PoolingFormer: Long Document Modeling with Pooling Attention

Hang Zhang, Yeyun Gong, Yelong Shen, Weisheng Li, Jiancheng Lv, Nan Duan, Weizhu Chen

Background

How to handle a long document with thousands tokens?

News Article

As in years past, a lot of the food trends of the year were based on creating perfectly photogenic dishes. An aesthetically pleasing dish, however, doesn't mean it will stand the test of time. In fact, it's not uncommon for food trends to be all the hype one year and die out the next. From broccoli coffee to "bowl food," here are 10 food trends that you likely won't see in 2019.

...[15 sentences with 307 words are abbreviated from here.]

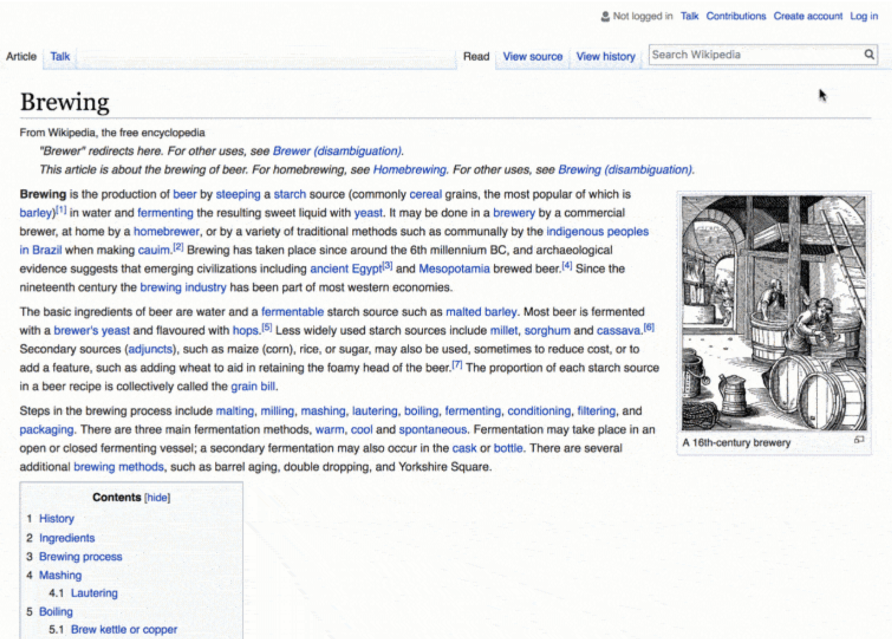
In 2018, restaurants all over the US decided it was a good idea to place gold foil on everything from ice cream to chicken wings to pizza resulting in an expensive food trend. For example, the Ainsworth in New York City sells $1,000 worth of gold covered chicken wings. It seems everyone can agree that this is a food trend that might soon disappear.

10 food trends that you likely won't see in 2019

Summarization

Select a question:

- when are hops added to the brewing process?
- what does the word china mean in chinese?
- where is the world s largest ice sheet located today?
- who lives in the imperial palace in tokyo?
- where does the last name hogan come from?
- who is the author of the book arabian nights?
- how many episodes in season 2 breaking bad?
- where is blood pumped after it leaves the right ventricle?
- where is the bowling hall of fame



Question:

when are hops added to the brewing process?

Short Answer:

The boiling process

Long Answer:

After mashing , the beer wort is boiled with hops (and other flavourings if used) in a large tank known as a " copper " or brew kettle – though historically the mash vessel was used and is still in some small breweries . The boiling process is where chemical reactions take place , including sterilization of the wort to remove unwanted bacteria , releasing of hop flavours , bitterness and aroma compounds through isomerization , stopping of enzymatic processes ,

QA

Long Document Modeling

Task: Long document QA, Summarization, Translation, even Vision Task (64*64 pixels) ...

Naive ways:

1. Truncation
   (Extract the beginning or end of the document, drop others)

2. N-stage
   (Split the input into small units then hierarchical modeling)

Poor performance because lots of information is dropped

Long Document Modeling

Question:

What’ s the key problem in Long Document Modeling ?

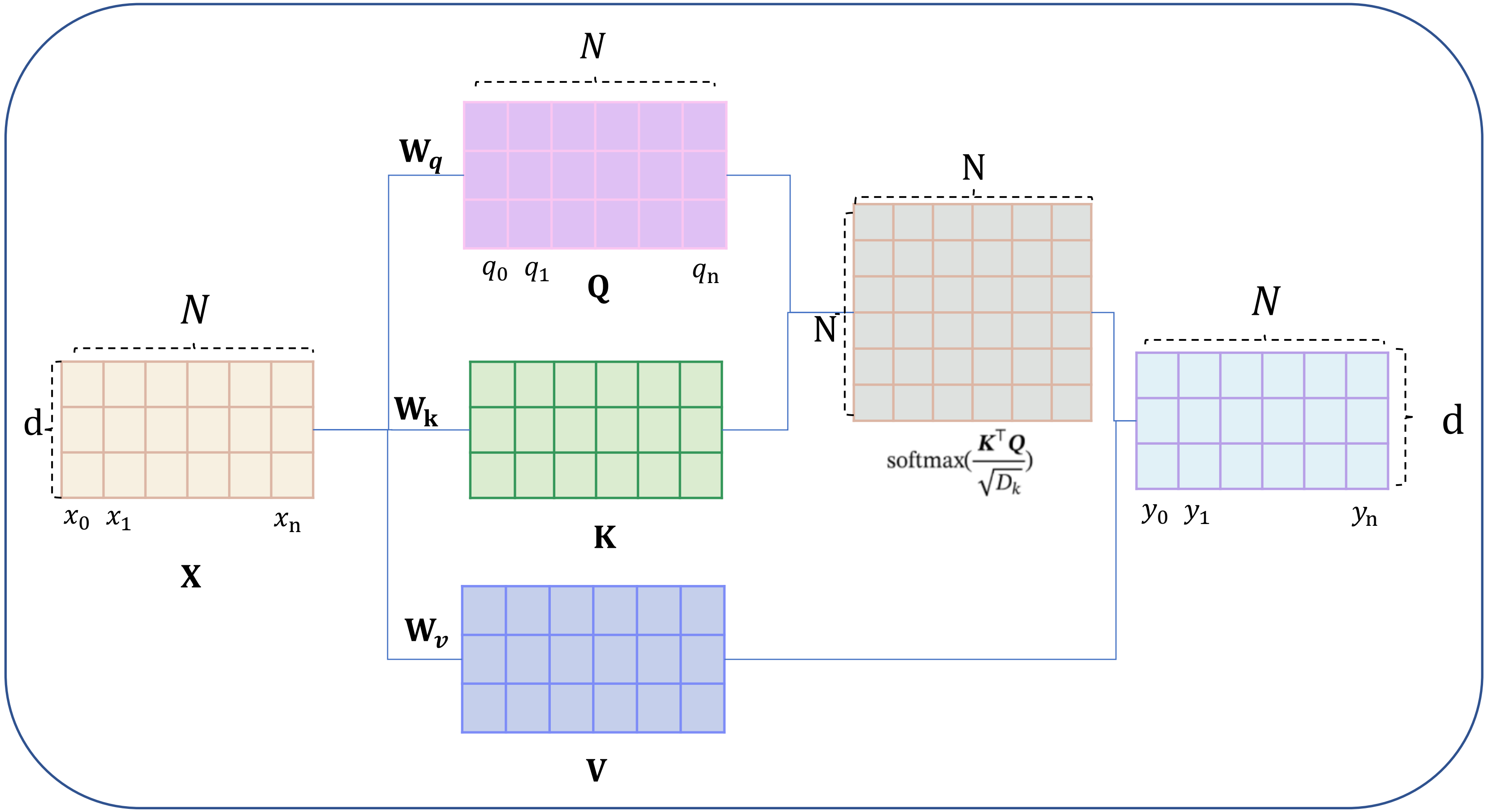
Why pretrained models (Bert,Roberta..) constrained input length of 512?

Self-Attention is **quadratic dependency** on the sequence length

Self-Attention

Sequence of text embeddings: $\mathbf{X} = (x_1.x_2,\ldots,x_\mathrm{N})$

Out: $\mathbf{Y} = (\mathrm{y}_1.\mathrm{y}_2,\ldots,\mathrm{y}_\mathrm{N})$

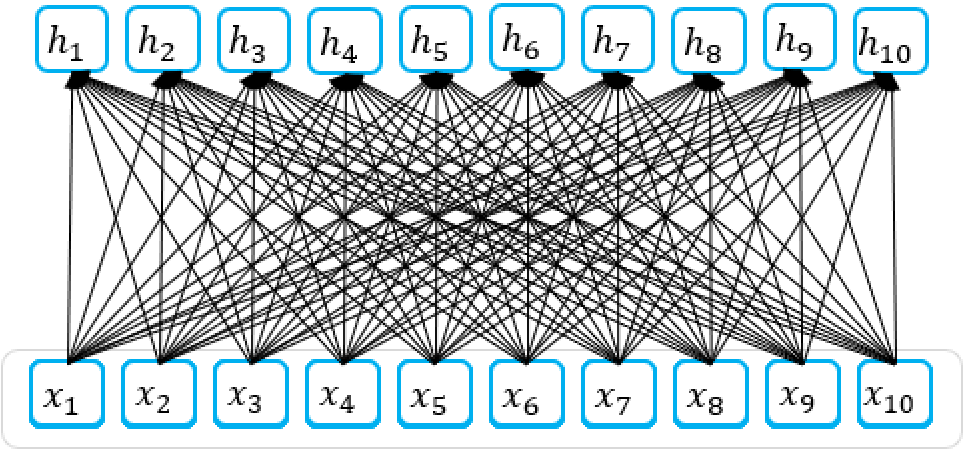


$$\begin{pmatrix}\mathbf{Q}\\ \mathbf{K}\\ \mathbf{V}\end{pmatrix} = \begin{pmatrix}\mathbf{W_q}\\ \mathbf{W_k}\\ \mathbf{W_v}\end{pmatrix}\mathbf{X} + \begin{pmatrix}\mathbf{b_q}\\ \mathbf{b_k}\\ \mathbf{b_v}\end{pmatrix}$$

$$\boldsymbol{y}_i^T = \mathrm{Softmax}\left(\alpha \boldsymbol{q}_i^T \mathbf{K}\right)\mathbf{V}^T$$

Self-Attention

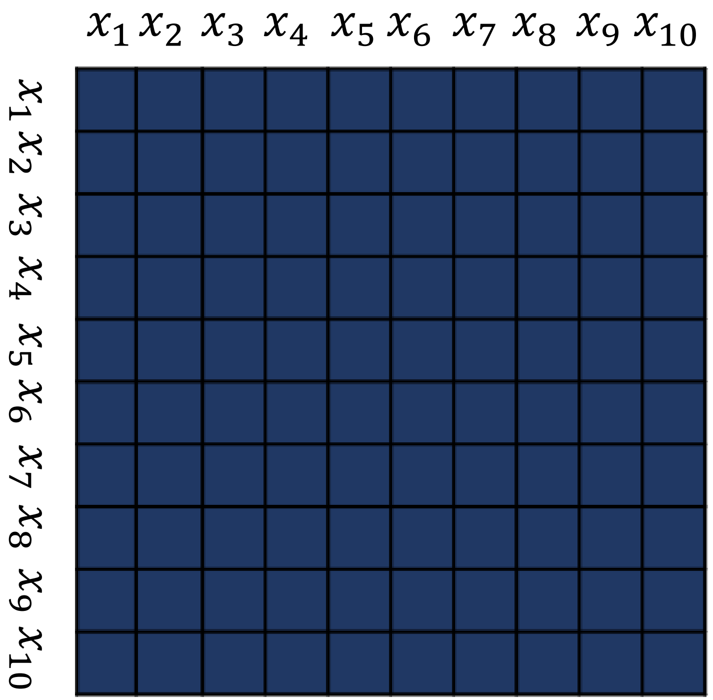
Inputs



Quadratic dependency on the sequence length

(computational cost and memory consumption)

$\mathcal{O}(n^2)$

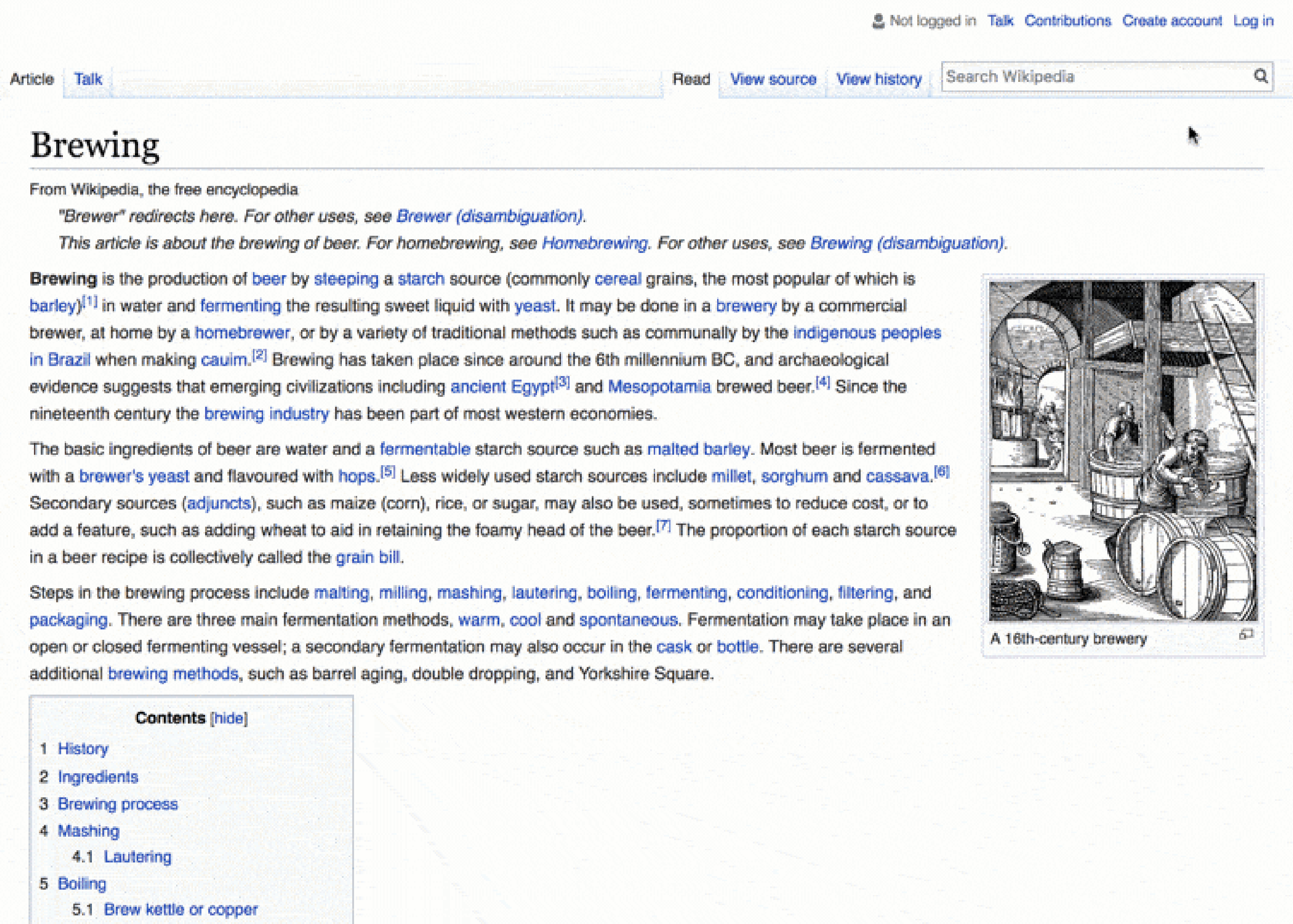


$$\begin{pmatrix}\mathbf{Q}\\ \mathbf{K}\\ \mathbf{V}\end{pmatrix} = \begin{pmatrix}\mathbf{W_q}\\ \mathbf{W_k}\\ \mathbf{W_v}\end{pmatrix}\mathbf{X} + \begin{pmatrix}\mathbf{b_q}\\ \mathbf{b_k}\\ \mathbf{b_v}\end{pmatrix}$$

$$\boldsymbol{y}_i^T = \text{Softmax}\left(\alpha \boxed{\boldsymbol{q}_i^T \mathbf{K}}\right)\mathbf{V}^T$$

Self-Attention

Hard to handle long sequence !



Not logged in Talk Contributions Create account Log in

Article Talk Read View source View history Search Wikipedia

Brewing

From Wikipedia, the free encyclopedia

"Brewer" redirects here. For other uses, see Brewer (disambiguation).

This article is about the brewing of beer. For homebrewing, see Homebrewing. For other uses, see Brewing (disambiguation).

Brewing is the production of beer by steeping a starch source (commonly cereal grains, the most popular of which is barley)[1] in water and fermenting the resulting sweet liquid with yeast. It may be done in a brewery by a commercial brewer, at home by a homebrewer, or by a variety of traditional methods such as communally by the indigenous peoples in Brazil when making cauim.[2] Brewing has taken place since around the 6th millennium BC, and archaeological evidence suggests that emerging civilizations including ancient Egypt[3] and Mesopotamia brewed beer.[4] Since the nineteenth century the brewing industry has been part of most western economies.

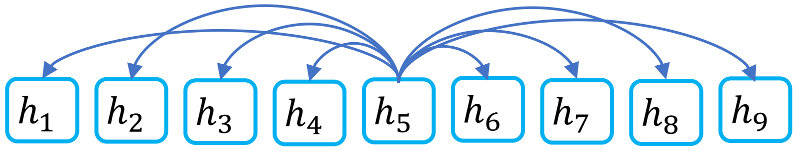
The basic ingredients of beer are water and a fermentable starch source such as malted barley. Most beer is fermented with a brewer's yeast and flavoured with hops.[5] Less widely used starch sources include millet, sorghum and cassava.[6] Secondary sources (adjuncts), such as maize (corn), rice, or sugar, may also be used, sometimes to reduce cost, or to add a feature, such as adding wheat to aid in retaining the foamy head of the beer.[7] The proportion of each starch source in a beer recipe is collectively called the grain bill.

Steps in the brewing process include malting, milling, mashing, lautering, boiling, fermenting, conditioning, filtering, and packaging. There are three main fermentation methods, warm, cool and spontaneous. Fermentation may take place in an open or closed fermenting vessel; a secondary fermentation may also occur in the cask or bottle. There are several additional brewing methods, such as barrel aging, double dropping, and Yorkshire Square.

A 16th-century brewery

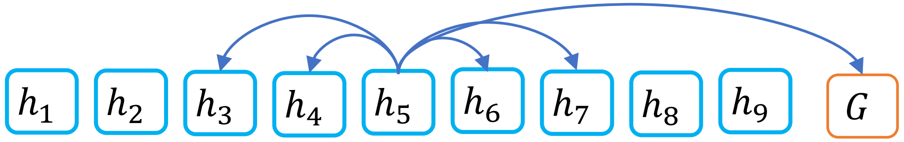
Contents [hide]

Sparse Attention



Tokens “see” the entire sequence

Original Self-Attention



Tokens “see” the local neighbors and global tokens

Sparse attention Longformer & Bigbird

Sparse attention

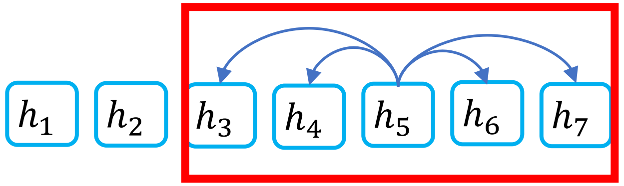
limit the receptive field of tokens:

$$\mathcal{N}(i, w_1) = \{i - w_1, ..., i, ..., i + w_1\}$$

$$\boldsymbol{y}_i^T = \mathrm{Softmax}\left(\alpha \boldsymbol{q}_i^T \mathbf{K}_{\mathcal{N}(i,w_1)}\right) \mathbf{V}_{\mathcal{N}(i,w_1)}^T$$

Motivation

Intuitively:



How to see further efficiently ?

1. the farther tokens "see", the better performance

2. the farther tokens "see", the higher computational complexity

3. Local neighbors are import. Farther neighbors contain more redundant information

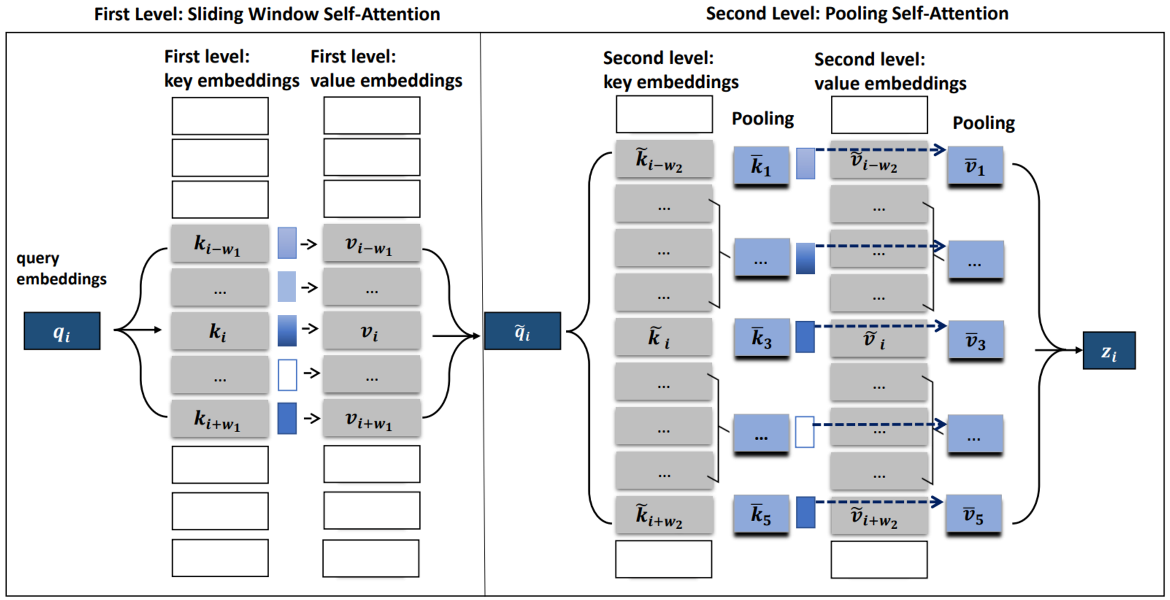
Motivation:

Different attention strategies for different distance neighbors

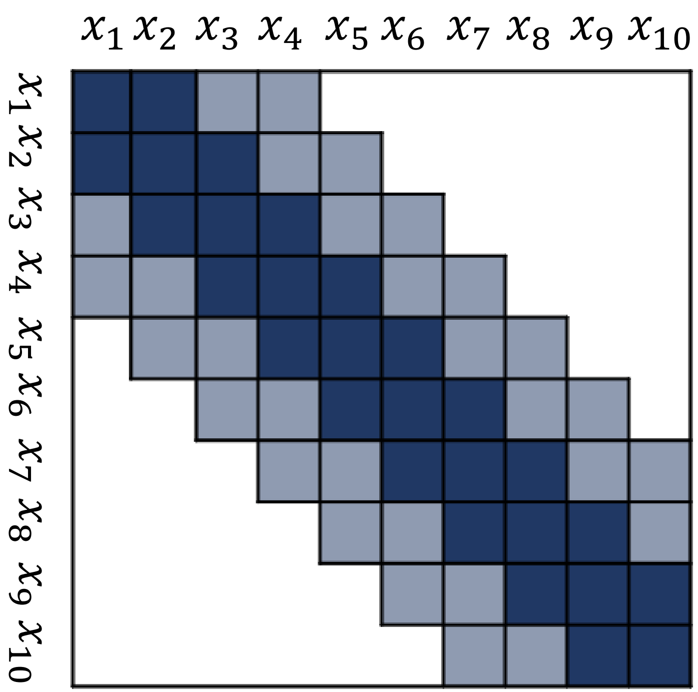
PoolingFormer

Two-level attention schema

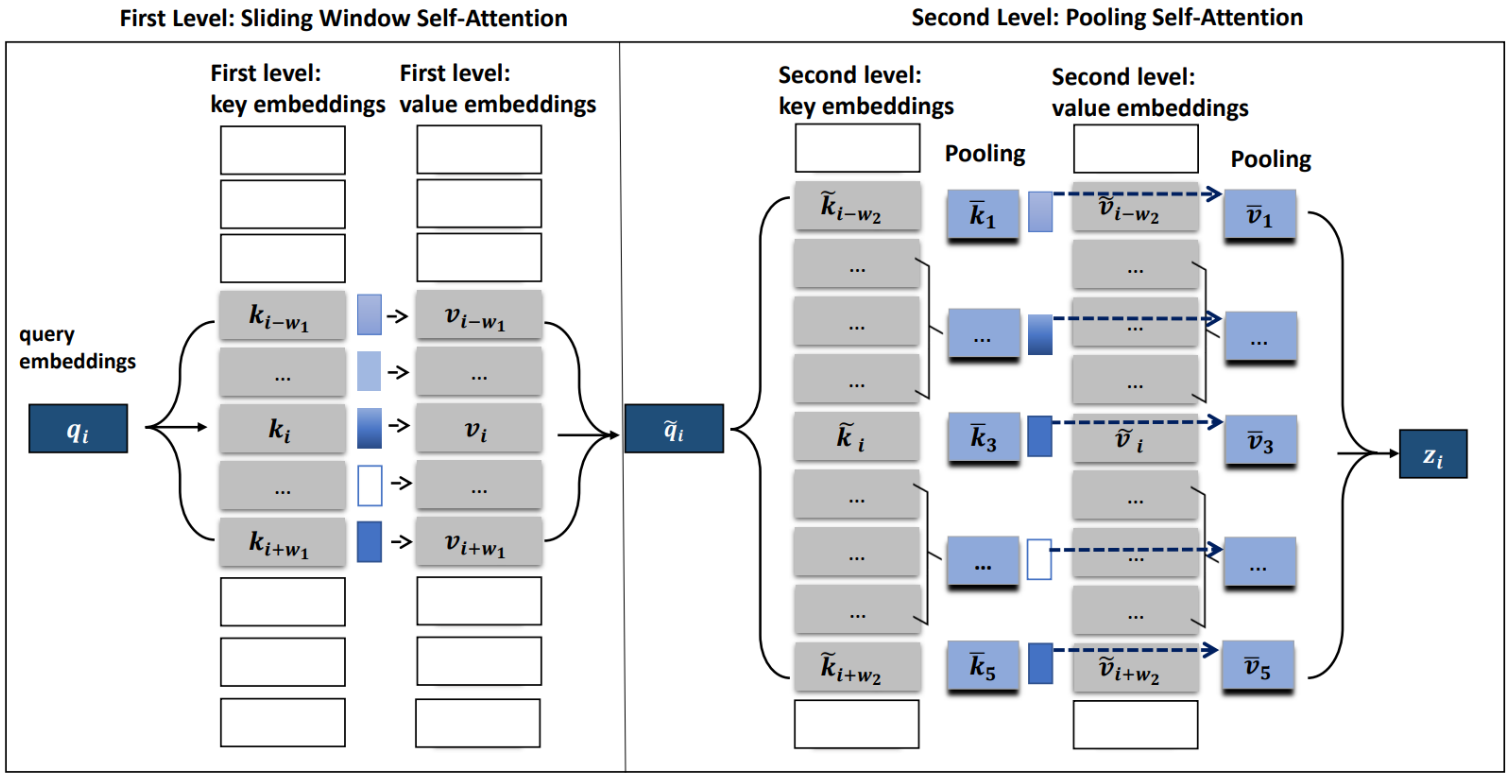
1. For closer neighbors: Full Attention
2. For farther neighbors: Pooling attention



$\mathcal{O}(kn)$



PoolingFormer



First Level

$$\mathcal{N}(i, w_1) = \{i - w_1, \dots, i, \dots . i + w_1\}$$

$$y_i^T = Softmax\left(a q_i^T K_{\mathcal{N}(i,w_1)}\right) V_{\mathcal{N}(i,w_1)}^T$$

Second Level:

$$\mathcal{N}(i, w_2) = \{i - w_2, \dots, i, \dots . i + w_2\}$$

$$\bar{\mathbf{K}}_i = \text{Pooling}(\widetilde{\mathbf{K}}_{\mathcal{N}(i,w_2)}; \kappa, \xi)$$

$$\bar{\mathbf{V}}_i = \text{Pooling}(\widetilde{\mathbf{V}}_{\mathcal{N}(i,w_2)}; \kappa, \xi)$$

$$z_i^T = \text{Softmax}\left(\alpha \tilde{\boldsymbol{q}}_i^T \bar{\mathbf{K}}_i\right) \bar{\mathbf{V}}_i^T$$

PoolingFormer

- **Trainable Pooling Mechanisms : LDConv**

$$\mathbf{V} = (\boldsymbol{v}_1, ..., \boldsymbol{v}_m) \rightarrow ((\boldsymbol{v}_1, ..., \boldsymbol{v}_\kappa), (\boldsymbol{v}_{1+\xi}, ..., \boldsymbol{v}_{1+\xi+\kappa}), ...)$$

$$(\delta_1, ..., \delta_\kappa)^T = \mathrm{Softmax}(\mathbf{W_p} \boldsymbol{v}_i)$$

$$\mathrm{LDConv}(\boldsymbol{v}_1, ..., \boldsymbol{v}_\kappa) = \sum_{i=1}^{\kappa} \delta_i \cdot \boldsymbol{v}_i$$

Mean-LDConv : $(\delta_1, ..., \delta_\kappa)^T = \mathrm{Softmax}(\mathbf{W_p} \bar{\boldsymbol{v}})$

PoolingFormer for Document-level Summarization

Model	ROUGE-1	ROUGE-2	ROUGE-L
Sent-PTR-512	42.32	15.63	38.06
Extr-Abst-TLM-512	41.62	14.69	38.03
PEGASUS-512	44.21	16.95	38.83
Dancer-512	45.01	17.60	40.56
BigBird-16k	46.63	19.02	41.77
LED-4k	44.40	17.94	39.76
LED-16k	46.63	19.62	41.83
Poolingformer-4k	47.86	19.54	42.35
Poolingformer-16k	**48.47**	**20.23**	**42.69**

PoolingFormer for Document-level QA

Google's NQ

Long Answer Leaderboard

Rank	Model	Participant	Affiliation	Attempt Date	F1	Precision	Recall	R@P = 90	R@P = 75	R@P = 50
1	PoolingFormer	PoolingFormer_Team	To-Be-Released	1/14/21	0.79823	0.7847	0.81224	0.62448	0.8379	0.88748
2	ClusterFormer	Dynamics_365_AI_Research	Microsoft	9/28/20	0.77969	0.78471	0.77473	0.56986	0.80654	0.86686
3	ETC-large	philly_pham	Google Research	5/25/20	0.7778	0.77476	0.78087	0.52204	0.79864	0.86598
4	ReflectionNet-ensemble	Wide_Field	Microsoft STCA NLP Group	2/9/20	0.77185	0.76791	0.77583	0.53345	0.78526	0.85238
5	ClusterFormer	Dynamics_365_AI_Research	Microsoft	9/19/20	0.76351	0.76151	0.76552	0.54354	0.77495	0.85852

Short Answer Leaderboard

Rank	Model	Participant	Affiliation	Attempt Date	F1	Precision	Recall	R@P = 90	R@P = 75	R@P = 50
1	ReflectionNet-ensemble	Wide_Field	Microsoft STCA NLP Group	2/9/20	0.64114	0.70445	0.58827	0.35046	0.54355	0.66144
2	PoolingFormer	PoolingFormer_Team	To-Be-Released	1/14/21	0.61629	0.70369	0.5482	0.17567	0.509	0.63995
3	RoBERTa-mnlp-ensemble	GAAMA	IBM Research AI	1/6/20	0.61409	0.6961	0.54936	0.28223	0.50436	0.62747
4	RikiNet_v2	DREAM_Losin	anonymous	11/29/19	0.61302	0.67612	0.56069	0.18089	0.48432	0.6417
5	ClusterFormer	Dynamics_365_AI_Research	Microsoft	9/28/20	0.60944	0.62149	0.59785	0.29326	0.49506	0.64053

TyDi QA

Passage Answer Task

Rank	Model	Participant	Affiliation	Attempt Date	F1	Precision	Recall
1	PoolingFormer	PoolingFormer_Team	To-Be-Released	1/25/2021	79.53	80.42	78.79
2	BERT with language-clustered vocab	Google-Research	Google Research	6/4/2020	77.65	77.43	78.00
3	GAAMA (XLM-R) with ARES system	GAAMA	IBM Research AI	11/13/2020	72.56	73.55	72.12
4	tydiqa-baseline	tydiqa-team	Google Research	2/15/2020	64.40	62.32	67.13
			IBM				

Minimal Answer Task

Rank	Model	Participant	Affiliation	Attempt Date	F1	Precision	Recall
1	PoolingFormer	PoolingFormer_Team	To-Be-Released	1/25/2021	67.65	73.48	63.27
2	GAAMA (XLM-R) with ARES system	GAAMA	IBM Research AI	11/13/2020	66.08	70.78	62.20
3	BERT with language-clustered vocab	Google-Research	Google Research	6/4/2020	63.40	67.19	60.21
4	mBERT-mnlp-single	GAAMA	IBM Research AI	8/12/2020	53.19	61.47	47.28

Ablation Study

Table 5. Ablation study of Poolingformer$_{base}$ with different window lengths on NQ dev set. w_1: the size of the first level window. w_2: the size of the second level window. C: the compression rate of the second level window controlled by adjusting the kernel size and stride size of the pooling.

Setting	w_1	w_2	C	LA F1	SA F1
RoBERTa$_{base}$	-	-	-	63.8	43.2
Poolingformer$_{base}$	128	-	-	66.3	43.1
Poolingformer$_{base}$	256	-	-	67.4	43.4
Poolingformer$_{base}$	512	-	-	66.1	42.6
Poolingformer$_{base}$	128	256	4	67.9	45.0
Poolingformer$_{base}$	128	512	4	**68.7**	**45.2**
Poolingformer$_{base}$	128	2,048	8	66.9	42.6
Poolingformer$_{base}$	128	2,048	16	67.0	44.4

Table 6. Ablation study of pooling and fusion approaches.

Setting	LA F1	SA F1
Poolingformer$_{base}$(Without $2nd$ level window)	66.3	43.1
Poolingformer$_{base}$(MEAN)	68.5	43.7
Poolingformer$_{base}$(MAX)	68.6	**45.3**
Poolingformer$_{base}$(LDConv)	**68.7**	45.2
Poolingformer$_{base}$(MeanLDConv)	67.7	44.1
Poolingformer$_{base}$(LDConv, *Mix*)	67.5	44.6
Poolingformer$_{base}$(LDConv, *Weight Sharing*)	67.2	44.2

Thank you!