Reading Comprehension

Arjun Krishnan and Seyoon Ragavan

What is Reading Comprehension?

Q: When did Kendrick Lamar's first album come out?

Article | Talk | Read | View source | View history | Search Wikipedia

Kendrick Lamar

From Wikipedia, the free encyclopedia

An accepted version of this page, accepted on *28 February 2018*, was based on this revision.

Kendrick Lamar Duckworth (born June 17, 1987) is an American rapper and songwriter. Raised in Compton, California, Lamar embarked on his musical career as a teenager under the stage name **K-Dot**, releasing a mixtape that garnered local attention and led to his signing with indie record label Top Dawg Entertainment (TDE).

He began to gain recognition in 2010, after his first retail release, *Overly Dedicated*. The following year, he independently released his first studio album, *Section.80*, which included his debut single, "HiiiPoWeR". By that time, he had amassed a large online following and collaborated with several prominent artists in the hip hop industry, including The Game, Busta Rhymes, and Snoop Dogg.

Lamar's major label debut album, *good kid, m.A.A.d city*, was released in 2012 by TDE, Aftermath, and Interscope Records to critical success. It debuted at number two on the US *Billboard* 200 chart and was later certified platinum by the Recording Industry Association of America (RIAA). The record contained the top 40 singles "Swimming Pools (Drank)", "Bitch, Don't Kill My Vibe", and "Poetic Justice".

His critically acclaimed third album *To Pimp a Butterfly* (2015) comprised funk, soul, and spoken word, debuted atop the charts in the US and the UK, and won the Grammy Award for Best Rap Album at the 58th ceremony. In 2016, Lamar released *Untitled Unmastered*, a collection of unreleased demos that originated during the recording sessions for *Butterfly*. He released his fourth album *Damn* in 2017 to further acclaim; its lead single "Humble" topped the US *Billboard* Hot 100 chart.

Lamar has received a number of accolades over the course of his career, including twelve Grammy Awards. In early 2013, MTV named him the number one "Hottest MC in the Game", on their annual list.[1] *Time* named him one of the 100 most influential people in the world in 2016.[2] Aside from his solo career, Lamar is also known as a member of the West Coast hip hop supergroup Black Hippy, alongside his TDE label-mates and fellow South Los Angeles–based rappers Ab-Soul, Jay Rock, and Schoolboy Q.

Contents [hide]

Kendrick Lamar

Lamar in 2016

Born	Kendrick Lamar Duckworth June 17, 1987 (age 31) Compton, California, U.S.

"the ability to read and **understand** unstructured text and then **answer questions** about it"

Source: https://ai.googleblog.com/2019/01/natural-questions-new-corpus-and.html

What do RC Problems Look Like?

- Input: context (passage of text) and query
- Output: answer
  - Abstractive: free-form answer
  - Extractive: substring of the content

RC Necessitates Language Understanding

Alyssa got to the beach after a long trip. She's from Charlotte. She traveled from Atlanta. She's now in Miami. She went to Miami to visit some friends. But she wanted some time to herself at the beach, so she went there first. After going swimming and laying out, she went to her friend **Ellen**'s house. **Ellen** greeted **Alyssa** and they both had some lemonade to drink. **Alyssa** called her friends **Kristen** and **Rachel** to meet at **Ellen**'s house. The girls traded stories and caught up on their lives. It was a happy time for everyone. The girls went to a restaurant for dinner. The restaurant had a special on catfish. **Alyssa** enjoyed the restaurant's special. **Ellen** ordered a salad. **Kristen** had soup. **Rachel** had a steak. After eating, the ladies went back to **Ellen**'s house to have fun. They had lots of fun. They stayed the night because they were tired. **Alyssa** was happy to spend time with her friends again.

(a) **Question:** What city is Alyssa in?
Answer: Miami

(b) **Question**: What did Alyssa eat at the restaurant?
Answer: catfish

(c) **Question**: How many friends does Alyssa have in this story?
Answer: 3

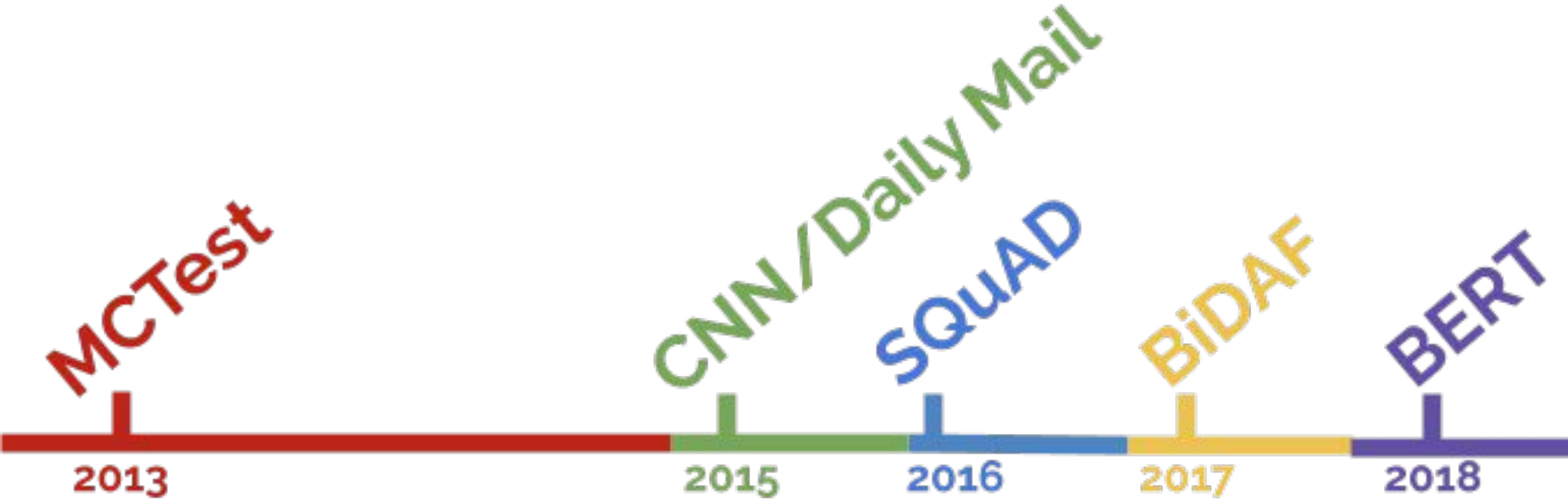
- **Coreference resolution:** understanding that "she" = Alyssa
- Inferring that **"special" = catfish** so this must be what Alyssa ate
- Identify **which entities in the text are people** and among these which are Alyssa's friends

(Richardson et. al. 2013, Chen 2018)

Outline

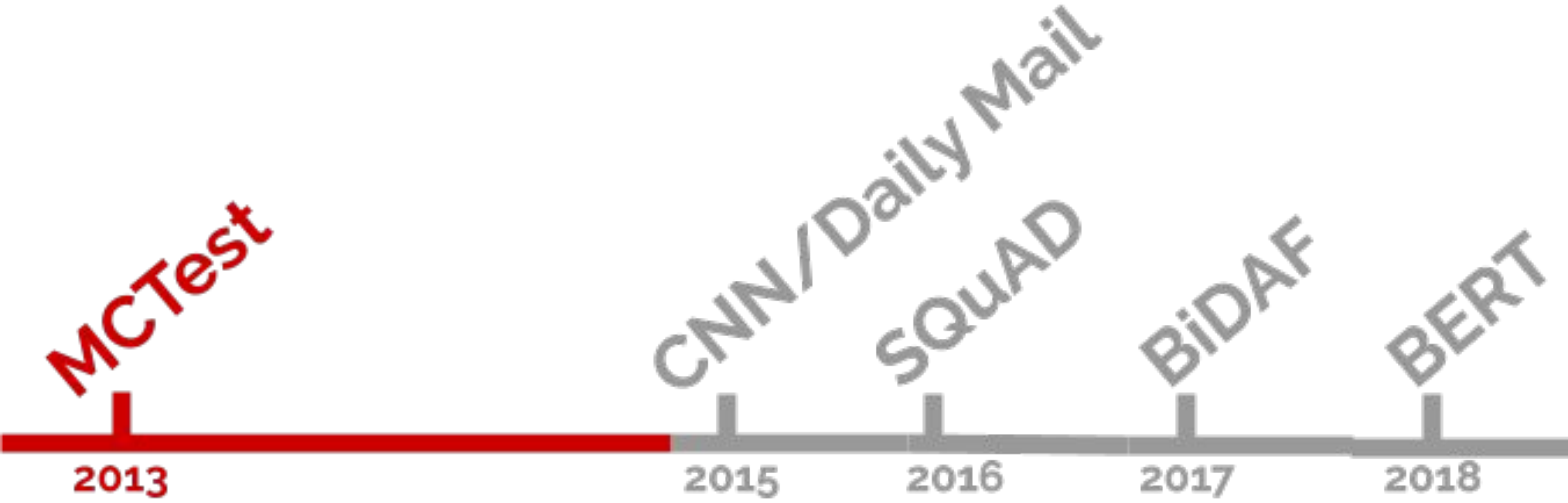
- **RC Pre-2015**
- **Paper 1:** Teaching Machines to Read and Comprehend (Hermann et al, 2015)
- **Paper 2:** Bi-directional Attention Flow for Machine Comprehension (Seo et al, 2017)
- **Current State of the Art**
- **Further Challenging Datasets**

Timeline



MCTest
CNN/Daily Mail
SQuAD
BiDAF
BERT
2013
2015
2016
2017
2018

Before 2015


MCTest
CNN/Daily Mail
SQuAD
BiDAF
BERT
2013
2015
2016
2017
2018

Before 2015: Datasets

- Challenge: generating several question-answer pairs for text corpora
- MCTest: a first attempt
  - 660 fictional stories
  - 4 multiple choice questions per story
  - Total: < 3000 questions
  - Enough for testing, not for training

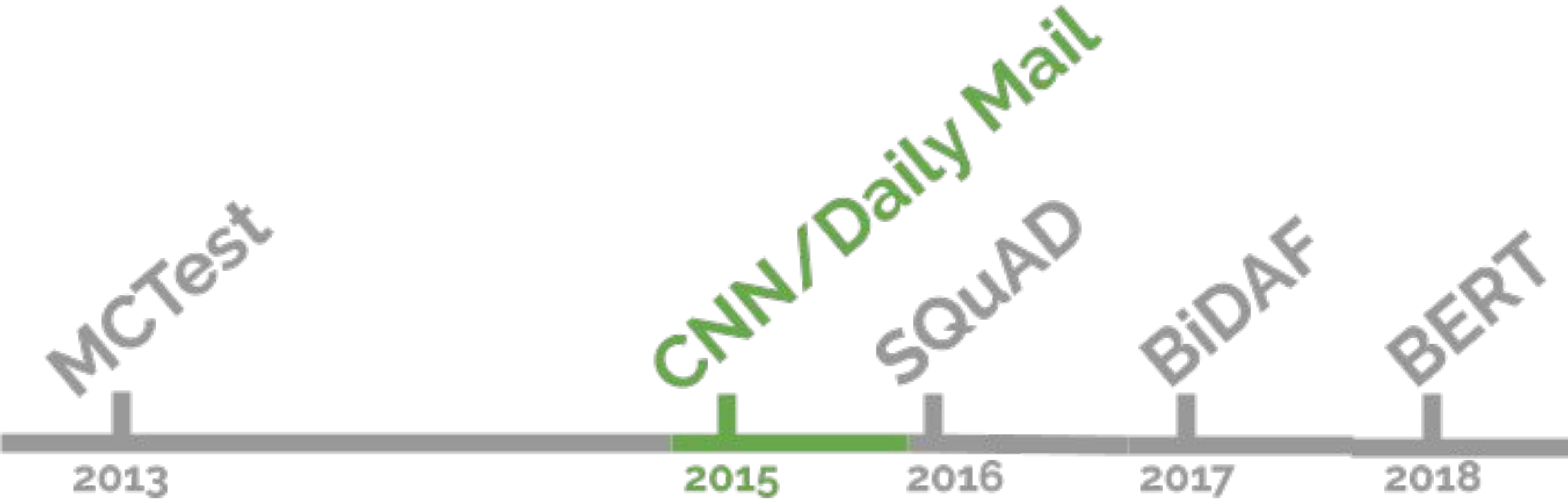
Before 2015: Models

- Rule-based approaches (no training)
- Simple ML models built on top of hand-engineered linguistic features
    - Syntactic dependencies
    - Coreference resolution
    - Word embeddings

Teaching Machines to Read and Comprehend

Hermann et. al. (2015)

CNN and Daily Mail



Datasets: CNN/Daily Mail

- Key idea: find a naturally occurring distribution of (context, query, answer) triples rather than generating them!

London (CNN) — The BBC producer allegedly struck by Jeremy Clarkson will not press charges against the "Top Gear" host, his lawyer said Friday.

Clarkson, who hosted one of the most-watched television shows in the world, was dropped by the BBC Wednesday after an internal investigation by the British broadcaster found he had subjected producer Oisin Tymon "to an unprovoked physical and verbal attack."

Story highlights

- Producer Oisin Tymon will not press charges against Jeremy Clarkson, his lawyer says
- An internal BBC investigation found Clarkson had struck Tymon in an "unprovoked attack"
- The BBC dropped Clarkson as "Top Gear" host Wednesday and police asked for the report

(Cullinane, 2015)

- Cloze style questions
- Summary sentence → query/answer pair

Context

The BBC producer allegedly struck by Jeremy Clarkson will not press charges against the "Top Gear" host, his lawyer said Friday. Clarkson, who hosted one of the most-watched television shows in the world, was dropped by the BBC Wednesday after an internal investigation by the British broadcaster found he had subjected producer Oisin Tymon "to an unprovoked physical and verbal attack." …

Query

Producer **X** will not press charges against Jeremy Clarkson, his lawyer says.

Answer

Oisin Tymon

CNN : 93,000 articles

Daily Mail; 220,000 articles

1 million data points

(Hermann et al, 2015)

But you can “cheat” on this

- “The hi-tech bra that helps you beat breast **X**”
- “Could Saccharin help beat **X**?”
- “Can fish oils help fight prostate **X**?”
- ^ All doable with an n-gram language model without absorbing any information from the context document

Solution: anonymise

Context: ent01 won't have his contract renewed as host of "**ent02**" after he apparently busted **ent03**'s lip and verbally abused him, **ent04** announced Wednesday.

ent01, who hosted one of the most-watched television shows in the world, was suspended on March 10 after what **ent04** previously described as a "fracas" with **ent03** on March 4.

Query: ent05 confirms [X] sacked

This helps... a little

- “The hi-tech bra that helps you beat breast **X**” ❌
- “Could Saccharin help beat **X**?” ✔️
- “Can fish oils help fight prostate **X**?” ❌

Previous Non-Neural Models: Symbolic Matching

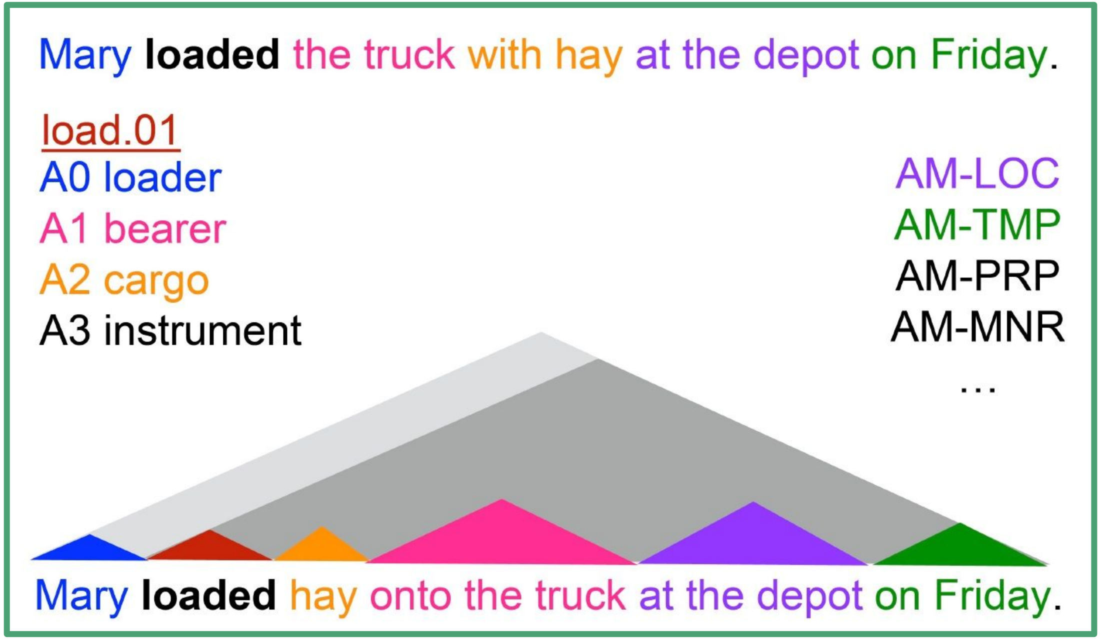
- **Frame-Semantic Models:** Statistical models that derives predicate-argument structures

  Entity-predicate triples:

  (e1, V, e2)

  e.g.

  (Alice, loves, Bob)



Source: https://devopedia.org/semantic-role-labelling

(Lascarides 2019, slide 10)

Previous Non-Neural Models: Symbolic Matching

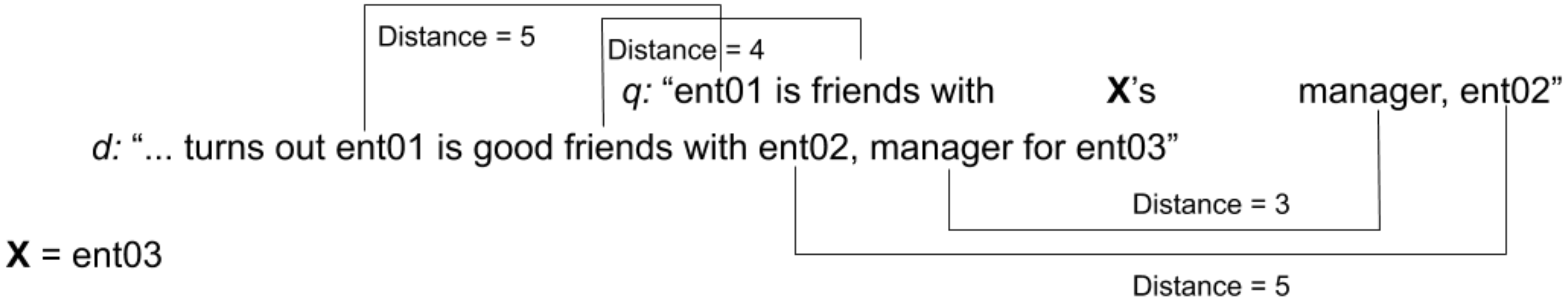
- **Frame-Semantic Models:** Statistical models that derives predicate-argument structures

	Strategy	Pattern $\in q$	Pattern $\in d$	Example (Cloze / Context)
1	Exact match	(p, V, y)	$(\boldsymbol{x}, V, y)$	X loves Suse / **Kim** loves Suse
2	be.01.V match	$(p, be.01.V, y)$	$(\boldsymbol{x}, be.01.V, y)$	X is president / **Mike** is president
3	Correct frame	(p, V, y)	$(\boldsymbol{x}, V, z)$	X won Oscar / **Tom** won Academy Award
4	Permuted frame	(p, V, y)	$(y, V, \boldsymbol{x})$	X met Suse / Suse met **Tom**
5	Matching entity	(p, V, y)	$(\boldsymbol{x}, Z, y)$	X likes candy / **Tom** loves candy
6	Back-off strategy	*Pick the most frequent entity from the context that doesn't appear in the query*		

(Hermann et al, 2015)

Previous Non-Neural Models: Symbolic Matching

- **Word Distance Benchmark:**
  - Align the placeholder with every possible entity in the document and then sum up the distance of every word in the question to their nearest aligned word in the document
  - "Aligned word" = same word or coreferent



Neural Network Models

High level overview:

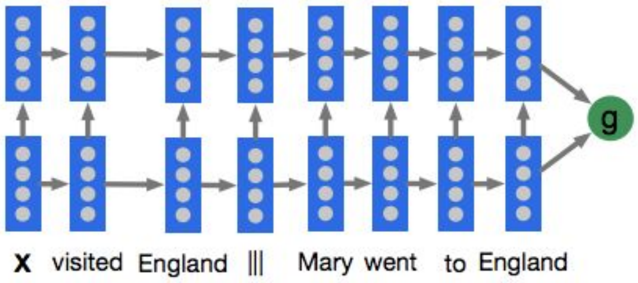
- NN (coming up): compute embedding g(d, q) for a given document-query pair (d, q)
  - Deep LSTM Reader
  - Attentive Reader
  - Impatient Reader
- Trainable matrix W of vectors for each word
- Softmax over output word types to get probabilities:

$$p(a|d,q) \propto \exp\left(W(a)g(d,q)\right)$$

(Hermann et al, 2015)

Deep LSTM reader

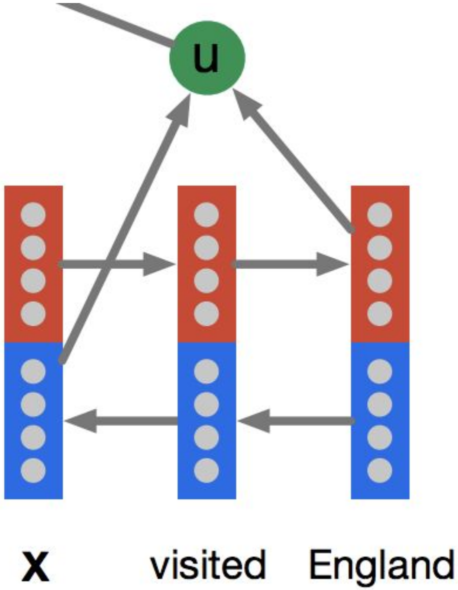
- Longer than usual input to LSTM (700-800 tokens):
  - Document word by word
  - Delimiter
  - Query word by word
  - Or query then document



(Hermann et al, 2015)

Attentive Reader

Step 1: encode the query by passing it through forward and backward LSTMs and concatenating the outputs

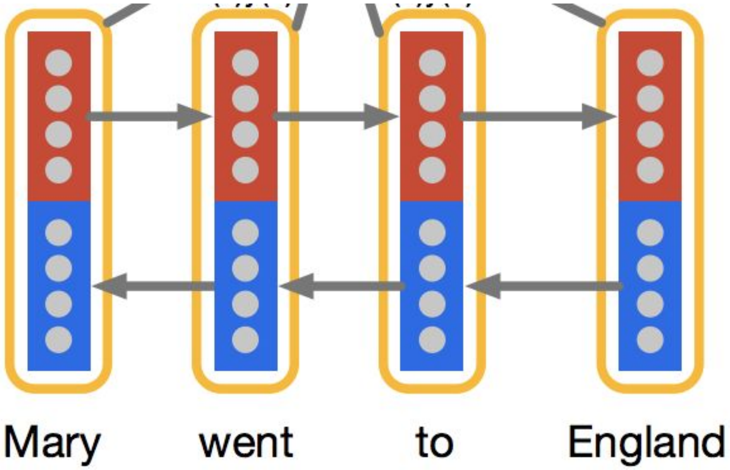


$$u = \overrightarrow{y_q}(|q|) \,||\, \overleftarrow{y_q}(1)$$

(Hermann et al, 2015)

Attentive Reader

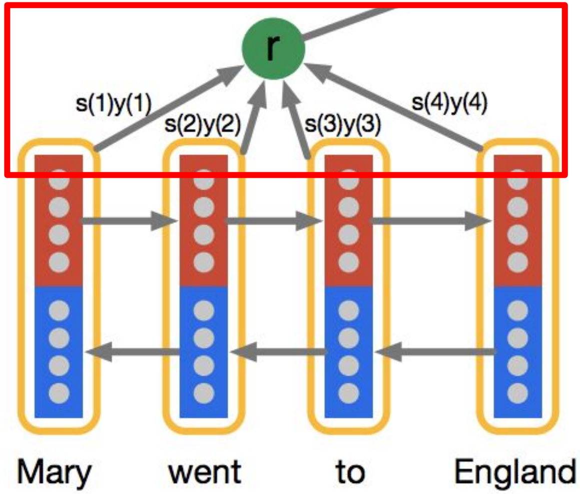
Step 2: same drill with the document, but this time obtaining an embedding for every token



(Hermann et al, 2015)

Attentive Reader

Step 3: use attention with the query embedding and document token embeddings as input to determine which tokens in the document to attend to



$$m(t) = \tanh(W_{ym} y_d(t) + W_{um} u)$$

$$s(t) \propto \exp(W_{m,s}^T m(t))$$

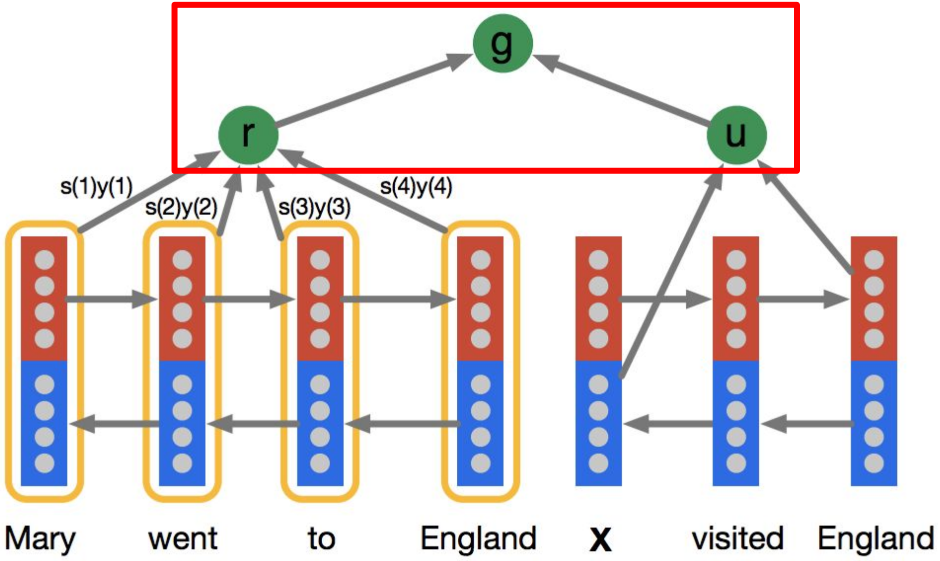
$$r = y_d s$$

$$= \sum_t s(t) y_d(t)$$

(Hermann et al, 2015)

Attentive Reader

Step 4: one layer to combine the final document and query embeddings

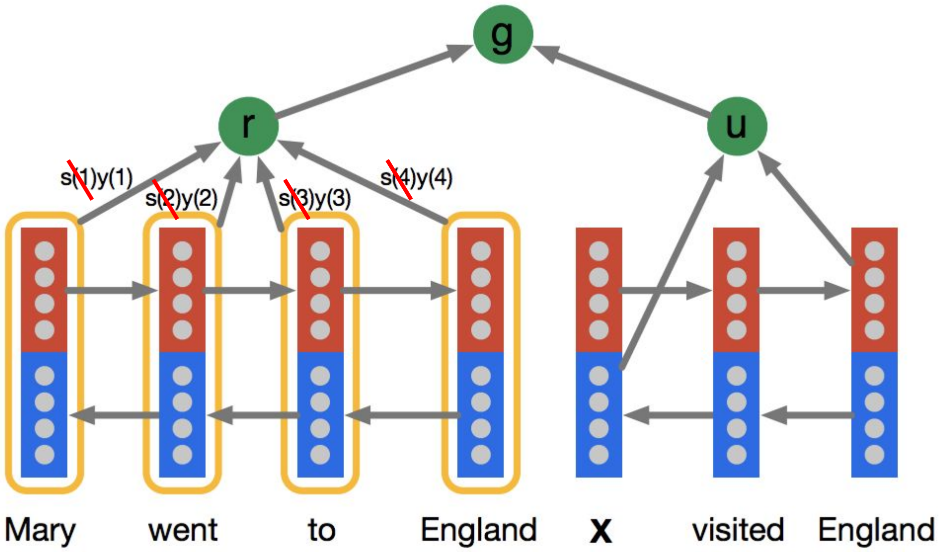


$$g^{\mathrm{AR}}(d,q) = \tanh\left(W_{rg}r + W_{ug}u\right)$$

(Hermann et al, 2015)

Uniform Reader (baseline)

- Same as attentive reader but without the attention part; instead it averages uniformly over the document token embeddings

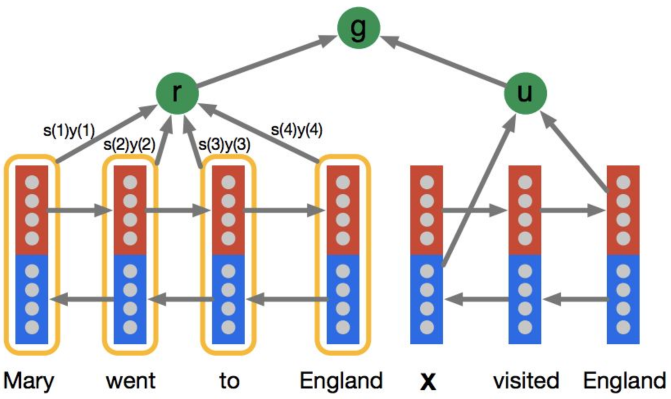


$$r = \frac{1}{|d|} \sum_t y_d(t)$$

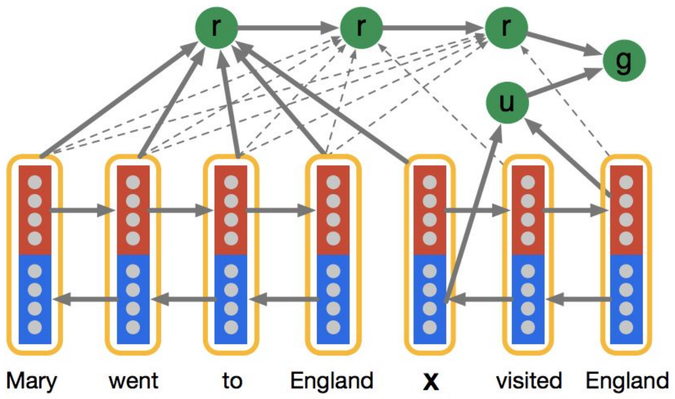
(Hermann et al, 2015)

Impatient Reader

Same as attentive reader but rereads from the document as each token is read, so attention is repeatedly applied:



(a) Attentive Reader.



(b) Impatient Reader.

(Hermann et al, 2015)

Experiments - predictions?

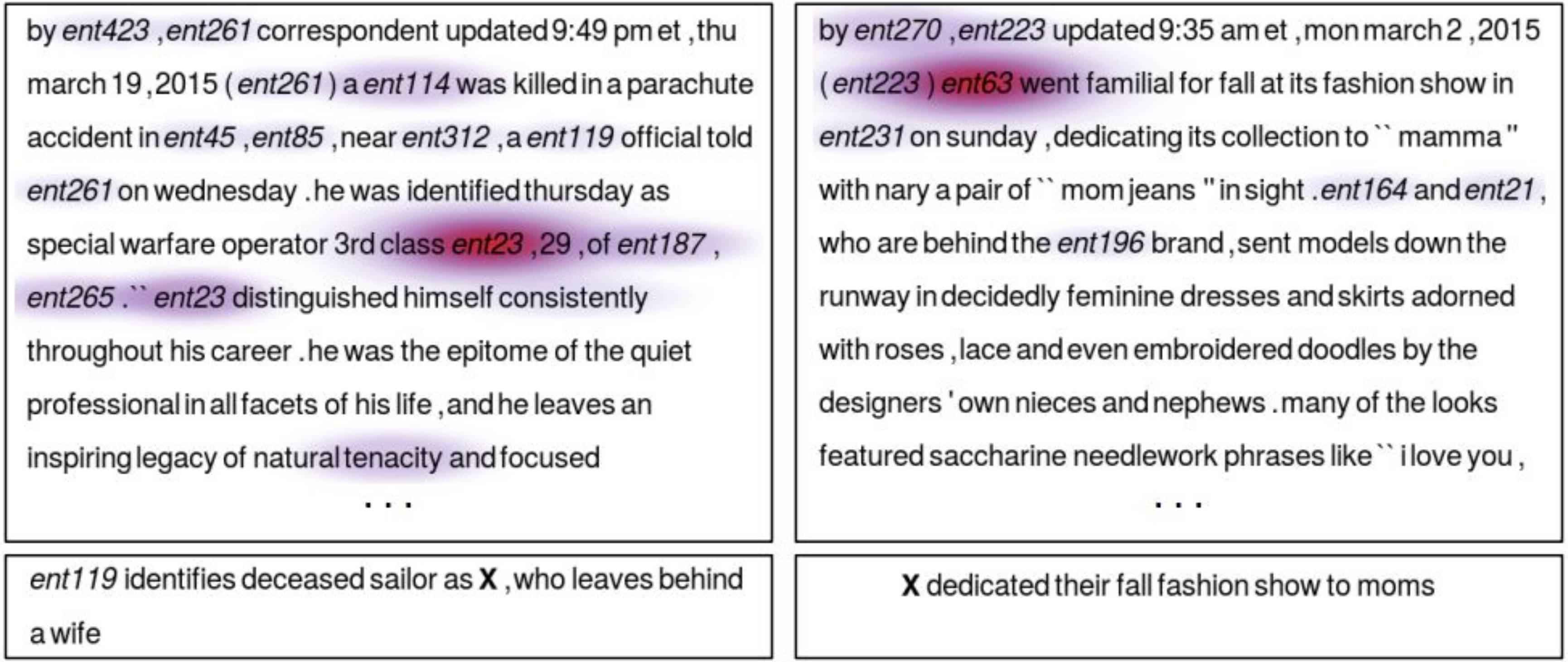
- Traditional vs. neural models?
  - Should the entity anonymisation complicate this?
- LSTM vs. attention-based approaches?
- Attentive vs. impatient vs. uniform reader?
- Word distance vs. frame-semantic?

CNN/Daily Mail: Results

	CNN		Daily Mail	
	valid	test	valid	test
Maximum frequency	30.5	33.2	25.6	25.5
Exclusive frequency	36.6	39.3	32.7	32.8
Frame-semantic model	36.3	40.2	35.5	35.5
Word distance model	50.5	50.9	56.4	55.5
Deep LSTM Reader	55.0	57.0	63.3	62.2
Uniform Reader	39.0	39.4	34.6	34.4
Attentive Reader	61.6	63.0	**70.5**	**69.0**
Impatient Reader	**61.8**	**63.8**	69.0	68.0

(Hermann et al, 2015)

Attention heatmaps for attention reader



(Hermann et al, 2015)

Main Takeaways

- Revolutionary dataset in its time
- Small heuristic allowed authors to capitalize on naturally existing dataset
- Attention helps significantly
- However, poor baseline models do better than expected (Word distance benchmark)

Bi-directional Attention Flow For Machine Comprehension

Seo et. al. (2017)

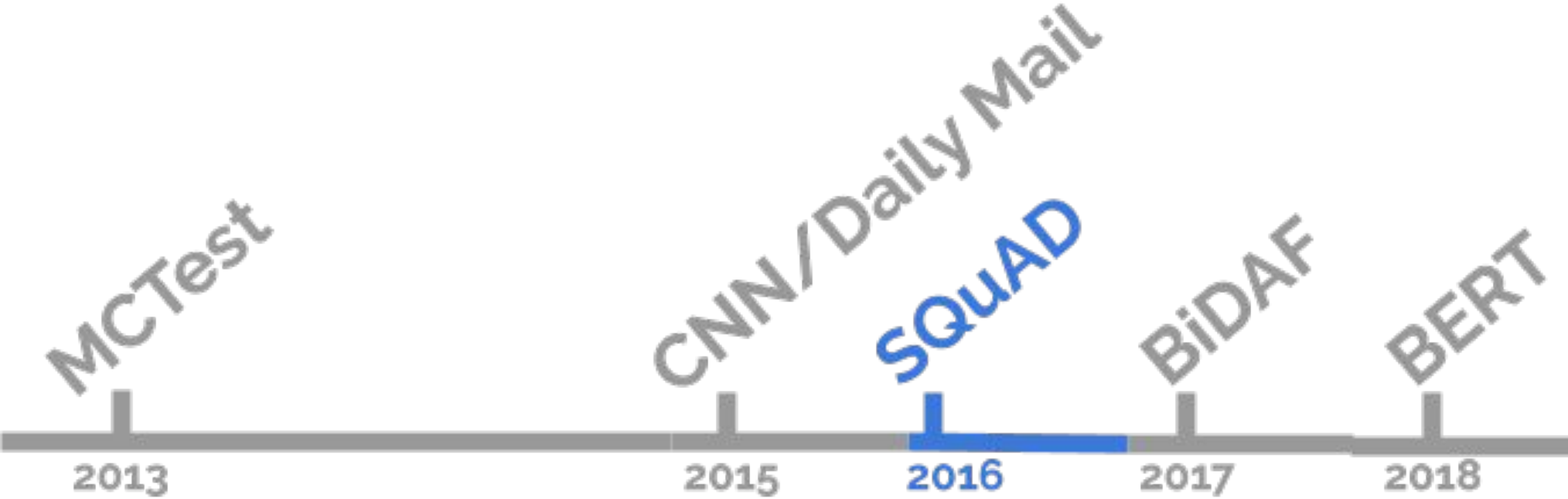
Discussion

Q: CNN/Daily Mail was the first large-scale reading comprehension dataset available in this field. What is good about this dataset and what is its main limitation?

Motivation: Datasets

- High quality human-written databases not very large (on the order 10^3 in size)
- Cloze-form questions better, but not very natural
  - Semi-synthetic (As in Cloze)
  - Not explicit question answering
- Heuristically created ➔ noisy

SQuAD: Timeline



SQuAD: Basics

- Questions posed by crowdworkers on a set of Wikipedia articles
- 100,000 query-context-answer triples
- Extractive question answering: all answers a *span* of text

Computational complexity theory is a branch of the theory of computation in theoretical computer science that focuses on classifying computational problems according to their inherent difficulty, and relating those classes to each other. A computational problem is understood to be a task that is in principle amenable to being solved by a computer, which is equivalent to stating that the problem may be solved by mechanical application of mathematical steps, such as an algorithm.

By what main attribute are computational problems classified utilizing computational complexity theory?

Ground Truth Answers: inherent difficulty | their inherent difficulty | inherent difficulty

Prediction: inherent difficulty

3 gold answers are collected for each answer

100,000 data points

Source: https://rajpurkar.github.io/SQuAD-explorer/explore/v2.0/dev/Computational_complexity_theory.html

Source: https://rajpurkar.github.io/SQuAD-explorer/explore/v2.0/dev/Computational_complexity_theory.html

SQuAD: Example

The Amazon rainforest (Portuguese: Floresta Amazônica or Amazônia; Spanish: Selva Amazónica, Amazonía or usually Amazonia; French: Forêt amazonienne; Dutch: Amazoneregenwoud), also known in English as Amazonia or the Amazon Jungle, is a moist broadleaf forest that covers most of the Amazon basin of South America. This basin encompasses 7,000,000 square kilometres (2,700,000 sq mi), of which 5,500,000 square kilometres (2,100,000 sq mi) are covered by the rainforest. This region includes territory belonging to nine nations. The majority of the forest is contained within Brazil, with 60% of the rainforest, followed by Peru with 13%, Colombia with 10%, and with minor amounts in Venezuela, Ecuador, Bolivia, Guyana, Suriname and French Guiana. States or departments in four nations contain "Amazonas" in their names. The Amazon represents over half of the planet's remaining rainforests, and comprises the largest and most biodiverse tract of tropical rainforest in the world, with an estimated 390 billion individual trees divided into 16,000 species.

How many square kilometers of rainforest is covered in the basin?

Source: https://rajpurkar.github.io/SQuAD-explorer/explore/v2.0/dev/Computational_complexity_theory.html

SQuAD: Example

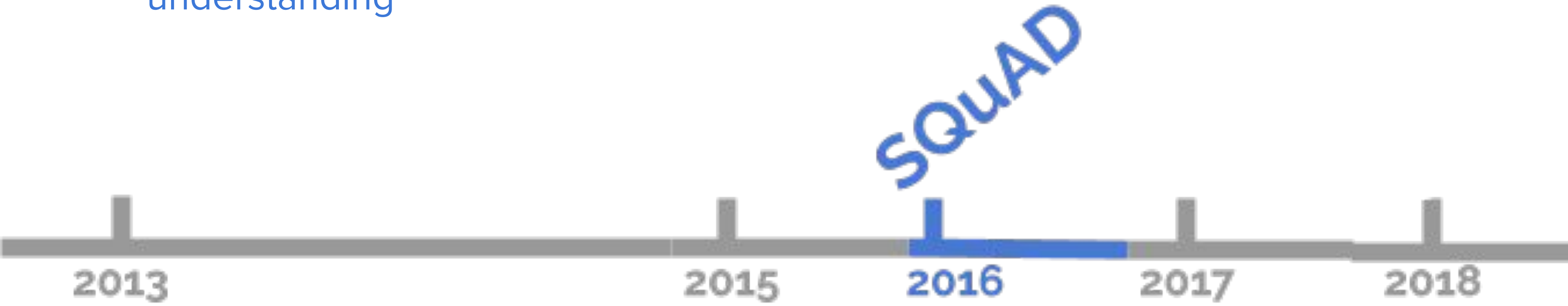
The Amazon rainforest (Portuguese: Floresta Amazônica or Amazônia; Spanish: Selva Amazónica, Amazonía or usually Amazonia; French: Forêt amazonienne; Dutch: Amazoneregenwoud), also known in English as Amazonia or the Amazon Jungle, is a moist broadleaf forest that covers most of the Amazon basin of South America. This basin encompasses 7,000,000 square kilometres (2,700,000 sq mi), of which 5,500,000 square kilometres (2,100,000 sq mi) are covered by the rainforest. This region includes territory belonging to nine nations. The majority of the forest is contained within Brazil, with 60% of the rainforest, followed by Peru with 13%, Colombia with 10%, and with minor amounts in Venezuela, Ecuador, Bolivia, Guyana, Suriname and French Guiana. States or departments in four nations contain "Amazonas" in their names. The Amazon represents over half of the planet's remaining rainforests, and comprises the largest and most biodiverse tract of tropical rainforest in the world, with an estimated 390 billion individual trees divided into 16,000 species.

How many square kilometers of rainforest is covered in the basin?

Ground Truth Answers: 5,500,000 square kilometres (2,100,000 sq mi) are covered by the rainforest. 5,500,000 5,500,000

Why is SQuAD better?

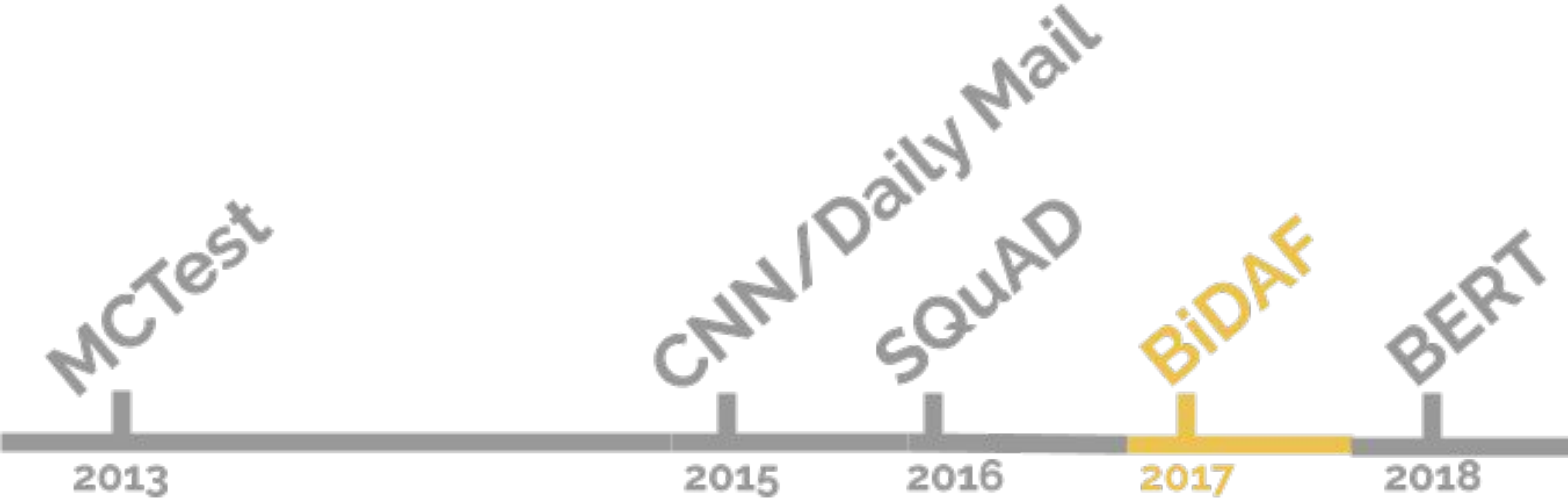
- Human-written, human curated ➔ less noisy than CNN/DM
- Not Cloze-form
- Step towards better language understanding



BiDAF: Motivations

- Incorporating attention better into Question Answering
- What are the problems with prior models?
    - Unidirectional attention
    - Summarising context into fixed-size vectors
- How does current paper seek to address these?
    - Bidirectional attention: query-to-context and context-to-query
    - Includes character-level, word-level, and contextual embeddings
    - Attended vectors are passed along together with original embeddings

BiDAF: Timeline



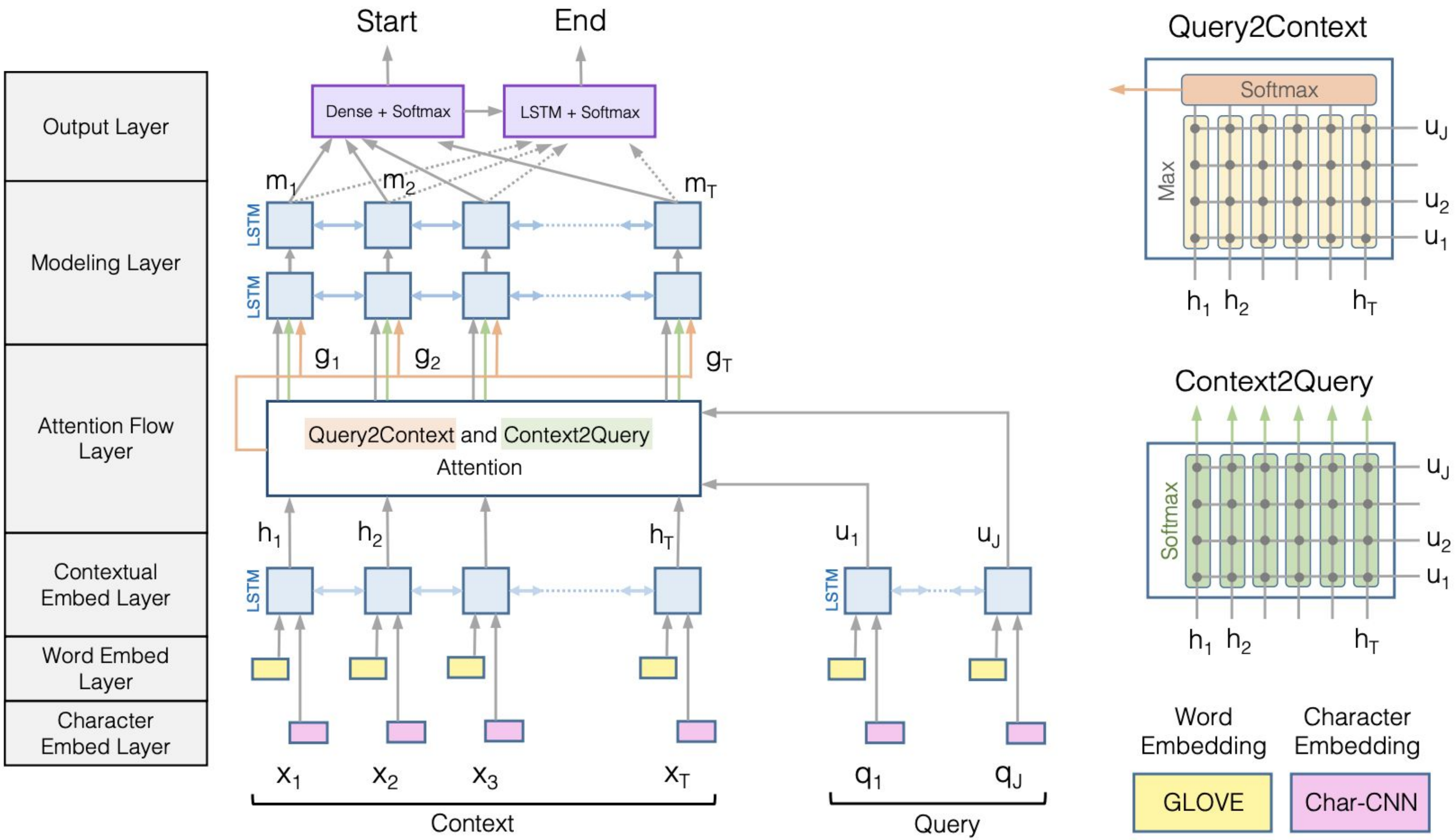


Figure 1: BiDirectional Attention Flow Model *(best viewed in color)*

(Seo et al, 2017)

Basic Components of the Model

- **Character Embedding Layer**
- **Word Embedding Layer**
- **Contextual Embedding Layer**
- **Attention Flow Layer**
- **Modeling Layer**
- **Output Layer**

Basic Components of the Model

- **Character Embedding Layer ➜** Embeds each word using character-level CNNs.
- Word Embedding Layer
- Contextual Embedding Layer
- Attention Flow Layer
- Modeling Layer
- Output Layer

Basic Components of the Model

- Character Embedding Layer
- **Word Embedding Layer** → GloVe
- Contextual Embedding Layer
- Attention Flow Layer
- Modeling Layer
- Output Layer

Basic Components of the Model

- Character Embedding Layer
- Word Embedding Layer
- **Contextual Embedding Layer** → Character and word embeddings passed through bi-LSTM to obtain contextual embeddings for query and context.
- Attention Flow Layer
- Modeling Layer
- Output Layer

Basic Components of the Model

- Character Embedding Layer
- Word Embedding Layer
- Contextual Embedding Layer
- **Attention Flow Layer** ➜ Produces a set of query-aware feature vectors for each word in the context (C2Q) and a context-aware vector for the query (Q2C).
- Modeling Layer
- Output Layer

Basic Components of the Model

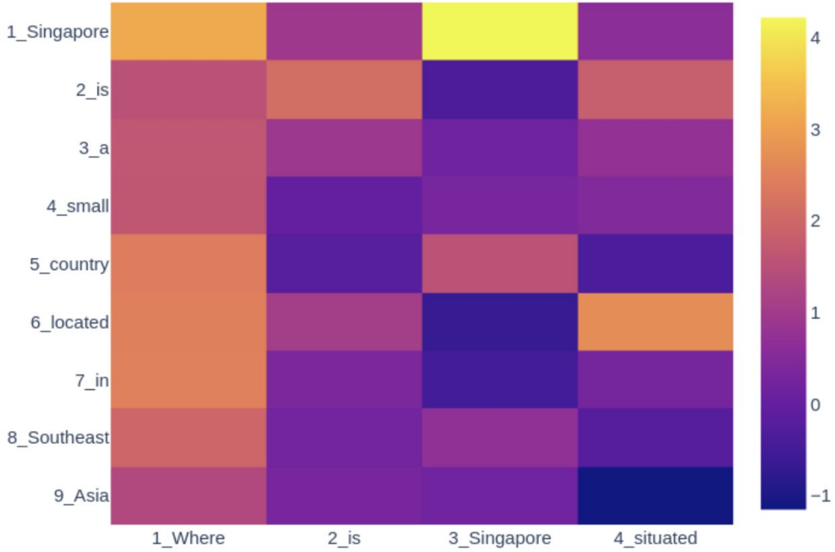
- Character Embedding Layer
- Word Embedding Layer
- Contextual Embedding Layer
- Attention Flow Layer
- **Modeling Layer ➔** Contextual embeddings and attended vectors passed through two-layer bi-LSTM for even more refined representation.
- Output Layer

Basic Components of the Model

- Character Embedding Layer
- Word Embedding Layer
- Contextual Embedding Layer
- Attention Flow Layer
- Modeling Layer
- **Output Layer** ➔ Linear layer then softmax to obtain a start probability distribution and an end probability distribution over the indices.

A Closer Look: Attention

- Compute a similarity matrix S from context embeddings H and query embeddings U



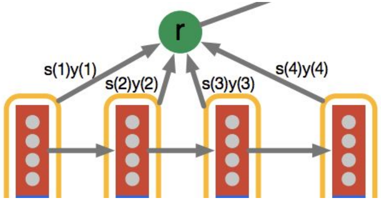
$$S_{tj} = \alpha(H_{:t}, U_{:j}) \in \mathbb{R}$$

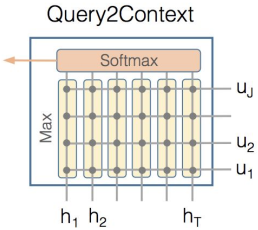
$$\alpha(h, u) = w_{(S)}^T [h; u; h \circ u]$$

Source: https://towardsdatascience.com/the-definitive-guide-to-bidaf-part-3-attention-92352bbdcb07

A Closer Look: Attention

- Q2C: query ➜ which tokens in the context to attend to

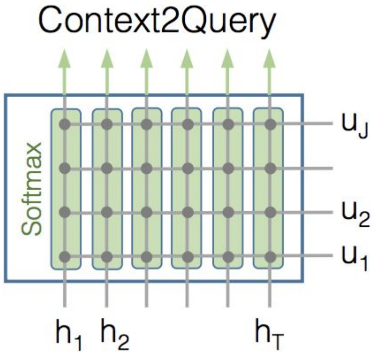




$$\tilde{h} = \sum_t b_t H_{:t}$$

$$b_t \propto \exp(\max_j S_{tj})$$

- **C2Q**: each context token ➜ which tokens in the query it should attend to

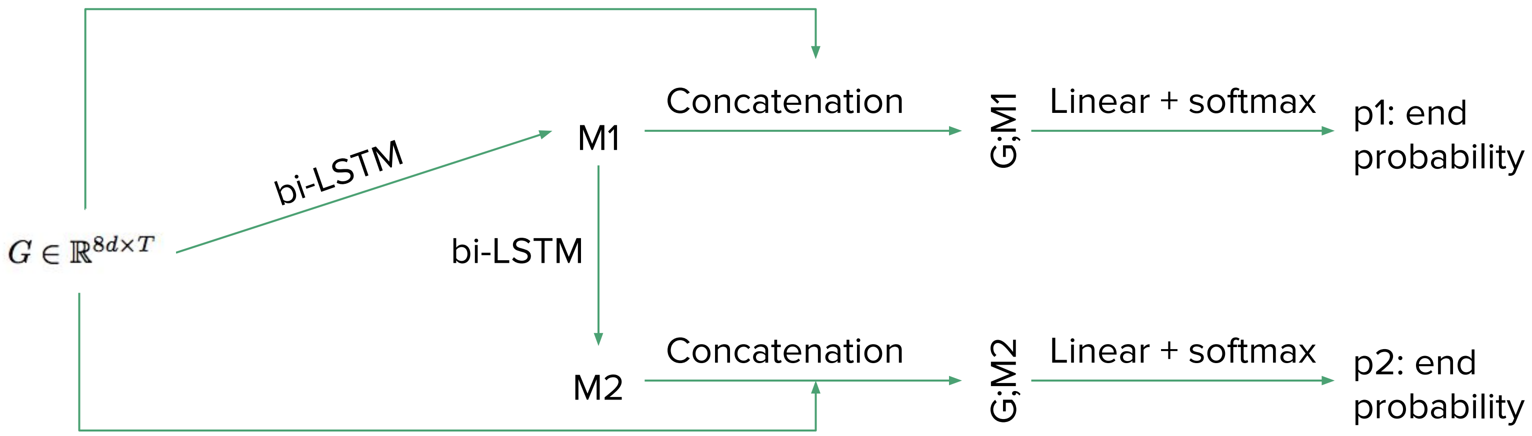


$$\widetilde{U}_{:t} = \sum_{j=1}^{J} a_{tj} U_{:j}$$

$$a_{tj} \propto \exp(S_{tj})$$

(Hermann et al 2015, Seo et al, 2017)

A Closer Look: Output



Performance Metrics

- Training: log likelihood of correct start/end indices $L(\theta) = -\frac{1}{N}\sum_{i}^{N}\log(\mathbf{p}^1_{y_i^1}) + \log(\mathbf{p}^2_{y_i^2})$
- Testing: choose start-end index pair (i, j) with i < j maximising p1(i) * p2(j)
    - Remove all articles (a, an, the)
    - Exact Match (EM): choosing exactly the same start and end index as some gold answer
    - F1: treat predicted and gold answers as bags of tokens, then take harmonic mean of precision and recall

$$F_1 = \left(\frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}}\right) = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

$$\mathrm{precision} = \frac{\#\text{ of correctly predicted tokens}}{\#\text{ of predicted tokens}} \qquad \mathrm{recall} = \frac{\#\text{ of correctly predicted tokens}}{\#\text{ of gold tokens}}$$

(Seo et al, 2017, https://en.wikipedia.org/wiki/F1_score)

Results on SQuAD: vs. other methods (test set)

	Single Model		Ensemble	
	EM	F1	EM	F1
Logistic Regression Baseline[a]	40.4	51.0	-	-
Dynamic Chunk Reader[b]	62.5	71.0	-	-
Fine-Grained Gating[c]	62.5	73.3	-	-
Match-LSTM[d]	64.7	73.7	67.9	77.0
Multi-Perspective Matching[e]	65.5	75.1	68.2	77.2
Dynamic Coattention Networks[f]	66.2	75.9	71.6	80.4
R-Net[g]	**68.4**	**77.5**	72.1	79.7
BiDAF (Ours)	68.0	77.3	**73.3**	**81.1**

Ensemble: train 12 models, choose start and end indices with the highest sum of confidence scores

(Seo et al, 2017)

Results on SQuAD: vs. ablations (dev set)

Character-level embedding: effective in handling out-of-vocab or rare words

Word-level embedding: better at capturing the overall semantics of words

	EM	F1
No char embedding	65.0	75.4
No word embedding	55.5	66.8
No C2Q attention	57.2	67.7
No Q2C attention	63.6	73.7
Dynamic attention	63.5	73.6
BIDAF (single)	67.7	77.3
BIDAF (ensemble)	72.6	80.7

(Seo et al, 2017)

Results: SQuAD vs. ablations

C2Q ablation: attended query vector for each context word is a uniform average over the word vectors

Q2C ablation: remove any terms incorporating attended context vectors for each query word

	EM	F1
No char embedding	65.0	75.4
No word embedding	55.5	66.8
No C2Q attention	57.2	67.7
No Q2C attention	63.6	73.7
Dynamic attention	63.5	73.6
BIDAF (single)	67.7	77.3
BIDAF (ensemble)	72.6	80.7

(Seo et al, 2017)

Results on SQuAD: vs. ablations (dev set)

Dynamic attention: Update attention throughout the modelling layer

Intuition: Separating out the attention layer gives a richer set of features to feed into the modelling layer

	EM	F1
No char embedding	65.0	75.4
No word embedding	55.5	66.8
No C2Q attention	57.2	67.7
No Q2C attention	63.6	73.7
Dynamic attention	63.5	73.6
BIDAF (single)	67.7	77.3
BIDAF (ensemble)	72.6	80.7

(Seo et al, 2017)

Results on CNN/Daily Mail

- Only predict start index
- Mask out non-entity words in classification layer
- For loss function: sum probability over all instances of the correct entity

	CNN		DailyMail	
	val	test	val	test
Attentive Reader (Hermann et al., 2015)	61.6	63.0	70.5	69.0
MemNN (Hill et al., 2016)	63.4	6.8	-	-
AS Reader (Kadlec et al., 2016)	68.6	69.5	75.0	73.9
DER Network (Kobayashi et al., 2016)	71.3	72.9	-	-
Iterative Attention (Sordoni et al., 2016)	72.6	73.3	-	-
EpiReader (Trischler et al., 2016)	73.4	74.0	-	-
Stanford AR (Chen et al., 2016)	73.8	73.6	77.6	76.6
GAReader (Dhingra et al., 2016)	73.0	73.8	76.7	75.7
AoA Reader (Cui et al., 2016)	73.1	74.4	-	-
ReasoNet (Shen et al., 2016)	72.9	74.7	77.6	76.6
BIDAF (Ours)	**76.3**	**76.9**	**80.3**	**79.6**
MemNN* (Hill et al., 2016)	66.2	69.4	-	-
ASReader* (Kadlec et al., 2016)	73.9	75.4	78.7	77.7
Iterative Attention* (Sordoni et al., 2016)	74.5	75.7	-	-
GA Reader* (Dhingra et al., 2016)	76.4	77.4	79.1	78.1
Stanford AR* (Chen et al., 2016)	77.2	77.6	80.2	79.2

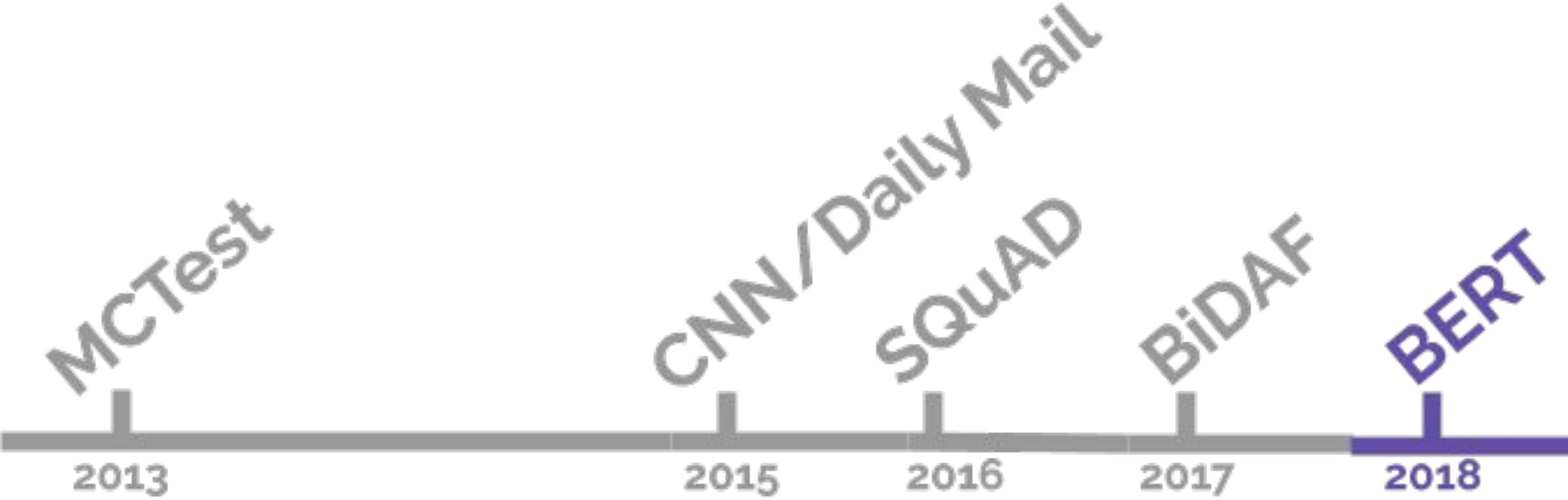
(Seo et al, 2017)

BiDAF: Takeaways

- Embeddings on multiple levels of granularity
- SQuAD: Facilitated much more natural Q&A
- Bi-directional attention was new: **C2Q** + Q2C
- Query aware context representation without early summarization
- SOTA performance at the time

Current SOTA: Pre-Trained Models

BERT: Timeline

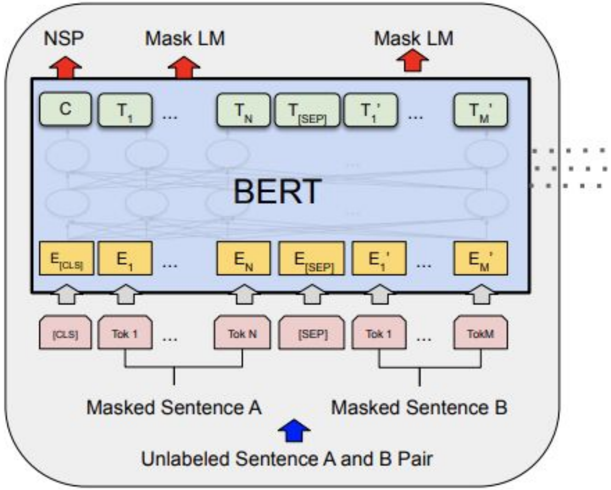


SQuAD: Leaderboard

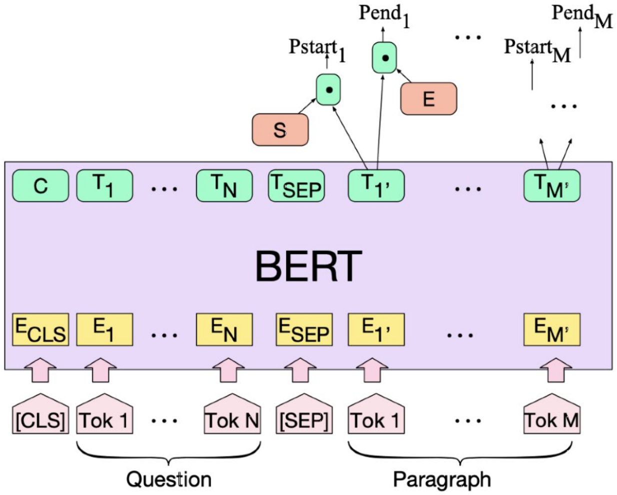
Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 May 21, 2019	XLNet (single model) *Google Brain & CMU*	**89.898**	**95.080**
2 Dec 11, 2019	XLNET-123++ (single model) *MST/EOI* http://tia.today	89.856	94.903
2 Aug 11, 2019	XLNET-123 (single model) *MST/EOI*	89.646	94.930
3 Sep 25, 2019	BERTSP (single model) *NEUKG* http://www.techkg.cn/	88.912	94.584
3 Jul 21, 2019	SpanBERT (single model) *FAIR & UW*	88.839	94.635
4 Jul 03, 2019	BERT+WWM+MT (single model) *Xiaoi Research*	88.650	94.393
5 Jul 21, 2019	Tuned BERT-1seq Large Cased (single model) *FAIR & UW*	87.465	93.294
6 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	87.433	93.160
7 May 14, 2019	ATB (single model) *Anonymous*	86.940	92.641
8 Jul 21, 2019	Tuned BERT Large Cased (single model) *FAIR & UW*	86.521	92.617
8 Jul 04, 2019	BERT+MT (single model) *Xiaoi Research*	86.458	92.645
9 Feb 14, 2019	KT-NET (single model) *Baidu NLP*	85.944	92.425
9 Sep 26, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.954	91.677

Source: https://rajpurkar.github.io/SQuAD-explorer/

BERT for Reading Comprehension - Recap



Pretraining



Finetuning

$$Pstart_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}$$

$$Pend_i = \frac{e^{E \cdot T_i}}{\sum_j e^{E \cdot T_j}}$$

(Devlin et. al. 2018, Chen 2019)

Discussion

Q2: Comparing the BiDAF model proposed in (Seo et al, 2017) with the BERT model applied to question answering that we have already learned in the class, can you identify the key differences between the two models?

Discussion

Q2: Comparing the BiDAF model proposed in (Seo et al, 2017) with the BERT model applied to question answering that we have already learned in the class, can you identify the key differences between the two models?

- Self-attention in BERT: C2Q and Q2C attention but also C2C and Q2Q
- BERT is pre-trained
- Multistage dynamic attention

Challenging Datasets

Discussion

Q:Can you think of any limitations of SQuAD (which was constructed one year after the CNN/DM work and consisting of 100,000+ questions annotated by crowd-workers)?

Limitations of SQuAD

- Only span-based answers (no yes/no, counting, implicit why)
- Questions were constructed looking at passages
- Not genuine information needs
- Generally greater lexical and syntactic matching between question and answer span
- Barely any multi-fact/sentence inference beyond coreference

(Chen, 2019)

DROP: Discrete **R**easoning **O**ver **P**aragraphs

"Force a structured analysis of the content of the paragraph that is detailed enough to permit reasoning."

Alyssa got to the beach after a long trip. She's from Charlotte. She traveled from Atlanta. She's now in Miami. She went to Miami to visit some friends. But she wanted some time to herself at the beach, so she went there first. After going swimming and laying out, she went to her friend **Ellen**'s house. **Ellen** greeted **Alyssa** and they both had some lemonade to drink. **Alyssa** called her friends **Kristen** and **Rachel** to meet at **Ellen**'s house. The girls traded stories and caught up on their lives. It was a happy time for everyone. The girls went to a restaurant for dinner. The restaurant had a special on catfish. **Alyssa** enjoyed the restaurant's special. **Ellen** ordered a salad. **Kristen** had soup. **Rachel** had a steak. After eating, the ladies went back to **Ellen**'s house to have fun. They had lots of fun. They stayed the night because they were tired. **Alyssa** was happy to spend time with her friends again.

(a) **Question:** What city is Alyssa in?
Answer: Miami

(b) **Question**: What did Alyssa eat at the restaurant?
Answer: catfish

(c) **Question**: How many friends does Alyssa have in this story?
Answer: 3

(Richardson et al, 2013, Dua et al, 2019)

DROP ctd.

Reasoning	**Passage** (some parts shortened)	**Question**	**Answer**	**BiDAF**
Subtraction (28.8%)	That year, his **Untitled (1981)**, a painting of a haloed, black-headed man with a bright red skeletal body, depicted amid the artists signature scrawls, was **sold by Robert Lehrman for $16.3 million, well above its $12 million high estimate**.	How many more dollars was the Untitled (1981) painting sold for than the 12 million dollar estimation?	4300000	$16.3 million
Comparison (18.2%)	In **1517, the seventeen-year-old King sailed to Castile**. There, his Flemish court .... **In May 1518, Charles traveled to Barcelona in Aragon**.	Where did Charles travel to first, Castile or Barcelona?	Castile	Aragon
Selection (19.4%)	In 1970, to commemorate the 100th anniversary of the founding of Baldwin City, **Baker University professor and playwright Don Mueller and Phyllis E. Braun, Business Manager, produced a musical play entitled The Ballad Of Black Jack** to tell the story of the events that led up to the battle.	Who was the University professor that helped produce The Ballad Of Black Jack, Ivan Boyd or Don Mueller?	Don Mueller	Baker
Addition (11.7%)	Before the UNPROFOR fully deployed, the HV clashed with an armed force of the RSK in the village of Nos Kalik, located in a pink zone near Šibenik, and captured the village at 4:45 p.m. on **2 March 1992**. The JNA formed a battlegroup to counterattack the **next day**.	What date did the JNA form a battlegroup to counterattack after the village of Nos Kalik was captured?	3 March 1992	2 March 1992

(Dua et al, 2019)

CoQA

The Virginia governor's race, billed as the marquee battle of an otherwise anticlimactic 2013 election cycle, is shaping up to be a foregone conclusion. Democrat Terry McAuliffe, the longtime political fixer and moneyman, hasn't trailed in a poll since May. Barring a political miracle, Republican Ken Cuccinelli will be delivering a concession speech on Tuesday evening in Richmond. In recent ...

Q_1: What are the candidates **running** for?
A_1: Governor
R_1: The Virginia governor's race

Q_2: **Where**?
A_2: Virginia
R_2: The Virginia governor's race

Q_3: Who is the democratic candidate?
A_3: **Terry McAuliffe**
R_3: Democrat Terry McAuliffe

Q_4: Who is **his** opponent?
A_4: **Ken Cuccinelli**
R_4 Republican Ken Cuccinelli

Q_5: What party does **he** belong to?
A_5: Republican
R_5: Republican Ken Cuccinelli

Q_6: Which of **them** is winning?
A_6: Terry McAuliffe
R_6: Democrat Terry McAuliffe, the longtime political fixer and moneyman, hasn't trailed in a poll since May

Source: http://ai.stanford.edu/blog/beyond-local-pattern-matching/

HotpotQA

Paragraph A: Ricardo Rodríguez Saá

Ricardo Rodríguez Saá was Governor of the San Luis Province in Argentina from 1934 to 1938. His great-nephew, Adolfo Rodríguez Saá, would become President of Argentina. His brother, Adolfo, and another great-nephew, Alberto, have also served as Governors of the San Luis Province.

Paragraph B: Adolfo Rodríguez Saá

Adolfo Rodríguez Saá (born July 25, 1947) is an Argentine Peronist politician. Born in a family that was highly influential in the history of the San Luis Province, he became governor in 1983, after the end of the National Reorganization Process military dictatorship. He remained governor up to 2001, being re-elected in successive elections.

Q: Which one of Ricardo Rodríguez Saá's relatives would become governor from 1983 to 2001?

A: Adolfo Rodríguez Saá

Source: http://ai.stanford.edu/blog/beyond-local-pattern-matching/

HotpotQA: What state was Yahoo founded in?

History of Yahoo!

From Wikipedia, the free encyclopedia

See also: Timeline of Yahoo!

This article needs to be **updated**. Please update this article to reflect recent events or newly available information. *(May 2016)*

Yahoo! was started at Stanford University. It was founded in January 1994 by Jerry Yang and David Filo, who were Electrical Engineering graduate students when they created a website named "Jerry and David's Guide to the World Wide Web". The Guide was a directory of other websites, organized in a hierarchy, as opposed to a searchable index of pages. In April 1994, Jerry and David's Guide to the World Wide Web was renamed "Yahoo!".[1][2] The word "YAHOO" is a backronym for "Yet Another Hierarchically Organized Oracle"[3] or "Yet Another Hierarchical Officious Oracle."[4] The yahoo.com domain was created on January 18, 1995.[5]

Source: http://ai.stanford.edu/blog/beyond-local-pattern-matching/

Overall Takeaways

- RC is an important task that draws on several other components of language understanding
- Datasets are critical for reading comprehension
  - Hard to create large datasets
  - Hard to create datasets on which high performance requires “true” language understanding
- We can do well on the easier datasets but not the tougher ones yet
- The more attention, the better
  - LSTM < Attentive Reader < BiDAF < BERT
- Pre-training helps A LOT!

References

MCTest paper: https://www.aclweb.org/anthology/D13-1020.pdf

CNN/Daily Mail paper: https://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.pdf

BiDAF paper: https://arxiv.org/pdf/1611.01603.pdf

BERT paper: https://arxiv.org/pdf/1810.04805.pdf

DROP paper: https://www.semanticscholar.org/paper/DROP%3A-A-Reading-Comprehension-Benchmark-Requiring-Dua-Wang/dda6fb309f62e2557a071522354d8c2c897a2805

Thank you!

Impatient reader

Same as attentive reader but rereads from the document as each token is read, so attention is repeatedly applied:

$$m(t) = \tanh(W_{ym} y_d(t) + W_{um} u)$$
$$s(t) \propto \exp(W_{m,s}^T m(t))$$
$$r = \sum_t s(t) y_d(t)$$

Attentive reader

$$m(i,t) = \tanh(W_{dm} y_d(t) + W_{rm} r(i-1) + W_{qm} y_q(i)), 1 \leq i \leq |q|$$
$$s(i,t) \propto \exp(W_{ms}^T m(i,t))$$
$$r(i) = \tanh(W_{rr} r(i-1)) + \sum_t s(i,t) y_d(t)$$

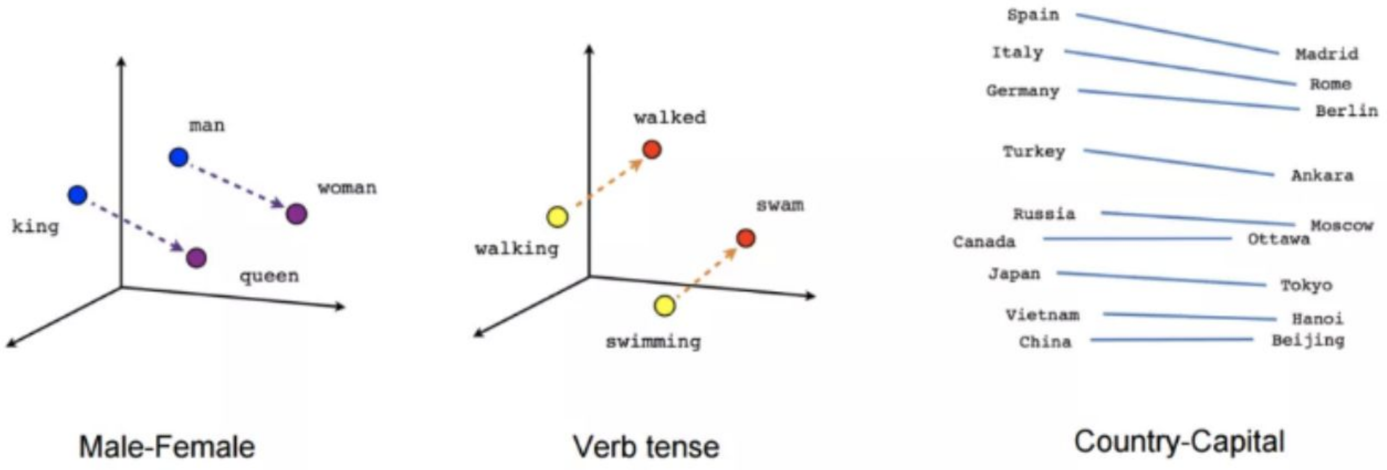
Impatient reader

A Closer Look: Embedding Layers

- Step 1: Tokenization

A Closer Look: Embedding Layers

- Step 2: Word Embeddings (GloVe)



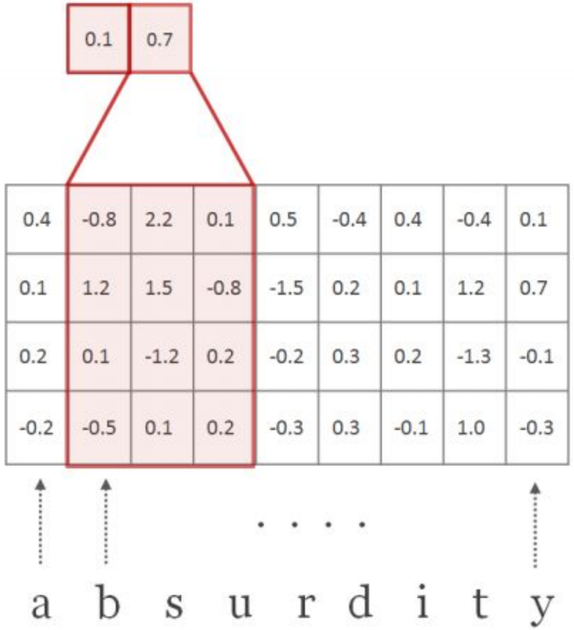
The distance between two GloVe vectors in space encapsulates a meaningful concept, such as gender, tense variation and country-capital relationship.

A Closer Look: Embedding Layers

- Step 3: Character embeddings (CNN)
  - Input: T words from context, J words from query
  - Output: vector of fixed size of each word

Randomly initialized d x L matrix → convolutional filter → Hadamard product → summary scalar

$$\mathbf{f}[2] = \langle \mathbf{C}[*, 2:4], \mathbf{H} \rangle$$



A Closer Look: Embedding Layers

- Step 3: Character embeddings (CNN)
  - Input: T words from context, J words from query
  - Output: vector of fixed size of each word

Analogous to feature extraction in vision!

anti | dis | establish | ment | arian | ism

A Closer Look: Embedding Layers

- Step 4: Highway Network
  - Input: concatenation of character and word embeddings in R^d
  - Output: partial modification of this, all embeddings still in R^d

$$X \in \mathbb{R}^{d \times T}, Q \in \mathbb{R}^{d \times J}$$

- Regular feedforward NN: $\mathbf{y} = H(\mathbf{x}, \mathbf{W_H})$
- Highway NN: $\mathbf{y} = H(\mathbf{x}, \mathbf{W_H}) \cdot T(\mathbf{x}, \mathbf{W_T}) + \mathbf{x} \cdot (1 - T(\mathbf{x}, \mathbf{W_T}))$
- Generalisation of a ResNet block
  - For ResNet, effectively T(x, W_T) = 1/2

A Closer Look: Embedding Layers

- Step 5: Contextual Embeddings
- Input: output from the highway network $X \in \mathbb{R}^{d \times T}, Q \in \mathbb{R}^{d \times J}$
- Feed through forward and backward LSTMs and concatenate
- Output: $H \in \mathbb{R}^{2d \times T}, U \in \mathbb{R}^{2d \times J}$

Source: https://towardsdatascience.com/the-definitive-guide-to-bidaf-part-3-attention-92352bbdcb07

A Closer Look: Attention Layers

- Similarity matrix encoding similarities between context and query embedding vectors

- Generalisation of inner products

$$S \in \mathbb{R}^{T \times J}$$

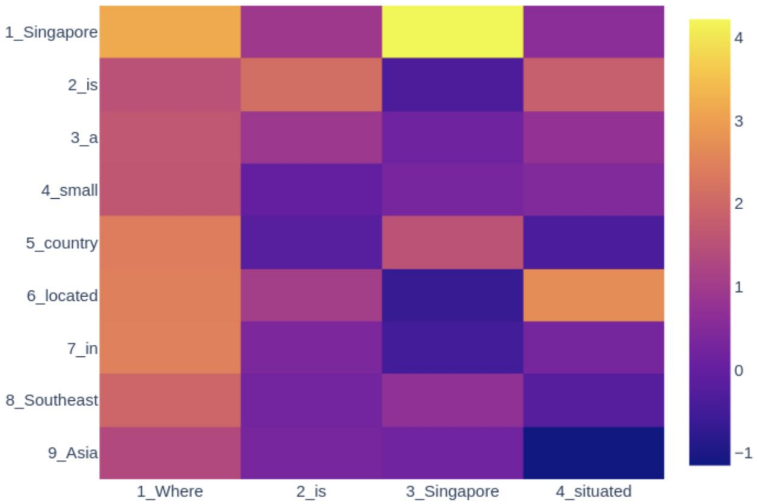
$$S_{tj} = \alpha(H_{:t}, U_{:j}) \in \mathbb{R}$$

$$\alpha(h, u) = w_{(S)}^{T}[h; u; h \circ u]$$

$$H_{:t} = t\text{th column of H}$$

$$\circ = \text{ elementwise multiplication}$$

$$w_{(S)} = \text{ trainable weight vector in } \mathbb{R}^{6d}$$



A Closer Look: Attention Layers

- Context-to-Query Attention
- Output: attended vector for each word in context $\widetilde{U} \in \mathbb{R}^{2d \times T}$
- a_tj: “how important is query word j to context word t”

$$\widetilde{U}_{:t} = \sum_{j=1}^{J} a_{tj} U_{:j}$$

$$a_{tj} \propto \exp(S_{tj})$$

$$\sum_{j=1}^{J} a_{tj} = 1$$

$$U = \text{query embeddings in } \mathbb{R}^{2d \times J}$$

$$\widetilde{U} \in \mathbb{R}^{2d \times T}$$

A Closer Look: Attention Layers

- Query-to-Context Attention
- Output: attended vector for the <u>overall query</u>
- b_t: “how important is document word t to the query”

$$\tilde{h} = \sum_t b_t H_{:t}$$

$$b_t \propto \exp(\max_j S_{tj})$$

$$\sum_t b_t = 1$$

$$\tilde{h} \in \mathbb{R}^{2d}$$

$$\tilde{H} = \tilde{h} \text{ tiled } T \text{ times in a row} \in \mathbb{R}^{2d \times T}$$

$$H = \text{document embeddings in } \mathbb{R}^{2d \times T}$$

A Closer Look: Attention Layers

- Megamerge
- Inputs:
  - Context word embeddings (before attention): $H \in \mathbb{R}^{2d \times T}$
  - Attended C2Q embeddings (weighted sums of query word embeddings): $\widetilde{U} \in \mathbb{R}^{2d \times T}$
  - Attended Q2C embedding (weighted sum of context word embeddings): $\tilde{h} \in \mathbb{R}^{2d} \rightarrow \widetilde{H} \in \mathbb{R}^{2d \times T}$
- Output: $G \in \mathbb{R}^{8d \times T}$
  - Concatenation and element-wise multiplication

$$G_{:t} = \beta(H_{:t}, \widetilde{U}_{:t}, \widetilde{H}_{:t})$$

$$\beta(h, \tilde{u}, \tilde{h}) = [h; \tilde{u}; h \circ \tilde{u}; h \circ \tilde{h}] \in \mathbb{R}^{8d}$$