Improving Cross-Encoders Through Task-Specific Attention Modifications

Glasgow IR Seminar, 03.06.2024

Ferdinand Schlatt

ferdinand.schlatt@uni-jena.de

webis.de

FRIEDRICH-SCHILLER-UNIVERSITÄT JENA

Motivation

Improving Cross-Encoder Models

Transformer-based encoder models (e.g. BERT) are trained for general NLU.

Motivation

Improving Cross-Encoder Models

Transformer-based encoder models (e.g. BERT) are trained for general NLU.

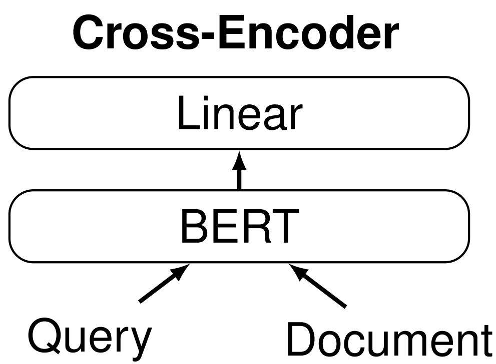
➜ We can further fine-tune the models for specific tasks, for example, re-ranking.

Motivation

Improving Cross-Encoder Models

Transformer-based encoder models (e.g. BERT) are trained for general NLU.

➜ We can further fine-tune the models for specific tasks, for example, re-ranking.

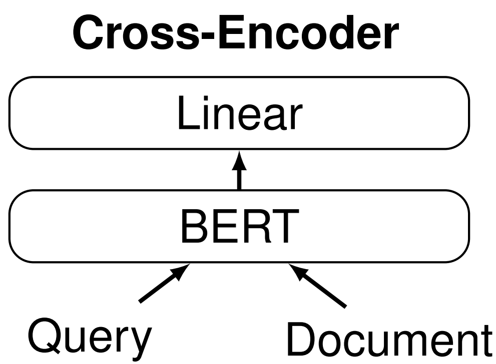


Motivation

Improving Cross-Encoder Models

Transformer-based encoder models (e.g. BERT) are trained for general NLU.

➜ We can further fine-tune the models for specific tasks, for example, re-ranking.



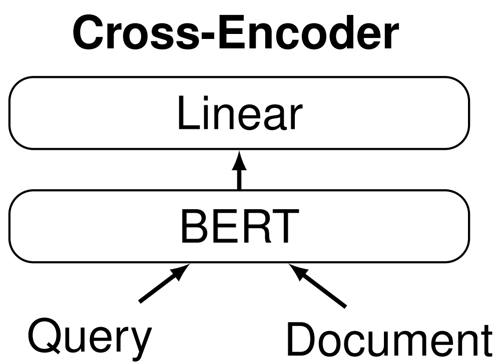
Can we “fine-tune” the architecture to gain efficiency / effectiveness for re-ranking?

Motivation

Improving Cross-Encoder Models

Transformer-based encoder models (e.g. BERT) are trained for general NLU.

➜ We can further fine-tune the models for specific tasks, for example, re-ranking.



Can we "fine-tune" the architecture to gain efficiency / effectiveness for re-ranking?

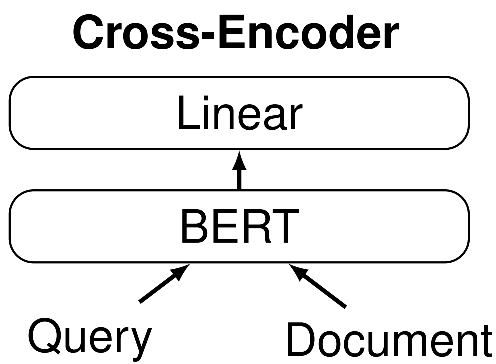
- **Efficiency**: Is full attention between all tokens necessary?
  ➜ Sparse Cross-Encoder

Motivation

Improving Cross-Encoder Models

Transformer-based encoder models (e.g. BERT) are trained for general NLU.

➜ We can further fine-tune the models for specific tasks, for example, re-ranking.



Can we “fine-tune” the architecture to gain efficiency / effectiveness for re-ranking?

- **Efficiency**: Is full attention between all tokens necessary?
  ➜ Sparse Cross-Encoder

- **Effectiveness**: Can we enable document interactions in re-ranking?
  ➜ Set-Encoder

Standard Cross-Encoder

Attention Mechanism

Query: python course

Document: Python is a great language to learn.

Standard Cross-Encoder

Attention Mechanism

[CLS] python course [SEP] Python is a great language to learn . [SEP]

Standard Cross-Encoder

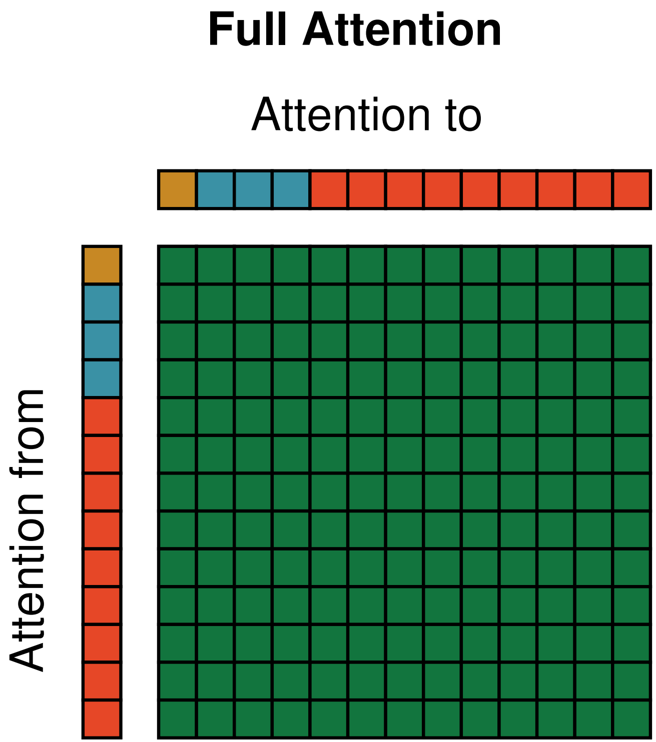
Attention Mechanism

[CLS] python course [SEP] Python is a great language to learn . [SEP]

Standard Cross-Encoder

Attention Mechanism

[CLS] python course [SEP] Python is a great language to learn . [SEP]



Sparse Cross-Encoder

Making Cross-Encoders More Efficient

One paradigm that improves cross-encoder efficiency is reducing the number of tokens that interact with each other. [Sekulic et al., TREC'20; Jiang et al., EMNLP'20]

Sparse Cross-Encoder

Making Cross-Encoders More Efficient

One paradigm that improves cross-encoder efficiency is reducing the number of tokens that interact with each other. [Sekulic et al., TREC’20; Jiang et al., EMNLP’20]

[CLS] python course [SEP] Python is a great language to learn . [SEP]

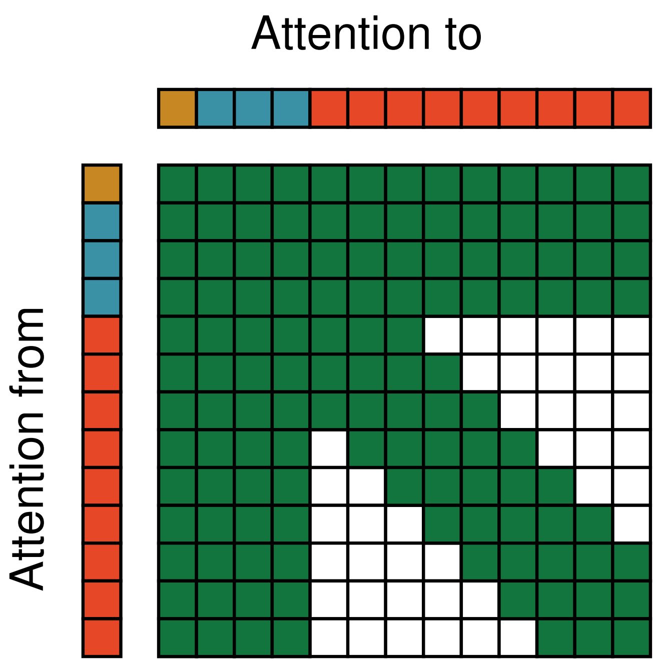
Sparse Cross-Encoder

Making Cross-Encoders More Efficient

One paradigm that improves cross-encoder efficiency is reducing the number of tokens that interact with each other. [Sekulic et al., TREC'20; Jiang et al., EMNLP'20]

[CLS] python course [SEP] Python is a great language to learn . [SEP]

Longformer [Beltagy et al., arXiv'20]



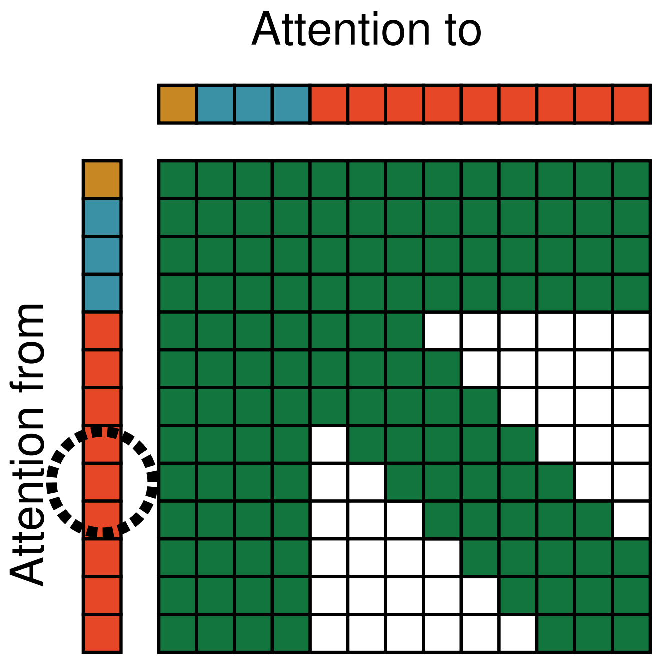
Sparse Cross-Encoder

Making Cross-Encoders More Efficient

One paradigm that improves cross-encoder efficiency is reducing the number of tokens that interact with each other. [Sekulic et al., TREC'20; Jiang et al., EMNLP'20]

[CLS] python course [SEP] Python is a great <u>language</u> to learn . [SEP]

Longformer [Beltagy et al., arXiv'20]



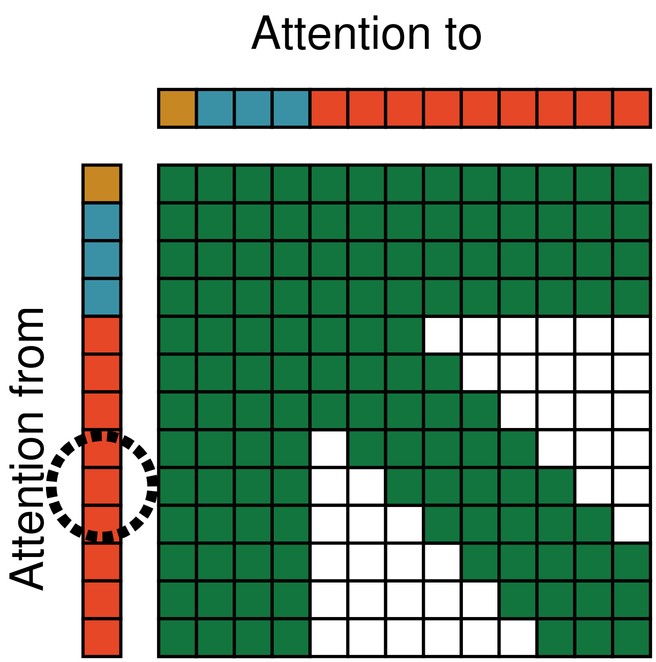
Sparse Cross-Encoder

Making Cross-Encoders More Efficient

One paradigm that improves cross-encoder efficiency is reducing the number of tokens that interact with each other. [Sekulic et al., TREC'20; Jiang et al., EMNLP'20]

[CLS] python course [SEP] Python is a great language to learn . [SEP]

Longformer [Beltagy et al., arXiv'20]



- ❑ Document tokens' attention restricted to context window of length w

➜ Semantic "gist" suffices to determine the relevance of a document token

- ❑ Previous work used $w = 64$ to save memory and re-rank longer documents

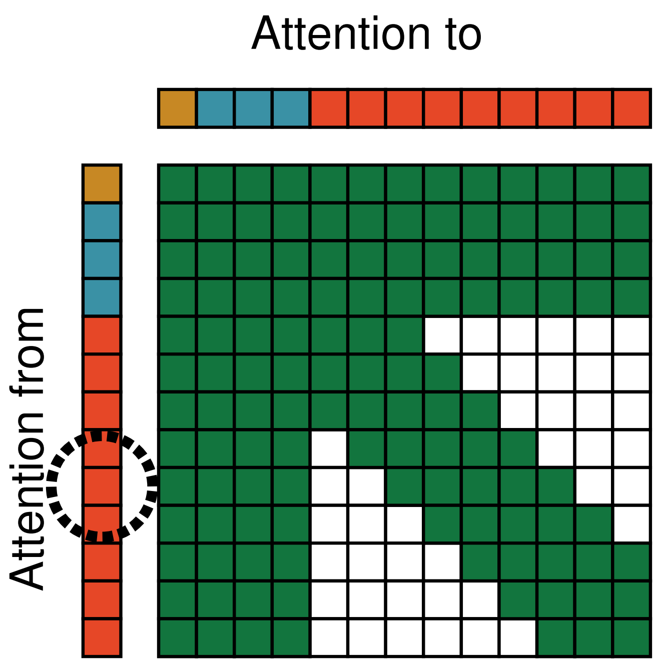
Sparse Cross-Encoder

Making Cross-Encoders More Efficient

One paradigm that improves cross-encoder efficiency is reducing the number of tokens that interact with each other. [Sekulic et al., TREC'20; Jiang et al., EMNLP'20]

[CLS] python course [SEP] Python is a great language to learn . [SEP]

Longformer [Beltagy et al., arXiv'20]



- Document tokens' attention restricted to context window of length w

➜ Semantic "gist" suffices to determine the relevance of a document token

- Previous work used $w = 64$ to save memory and re-rank longer documents

Hypothesis: Very small window sizes are as effective as full attention.

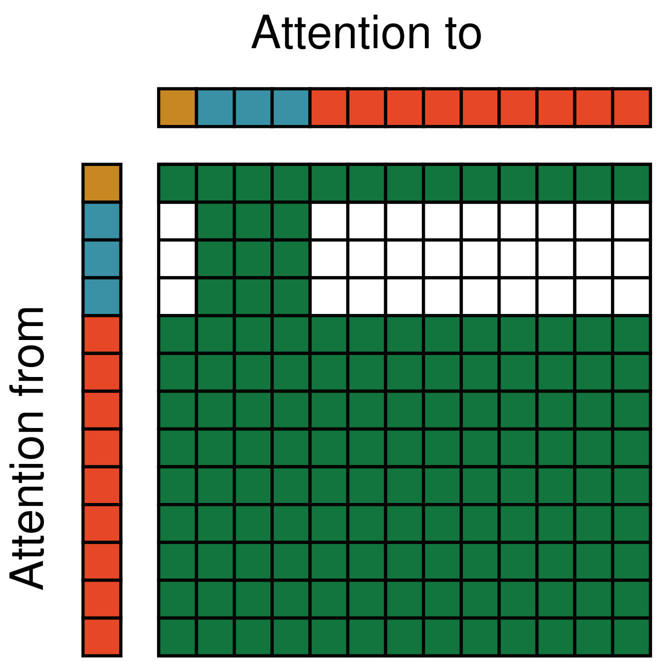
Sparse Cross-Encoder

Making Cross-Encoders More Efficient

One paradigm that improves cross-encoder efficiency is reducing the number of tokens that interact with each other. [Sekulic et al., TREC'20; Jiang et al., EMNLP'20]

[CLS] python course [SEP] Python is a great language to learn . [SEP]

Query Independent Attention



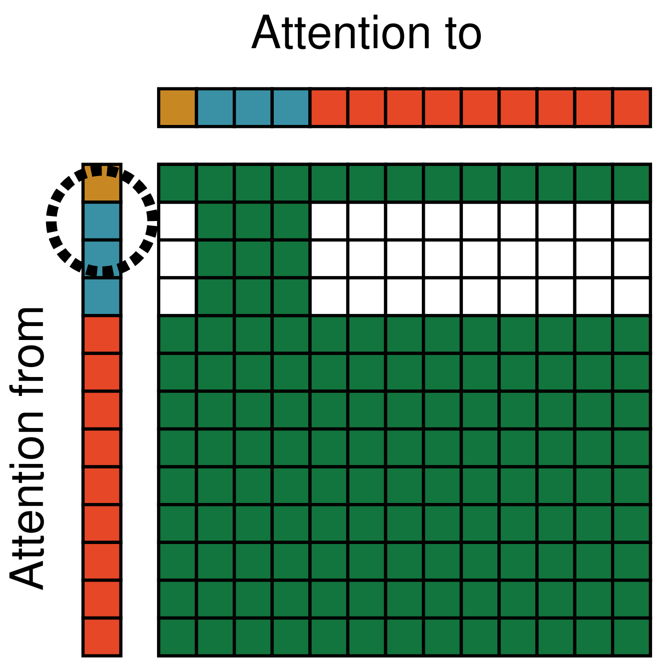
Sparse Cross-Encoder

Making Cross-Encoders More Efficient

One paradigm that improves cross-encoder efficiency is reducing the number of tokens that interact with each other. [Sekulic et al., TREC’20; Jiang et al., EMNLP’20]

[CLS] python course [SEP] Python is a great language to learn . [SEP]

Query Independent Attention



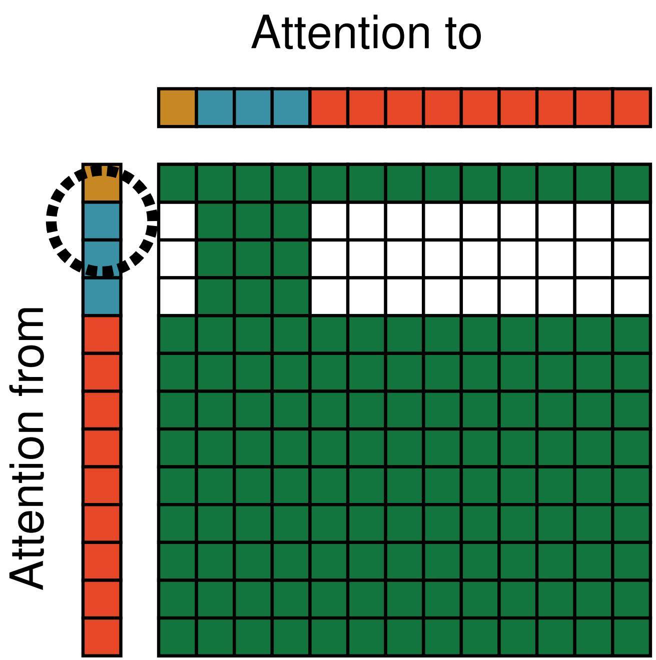
Sparse Cross-Encoder

Making Cross-Encoders More Efficient

One paradigm that improves cross-encoder efficiency is reducing the number of tokens that interact with each other. [Sekulic et al., TREC'20; Jiang et al., EMNLP'20]

[CLS] python course [SEP] Python is a great language to learn . [SEP]

Query Independent Attention



- ❑ A document is relevant to a query and not vice versa

➜ The query–document relevance relationship is asymmetric

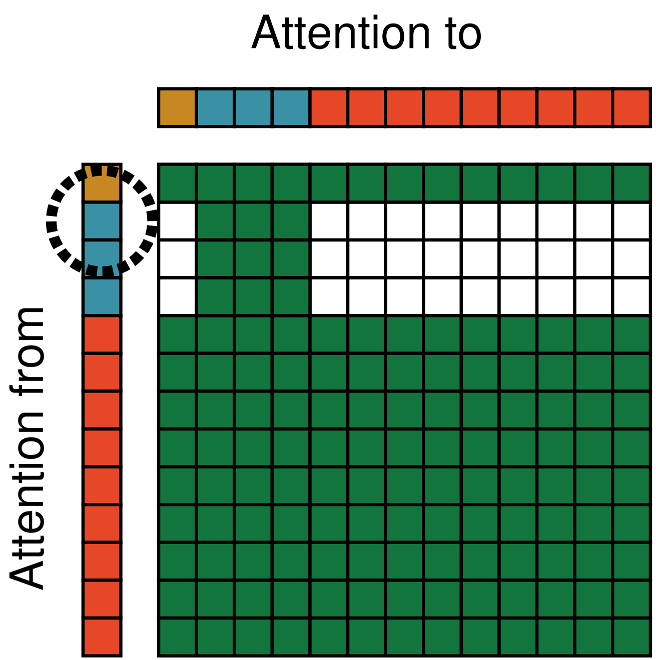
Sparse Cross-Encoder

Making Cross-Encoders More Efficient

One paradigm that improves cross-encoder efficiency is reducing the number of tokens that interact with each other. [Sekulic et al., TREC'20; Jiang et al., EMNLP'20]

[CLS] python course [SEP] Python is a great language to learn . [SEP]

Query Independent Attention



- ❑ A document is relevant to a query and not vice versa

➜ The query–document relevance relationship is asymmetric

Hypothesis: Deactivating attention from query tokens to other tokens is as effective as full attention.

Sparse Cross-Encoder

Attention Mechanism

Our sparse cross-encoder architecture combines windowed self-attention and asymmetric cross-attention between sub-sequences.

[CLS] python course [SEP] Python is a great language to learn . [SEP]

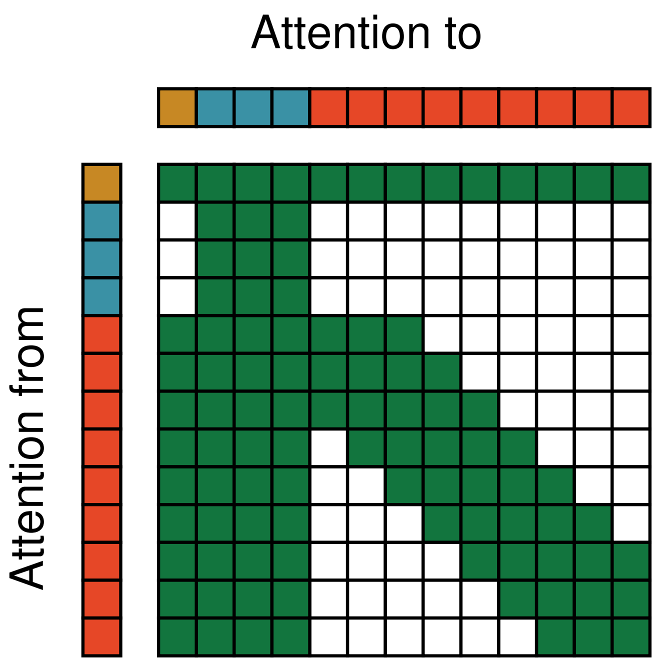
Sparse Cross-Encoder

Attention Mechanism

Our sparse cross-encoder architecture combines windowed self-attention and asymmetric cross-attention between sub-sequences.

[CLS] python course [SEP] Python is a great language to learn . [SEP]

Sparse Cross-Encoder



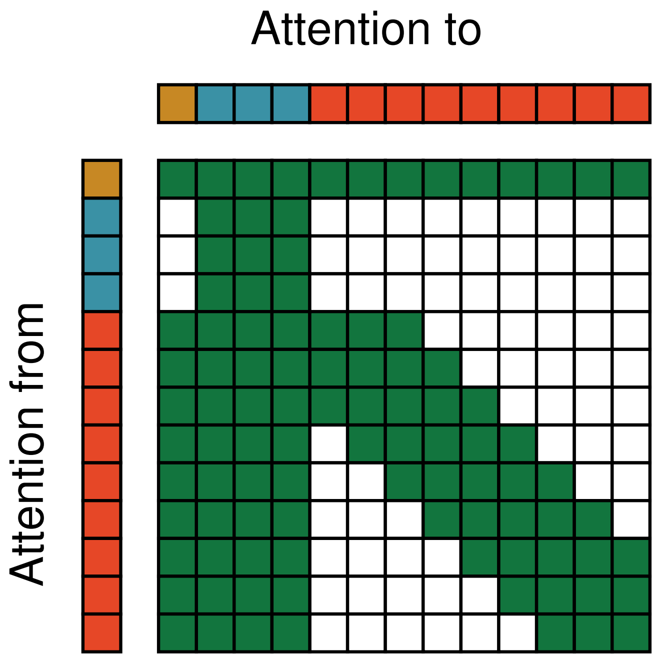
Sparse Cross-Encoder

Attention Mechanism

Our sparse cross-encoder architecture combines windowed self-attention and asymmetric cross-attention between sub-sequences.

[CLS] python course [SEP] Python is a great language to learn . [SEP]

Sparse Cross-Encoder



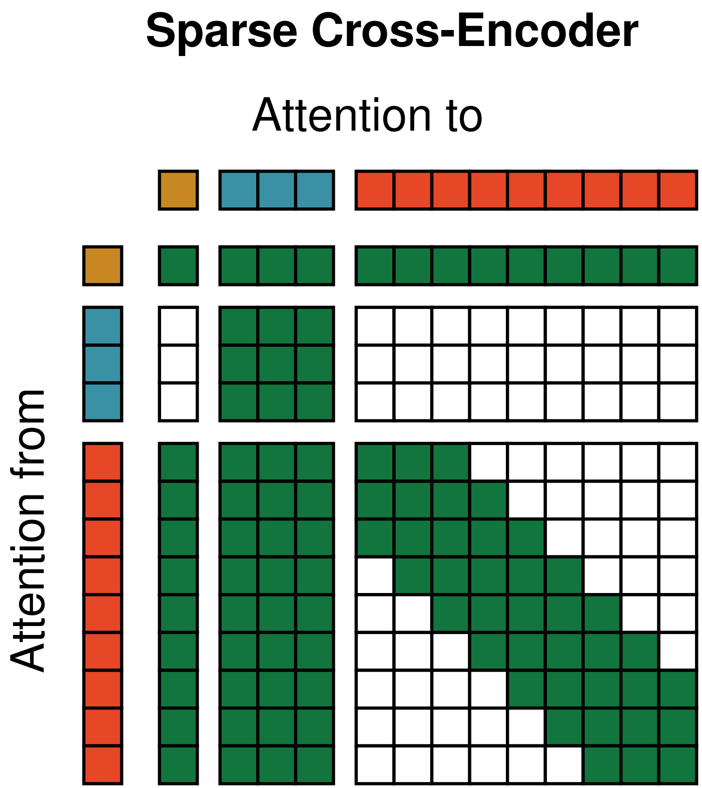
- Asymmetric attention not supported by standard transformer architectures

Sparse Cross-Encoder

Attention Mechanism

Our sparse cross-encoder architecture combines windowed self-attention and asymmetric cross-attention between sub-sequences.

[CLS] python course [SEP] Python is a great language to learn . [SEP]



- ❑ Asymmetric attention not supported by standard transformer architectures

➜ Custom architecture with cross-attention between sub-sequences

Sparse Cross-Encoder

Effectiveness

nDCG@10 on TREC Deep Learning 2019–2022 passage and document

Task	Full Attention / Longformer						Sparse Cross-Encoder					
$w =$	∞	64	16	4	1	0	∞	64	16	4	1	0
Passage	0.62	0.62†										
Document	0.58	0.58										

† denotes significant equivalence within ± 0.02 (paired TOST) with underlined score per row. MaxP results are grayed out.

Sparse Cross-Encoder
Effectiveness

nDCG@10 on TREC Deep Learning 2019–2022 passage and document

Task	Full Attention / Longformer						Sparse Cross-Encoder					
$w =$	∞	64	16	4	1	0	∞	64	16	4	1	0
Passage	<u>0.62</u>	0.62†					0.62†					
Document	0.58	<u>0.58</u>					0.57					

† denotes significant equivalence within ± 0.02 (paired TOST) with underlined score per row. MaxP results are grayed out.

1. Asymmetric query attention does not impact effectiveness ...

Sparse Cross-Encoder
Effectiveness

nDCG@10 on TREC Deep Learning 2019–2022 passage and document

Task	Full Attention / Longformer						Sparse Cross-Encoder					
$w =$	∞	64	16	4	1	0	∞	64	16	4	1	0
Passage	0.62	0.62†					0.62†	0.62†				
Document	0.58	0.58					0.57	0.59				

† denotes significant equivalence within ± 0.02 (paired TOST) with underlined score per row. MaxP results are grayed out.

1. Asymmetric query attention does not impact effectiveness even combined with windowed self-attention on documents

Sparse Cross-Encoder

Effectiveness

nDCG@10 on TREC Deep Learning 2019–2022 passage and document

Task	Full Attention / Longformer						Sparse Cross-Encoder					
$w =$	∞	64	16	4	1	0	∞	64	16	4	1	0
Passage	0.62	0.62†	0.62†				0.62†	0.62†	0.61			
Document	0.58	0.58	0.59†				0.57	0.59	0.59			

† denotes significant equivalence within ± 0.02 (paired TOST) with underlined score per row. MaxP results are grayed out.

1. Asymmetric query attention does not impact effectiveness even combined with windowed self-attention on documents

2. Window size of $w = 16$ is on par with full attention

Sparse Cross-Encoder

Effectiveness

nDCG@10 on TREC Deep Learning 2019–2022 passage and document

Task	Full Attention / Longformer						Sparse Cross-Encoder					
$w =$	∞	64	16	4	1	0	∞	64	16	4	1	0
Passage	<u>0.62</u>	0.62†	0.62†	0.62†			0.62†	0.62†	0.61	0.61†		
Document	0.58	<u>0.58</u>	0.59†	0.59			0.57	0.59	0.59	0.58		

† denotes significant equivalence within ± 0.02 (paired TOST) with underlined score per row. MaxP results are grayed out.

1. Asymmetric query attention does not impact effectiveness even combined with windowed self-attention on documents

2. Window size of $\cancel{w = 16}$ $w = 4$ is on par with full attention

Sparse Cross-Encoder

Effectiveness

nDCG@10 on TREC Deep Learning 2019–2022 passage and document

Task	Full Attention / Longformer						Sparse Cross-Encoder					
$w =$	∞	64	16	4	1	0	∞	64	16	4	1	0
Passage	<u>0.62</u>	0.62†	0.62†	0.62†	0.61		0.62†	0.62†	0.61	0.61†	0.60	
Document	0.58	<u>0.58</u>	0.59†	0.59	0.58†		0.57	0.59	0.59	0.58	0.59	

† denotes significant equivalence within ± 0.02 (paired TOST) with underlined score per row. MaxP results are grayed out.

1. Asymmetric query attention does not impact effectiveness even combined with windowed self-attention on documents
2. Window size of $\cancel{w = 16}$ $w = 4$ is on par with full attention
3. Window size of $w = 1$ still competitive

Sparse Cross-Encoder

Effectiveness

nDCG@10 on TREC Deep Learning 2019–2022 passage and document

Task	Full Attention / Longformer						Sparse Cross-Encoder					
$w =$	∞	64	16	4	1	0	∞	64	16	4	1	0
Passage	<u>0.62</u>	0.62†	0.62†	0.62†	0.61	0.57	0.62†	0.62†	0.61	0.61†	0.60	0.56
Document	0.58	<u>0.58</u>	0.59†	0.59	0.58†	0.56	0.57	0.59	0.59	0.58	0.59	0.56

† denotes significant equivalence within ± 0.02 (paired TOST) with underlined score per row. MaxP results are grayed out.

1. Asymmetric query attention does not impact effectiveness even combined with windowed self-attention on documents
2. Window size of $\cancel{w = 16}$ $w = 4$ is on par with full attention
3. Window size of $w = 1$ still competitive
4. Window size of $w = 0$ slightly less effective

Sparse Cross-Encoder

Effectiveness

nDCG@10 on TREC Deep Learning 2019–2022 passage and document

Task	**Full Attention / Longformer**						**Sparse Cross-Encoder**					
$w =$	∞	64	16	4	1	0	∞	64	16	4	1	0
Passage	<u>0.62</u>	0.62†	0.62†	0.62†	0.61	0.57	0.62†	0.62†	0.61	0.61†	0.60	0.56
Document	0.58	<u>0.58</u>	0.59†	0.59	0.58†	0.56	0.57	0.59	0.59	0.58	0.59	0.56

† denotes significant equivalence within ± 0.02 (paired TOST) with underlined score per row. MaxP results are grayed out.

1. Asymmetric query attention does not impact effectiveness even combined with windowed self-attention on documents

2. Window size of $\cancel{w = 16}$ $w = 4$ is on par with full attention

3. Window size of $w = 1$ still competitive

4. Window size of $w = 0$ slightly less effective

➜ Also translates to out-of-domain effectiveness on TIREx [Fröbe et al. SIGIR'23]

Sparse Cross-Encoder

Efficiency

Latency and memory consumption on synthetic query document pairs

Unit	**Full Attention**	**Longformer**	**Sparse CE**	**Sparse CE**
$w =$	∞	64	64	4
Query length 10, Passage length 164				
µs	368	980 $(+166\%)$		
MB	9	15 $(+67\%)$		
Query length 10, Document length 4086				
ms	49 $(+250\%)$	14		
MB	1608 $(+905\%)$	160		

Sparse Cross-Encoder

Efficiency

Latency and memory consumption on synthetic query document pairs

Unit	Full Attention	Longformer	Sparse CE	Sparse CE
$w =$	∞	64	64	4
Query length 10, Passage length 164				
µs	368	980 $(+166\%)$	527 $(+43\%)$	
MB	9	15 $(+67\%)$	9 $(+0\%)$	
Query length 10, Document length 4086				
ms	49 $(+250\%)$	14	12 (-14%)	
MB	1608 $(+905\%)$	160	111 (-31%)	

1. Sparse cross-encoder with $w = 64$ is more efficient than the Longformer

Sparse Cross-Encoder
Efficiency

Latency and memory consumption on synthetic query document pairs

Unit	Full Attention	Longformer	Sparse CE	Sparse CE
$w =$	∞	64	64	4
Query length 10, Passage length 164				
µs	368	980 $(+166\%)$	527 $(+43\%)$	364 (-1%)
MB	9	15 $(+67\%)$	9 $(+0\%)$	7 (-22%)
Query length 10, Document length 4086				
ms	49 $(+250\%)$	14	12 (-14%)	8 (-43%)
MB	1608 $(+905\%)$	160	111 (-31%)	66 (-59%)

1. Sparse cross-encoder with $w = 64$ is more efficient than the Longformer

2. Window size $w = 4$ is more efficient than full attention on passages

Sparse Cross-Encoder

Conclusion

We introduced a sparse cross-encoder architecture that combines windowed self-attention and asymmetric cross-attention between sub-sequences.

- Attention from query tokens to other tokens can be deactivated without losing effectiveness.
- Very small window sizes are still effective for re-ranking with cross-encoders.
- Our sparse cross-encoder reduces memory consumption and runtime.

Code, models, and paper @ https://github.com/webis-de/ECIR-24

Set-Encoder

Making Cross-Encoders More Effective

Query: python course

Documents:
Python is a great language to learn.
Pythons live in the rainforest.
Guido van Rossum invented Python.

Set-Encoder

Making Cross-Encoders More Effective

Query: python course

Documents:
Python is a great language to learn.
Pythons live in the rainforest.
Guido van Rossum invented Python.

monoBERT (Pointwise) [Nogueira and Cho, arXiv'19]

[CLS] python course [SEP] Python is . . . [SEP] → BERT → **2.4**
[CLS] python course [SEP] Pythons live . . . [SEP] → BERT → **0.1**
[CLS] python course [SEP] Guido van . . . [SEP] → BERT → **1.9**

Set-Encoder

Making Cross-Encoders More Effective

Query: python course

Documents: Python is a great language to learn.
Pythons live in the rainforest.
Guido van Rossum invented Python.

monoBERT (Pointwise) [Nogueira and Cho, arXiv'19]

[CLS] python course [SEP] Python is . . . [SEP] → BERT → **2.4**
[CLS] python course [SEP] Pythons live . . . [SEP] → BERT → **0.1**
[CLS] python course [SEP] Guido van . . . [SEP] → BERT → **1.9**

Issue: The model scores each document independently.

Set-Encoder

Making Cross-Encoders More Effective

Query: python course

Documents:
Python is a great language to learn.
Pythons live in the rainforest.
Guido van Rossum invented Python.

monoBERT (Pointwise) [Nogueira and Cho, arXiv'19]

[CLS] python course [SEP] Python is . . . [SEP]
[CLS] python course [SEP] Pythons live . . . [SEP] ➜ BERT ➜
[CLS] python course [SEP] Guido van . . . [SEP]

2.4
0.1
1.9

Issue: The model scores each document independently.

➜ Listwise (and pairwise) models enable interactions between documents.

Set-Encoder

Making Cross-Encoders More Effective

Query: python course

Documents:
Python is a great language to learn.
Pythons live in the rainforest.
Guido van Rossum invented Python.

duoBERT (Pairwise) [Nogueira et al., arXiv'20]

[CLS] python course [SEP] . . . [SEP] . . . [SEP]
[CLS] python course [SEP] . . . [SEP] . . . [SEP]
[CLS] python course [SEP] . . . [SEP] . . . [SEP]
[CLS] python course [SEP] . . . [SEP] . . . [SEP]
⋮

→ BERT →

1.2
-2.1
2.7
0.1
⋮

Set-Encoder

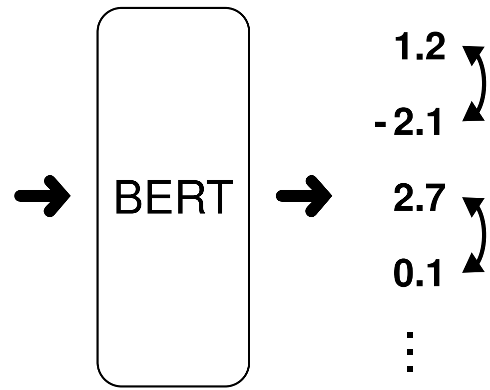
Making Cross-Encoders More Effective

Query: python course

Documents:
Python is a great language to learn.
Pythons live in the rainforest.
Guido van Rossum invented Python.

duoBERT (Pairwise) [Nogueira et al., arXiv'20]

[CLS] python course [SEP] . . . [SEP] . . . [SEP]
[CLS] python course [SEP] . . . [SEP] . . . [SEP]
[CLS] python course [SEP] . . . [SEP] . . . [SEP]
[CLS] python course [SEP] . . . [SEP] . . . [SEP]
⋮



Issue: Relevance scores are not symmetric.

Set-Encoder

Making Cross-Encoders More Effective

Query: python course

Documents:
Python is a great language to learn.
Pythons live in the rainforest.
Guido van Rossum invented Python.

RankGPT (Listwise) [Sun et al., EMNLP'23]

Prompt: ... Query: ... [1]: ... [2]: ... [3]: ...			1 > 3 > 2
Prompt: ... Query: ... [1]: ... [3]: ... [2]: ...			1 > 3 > 2
Prompt: ... Query: ... [2]: ... [1]: ... [3]: ...	→	GPT →	1 > 3 > 2
Prompt: ... Query: ... [2]: ... [3]: ... [1]: ...			3 > 1 > 2
⋮			⋮

Set-Encoder

Making Cross-Encoders More Effective

Query: python course

Documents: Python is a great language to learn.
Pythons live in the rainforest.
Guido van Rossum invented Python.

RankGPT (Listwise) [Sun et al., EMNLP'23]

Prompt: . . . Query: . . . [1]: . . . [2]: . . . [3]: . . . → GPT → **1 > 3 > 2**
Prompt: . . . Query: . . . [1]: . . . [3]: . . . [2]: . . . → GPT → **1 > 3 > 2**
Prompt: . . . Query: . . . [2]: . . . [1]: . . . [3]: . . . → GPT → **1 > 3 > 2**
Prompt: . . . Query: . . . [2]: . . . [3]: . . . [1]: . . . → GPT → **3 > 1 > 2**
⋮

Issue: Relevance preference order is not consistent.

Set-Encoder

Making Cross-Encoders More Effective

Query: python course

Documents: Python is a great language to learn.
Pythons live in the rainforest.
Guido van Rossum invented Python.

RankGPT (Listwise) [Sun et al., EMNLP’23]

Prompt: ... Query: ... [1]: ... [2]: ... [3]: ... → GPT → 1 > 3 > 2
Prompt: ... Query: ... [1]: ... [3]: ... [2]: ... → GPT → 1 > 3 > 2
Prompt: ... Query: ... [2]: ... [1]: ... [3]: ... → GPT → 1 > 3 > 2
Prompt: ... Query: ... [2]: ... [3]: ... [1]: ... → GPT → 3 > 1 > 2
⋮

Issue: Relevance preference order is not consistent.

➜ No current transformer-based re-rankers are listwise and permutation invariant because input documents are processed independently or concatenated.

Set-Encoder

Attention Mechanism

[CLS] python course [SEP] Python is a great language to learn . [SEP]

[CLS] python course [SEP] Pythons live in the rainforest . [SEP]

[CLS] python course [SEP] Guido van Rossum invented Python . [SEP]

Set-Encoder

Attention Mechanism

[CLS] python course [SEP] Python is a great language to learn . [SEP]

[CLS] python course [SEP] Pythons live in the rainforest . [SEP]

[CLS] python course [SEP] Guido van Rossum invented Python . [SEP]

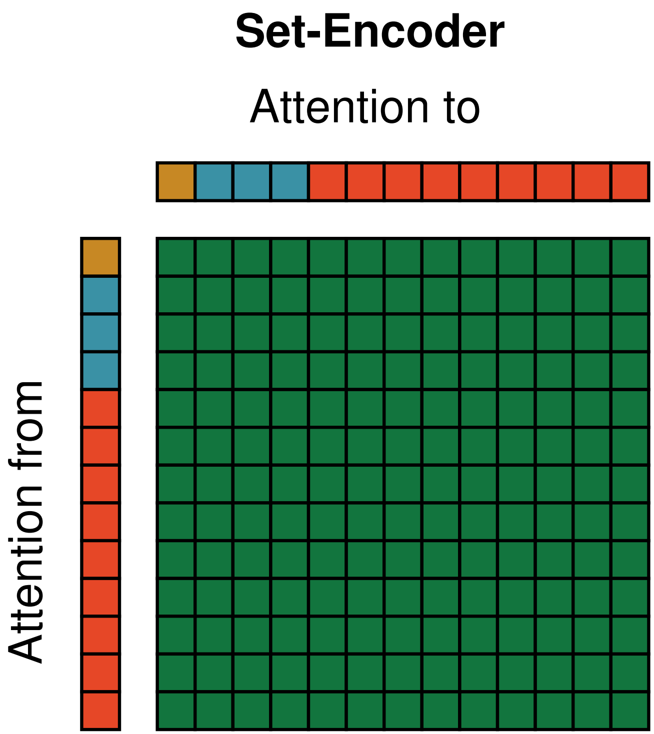
Set-Encoder

Attention Mechanism

[CLS] python course [SEP] Python is a great language to learn . [SEP]

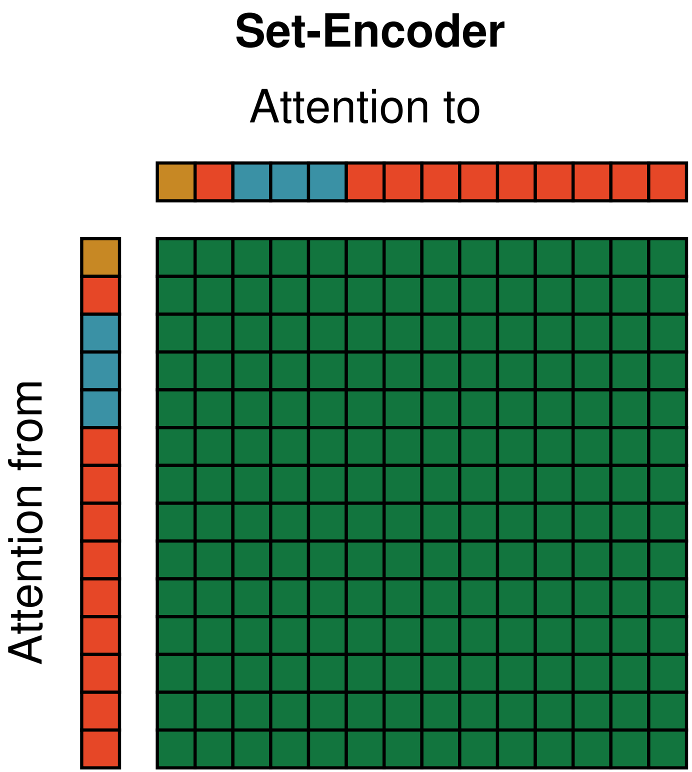
[CLS] python course [SEP] Pythons live in the rainforest . [SEP]

[CLS] python course [SEP] Guido van Rossum invented Python . [SEP]



Set-Encoder
Attention Mechanism

[CLS] [INT] python course [SEP] Python is a great language to learn . [SEP]
[CLS] [INT] python course [SEP] Pythons live in the rainforest . [SEP]
[CLS] [INT] python course [SEP] Guido van Rossum invented Python . [SEP]



1. Insert an extra [INT] token

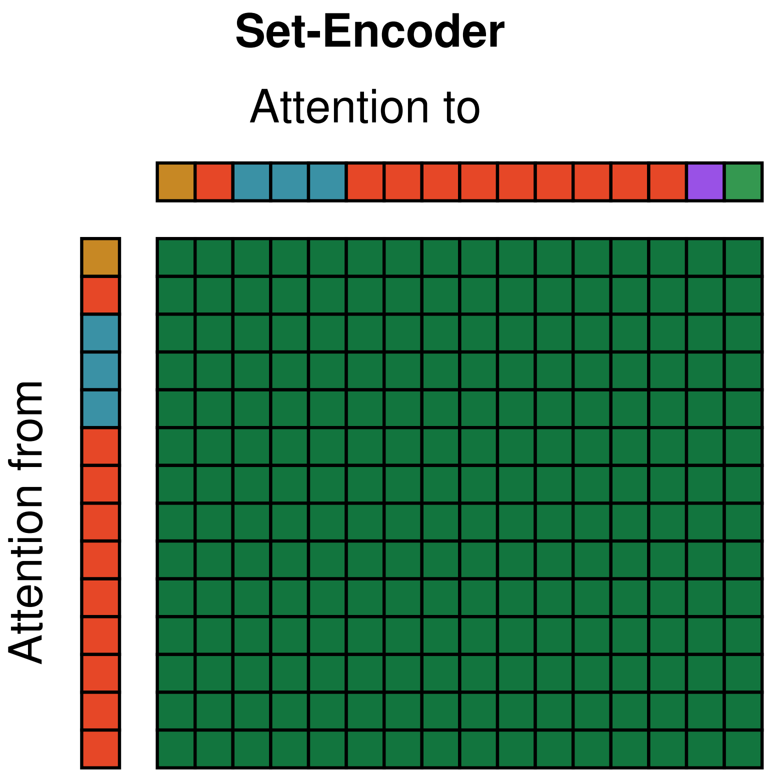
Set-Encoder

Attention Mechanism

[CLS] [INT] python course [SEP] Python is a great language to learn . [SEP]

[CLS] [INT] python course [SEP] Pythons live in the rainforest . [SEP]

[CLS] [INT] python course [SEP] Guido van Rossum invented Python . [SEP]



1. Insert an extra [INT] token
2. Allow a document to attend to all other documents' [INT] tokens

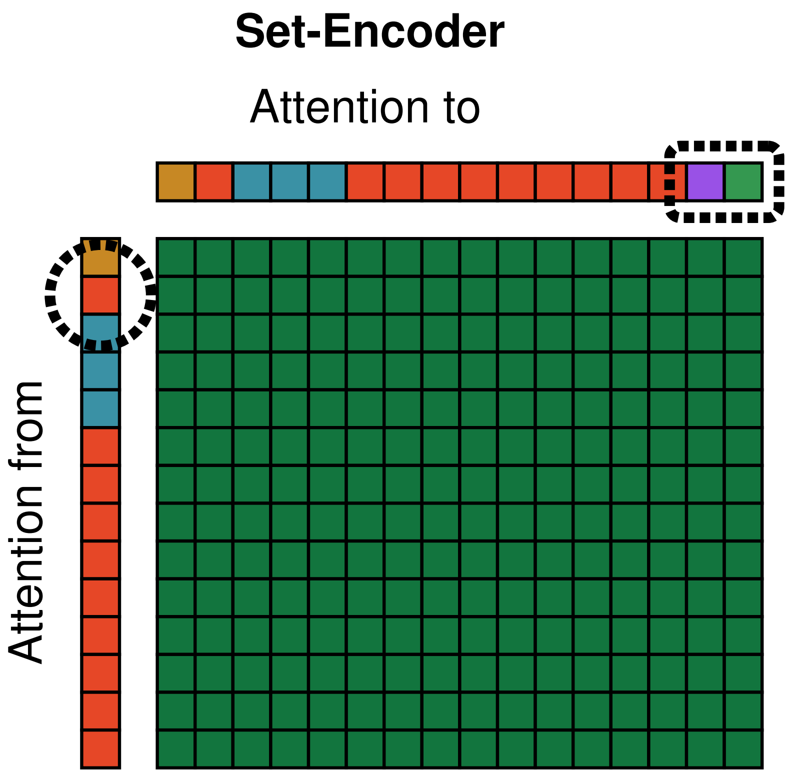
Set-Encoder

Attention Mechanism

[CLS] [INT] python course [SEP] Python is a great language to learn . [SEP]

[CLS] [INT] python course [SEP] Pythons live in the rainforest . [SEP]

[CLS] [INT] python course [SEP] Guido van Rossum invented Python . [SEP]



1. Insert an extra [INT] token
2. Allow a document to attend to all other documents' [INT] tokens

- [INT] tokens aggregate semantic information and shares information with other documents

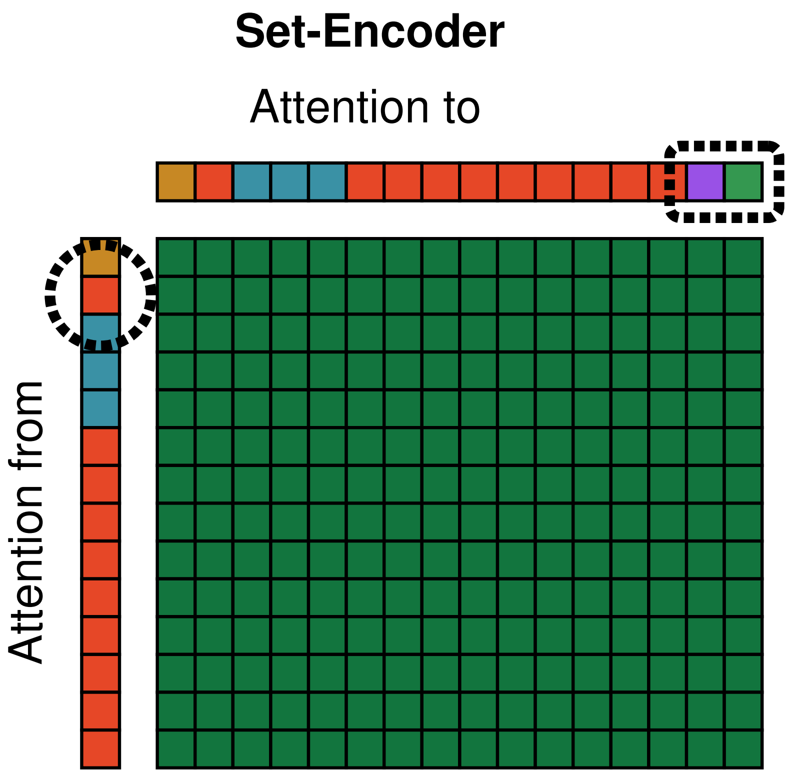
Set-Encoder

Attention Mechanism

[CLS] [INT] python course [SEP] Python is a great language to learn . [SEP]

[CLS] [INT] python course [SEP] Pythons live in the rainforest . [SEP]

[CLS] [INT] python course [SEP] Guido van Rossum invented Python . [SEP]



1. Insert an extra [INT] token
2. Allow a document to attend to all other documents' [INT] tokens

- [INT] tokens aggregate semantic information and shares information with other documents
- Permutation-invariant because all [INT] tokens share the same positional encoding

Set-Encoder

Distilling Cross-Encoders from LLMs

Cross-encoders are typically fine-tuned on MS MARCO.
[Nguyen et al., COCO@NeurIPS'16]

Cross-Encoder
MS MARCO

Set-Encoder

Distilling Cross-Encoders from LLMs

Cross-encoders are typically fine-tuned on MS MARCO.
[Nguyen et al., COCO@NeurIPS'16]

Zero-shot LLMs are more effective than cross-encoders fine-tuned on MS MARCO.
[Sun et al., EMNLP'23, Pradeep et al., arXiv'23]

Cross-Encoder MS MARCO	$<$	RankGPT

Set-Encoder

Distilling Cross-Encoders from LLMs

Cross-encoders are typically fine-tuned on MS MARCO.
[Nguyen et al., COCO@NeurIPS'16]

Zero-shot LLMs are more effective than cross-encoders fine-tuned on MS MARCO.
[Sun et al., EMNLP'23, Pradeep et al., arXiv'23]

Cross-encoders distilled from LLMs sit in between.
[Sun et al., EMNLP'23, Baldelli et al., ECIR'24]

Cross-Encoder MS MARCO	<	Cross-Encoder Distilled from LLM	<	RankGPT

Set-Encoder

Distilling Cross-Encoders from LLMs

Cross-encoders are typically fine-tuned on MS MARCO.
[Nguyen et al., COCO@NeurIPS'16]

Zero-shot LLMs are more effective than cross-encoders fine-tuned on MS MARCO.
[Sun et al., EMNLP'23, Pradeep et al., arXiv'23]

Cross-encoders distilled from LLMs sit in between.
[Sun et al., EMNLP'23, Baldelli et al., ECIR'24]

Cross-Encoder		Cross-Encoder		RankGPT
MS MARCO	<	Distilled from LLM	<	

Rank-DistiLLM

Set-Encoder

Distilling Cross-Encoders from LLMs

Cross-encoders are typically fine-tuned on MS MARCO.
[Nguyen et al., COCO@NeurIPS'16]

Zero-shot LLMs are more effective than cross-encoders fine-tuned on MS MARCO.
[Sun et al., EMNLP'23, Pradeep et al., arXiv'23]

Cross-encoders distilled from LLMs sit in between.
[Sun et al., EMNLP'23, Baldelli et al., ECIR'24]

Cross-Encoder		Cross-Encoder		RankGPT
MS MARCO	<	Distilled from LLM	<	

Rank-DistiLLM

Data and paper @ https://github.com/webis-de/msmarco-llm-distillation

Set-Encoder

Effectiveness

nDCG@10 on TREC Deep Learning 2019 and 2020 passage and TIREx

Model	**TREC DL 19**		**TREC DL 20**		**TIREx**
First Stage	BM25	CBv2	BM25	CBv2	
First Stage	0.480	0.732	0.494	0.724	0.394
RankGPT-4o	0.725	<u>0.784</u>	0.719	0.793	–
RankGPT-4o Full	<u>0.732</u>	0.781	0.711	**0.799**	–
RankZephyr	0.719	0.749	<u>0.720</u>	<u>0.798</u>	**0.478**
monoELECTRA$_{\texttt{BASE}}$					
monoELECTRA$_{\texttt{LARGE}}$					
Set-Encoder$_{\texttt{BASE}}$					
Set-Encoder$_{\texttt{LARGE}}$		Still training :(			

Bold / underlined scores are the highest / second highest per task. TIREx scores are reported as geometric mean.

Set-Encoder
Effectiveness

nDCG@10 on TREC Deep Learning 2019 and 2020 passage and TIREx

Model	TREC DL 19		TREC DL 20		TIREx
First Stage	BM25	CBv2	BM25	CBv2	
First Stage	0.480	0.732	0.494	0.724	0.394
RankGPT-4o	0.725	0.784	0.719	0.793	–
RankGPT-4o Full	0.732	0.781	0.711	**0.799**	–
RankZephyr	0.719	0.749	0.720	0.798	**0.478**
monoELECTRA$_{\mathrm{BASE}}$					
monoELECTRA$_{\mathrm{LARGE}}$					
Set-Encoder$_{\mathrm{BASE}}$	0.724	**0.788**	0.710	0.777	0.459
Set-Encoder$_{\mathrm{LARGE}}$	Still training :(				

Bold / underlined scores are the highest / second highest per task. TIREx scores are reported as geometric mean.

1. Set-Encoder is competitive with state-of-the-art zero-shot LLM re-rankers.

Set-Encoder

Effectiveness

nDCG@10 on TREC Deep Learning 2019 and 2020 passage and TIREx

Model	**TREC DL 19**		**TREC DL 20**		**TIREx**
First Stage	BM25	CBv2	BM25	CBv2	
First Stage	0.480	0.732	0.494	0.724	0.394
RankGPT-4o	0.725	<u>0.784</u>	0.719	0.793	–
RankGPT-4o Full	<u>0.732</u>	0.781	0.711	**0.799**	–
RankZephyr	0.719	0.749	<u>0.720</u>	<u>0.798</u>	**0.478**
monoELECTRA$_{\texttt{BASE}}$	0.720	0.768	0.711	0.770	0.457
monoELECTRA$_{\texttt{LARGE}}$					
Set-Encoder$_{\texttt{BASE}}$	0.724	**0.788**	0.710	0.777	0.459
Set-Encoder$_{\texttt{LARGE}}$	Still training :(				

Bold / underlined scores are the highest / second highest per task. TIREx scores are reported as geometric mean.

1. Set-Encoder is competitive with state-of-the-art zero-shot LLM re-rankers.
2. But so is a plain pointwise monoELECTRA.

Set-Encoder

Effectiveness

nDCG@10 on TREC Deep Learning 2019 and 2020 passage and TIREx

Model	**TREC DL 19**		**TREC DL 20**		**TIREx**
First Stage	BM25	CBv2	BM25	CBv2	
First Stage	0.480	0.732	0.494	0.724	0.394
RankGPT-4o	0.725	<u>0.784</u>	0.719	0.793	–
RankGPT-4o Full	<u>0.732</u>	0.781	0.711	**0.799**	–
RankZephyr	0.719	0.749	<u>0.720</u>	<u>0.798</u>	**0.478**
monoELECTRA$_{\texttt{BASE}}$	0.720	0.768	0.711	0.770	0.457
monoELECTRA$_{\texttt{LARGE}}$	**0.733**	0.765	**0.727**	**0.799**	<u>0.475</u>
Set-Encoder$_{\texttt{BASE}}$	0.724	**0.788**	0.710	0.777	0.459
Set-Encoder$_{\texttt{LARGE}}$	Still training :(				

Bold / underlined scores are the highest / second highest per task. TIREx scores are reported as geometric mean.

1. Set-Encoder is competitive with state-of-the-art zero-shot LLM re-rankers.
2. But so is a plain pointwise monoELECTRA.
3. A large monoELECTRA is on par with LLMs even in out-of-domain re-ranking.

Set-Encoder

Listwise Re-Ranking

Three hypotheses why the Set-Encoder does not improve over monoELECTRA:

Set-Encoder

Listwise Re-Ranking

Three hypotheses why the Set-Encoder does not improve over monoELECTRA:

1. The Set-Encoder cannot model interactions between documents.

Set-Encoder

Listwise Re-Ranking

Three hypotheses why the Set-Encoder does not improve over monoELECTRA:

1. The Set-Encoder cannot model interactions between documents.
2. The training data does not provide signals that listwise models profit from.

Set-Encoder

Listwise Re-Ranking

Three hypotheses why the Set-Encoder does not improve over monoELECTRA:

1. The Set-Encoder cannot model interactions between documents.
2. The training data does not provide signals that listwise models profit from.
3. Assessing topical relevance does not require document interactions.

Set-Encoder

Listwise Re-Ranking

Three hypotheses why the Set-Encoder does not improve over monoELECTRA:

1. The Set-Encoder cannot model interactions between documents.
2. The training data does not provide signals that listwise models profit from.
3. Assessing topical relevance does not require document interactions.

Set-Encoder

Listwise Re-Ranking

Three hypotheses why the Set-Encoder does not improve over monoELECTRA:

1. The Set-Encoder cannot model interactions between documents.
2. The training data does not provide signals that listwise models profit from.
3. Assessing topical relevance does not require document interactions.

➜ We build a synthetic task which requires document interactions.

MS MARCO contains many lexical near-duplicates.

Python is a great language to learn.

Python is a great language to learn now.

Pythons live in the rainforest.

Guido van Rossum invented Python.

Set-Encoder

Listwise Re-Ranking

Three hypotheses why the Set-Encoder does not improve over monoELECTRA:

1. The Set-Encoder cannot model interactions between documents.
2. The training data does not provide signals that listwise models profit from.
3. Assessing topical relevance does not require document interactions.

➜ We build a synthetic task which requires document interactions.

MS MARCO contains many lexical near-duplicates.

Python is a great language to learn.
Python is a great language to learn now.
Pythons live in the rainforest.
Guido van Rossum invented Python.

Fine-tune models to rank according to relevance and put duplicates at the end.

Set-Encoder

Listwise Re-Ranking

Three hypotheses why the Set-Encoder does not improve over monoELECTRA:

1. The Set-Encoder cannot model interactions between documents.
2. The training data does not provide signals that listwise models profit from.
3. Assessing topical relevance does not require document interactions.

➜ We build a synthetic task which requires document interactions.

α-nDCG@10 ($\alpha = 0.99$) on the synthetic task

Model	**TREC DL 19**	**TREC DL 20**
monoELECTRA	0.794	0.765
Set-Encoder	**0.830**†	**0.803**†

Set-Encoder

Listwise Re-Ranking

Three hypotheses why the Set-Encoder does not improve over monoELECTRA:

1. ~~The Set-Encoder cannot model interactions between documents.~~
2. The training data does not provide signals that listwise models profit from.
3. Assessing topical relevance does not require document interactions.

➜ We build a synthetic task which requires document interactions.

α-nDCG@10 ($\alpha = 0.99$) on the synthetic task

Model	**TREC DL 19**	**TREC DL 20**
monoELECTRA	0.794	0.765
Set-Encoder	**0.830**†	**0.803**†

Set-Encoder

Listwise Re-Ranking

Three hypotheses why the Set-Encoder does not improve over monoELECTRA:

1. ~~The Set-Encoder cannot model interactions between documents.~~
2. The training data does not provide signals that listwise models profit from.
3. Assessing topical relevance does not require document interactions.

Set-Encoder

Listwise Re-Ranking

Three hypotheses why the Set-Encoder does not improve over monoELECTRA:

1. ~~The Set-Encoder cannot model interactions between documents.~~

2. The training data does not provide signals that listwise models profit from.

3. Assessing topical relevance does not require document interactions.

Model	**TREC DL 19**		**TREC DL 20**		**TIREx**
First Stage	BM25	CBv2	BM25	CBv2	
First Stage	0.480	0.732	0.494	0.724	0.394
RankGPT-4o	0.725	<u>0.784</u>	0.719	0.793	–
RankGPT-4o Full	<u>0.732</u>	0.781	0.711	**0.799**	–
RankZephyr	0.719	0.749	<u>0.720</u>	<u>0.798</u>	**0.478**
monoELECTRA$_{\mathrm{BASE}}$	0.720	0.768	0.711	0.770	0.457
monoELECTRA$_{\mathrm{LARGE}}$	**0.733**	0.765	**0.727**	**0.799**	<u>0.475</u>
Set-Encoder$_{\mathrm{BASE}}$	0.724	**0.788**	0.710	0.777	0.459
Set-Encoder$_{\mathrm{LARGE}}$	Still training :(				

Set-Encoder

Permutation Invariance

Re-ordering input documents affects previous listwise model's ranking preferences.

Set-Encoder

Permutation Invariance

Re-ordering input documents affects previous listwise model's ranking preferences.

We create corrupted BM25 rankings to test a model's robustness to permutations.

1. Ideal ranking
2. Original BM25 ranking
3. Randomly shuffled ranking
4. Inverse ideal ranking

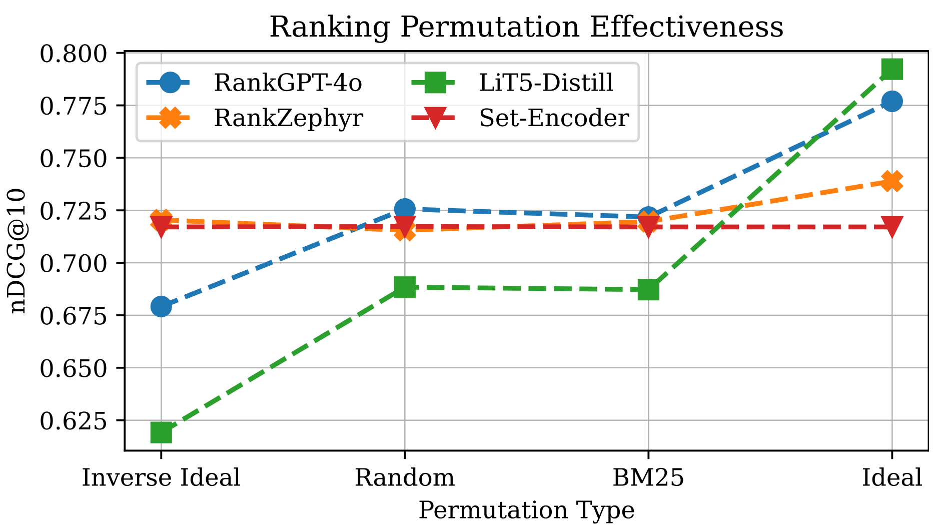
Set-Encoder

Permutation Invariance

Re-ordering input documents affects previous listwise model's ranking preferences.

We create corrupted BM25 rankings to test a model's robustness to permutations.

1. Ideal ranking
2. Original BM25 ranking
3. Randomly shuffled ranking
4. Inverse ideal ranking



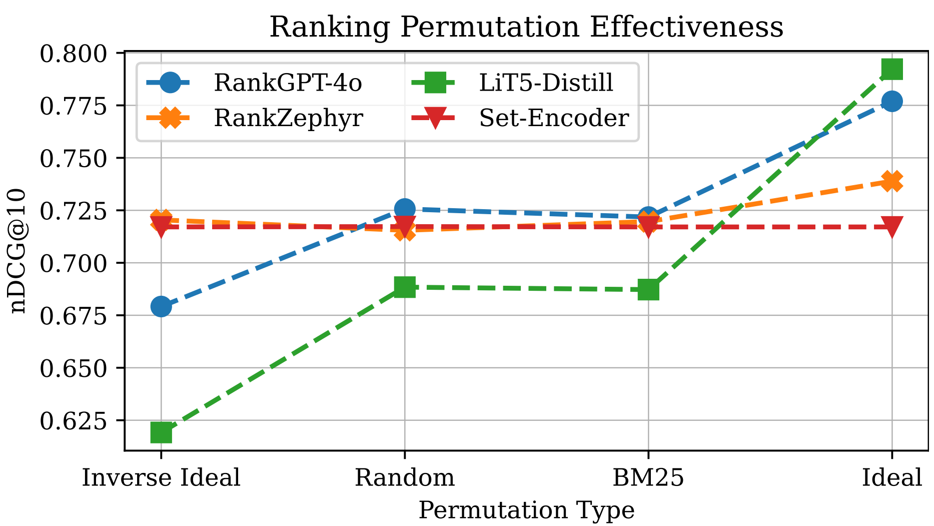
Set-Encoder

Permutation Invariance

Re-ordering input documents affects previous listwise model's ranking preferences.

We create corrupted BM25 rankings to test a model's robustness to permutations.

1. Ideal ranking
2. Original BM25 ranking
3. Randomly shuffled ranking
4. Inverse ideal ranking



- Set-Encoder is invariant to the order of the input documents.

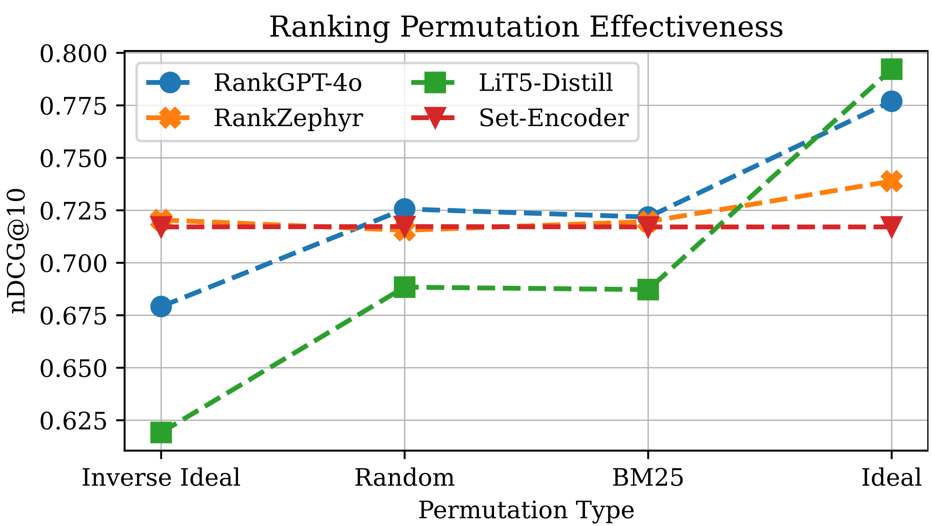
Set-Encoder

Permutation Invariance

Re-ordering input documents affects previous listwise model's ranking preferences.

We create corrupted BM25 rankings to test a model's robustness to permutations.

1. Ideal ranking
2. Original BM25 ranking
3. Randomly shuffled ranking
4. Inverse ideal ranking

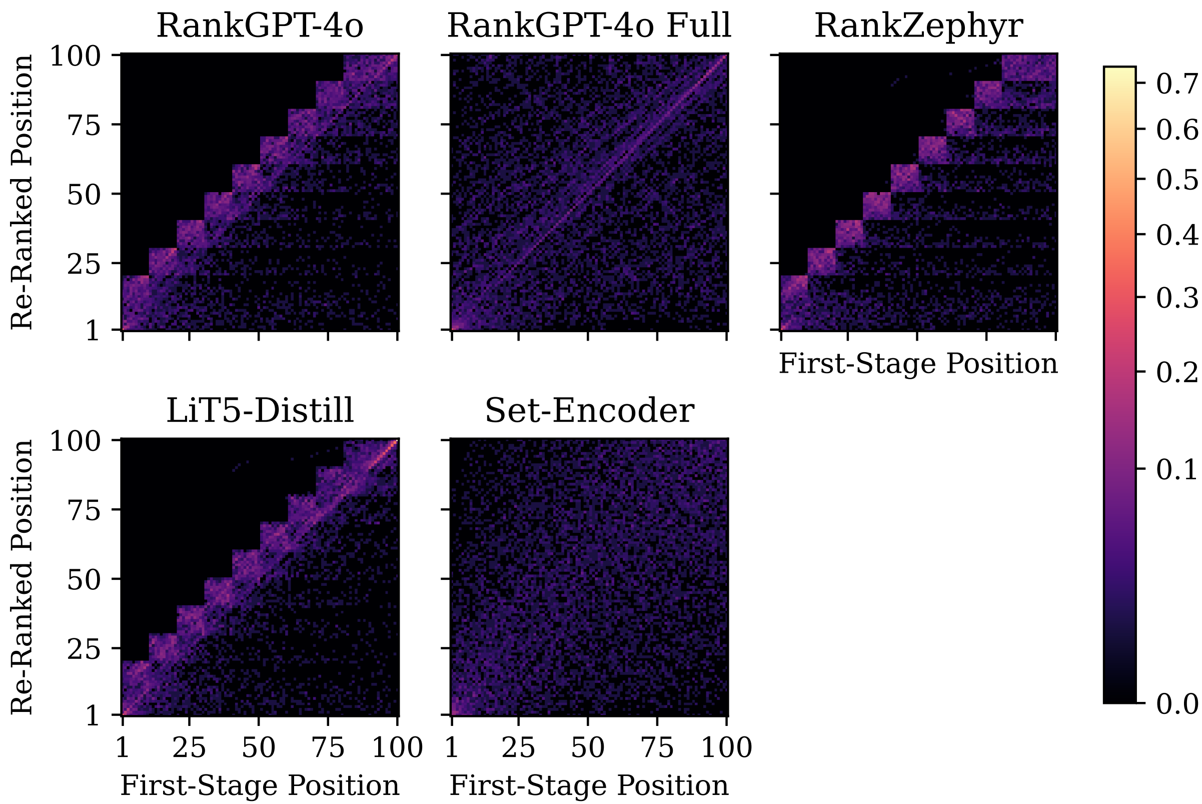


- Set-Encoder is invariant to the order of the input documents.
- Previous listwise re-rankers are biased by the order of the input documents.

Set-Encoder

Permutation Invariance

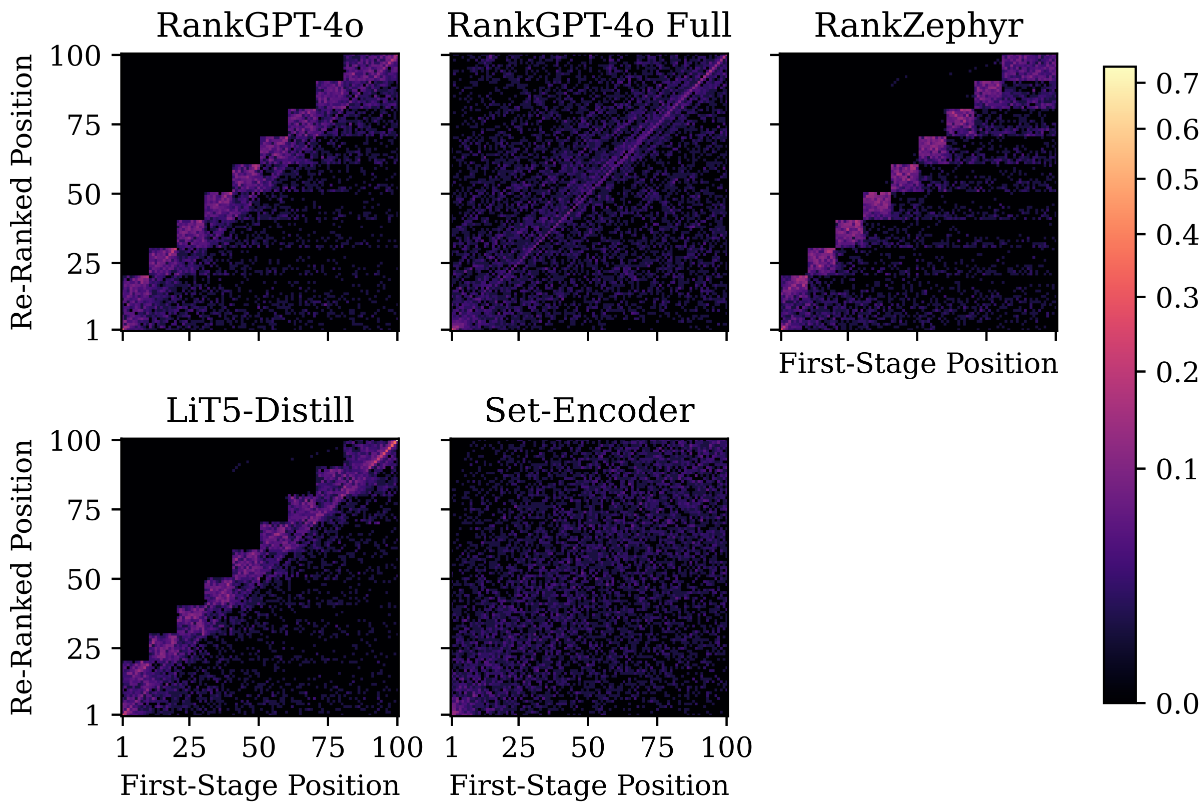
Previous listwise re-rankers are biased by the order of the input documents.



Set-Encoder

Permutation Invariance

Previous listwise re-rankers are biased by the order of the input documents.



A substantial number of previous works attempt to mitigate these positional biases.
[Zhuang et al., SIGIR'24; Parry et al., arXiv'24]

➜ Making the model permutation-invariant is a more principled approach.

Set-Encoder

Conclusion

We introduced the Set-Encoder architecture that enables inter-document interactions in a permutation-invariant way.

- Permutation invariance is crucial for robustness and efficiency.
- Inter-document interactions do not lead to more effective models when assessing topical relevance.
- For more complex tasks requiring inter-document interactions, the Set-Encoder is a promising architecture.

Code and paper @ https://github.com/webis-de/set-encoder

Improving Cross-Encoders

Conclusion

Bottom line:

1. Decoder-only is cool, but do not forget our friend, the encoder-only model.

Conclusion

Bottom line:

1. Decoder-only is cool, but do not forget our friend, the encoder-only model.

Improving Cross-Encoders

Conclusion

Bottom line:

1. Decoder-only is cool, but do not forget our friend, the encoder-only model.

2. “Architecture-fine-tuning” combined with parameter fine-tuning can significantly improve effectiveness and efficiency.

Improving Cross-Encoders

Conclusion

Bottom line:

1. Decoder-only is cool, but do not forget our friend, the encoder-only model.

2. “Architecture-fine-tuning” combined with parameter fine-tuning can significantly improve effectiveness and efficiency.

3. Our current evaluation setups are insufficient to determine if listwise models are better than pointwise ones.

Improving Cross-Encoders

Conclusion

Bottom line:

1. Decoder-only is cool, but do not forget our friend, the encoder-only model.

2. “Architecture-fine-tuning” combined with parameter fine-tuning can significantly improve effectiveness and efficiency.

3. Our current evaluation setups are insufficient to determine if listwise models are better than pointwise ones.

Thank you!

Sparse Cross-Encoder **Rank-DistiLLM** **Set-Encoder**

Sparse Cross-Encoder

Full TREC DL Table

Task		Full Att. / Longformer						Sparse Cross-Encoder						QDS
	$w =$	∞	64	16	4	1	0	∞	64	16	4	1	0	64
Passage	2019	.724	.719†	.725†	.719	.714	.694	.722	.717	.724	**.728**	.715	.696	.720†
Passage	2020	.674	.681†	.680	**.684**	.676	.632	.666	.672	.661	.665	.649	.605	.682
Passage	2021	**.656**	.653	.650	.645	.629	.602	**.656**	.650	.639	.647	.625	.593	**.656**†
Passage	2022	**.496**	.494†	.487	.486	.481	.441	.490	.492†	.479	.484	.471	.427	.495†
Passage	Avg.	.619	.619†	.616†	.615†	.607	.572	.615†	.615†	.607	.612†	.596	.560	**.620**†
Document	2019	.658	.683	.678	.667	.689	.663	.638	.672	.685	.669	.692	.646	**.697**
Document	2020	.622	.640	.639	**.661**	.655	.644	.636	.638	.650	.642	.657	.638	.639
Document	2021	.678	.671	.681	**.683**	**.683**	.629	.677	.681	.681	.670	.679	.644	.676
Document	2022	.424	.425	.431	.425	.409	.389	.421	**.446**	.443	.417	.424	.405	.428
Document	Avg.	.575	.582	.586†	.587	.584†	.556	.573	.590	**.594**	.577	.589	.561	.587†

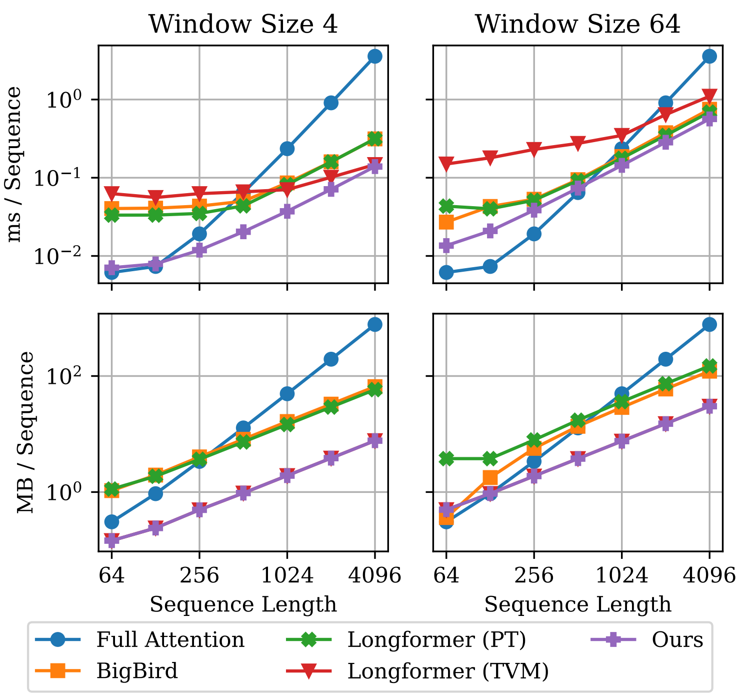
Sparse Cross-Encoder

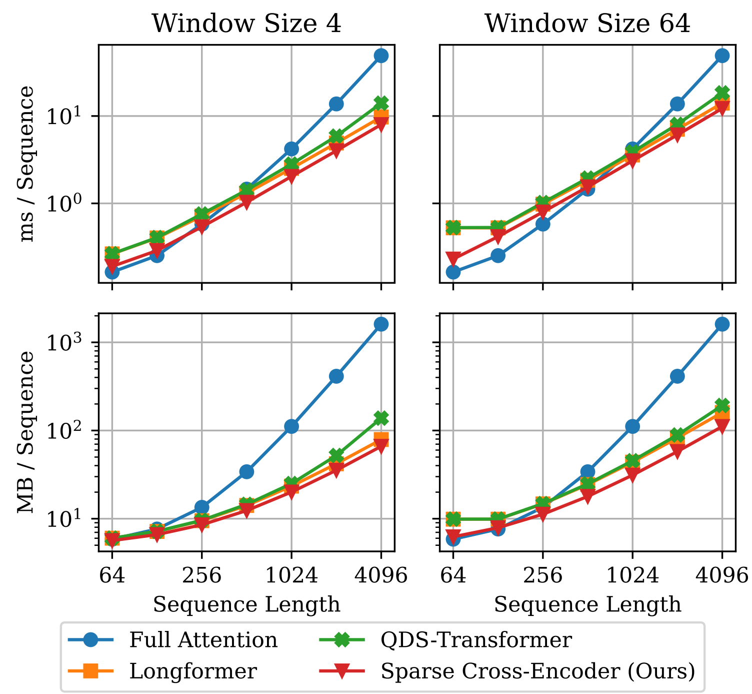
TIREx Table

Corpus	Doc. Len.	First Stage	monoT5			monoBERT		Sparse CE	
			Base	Large	3b	Base	Large	512	4096
Antique	49.9	.510	.505	.527	.537	.507	.484	**.540**	.174
Args.me	435.5	**.405**	.305	.338	.392	.314	.371	.313	.180
CW09	1132.6	.178	.186	.182	.201	.192	.134	.198	**.212**
CW12	5641.7	**.364**	.260	.266	.279	.263	.251	.312	.338
CORD-19	3647.7	.586	.688	.636	.603	**.690**	.625	.673	.642
Cranfield	234.8	.008	.006	.007	.007	.006	.006	**.009**	.003
Disks4+5	749.3	.429	.516	.548	**.555**	.514	.494	.487	.293
GOV	2700.5	.266	.320	.327	**.351**	.318	.292	.316	.292
GOV2	2410.3	.467	.486	.513	**.514**	.489	.474	.503	.460
MED.	309.1	**.366**	.264	.318	.350	.267	.298	.237	.180
NFCorpus	364.6	.268	.295	.296	**.308**	.295	.288	.284	.151
Vaswani	51.3	.447	.306	.414	.458	.321	**.476**	.436	.163
WaPo	713.0	.364	.451	**.492**	.476	.449	.438	.434	.296
Average	–	.358	.353	.374	**.387**	.356	.356	.365	.260

Cross-Encoder

Efficiency Graphs





Set-Encoder

Efficiency

Previous listwise re-rankers are also less efficient.

Model	# Parameters	Inference Time
RankGPT-4o (20,10)	?	≈35s
RankGPT-4o (100,0)	?	≈11s
RankZephyr	7B	21.1s
LiT5-Distill	248M	4.0s
monoELECTRA$_{\texttt{BASE}}$	109M	0.3s
monoELECTRA$_{\texttt{LARGE}}$	334M	?
Set-Encoder$_{\texttt{BASE}}$	109M	0.5s
Set-Encoder$_{\texttt{LARGE}}$	334M	?