PROP: Pre-training with Representative Words Prediction for Ad-hoc Retrieval

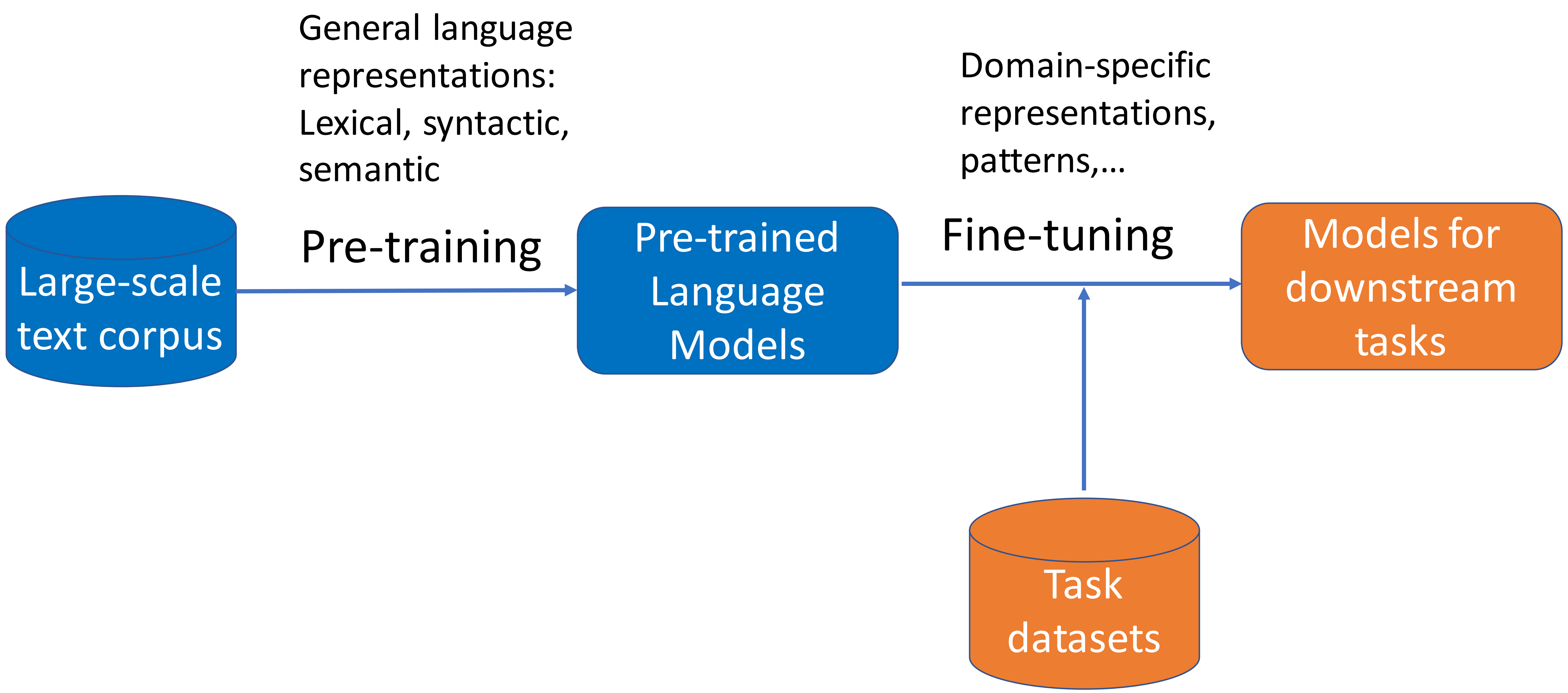
Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji and Xueqi Cheng

1. CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences

2. University of Chinese Academy of Sciences

New Paradigm of NLP

- Pre-training and then fine-tuning paradigm
- Significant benefit for tasks with limited training data

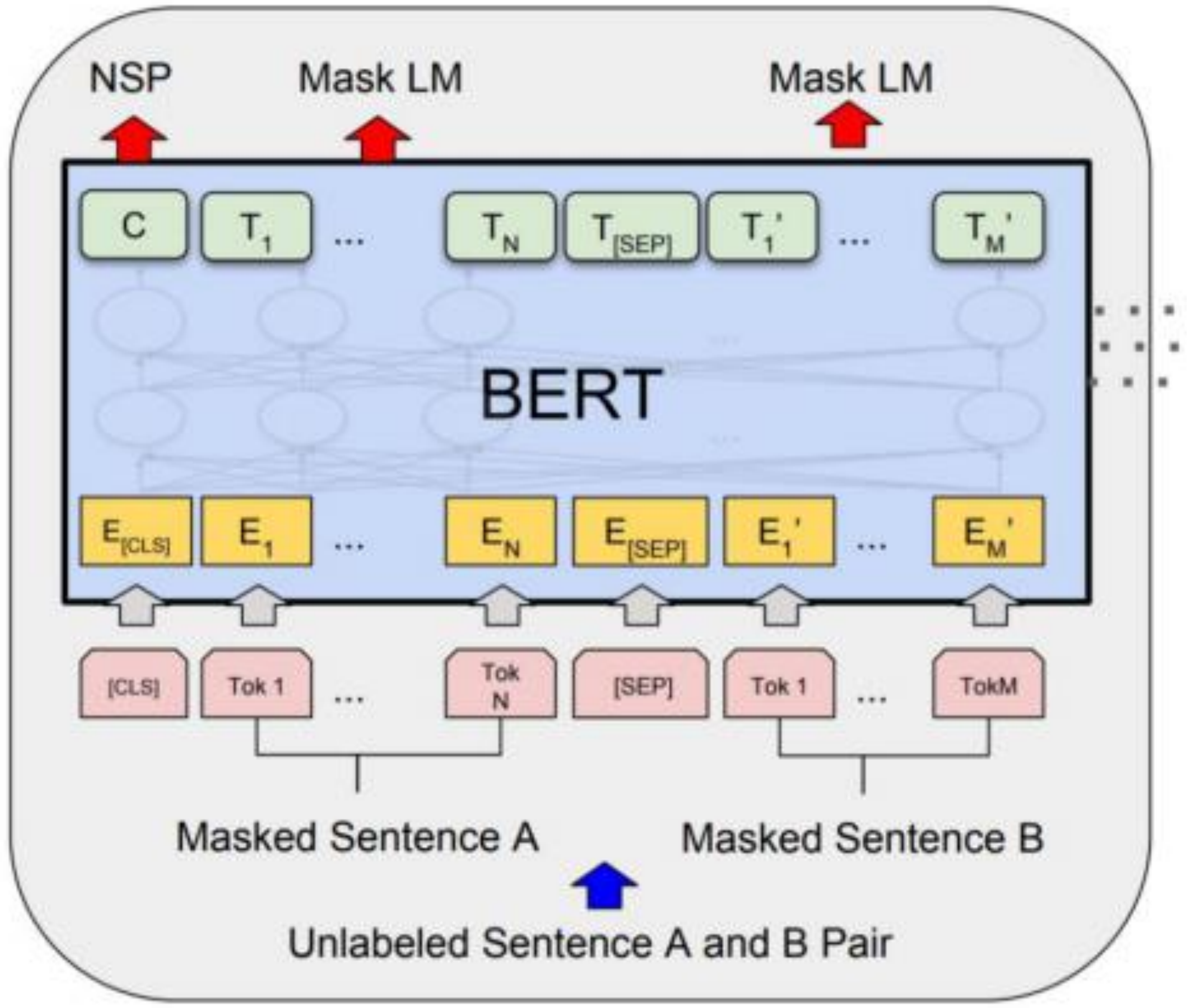


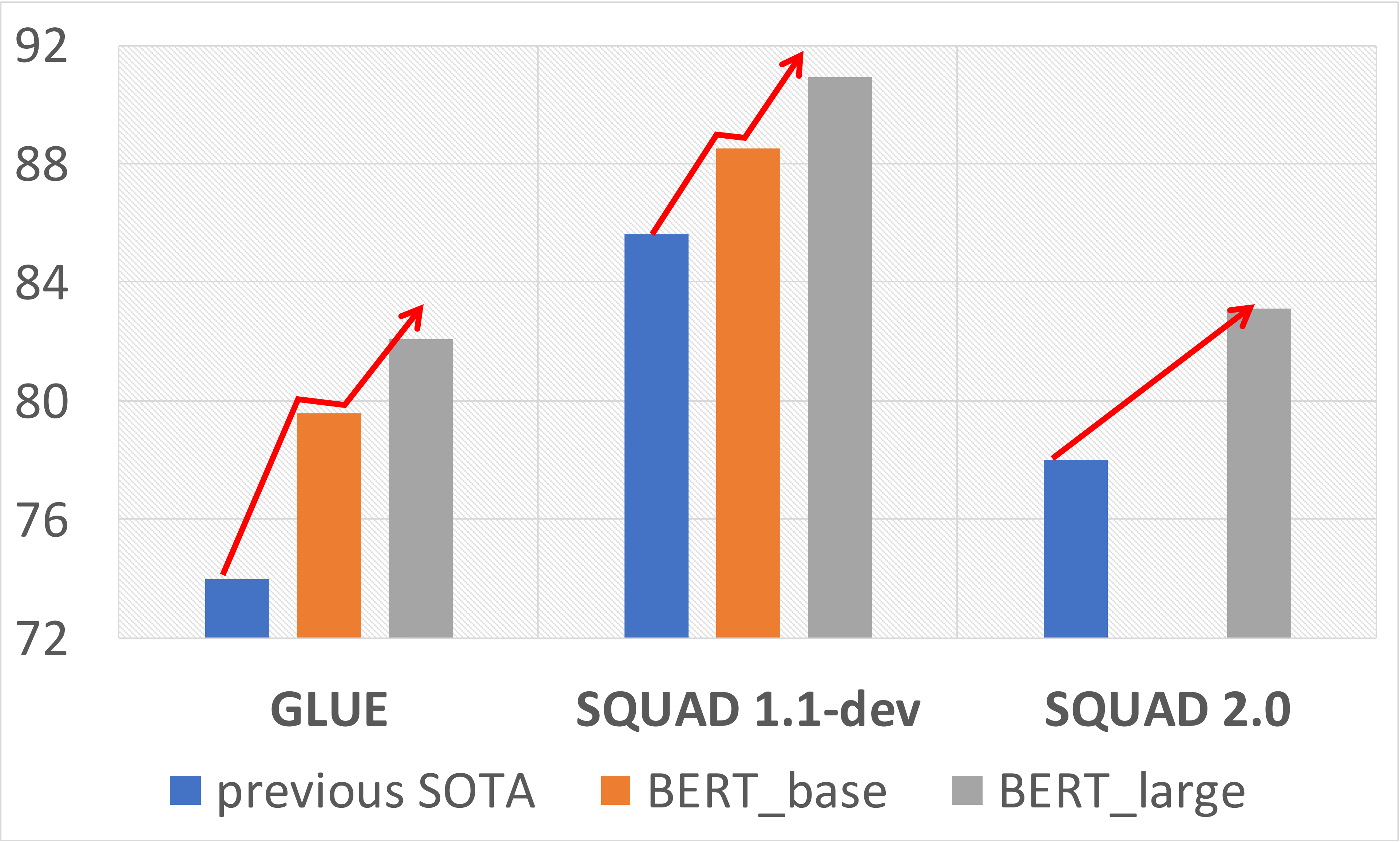
NLP Tasks

- Machine Translation
- Sentiment Analysis
- Question Answering
- Dialogue & Chatbot
- Textual Entailment
- Paraphrasing
- Semantic Parsing
- …

BERT

- BERT: Bidirectional Encoder Representations from Transformers



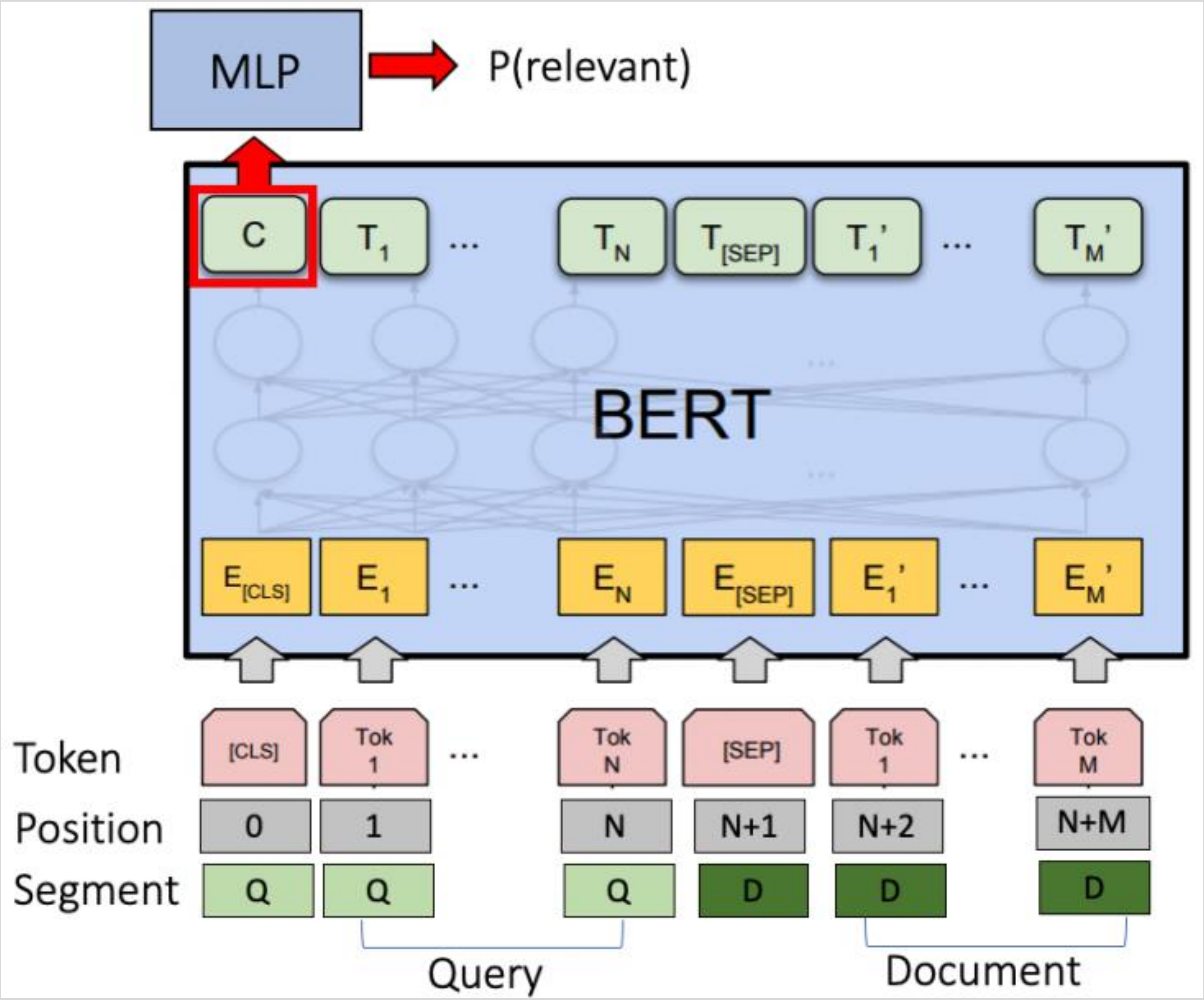


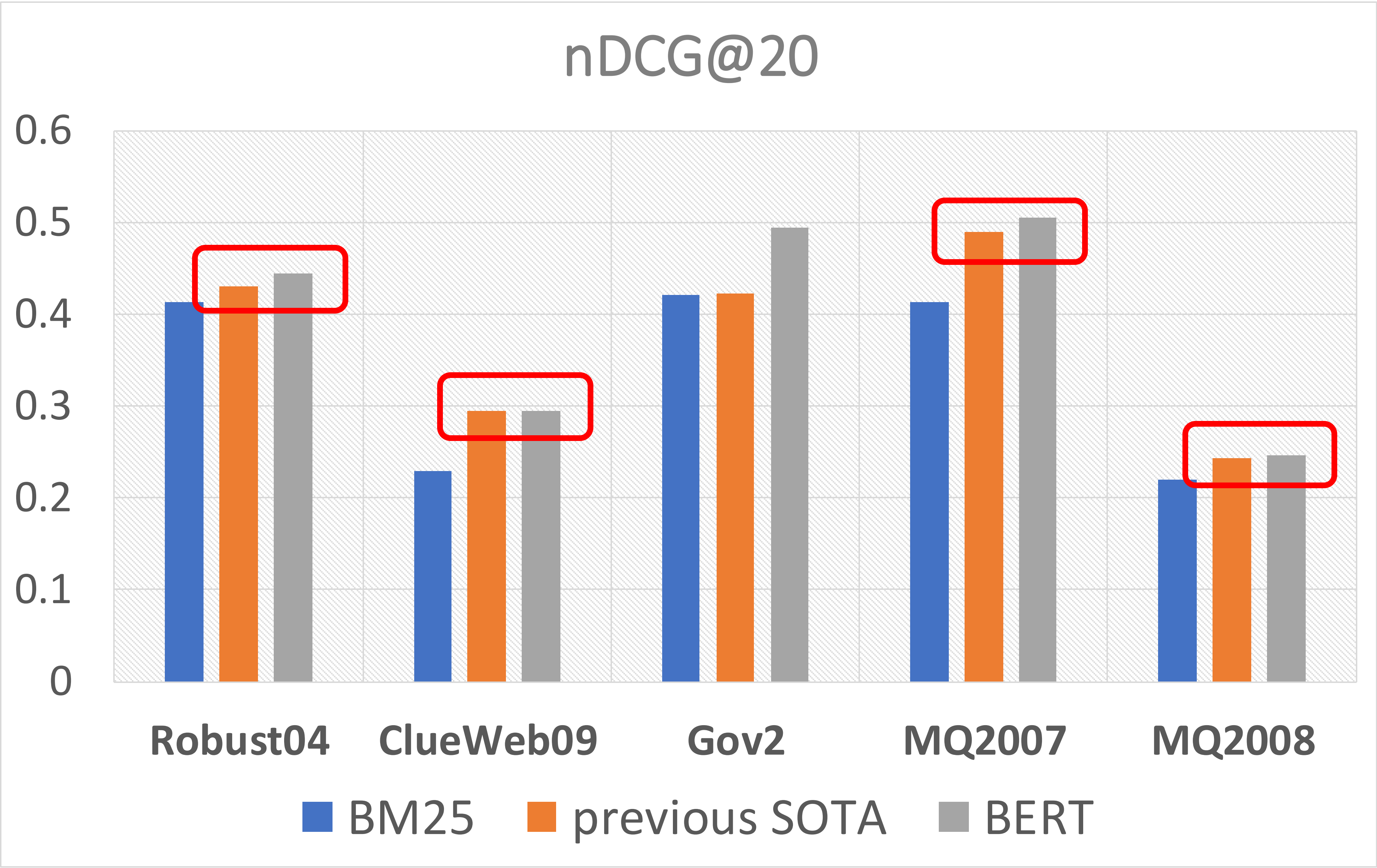
- Pre-trained with mask language model and next sentence prediction on Wikipedia and BookCorpus.
- A comparison of BERT with previous SOTA on GLUE, SQUAD 1.1, SQUAD 2.0, from Devlin et.al.

- BERT outperform previous SOTA on many natural language understanding tasks.

BERT for Information Retrieval

- Directly applying BERT to IR





- Usage of BERT for IR. Concatenate query and document, take [CLS] for relevance computation
- A comparison of BERT with BM25 and previous SOTA on downstream IR tasks.

- **Pre-trained models also benefit the search tasks, but not very significant**

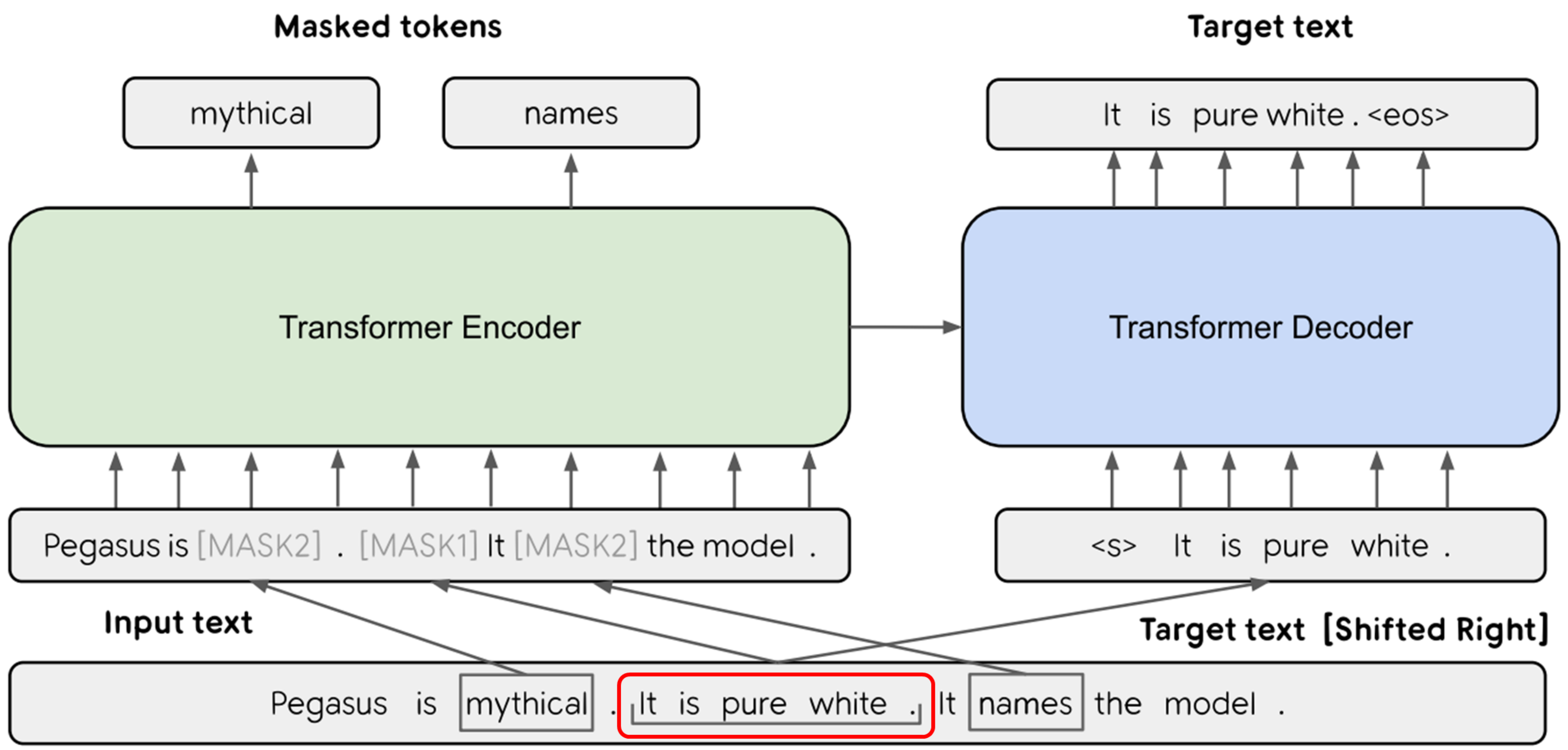
Observation

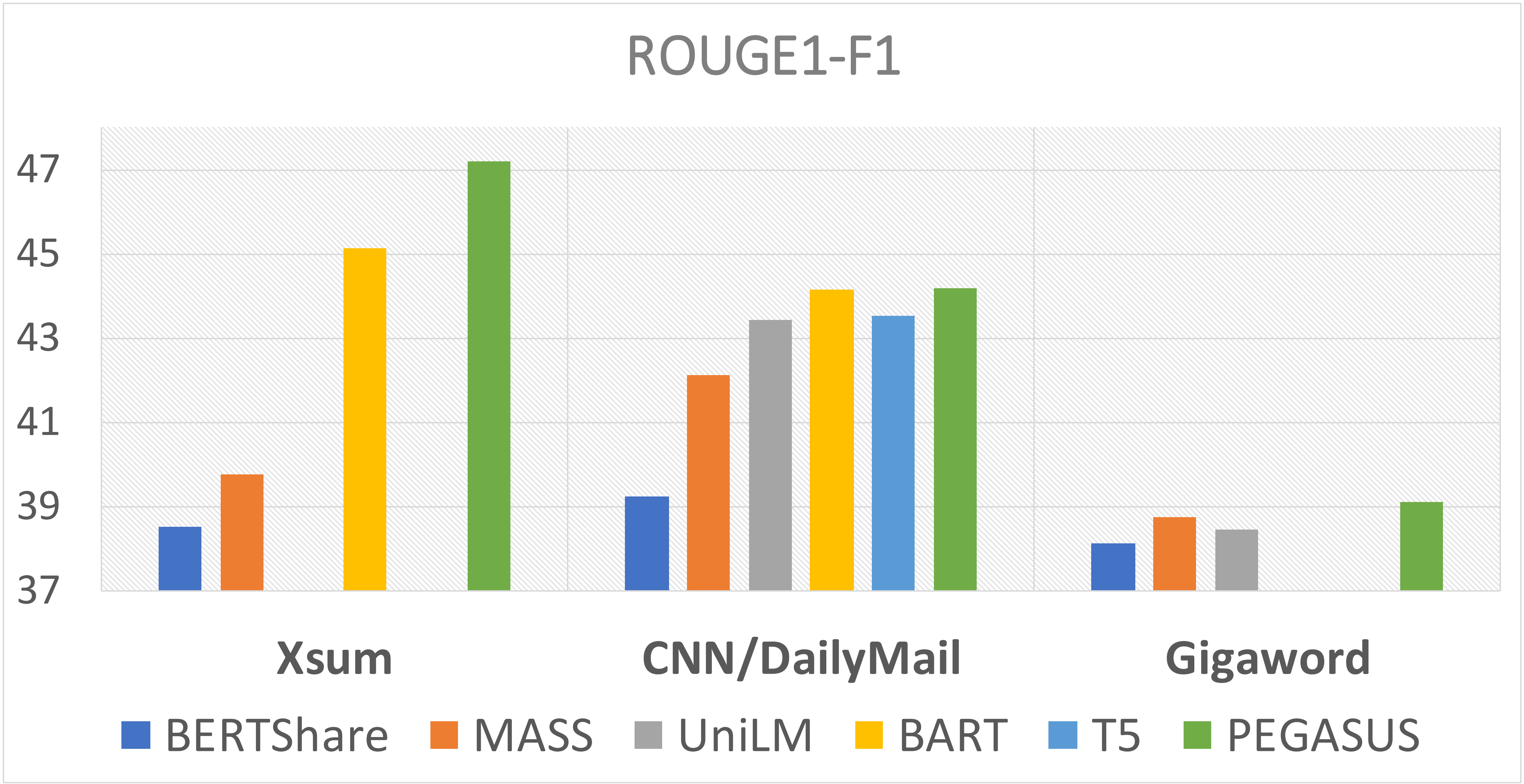
- Pegasus for Abstractive Summarization
- SSPT for Question Answering
- SentiLARE for sentiment analysis
- ERNIE (THU) for entity-related tasks
- …

The pre-training objective that more closely resembles the downstream tasks leads to better and faster fine-tuning performance.

Pegasus for Abstractive Summarization

- **Gap sentence generation** (GSG): selected by ROUGE scores





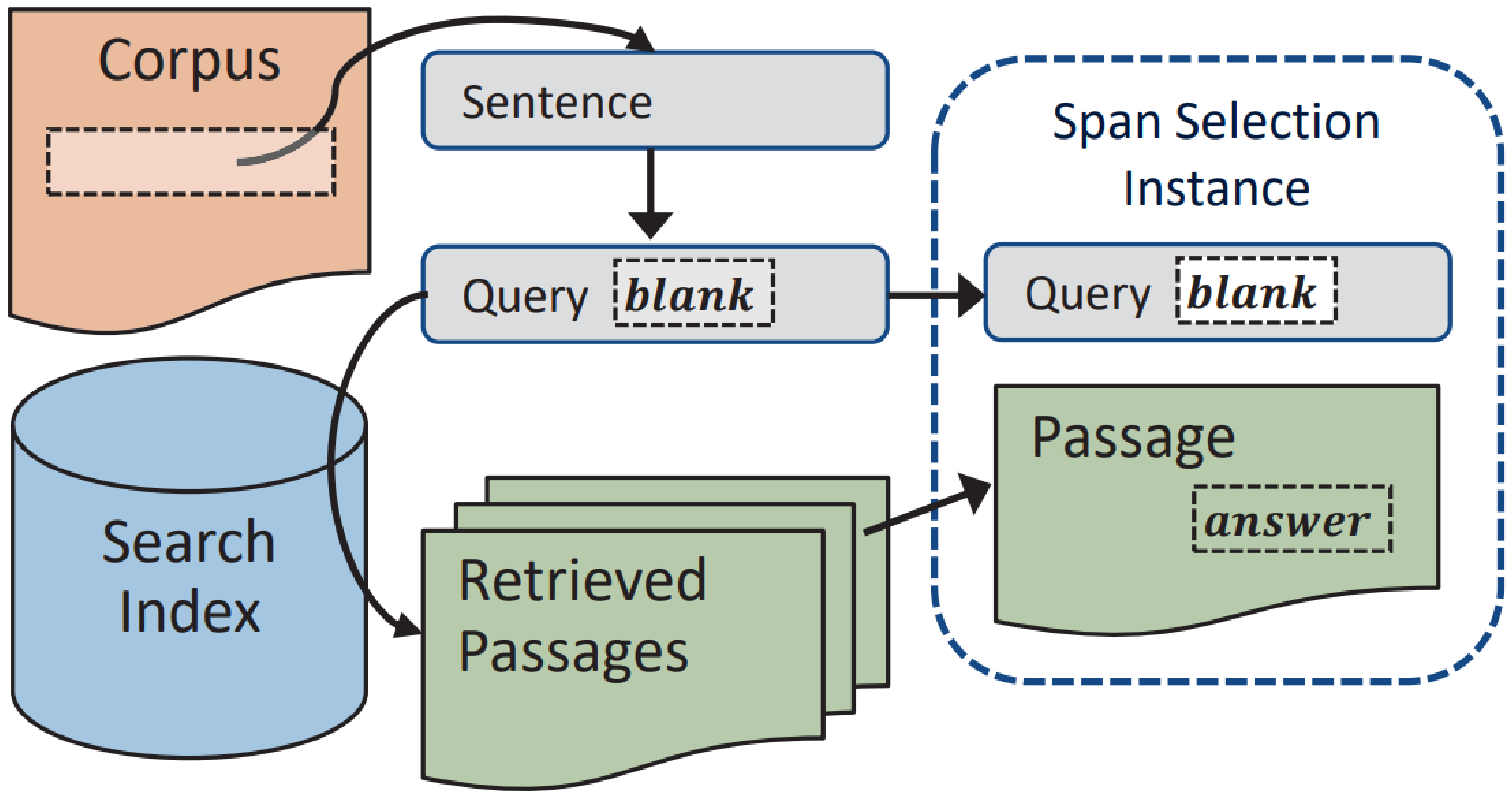
- One sentence is masked with [MASK1] and used as target generation text (GSG).
- A comparison of PEGASUS with other pretrained models on XSum, CNN/DailyMail and Gigaword.

- Pegasus is significantly better than other pre-trained models

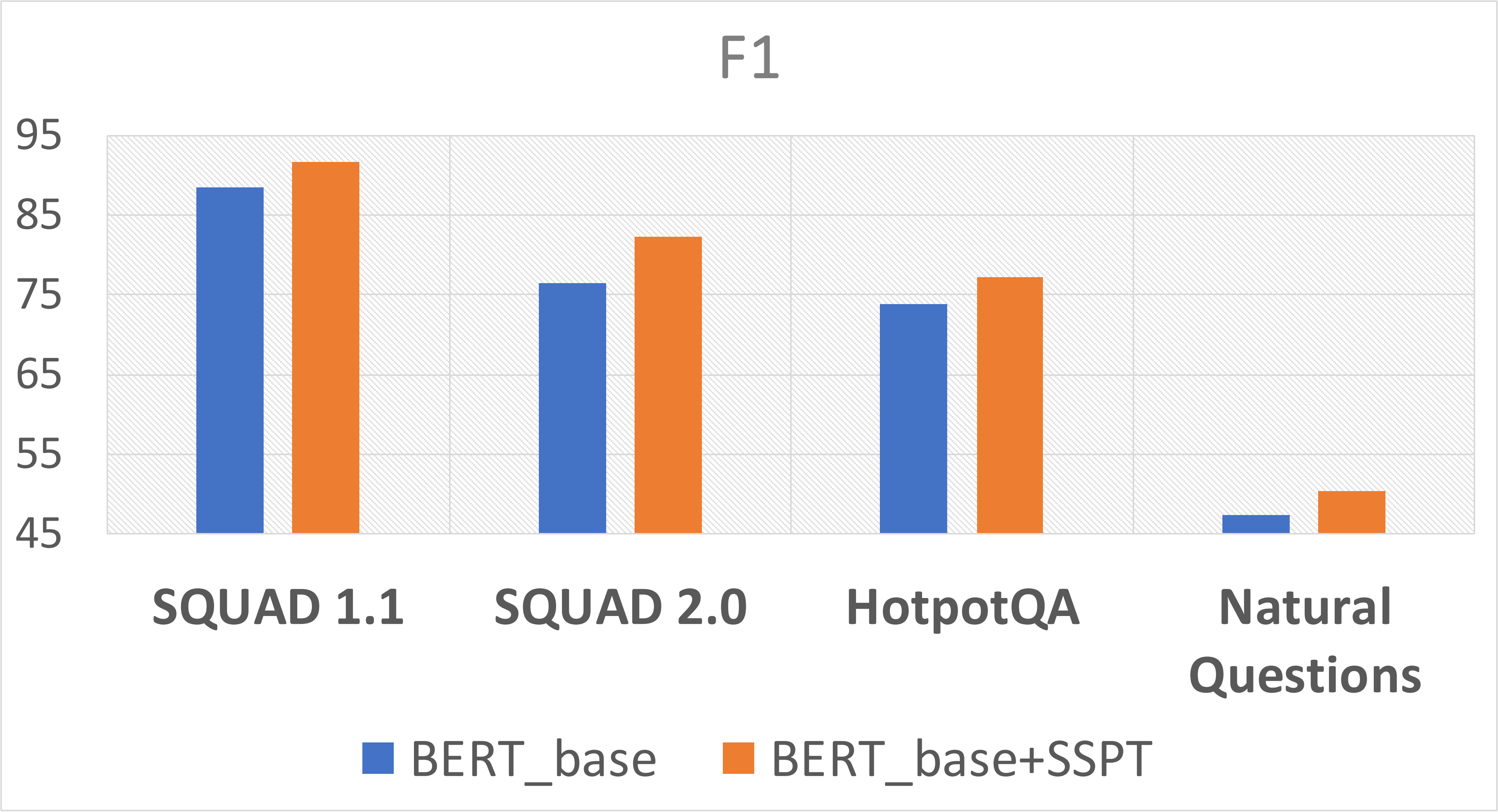
PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization, ICML, 2020

SSPT for Question Answering

- **Span Selection PreTraining** (SSPT): predict masked span(noun phrase or entity, pseudo answer), jointly pre-training with MLM



- Span selection training instance generation. Masked span will be predicted by the passage containing it.



- A comparison of BERT+SSPT with BERT on SQUAD 1.1, SQUAD 2.0, HotpotQA and Natural Questions.

- BERT+SSPT is significantly better than original BERT

Span Selection Pre-training for Question Answering, ACL, 2020

However, pre-training objectives tailored for ad-hoc retrieval have not been well explored.

Revisit the Pre-training Objectives

IR requirements

Sequence-based tasks:

- Masked Language Modeling
- Permuted Language Modeling

Learn contextual representations

✓ →

Good representations for the query and the document

Sequence pair-based tasks:

- Next Sentence Prediction
- Sentence Order Prediction

Learn inter-sequence coherence

✗ →

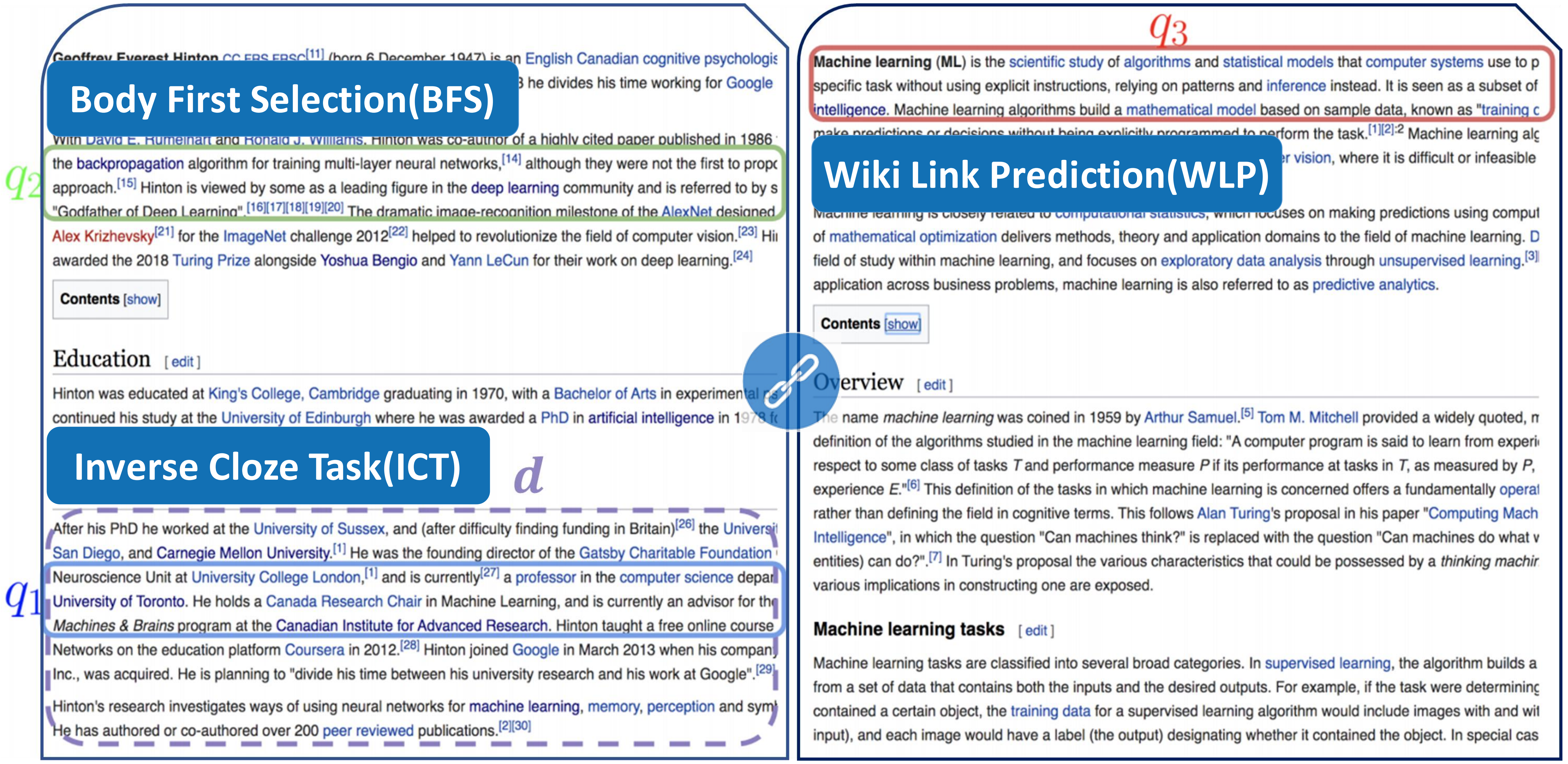
Relevance matching between short queries and long documents

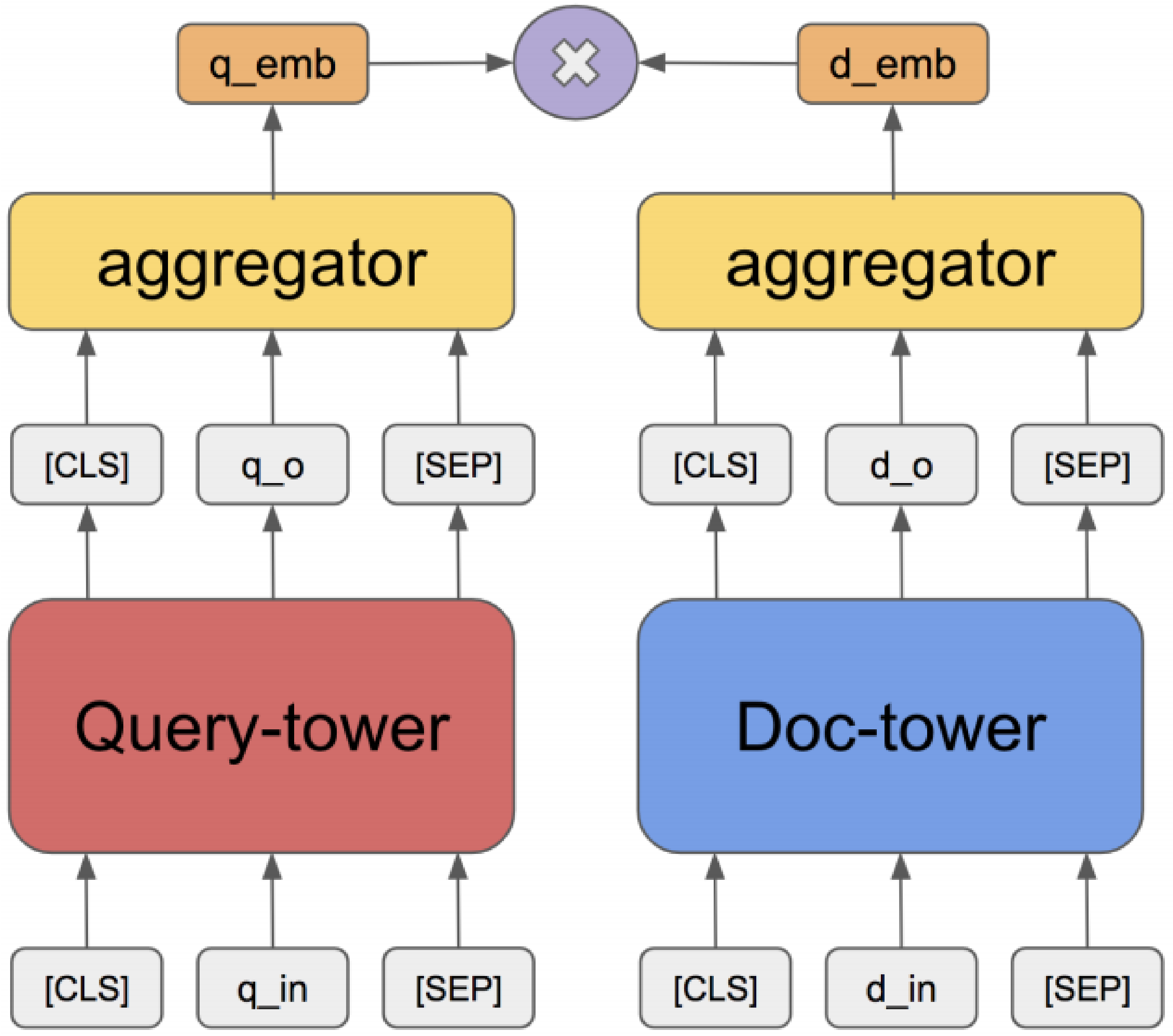
x **sentence-pair vs. query-document**

x **coherence vs. relevance**

Pre-training for Passage Retrieval in openQA

- Design three pre-training tasks that resemble the relevance relationship between **natural language questions** and **answer passages**





- Natural language questions-answer passages **vs.** short queries-long documents
- Depend on document structure, e.g., WLP
- Marginal benefit for ad-hoc retrieval

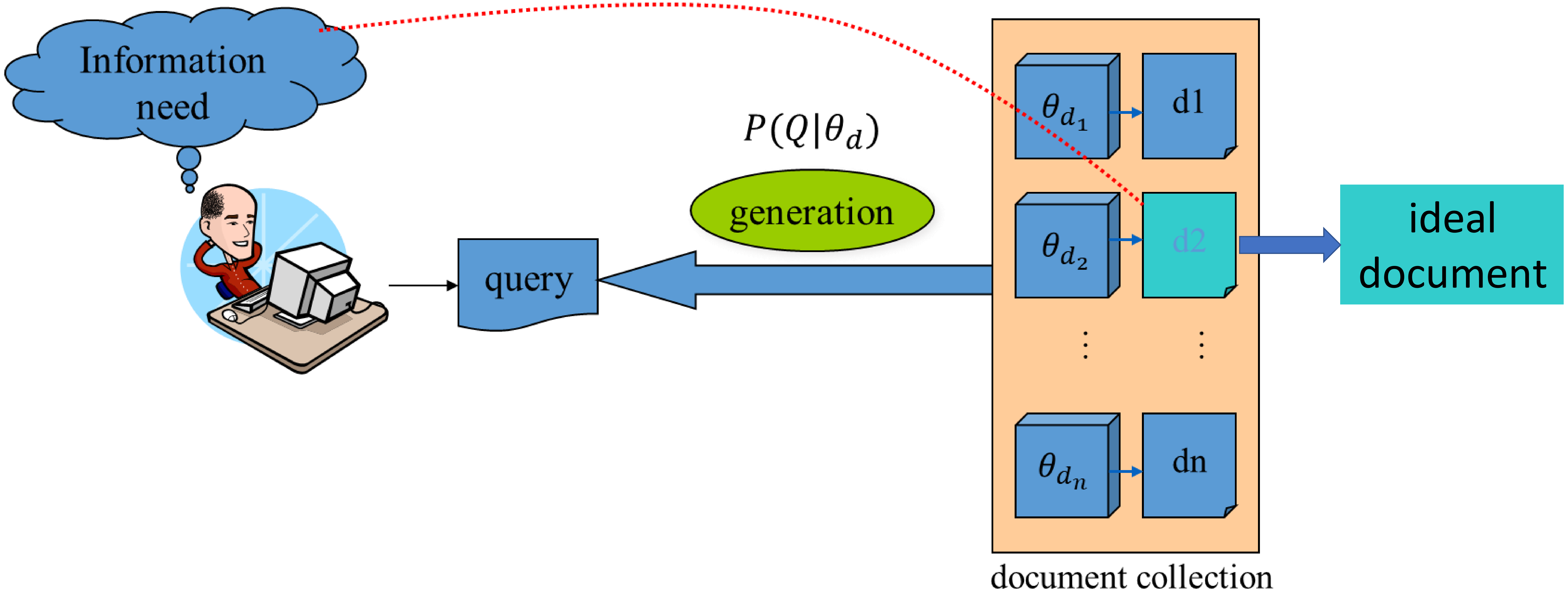
Pre-training Tasks for Embedding-based Large-scale Retrieval, ICLR, 2020

Our Goal

Design a novel pre-training objective tailored for IR, which more closely resembles the relevance relationship between query and document.

Back to Statistical LM for IR

- Classical SLM for IR: the Query Likelihood model



- The user has a reasonable idea of the terms that are likely to appear in the **"ideal" document** that can satisfy his/her information need
- The query is generated as **the piece of text representative** of the "ideal" document

Back to Statistical LM for IR

- Query likelihood scoring function derived by the Bayesian theorem

$$P(D/Q) \propto \underbrace{P(Q/\theta_D)}_{\text{Query generation probability}} \underbrace{P(D)}_{\text{Uniform distribution}} \propto P(Q/\theta_D)$$

Document language model

- Smoothing methods for zero probability problem
  - E.g., Jelinek-Mercer, Dirichlet prior, Absolute discounting
  - Query likelihood with Dirichlet smoothing is one of the most effective method (Zhai et.al. 2001)

$P(\boldsymbol{q_i}|\theta_D) = \frac{c(w,D)}{|D|}$ => $\frac{c(w,D)+\mu P(w|C)}{|D|+\mu}$, μ is smoothing parameter, $P(w|C)$ is collection language model

Pre-training Task for Ad-hoc Retrieval: ROP

- Representative words prediction (ROP) task
  - Given a document, sample word sets according to the document language model
  - The word set with higher likelihood is deemed as more “representative” of the document
  - Pre-train the Transformer model to predict the representativeness

Overview [edit]

An information retrieval process begins when a user enters a query into the system. Queries are formal statements of information needs, for example search strings in web search engines. In information retrieval a query does not uniquely identify a single object in the collection. Instead, several objects may match the query, perhaps with different degrees of relevancy.

An object is an entity that is represented by information in a content collection or database. User queries are matched against the database information. However, as opposed to classical SQL queries of a database, in information retrieval the results returned may or may not match the query, so results are typically ranked. This ranking of results is a key difference of information retrieval searching compared to database searching.[1]

Depending on the application the data objects may be, for example, text documents, images,[2] audio,[3] mind maps[4] or videos. Often the documents themselves are not kept or stored directly in the IR system, but are instead represented in the system by document surrogates or metadata.

Most IR systems compute a numeric score on how well each object in the database matches the query, and rank the objects according to this value. The top ranking objects are then shown to the user. The process may then be iterated if the user wishes to refine the query.[5]

From https://en.wikipedia.org/wiki/Information_retrieval

Representative Word Sets Sampling

1. Given document d, initialize document language model with Dirichlet smoothing θ_d

2. Choose length $l \sim Poisson(\lambda)$

3. **Paired Sampling**: Sample N pairs of word sets for each document where $w_i \sim P(w_i|\theta_d)$
   - Why? Likelihood comparable

4. Higher likelihood deemed as more representative

Algorithm 1 Sampling a Pair of Representative Word Sets

Input: Document D, Vocabulary $V = \{w_i\}_1^N$, probability of word w_i generated by the document language model with Dirichlet smoothing $P(w_i|D)$, Query likelihood score $QL(w_i, D)$

//Sample length

$l = Sample(X), x \sim Poisson(\lambda), x = 1, 2, 3...$

$S_1, S_2 = \emptyset, \emptyset$

//Sample a pair of word sets according to the document language model

for $k \leftarrow 1$ to l **do**

  $S_1 = S_1 \cup Sample(V), w_i \sim P(w_i|D)$

  $S_2 = S_2 \cup Sample(V), w_i \sim P(w_i|D)$

end for

//Decide which one is more representative

$S_1\_score = \prod_i^l QL(w_i, D), w_i \in S_1$

$S_2\_score = \prod_i^l QL(w_i, D), w_i \in S_2$

if $S_1\_score > S_2\_score$ **then**

  Output: (S_1^+, S_2^-, D)

else

  Output: (S_1^-, S_2^+, D)

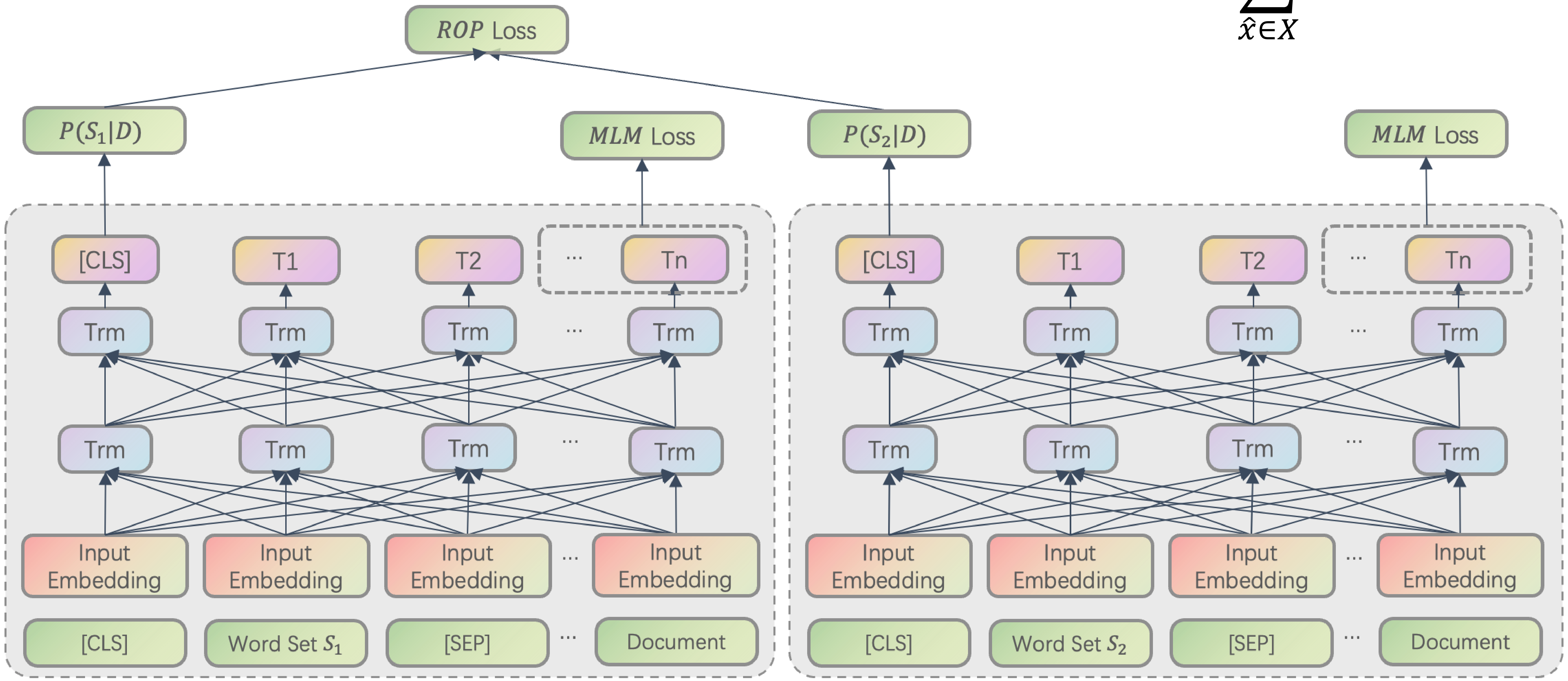
end if

Pre-training with the ROP task

- Pre-training Loss function

$$\mathcal{L}_{ROP} = \max(0, 1 - P(S_1|D) + P(S_2|D))$$

$$\mathcal{L}_{MLM} = -\sum_{\hat{x} \in X} \log p(\hat{x}|X_{\backslash \hat{x}})$$



Discussions

The ROP objective belongs to the category of model-based pre-training objective where the labels are produced by some automatic model rather than simple MASKs.

- **Electra** leverages a generative model to replace masked tokens
- **PEGASUS** leverages the ROUGE1-F1 score to select top-m sentences
- ……

Pre-training	**VS.**	**Weak supervision**
Only documents	What data is available?	Query and document, label is missing
MLM + ROP	Learning objective	Same as final ranking objective
A variety of retrieval tasks	Scope of application	Designed for each retrieval task

Experiment Setting

- Pretraining datasets：
  - Wikipedia, over 10 million documents
  - MS MARCO, about 3.4 million documents
- 5 downstream ad-hoc retrieval tasks：
  - Robust04, ClueWeb09-B, Gov2, MQ2007, MQ2008
- Baseline models:
  - Traditional retrieval models: BM25, QL
  - Previous state-of-the-art neural ranking models on each dataset: BERT-MaxP, HiNT et.al.
  - Other pretraining method: BERT, $\mathrm{Transformer}_{ICT}$

Experiments – Main Results

Table 2: Comparisons between PROP and the baselines. $*$, $\dagger$ and $\ddagger$ indicate statistically significance with $p - value \leq 0.05$ over BM25, BERT and Transformer$_{ICT}$, respectively.

Model	Robust04		ClueWeb09-B		Gov2		MQ2007		MQ2008	
	nDCG@20	P@20	nDCG@20	P@20	nDCG@20	P@20	nDCG@10	P@10	nDCG@10	P@10
QL	0.413	0.367	0.225	0.326	0.409	0.510	0.423	0.371	0.223	0.241
BM25	0.412	0.363	0.230	0.334	0.421	0.523	0.414	0.366	0.220	0.245
Previous SOTA	**0.538**	**0.467**	0.296	-	0.422	0.524	0.490	0.418	0.244	0.255
BERT	0.459^{*}	0.389^{*}	0.295^{*}	0.367^{*}	0.495^{*}	0.586^{*}	0.506^{*}	0.419^{*}	0.247^{*}	0.256^{*}
Transformer$_{ICT}$	0.460^{*}	0.388^{*}	0.298^{*}	0.369^{*}	$0.499^{*\dagger}$	0.587^{*}	0.508^{*}	0.420^{*}	0.245^{*}	0.256^{*}
PROP$_{Wikipedia}$	$\mathbf{0.502}^{*\dagger\ddagger}$	$\mathbf{0.421}^{*\dagger\ddagger}$	$0.316^{*\dagger\ddagger}$	$0.384^{*\dagger\ddagger}$	$0.519^{*\dagger\ddagger}$	$0.593^{*\dagger\ddagger}$	$\mathbf{0.523}^{*\dagger\ddagger}$	$\mathbf{0.432}^{*\dagger\ddagger}$	$0.262^{*\dagger\ddagger}$	$0.267^{*\dagger\ddagger}$
PROP$_{MSMARCO}$	$0.484^{*\dagger\ddagger}$	$0.408^{*\dagger\ddagger}$	$\mathbf{0.329}^{*\dagger\ddagger}$	$\mathbf{0.391}^{*\dagger\ddagger}$	$\mathbf{0.525}^{*\dagger\ddagger}$	$\mathbf{0.594}^{*\dagger\ddagger}$	$0.522^{*\dagger\ddagger}$	$0.430^{*\dagger\ddagger}$	$\mathbf{0.266}^{*\dagger\ddagger}$	$\mathbf{0.269}^{*\dagger\ddagger}$

1. PROP significantly outperforms previous SOTA in 4 of 5 tasks (8.9%, 24.4%, 6.7% and 9% in terms of NDCG@20), except for the Robust04 (BERT + Neu-IR ensemble).
2. PROP is significantly better than BERT and Transformer$_{ICT}$.
3. Pre-training in related domain corpus is more effective.

Experiments – Impact of Pre-training Objectives

Table 3: Impact of pre-training objectives. † indicates statistically significance with $p - value < 0.05$.

	nDCG@20			nDCG@10	
	Robust04	ClueWeb09-B	Gov2	MQ2007	MQ2008
w/ MLM	0.467	0.306	0.503	0.511	0.249
w/ ROP	0.481†	0.321†	0.519†	0.520†	0.262†
w/ ROP+MLM	**0.484**†	**0.329**†	**0.525**†	**0.522**†	**0.266**†

1. Pretraining with ROP achieves significant improvements over MLM.
2. MLM and ROP are both helpful for downstream tasks.

Experiments – Impact of Sampling Strategies

- docLM–based vs. Random sampling

Table 5: Impact of Different Sampling Strategies. Two-tailed t-tests demonstrate the improvements of document language model-based sampling to the random sampling strategy are statistically significant († indicates p-value < 0.05).

	nDCG@20			nDCG@10	
	Robust04	ClueWeb09-B	Gov2	MQ2007	MQ2008
Random	0.471	0.304	0.505	0.513	0.252
docLM-based	0.493†	0.317†	0.517†	0.516†	0.257†

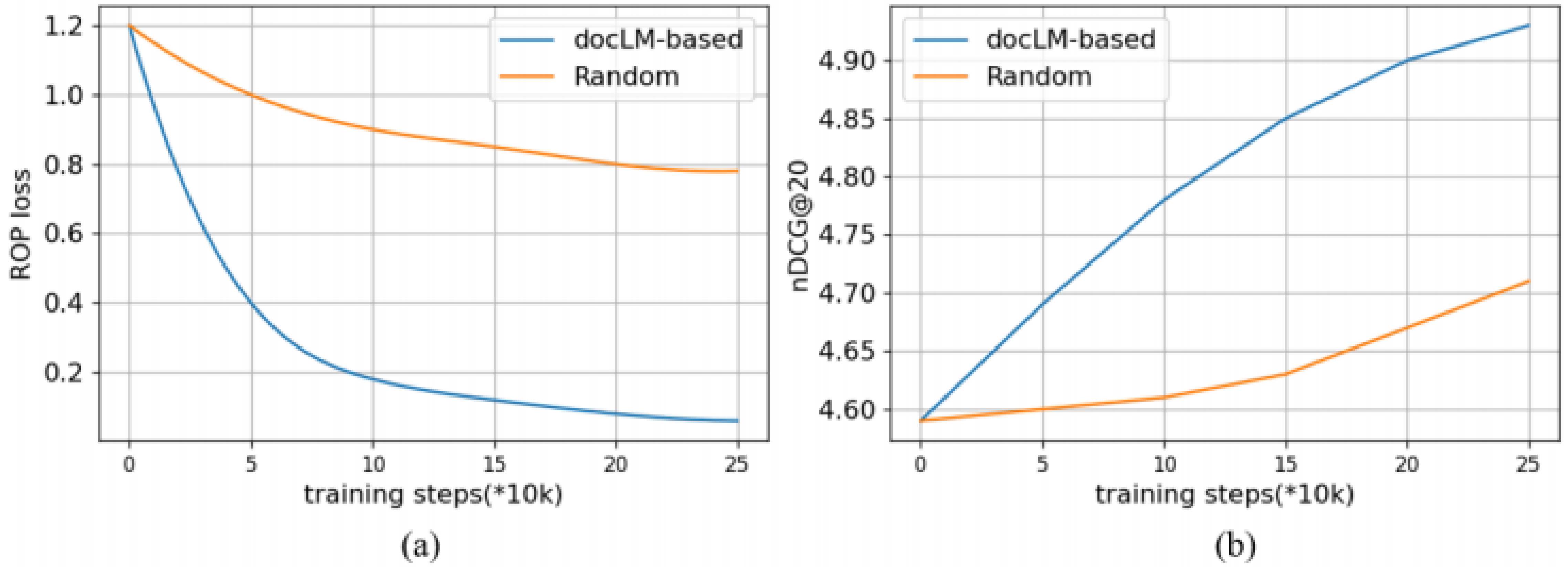


Figure 1: (a) ROP learning curve on Wikipedia over the pre-training steps. (b) The test performance curve on Robust04 in terms of nDCG@20 over pre-training steps.

1. docLM-based sampling converges faster and leads to better performance.
2. docLM-based sampling strategy is a more suitable way than the random sampling strategy to generate representative word sets for a document.

Experiments – Zero-shot and few-shot setting

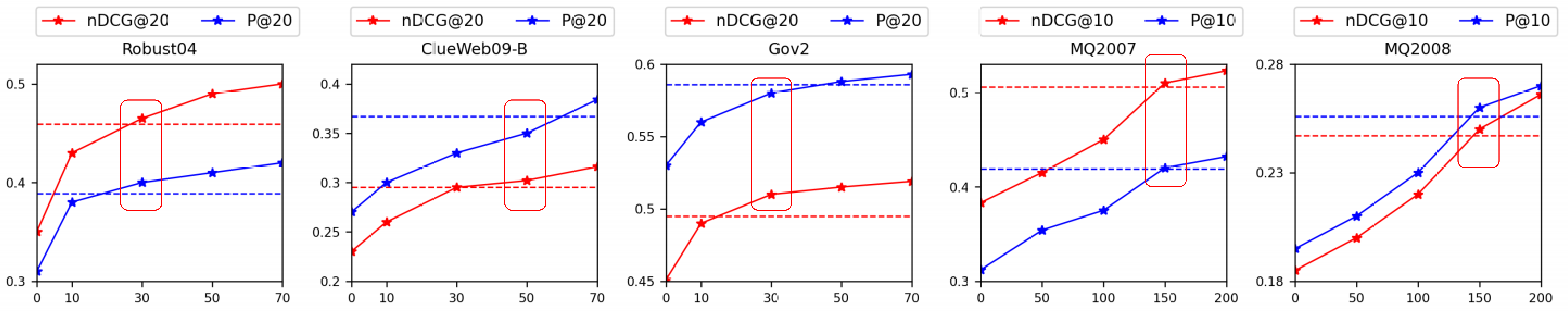


Figure 2: Fine-tuning with limited supervised data. The solid lines are PROP fine-tuned using 0 (zero shot), 10, 30, 50, and 70 queries for Robust04, ClueWeb09-B and Gov2 datasets, using 0 (zero shot), 50, 100, 150, and 200 queries for MQ2007 and MQ2008 datasets. The dashed lines are BERT fine-tuned using the full queries.

1. PROP fine-tuned on limited supervised data can achieve comparable performance with BERT fine-tuned on the full supervised datasets, e.g., 30 queries on Robust04.

2. Under the zero–shot setting, PROP also achieves exciting performance
   - On Gov2, PROP beats BM25 in terms of nDCG@20, and achieves about 90% performance of BERT fine-tuned on the full dataset

Conclusion & Future Work

- Conclusion
  - We proposed PROP, a new pre-training method tailed for ad-hoc retrieval
  - PROP achieved significant improvements over the baselines without pre-training or with other pre-training methods
  - PROP can achieve strong performance under both the zero- and low-resource IR settings

- Future work
  - Go beyond the ad-hoc retrieval, and test the ability of PROP over other downstream IR tasks, such as passage retrieval in QA or response retrieval in dialog systems
  - Investigate new ways to further enhance the pre-training objective tailored for IR

Code and the pre-training models are available at:
https://github.com/Albert-Ma/PROP

Thanks!

Xinyu Ma

maxinyu17g@ict.ac.cn