng S , et al. ERNIE 3.0: Large-scale Knowledge Enhanced Pre-training for Language Understanding and Generation[J]. arXiv e-prints, 2021.

[15] Liu T, Xiong Q, Zhang S. STI BM25 Sequence Dataset[Dataset]. figshare. 2023. https://doi.org/10.6084/m9.figshare.21321198.v2

[16] Robertson S, Zaragoza H . The Probabilistic Relevance Framework: BM25 and Beyond[J]. Foundations & Trends in Information

Retrieval, 2009, 3(4):333-389.

[17] Xiong Q. Ranking List of STI[Dataset]. figshare. 2023. https://doi.org/10.6084/m9.figshare.21835719.v2

[18] Liu T , Zhang S, Xiong Q. Sequence Encoder for Stopping Point Prediction of Autoregressive Processes[J]. ResearchGate, 2022.

[19] Gadget Versus. Processor[EB/OL]. February 25, 2021. https://gadgetversus.com/graphics-card/nvidia-tesla-v100-pcie-16gb-vs-nvidia-geforce-rtx-3060/