BiDAF and QANet

Sviatokum Polina

02.03.2019

QA as reading comprehension task

Extractive datasets

Answer to every question is a continuous span of context.

Multiple choice datasets

Questions about context with multiple choice.

Cloze datasets

Statements related to context with blanks to fill in.

Datasets

Dataset	Size	Source of Que.	Source of Docs	Answer Type
CNN/DailyMail	300K 1.4M	Synthetic	News	Fill in entity
RACE	50K 870K	English exam	English exam	Multiple choices
MCTest	500 2K	Crowd sourced	Fictional stories	Multiple choices
SQuAD	536 100K	Crowd sourced	Wiki.	Span of words
TrivaQA	40K 650K	Trivia websites	Wiki./Web doc.	Span of words

SQuAD leaderboard

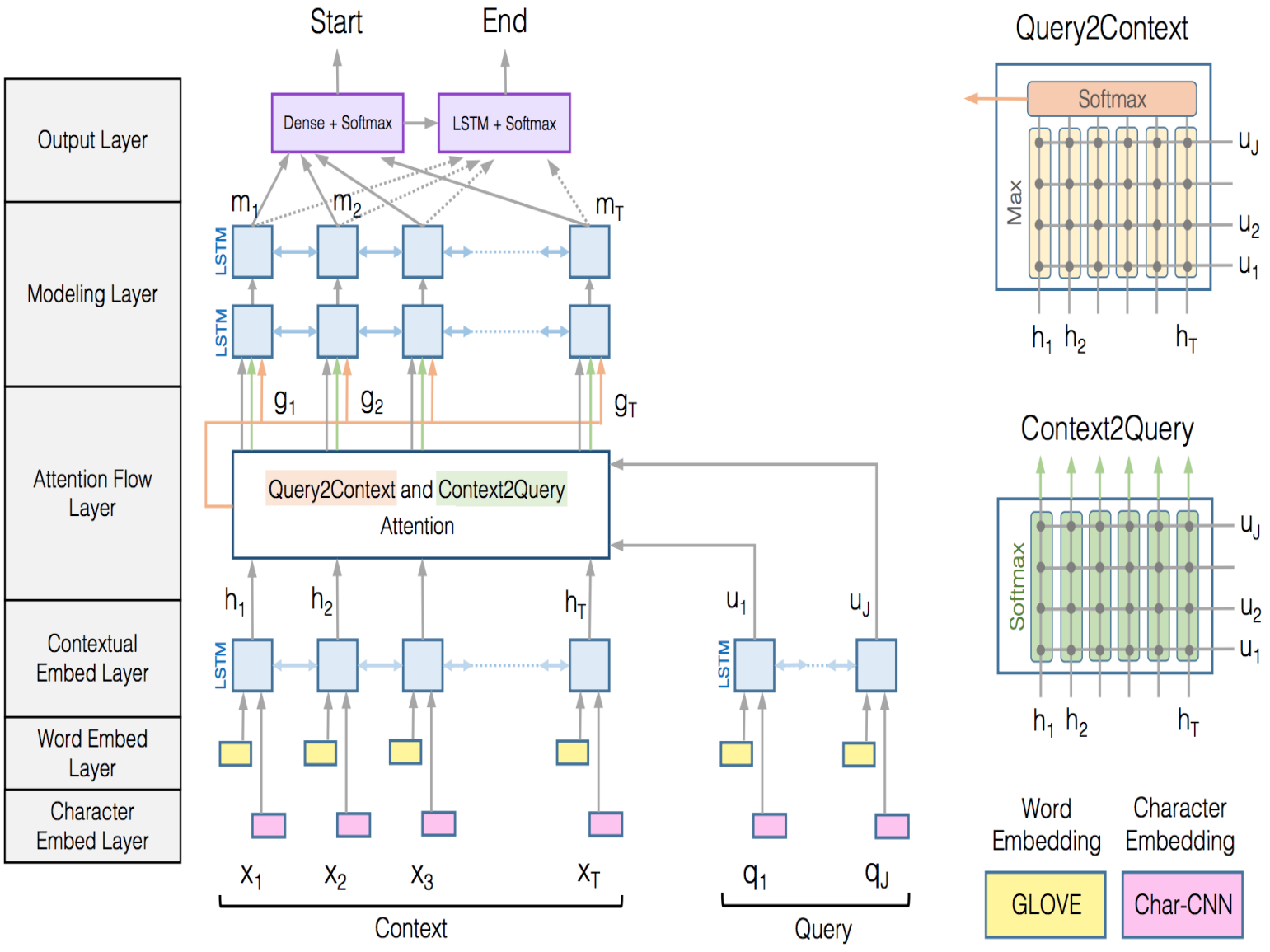
SQuAD v1.1

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	**87.433**	**93.160**
2 Feb 14, 2019	Knowledge-enhanced BERT (single model) *Anonymous*	85.944	92.425
2 Sep 26, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.954	91.677
3 Sep 09, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.356	91.202
3 Oct 05, 2018	BERT (single model) *Google AI Language* https://arxiv.org/abs/1810.04805	85.083	91.835
4 Feb 19, 2019	WD (single model) *Anonymous*	84.402	90.561
4 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490
5 Feb 21, 2019	WD1 (single model) *Anonymous*	83.804	90.429
5 Jul 08, 2018	r-net (ensemble) *Microsoft Research Asia*	84.003	90.147

SQuAD v2.0

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Jan 15, 2019	BERT + MMFT + ADA (ensemble) *Microsoft Research Asia*	**85.082**	**87.615**
2 Jan 10, 2019	BERT + Synthetic Self-Training (ensemble) *Google AI Language* https://github.com/google-research/bert	84.292	86.967
3 Dec 13, 2018	BERT finetune baseline (ensemble) *Anonymous*	83.536	86.096
4 Dec 16, 2018	Lunet + Verifier + BERT (ensemble) *Layer 6 AI NLP Team*	83.469	86.043
4 Dec 21, 2018	PAML+BERT (ensemble model) *PINGAN GammaLab*	83.457	86.122
5 Jan 10, 2019	BERT + Synthetic Self-Training (single model) *Google AI Language* https://github.com/google-research/bert	82.972	85.810

BiDAF


Output Layer
Modeling Layer
Attention Flow Layer
Contextual Embed Layer
Word Embed Layer
Character Embed Layer
Start
End
Dense + Softmax
LSTM + Softmax
LSTM
Query2Context and Context2Query Attention
Context
Query
Query2Context
Softmax
Max
Context2Query
Softmax
Word Embedding
Character Embedding
GLOVE
Char-CNN

Embedding Layers

Word Embeddings

Character Embedding Layer

Produces character-level embedding of each word using CNN. Characters are embedded into vectors, which can be considered as 1D inputs to the CNN.

Word Embedding Layer

Use pre-trained word vectors GloVe to obtain the fixed word embedding of each word.

Embedding Layers

Context embeddings

The concatenation of the character and word embedding vectors is passed to a two-layer Highway Network to obtain d-dimensional embedding vectors.

Contextual Embedding Layer

LSTM in both directions is placed on top of embeddings.

Hence we obtain $2d$-dimensional contextual embeddings for every token

Attention Flow Layer

The inputs to the layer are contextual vector representations of the context C and the query Q.
S – similarity matrix between C and Q.

$$S_{i,j} = \alpha(C_{:i}, Q_{:j})$$

, where α is a trainable scalar function.

$$\alpha(c, q) = W_S^T \mathrm{concat}[c, q, c \circ q]$$

Attention Flow Layer

Context-to-query Attention

$$S_{i,j} = \alpha(C_{:i}, Q_{:j})$$

Context-to-query Attention

Signifies which query words are most relevant to each context word.

$$a_t = \mathrm{softmax}(S_{t:})$$

a_t describes relation between t-th token of context and query

$$\hat{Q}_{:t} = \sum_j (a_t)_j Q_{:j}$$

$\hat{Q}$ – C2Q matrix

Attention Flow Layer

Query-to-context Attention

$$S_{i,j} = \alpha(C_{:i}, Q_{:j})$$

Query-to-context Attention

Signifies which context words have the closest similarity to one of the query words.

$$b = \mathrm{softmax}(\max_{\mathrm{col}}(S))$$

b describes relation between query tokens and their most similar tokens in the context

$$\hat{c} = \sum_i (b_i C_{:i})$$

$\hat{C}$ is tiled $\hat{c}$ vector – Q2C matrix

Attention Flow Layer

Attention Flow Layer

Output G combines C2Q and Q2C.

$$G_{:t} = \beta\left(C_{:t}, \hat{Q}_{:t}, \hat{C}_{:t}\right)$$

, where β is a (possibly) trainable vector function that fuses its (three) input vectors

$$\beta(c, \hat{q}, \hat{c}) = [h; \hat{u}; h \circ \hat{u}; h \circ \hat{h}]$$

Modeling Layer

This layer captures the interaction among the context words conditioned on the query.
We use two layers of bi-directional LSTM, with the output size of d for each direction.
Hence we obtain a matrix $M \in \mathbb{R}^{2d \times T}$, where T is context size

Output Layer

Output Layer

The output layer is application-specific. For SQuAD output layer predicts start and end point of answer span.

$$p^{start} = \mathrm{softmax}(W_{start}^T[G; M])$$

For the end index of the answer phrase, we pass M to another bidirectional LSTM layer and obtain M_2

$$p^{end} = \mathrm{softmax}(W_{end}^T[G; M_2])$$

Training & Testing

Training

$$L(\theta) = -\frac{1}{N}\sum_{i}^{N}\left(\log(p^{start}_{true\_start_i}) + \log(p^{end}_{true\_end_i})\right)$$

Testing

$$start\_idx, end\_idx = \underset{start\_idx<end\_idx}{\operatorname{argmax}} p^{start}_{start\_idx} \cdot p^{end}_{end\_idx}$$

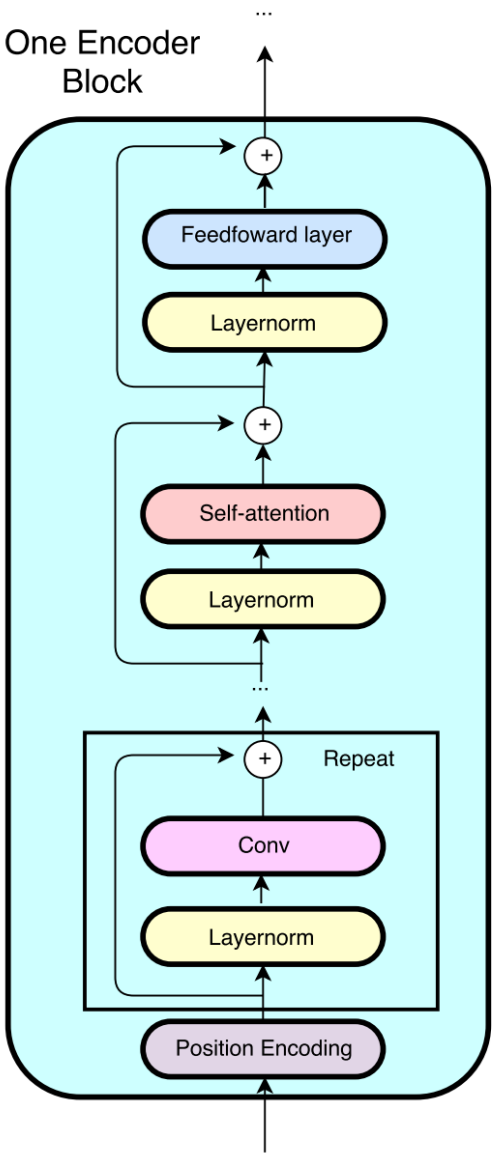
QANet

- What if we don't use RNN (not very new idea)?
- What if we use CNN + self-attention?

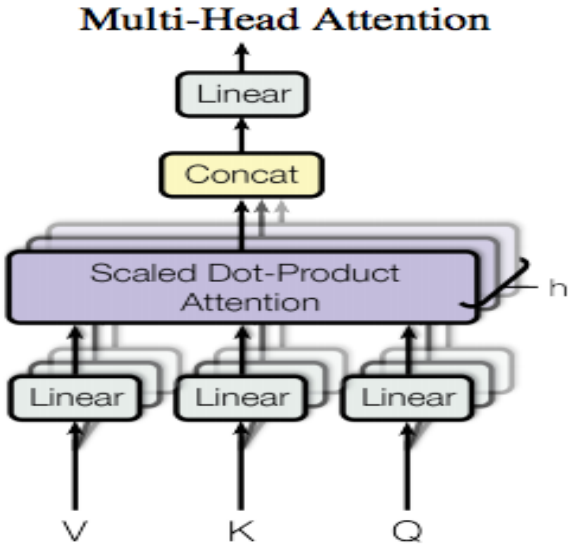
Benefits:

- Faster training and inference
- Training on larger datasets

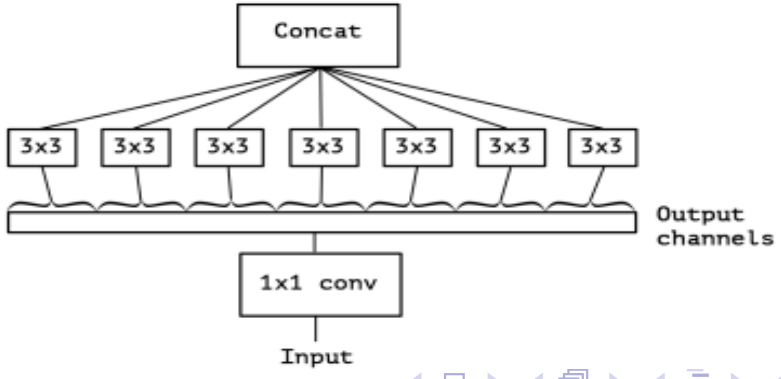
Encoder Block



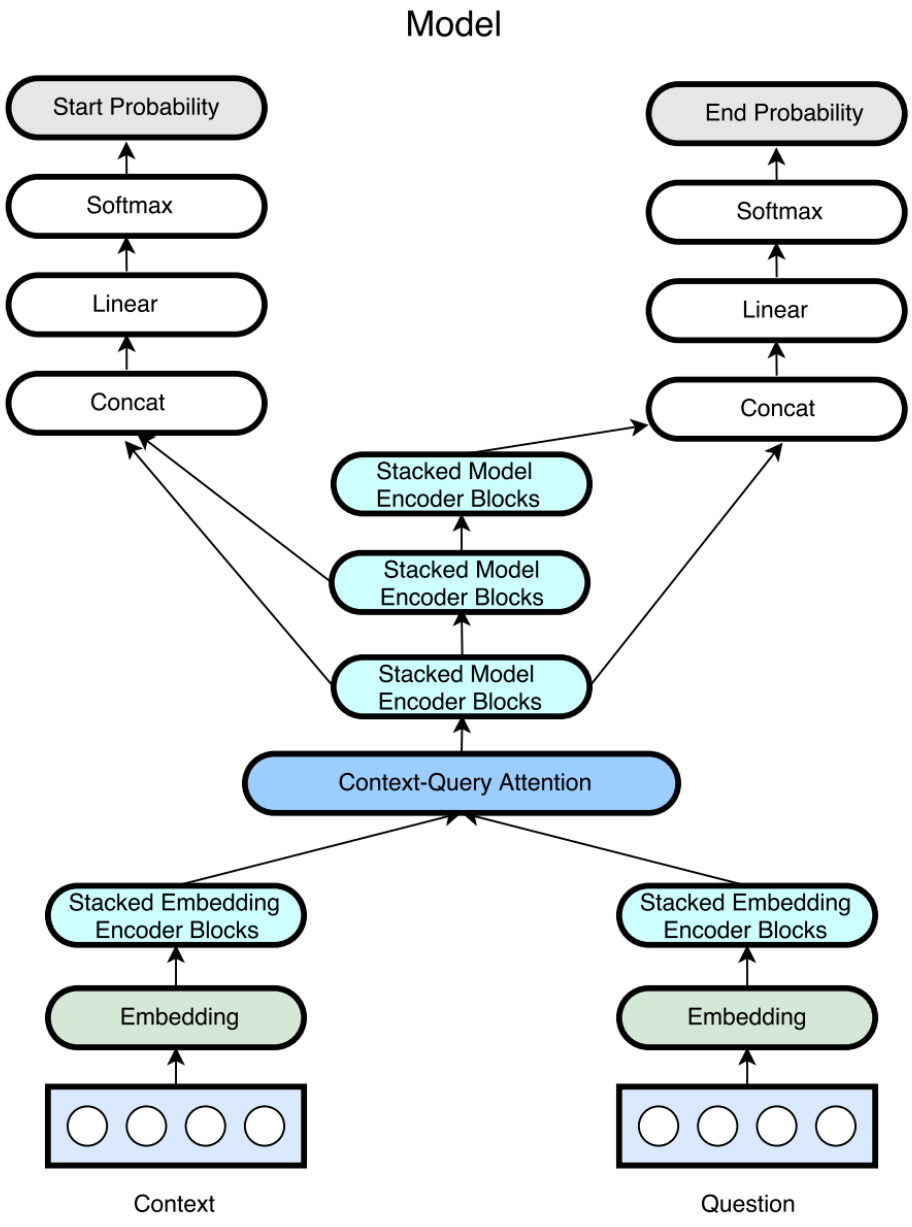
Multi-head attention mechanism



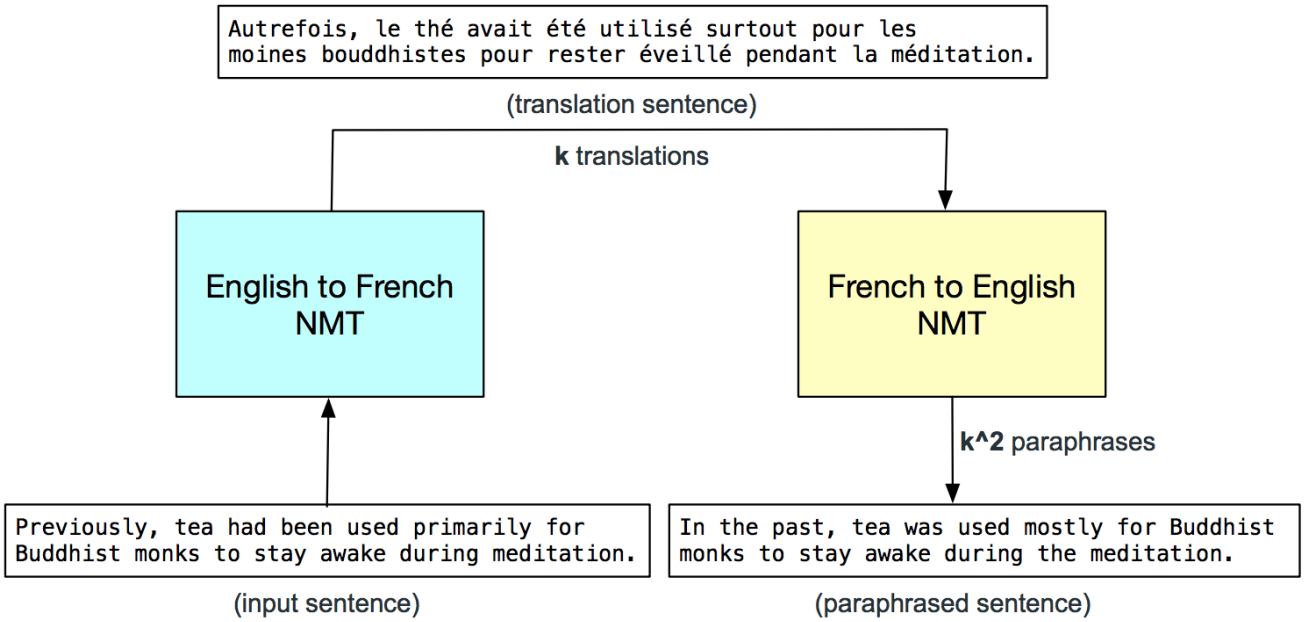
Depthwise separable convolution



Architecture



Data Augmentation by Backtranslation



- Translation model from English to French
- Translation model from French to English
- Obtain k French translations
- Obtain k English translations for each French phrase
- Total of k^2 paraphrases of the input sequence

Backtranslation & SQuAD

Document paraphrasing

- Question remains unchanged
- $k = 5$, so each sentence has 25 paraphrase choices
- New document is formed by replacing each sentence with a randomly-selected paraphrase

Answer extraction

- Sentence s contains original answer a
- Sentence s' is a paraphrase of s
- Compute character-level 2-gram scores between each word in s' and start/end words of a
- The highest scoring pair of words is a start and end of answers paraphrase

Results & Conclusions

- Best result as SQuAD v1.1 (at a time)
- 5 times speedup in training over BiDAF (*train time to get 77.0 F1 on SQuAD v1.1 Dev set*)

Effect of data augmentation

- Making the training data twice as large yields an increase in the F1 by 0.5 percent
- Adding same amout of data with another pivotal language brings another 0.2 improvement in F1
- Injecting more data beyond $\times 3$ does not benefit the model
- Increasing the ratio of the original data to (3:1:1) yields the best performance

Links


M. Seo, A. Kembhavi, A. Farhadi, H. Hajishirzi
Bidirectional Attention Flow for Machine Comprehension
https://arxiv.org/abs/1611.01603


D. Dohan, A. Wei Yu, M.T. Luonga, R. Zhao, K. Chen, M. Norouzi, Q. V. Le
QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension
https://arxiv.org/abs/1804.09541


François Chollet
Xception: Deep Learning with Depthwise Separable Convolutions
https://arxiv.org/abs/1610.02357


A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin
Attention Is All You Need
https://arxiv.org/abs/1706.03762