Memory, Reading, and Comprehension

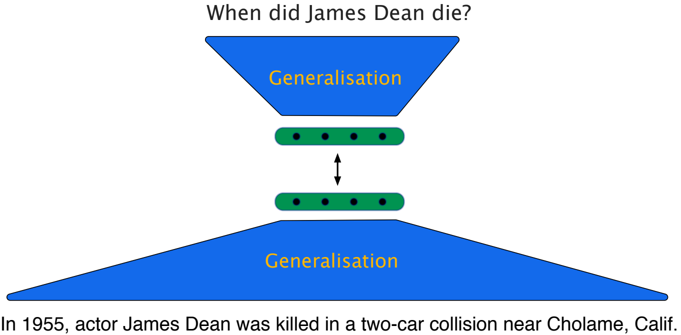
Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, Lei Yu, and **Phil Blunsom**

pblunsom@google.com

Google DeepMind

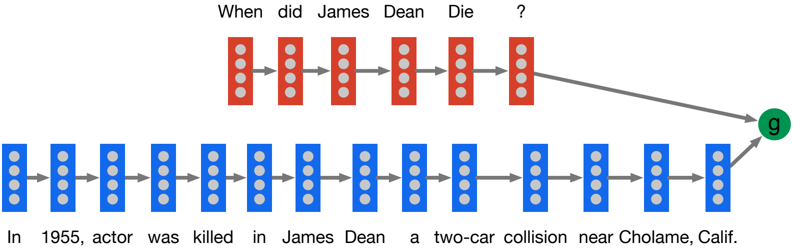
University of Oxford
Department of Computer Science

Deep Learning and NLP: Question Answer Selection



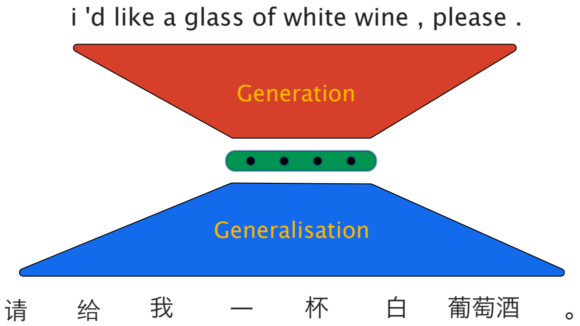
Beyond classification, deep models for embedding sentences have seen increasing success.

Deep Learning and NLP: Question Answer Selection



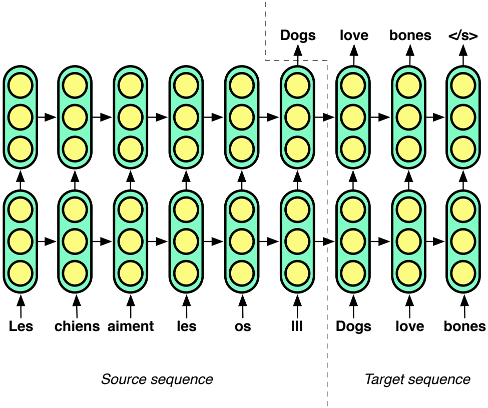
Recurrent neural networks provide a very practical tool for sentence embedding.

Deep Learning for NLP: Machine Translation



We can even view translation as encoding and decoding sentences.

Deep Learning for NLP: Machine Translation



Recurrent neural networks again perform surprisingly well.

NLP at Google DeepMind

Small steps towards NLU:

- reading and understanding text,
- connecting natural language, action, and inference in real environments.

Outline

Neural Unbounded Memory

Neural Machine Reading and Comprehension

Transduction and RNNs

Recently there have been many proposals to incorporate random access memories into recurrent networks:

- Memory Networks / Attention Models (Weston et al., Bahdanau et al. etc.)
- Neural Turing Machine (Graves et al.)

These are very powerful models, perhaps too powerful for many tasks. Here we will explore more restricted memory architectures with properties more suited to NLP tasks, along with better scalability.

Transduction and RNNs

Many NLP (and other!) tasks are castable as transduction problems. E.g.:

Translation: English to French transduction

Parsing: String to tree transduction

Computation: Input data to output data transduction

Transduction and RNNs

Generally, goal is to transform some source sequence

$$S = s_1\, s_2\, \ldots\, s_m,$$

into some target sequence

$$T = t_1\, t_2\, \ldots\, t_n,$$

Approach:

1. Model $P(t_{i+1}|t_1 \ldots t_n; S)$ with an RNN
2. Read in source sequences
3. Generate target sequences (greedily, beam search, etc).

Transduction and RNNs

1. Concatenate source and target sequences into joint sequences:

$$s_1\, s_2\, \ldots\, s_m \,|||\, t_1\, t_2\, \ldots\, t_n$$

2. Train a single RNN over joint sequences
3. Ignore RNN output until separator symbol (e.g. "|||")
4. Jointly learn to compose source and generate target sequences

Learning to Execute

Task (Zaremba and Sutskever, 2014):

1. Read simple python scripts character-by-character
2. Output numerical result character-by-character.

Input:

```
j=8584
for x in range(8):
  j+=920
b=(1500+j)
print((b+7567))
```

Target: 25011.

Input:

```
i=8827
c=(i-5347)
print((c+8704) if 2641<8500 else 5308)
```

Target: 12184.

Unbounded Neural Memory

Here we investigate memory modules that act like Stacks/Queues/DeQues:[1]

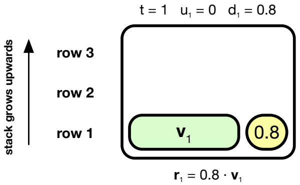
- Memory "size" grows/shrinks dynamically
- Continuous push/pop not affected by number of objects stored
- Can capture unboundedly long range dependencies*
- Propagates gradient flawlessly*

---

[1]

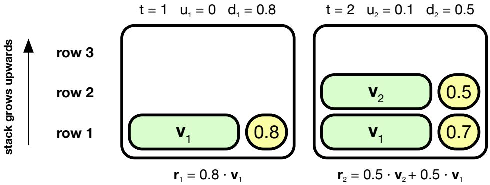
Sun et al. The neural network pushdown automaton: Model, stack and learning simulations. 1998
Joulin and Mikolov. Inferring algorithmic patterns with stack-augmented recurrent nets. 2015
Grefenstette et al. Learning to Transduce with Unbounded Memory. 2015

Example: A Continuous Stack



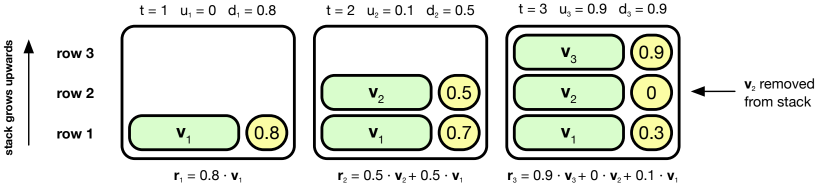
where t is the timestep, v_t is the value vector, u_t is the pop depth, d_t is the push strength, and r_t is the read vector. The read depth is always define to be 1.

Example: A Continuous Stack



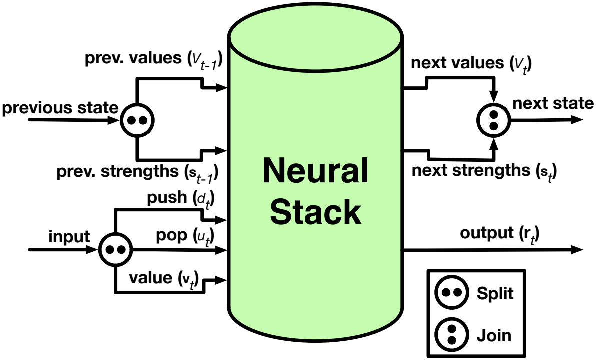
where t is the timestep, v_t is the value vector, u_t is the pop depth, d_t is the push strength, and r_t is the read vector. The read depth is always define to be 1.

Example: A Continuous Stack

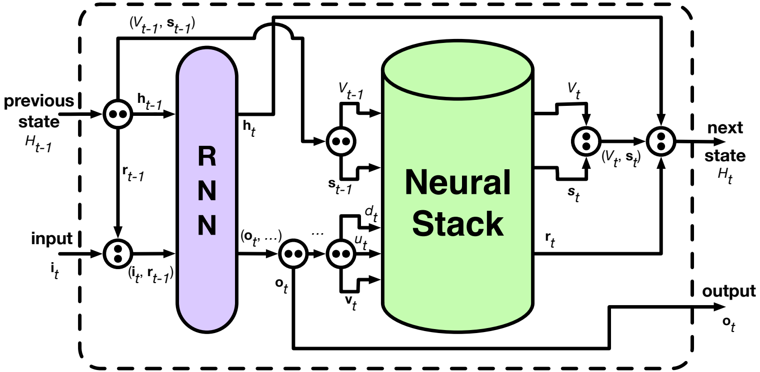


where t is the timestep, v_t is the value vector, u_t is the pop depth, d_t is the push strength, and r_t is the read vector. The read depth is always define to be 1.

Example: A Continuous Stack



Example: Controlling a Neural Stack



Synthetic Transduction Tasks

Copy

$$a_1\, a_2\, a_3\, \ldots\, a_n \rightarrow a_1\, a_2\, a_3\, \ldots\, a_n$$

Reversal

$$a_1\, a_2\, a_3\, \ldots\, a_n \rightarrow a_n\, a_3\, a_2\, a_1$$

Bigram Flipping

$$a_1\, a_2\, a_3\, a_4\, \ldots\, a_{n-1}\, a_n \rightarrow a_2\, a_1\, a_4\, a_3\, \ldots\, a_n\, a_{n-1}$$

Synthetic ITG Transduction Tasks

Subject–Verb–Object to Subject–Object–Verb Reordering

*si*1 *vi*28 *oi*5 *oi*7 *si*15 *rpi* *si*19 *vi*16 *oi*10 *oi*24

↓

*so*1 *oo*5 *oo*7 *so*15 *rpo* *so*19 *vo*16 *oo*10 *oo*24 *vo*28

Genderless to Gendered Grammar

*we*11 *the* *en*19 *and* *the* *em*17

↓

*wg*11 *das* *gn*19 *und* *der* *gm*17

Coarse and Fine Grained Accuracy

Coarse–grained accuracy Proportion of entirely correctly predicted sequences in test set

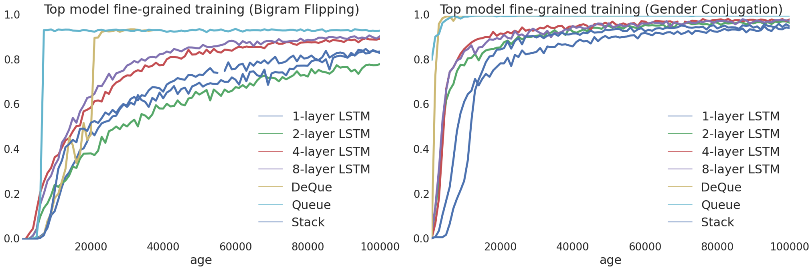
Fine–grained accuracy Average proportion of sequence correctly predicted before first error

Results

Experiment	Stack	Queue	DeQue	Deep LSTM
Copy	Poor	**Solved**	**Solved**	Poor
Reversal	**Solved**	Poor	**Solved**	Poor
Bigram Flip	Converges	**Best Results**	**Best Results**	Converges
SVO-SOV	**Solved**	**Solved**	**Solved**	Converges
Conjugation	Converges	**Solved**	**Solved**	Converges

Every Neural Stack/Queue/DeQue that solves a problem preserves the solution for longer sequences (tested up to 2x length of training sequences).

Rapid Convergence



Outline

Neural Unbounded Memory

Neural Machine Reading and Comprehension

Supervised Reading Comprehension

To achieve our aim of training supervised machine learning models for machine reading and comprehension, we must first find data.

Supervised Reading Comprehension: MCTest

Document

James the Turtle was always getting in trouble. Sometimes he'd reach into the freezer and empty out all the food. Other times he'd sled on the deck and get a splinter. His aunt Jane tried as hard as she could to keep him out of trouble, but he was sneaky and got into lots of trouble behind her back.
One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn't pay, and instead headed home.

. . .

Question

Where did James go after he went to the grocery store?

1. his deck
2. his freezer
3. a fast food restaurant
4. his room

Supervised Reading Comprehension: FB Synthetic

Synthetic example from the FaceBook data set

John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? A:office

An alternative to real language is to generate scripts from a synthetic grammar.

Supervised Reading Comprehension

The CNN and DailyMail websites provide paraphrase summary sentences for each full news story.

Supervised Reading Comprehension

CNN article:

Document 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

We formulate *Cloze* style queries from the story paraphrases.

Supervised Reading Comprehension

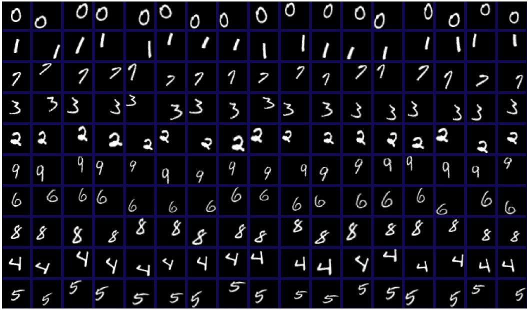
From the Daily Mail:

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

An ngram language model would correctly predict (**X** = *cancer*), regardless of the document, simply because this is a frequently cured entity in the Daily Mail corpus.

Supervised Reading Comprehension

MNIST example generation:



We generate quasi-synthetic examples from the original document-query pairs, obtaining exponentially more training examples by anonymising and permuting the mentioned entities.

Supervised Reading Comprehension

	Original Version	Anonymised Version
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." . . .	the *ent381* producer allegedly struck by *ent212* will not press charges against the " *ent153* " host , his lawyer said friday . *ent212* , who hosted one of the most - watched television shows in the world , was dropped by the *ent381* wednesday after an internal investigation by the *ent180* broadcaster found he had subjected producer *ent193* " to an unprovoked physical and verbal attack . " . . .
Query		
	Producer **X** will not press charges against Jeremy Clarkson, his lawyer says.	producer **X** will not press charges against *ent212* , his lawyer says .
Answer		
	Oisin Tymon	*ent193*

Original and anonymised version of a data point from the Daily Mail validation set. The anonymised entity markers are constantly permuted during training and testing.

Data Set Statistics

	CNN			Daily Mail		
	train	valid	test	train	valid	test
# months	95	1	1	56	1	1
# documents	108k	1k	1k	195k	12k	11k
# queries	438k	4k	3k	838k	61k	55k
Max # entities	456	190	398	424	247	250
Avg # entities	30	32	30	41	45	45
Avg tokens/doc	780	809	773	1044	1061	1066
Vocab size		125k			275k	

Articles were collected from April 2007 for CNN and June 2010 for the Daily Mail, until the end of April 2015. Validation data is from March, test data from April 2015.

Question difficulty

Category	**Sentences**		
	1	**2**	**≥3**
Simple	12	2	0
Lexical	14	0	0
Coref	0	8	2
Coref/Lex	10	8	4
Complex	8	8	14
Unanswerable		10	

Distribution (in percent) of queries over category and number of context sentences required to answer them based on a subset of the CNN validation data.

Frequency baselines (Accuracy)

	CNN		Daily Mail	
	valid	test	valid	test
Maximum frequency	26.3	27.9	22.5	22.7
Exclusive frequency	30.8	32.6	27.3	27.7

A simple baseline is to always predict the entity appearing most often in the document. A refinement of this is to exclude entities in the query.

Frame semantic matching

A stronger benchmark using a state-of-the-art frame semantic parser and rules with an increasing recall/precision trade-off:

	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	$(p, \mathit{be.01.V}, y)$	$(\boldsymbol{x}, \mathit{be.01.V}, y)$	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*		

$\boldsymbol{x}$ denotes the entity proposed as answer, V is a fully qualified PropBank frame (e.g. *give.01.V*). Strategies are ordered by precedence and answers determined accordingly.

Frame semantic matching

	CNN		Daily Mail	
	valid	test	valid	test
Maximum frequency	26.3	27.9	22.5	22.7
Exclusive frequency	30.8	32.6	27.3	27.7
Frame-semantic model	32.2	33.0	30.7	31.1

Failure modes:

- The Propbank parser has poor coverage with many relations not picked up as they do not adhere to the default predicate-argument structure.
- The frame-semantic approach does not trivially scale to situations where several frames are required to answer a query.

Word distance benchmark

Consider the query "*Tom Hanks is friends with* **X***'s manager, Scooter Brown*" where the document states "*... turns out he is good friends with Scooter Brown, manager for Carly Rae Jepson*."

The frame-semantic parser fails to pickup the friendship or management relations when parsing the query.

Word distance benchmark

Word distance benchmark:

- align the placeholder of the *Cloze* form question with each possible entity in the context document,
- calculate a distance measure between the question and the context around the aligned entity,
- sum the distances of every word in Q to its nearest aligned word in D.

Alignment is defined by matching words either directly or as aligned by the coreference system.

Word distance benchmark

	CNN		Daily Mail	
	valid	test	valid	test
Maximum frequency	26.3	27.9	22.5	22.7
Exclusive frequency	30.8	32.6	27.3	27.7
Frame-semantic model	32.2	33.0	30.7	31.1
Word distance model	46.2	46.9	55.6	54.8

This benchmark is robust to small mismatches between the query and answer, correctly solving most instances where the query is generated from a highlight which in turn closely matches a sentence in the context document.

Reading via Encoding

Use neural encoding models for estimating the probability of word type a from document d answering query q:

$$p(a|d,q) \propto \exp\left(W(a)g(d,q)\right), \quad \text{s.t. } a \in d.$$

where $W(a)$ indexes row a of weight matrix W and function $g(d,q)$ returns a vector embedding of a document and query pair.

Deep LSTM Reader

We employ a Deep LSTM cell with skip connections,

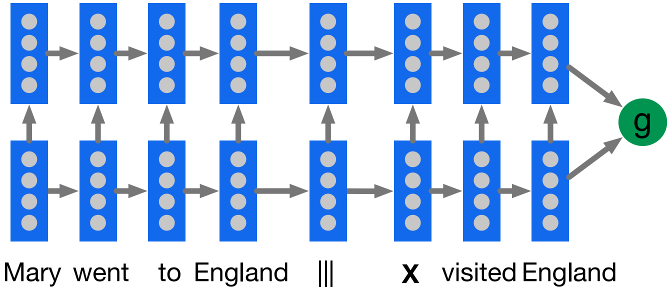
$$
\begin{aligned}
x'(t,k) &= x(t)||y'(t,k-1), \\
i(t,k) &= \sigma\left(W_{kxi}x'(t,k) + W_{khi}h(t-1,k) + W_{kci}c(t-1,k) + b_{ki}\right), \\
f(t,k) &= \sigma\left(W_{kxf}x'(t,k) + W_{khf}h(t-1,k) + W_{kcf}c(t-1,k) + b_{kf}\right), \\
c(t,k) &= f(t,k)c(t-1,k) + i(t,k)\tanh\left(W_{kxc}x'(t,k) + W_{khc}h(t-1,k) + b_{kc}\right), \\
o(t,k) &= \sigma\left(W_{kxo}x'(t,k) + W_{kho}h(t-1,k) + W_{kco}c(t,k) + b_{ko}\right), \\
h(t,k) &= o(t,k)\tanh\left(c(t,k)\right), \\
y'(t,k) &= W_{ky}h(t,k) + b_{ky}, \\
y(t) &= y'(t,1)||\ldots||y'(t,K),
\end{aligned}
$$

where $||$ indicates vector concatenation $h(t,k)$ is the hidden state for layer k at time t, and i, f, o are the input, forget, and output gates respectively.

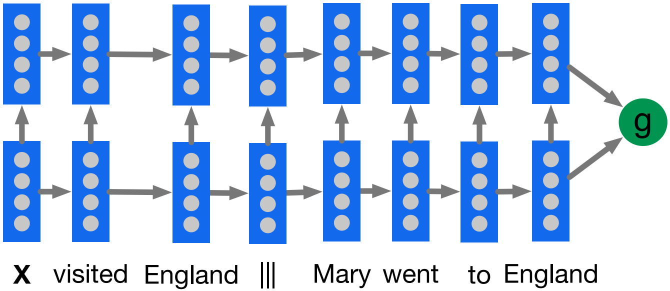
$$g^{\text{LSTM}}(d,q) = y(|d| + |q|)$$

with input $x(t)$ the concatenation of d and q separated by the delimiter $|||$.

Deep LSTM Reader



Deep LSTM Reader



Deep LSTM Reader

	CNN		Daily Mail	
	valid	test	valid	test
Maximum frequency	26.3	27.9	22.5	22.7
Exclusive frequency	30.8	32.6	27.3	27.7
Frame-semantic model	32.2	33.0	30.7	31.1
Word distance model	46.2	46.9	55.6	54.8
Deep LSTM Reader	49.0	49.9	57.1	57.3

Given the difficult of its task, the Deep LSTM Reader performs very strongly.

The Attentive Reader

Denote the outputs of a bidirectional LSTM as $\overrightarrow{y}(t)$ and $\overleftarrow{y}(t)$. Form two encodings, one for the query and one for each token in the document,

$$u = \overrightarrow{y_q}(|q|) \mid\mid \overleftarrow{y_q}(1), \qquad y_d(t) = \overrightarrow{y_d}(t) \mid\mid \overleftarrow{y_d}(t).$$

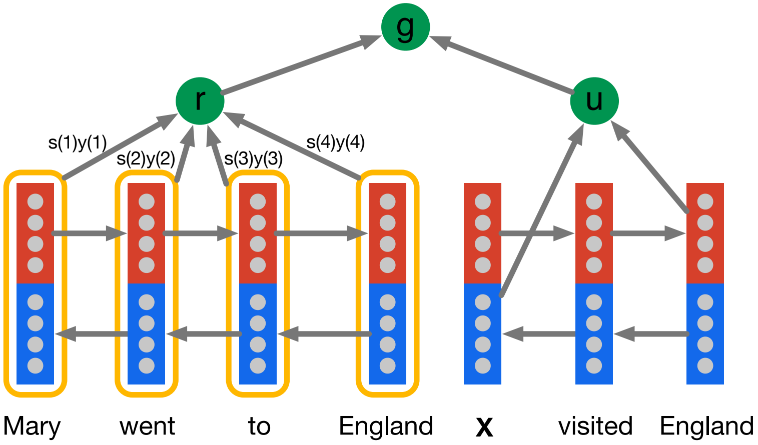
The representation r of the document d is formed by a weighted sum of the token vectors. The weights are interpreted as the model's attention,

$$\begin{aligned} m(t) &= \tanh\left(W_{ym} y_d(t) + W_{um} u\right), \\ s(t) &\propto \exp\left(\mathrm{w}_{ms}^{\intercal} m(t)\right), \\ r &= y_d s. \end{aligned}$$

Define the joint document and query embedding via a non-linear combination:

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

The Attentive Reader



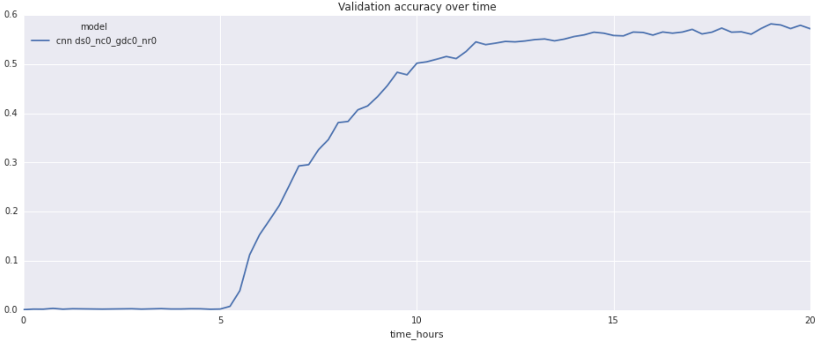
The Attentive Reader

	CNN		Daily Mail	
	valid	test	valid	test
Maximum frequency	26.3	27.9	22.5	22.7
Exclusive frequency	30.8	32.6	27.3	27.7
Frame-semantic model	32.2	33.0	30.7	31.1
Word distance model	46.2	46.9	55.6	54.8
Deep LSTM Reader	49.0	49.9	57.1	57.3
Uniform attention[2]	31.1	33.6	31.0	31.7
Attentive Reader	56.5	58.9	64.5	63.7

The attention variables effectively address the Deep LSTM Reader's inability to focus on part of the document.

[2] The Uniform attention baseline sets all $m(t)$ parameters to be equal.

Attentive Reader Training



Models were trained using asynchronous minibatch stochastic gradient descent (RMSProp) on approximately 25 GPUs.

The Attentive Reader: Predicted: *ent49*, Correct: *ent49*

by *ent40* , *ent62* correspondent updated 9:49 pm et , thu march 19 , 2015 (*ent62*) a *ent88* was killed in a parachute accident in *ent87* , *ent28* , near *ent66* , a *ent47* official told *ent62* on wednesday . he was identified thursday as special warfare operator 3rd class *ent49* , 29 , of *ent44* , *ent13* . `` *ent49* distinguished himself consistently throughout his career . he was the epitome of the quiet professional in all facets of his life , and he leaves an inspiring legacy of natural tenacity and focused commitment for posterity , '' the *ent47* said in a news release . *ent49* joined the seals in september after enlisting in the *ent47* two years earlier . he was married , the *ent47* said . initial indications are the parachute failed to open during a jump as part of a training exercise . *ent49* was part of a *ent57* - based *ent88* team .

ent47 identifies deceased sailor as **X** , who leaves behind a wife

The Attentive Reader: Predicted: *ent27*, Correct: *ent27*

by *ent82* , *ent38* updated 9:35 am et , mon march 2 , 2015 (*ent38*) *ent27* went familial for fall at its fashion show in *ent23* on sunday , dedicating its collection to `` mamma '' with nary a pair of `` mom jeans '' in sight . *ent57* and *ent78* , who are behind the *ent72* brand , sent models down the runway in decidedly feminine dresses and skirts adorned with roses , lace and even embroidered doodles by the designers ' own nieces and nephews . many of the looks featured saccharine needlework phrases like `` i love you , mamma '' and `` *ent46* '' (for the most beautiful mother in the world) as a tableau vivant of moms and daughters stood and posed as a backdrop for the runway . our little munchkins backstage *ent44* babies # friends # _UNK_ a photo posted by *ent58* (@ _UNK_) on mar 1 , 2015 at _UNK_ *ent17* even the usually stoic - faced front row could n't help but applaud and smile as a few models carried their own high - fashion progeny down the runway . almost ready for the show : watch the *ent87* live today at *ent8* (*ent65*) on *ent87* website . # _UNK_ # _UNK_ # _UNK_ # _UNK_ # _UNK_ a photo posted by *ent27* (@ _UNK_) on mar 1 , 2015 at _UNK_ *ent17*

X dedicated their fall fashion show to moms

The Attentive Reader: Predicted: *ent85*, Correct: *ent37*

by *ent52* and *ent22* , *ent43* updated 7:12 am et , fri march 20 , 2015 *ent74* , *ent37* (*ent43*) a passenger train overshot a stop and jumped its tracks in northern *ent37* on friday , killing at least 30 people and injuring more than 50 others , a railway spokesman said . the train was headed from *ent85* to the *ent27* holy city of *ent13* when it overshot an intended stop more than halfway along the route , about 35 kilometers (22 miles) east of *ent11* in the northern state of *ent56* , railway spokesman *ent20* said . two coaches and the locomotive derailed . video from the site , shown by *ent43* affiliate *ent33* , showed emergency workers pulling passengers from the train as a crowd looked on . the cause of the incident will be investigated , *ent20* said . *ent43* 's *ent52* reported from *ent74* . *ent43* 's *ent22* wrote in *ent15* .

a passenger train derails about 35 kilometers (22 miles) east of *ent11* in northern **X**

The Attentive Reader: Predicted: *ent24*, Correct: *ent2*

by *ent37* , *ent61* updated 11:44 am et , tue march 10 , 2015 (*ent61*) a suicide attacker detonated a car bomb near a police vehicle in the capital of southern *ent12* 's *ent24* on tuesday , killing seven people and injuring 23 others , the province 's deputy governor said . the attack happened at about 6 p.m. in the *ent27* area of *ent2* city , said *ent66* , deputy governor of *ent24* . several children were among the wounded , and the majority of casualties were civilians , *ent66* said . details about the attacker 's identity and motive were n't immediately available .

car bomb detonated near police vehicle in **X** , deputy governor says

The Impatient Reader

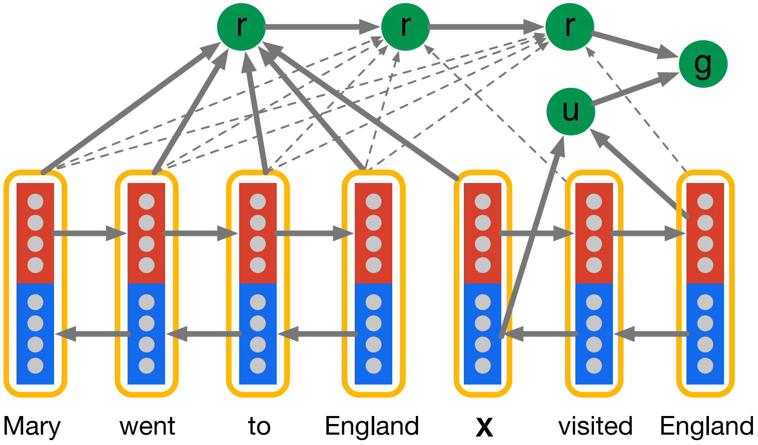
At each token i of the query q compute a representation vector $r(i)$ using the bidirectional embedding $y_q(i) = \overrightarrow{y_q}(i) \mid\mid \overleftarrow{y_q}(i)$:

$$m(i,t) = \tanh\left(W_{dm} y_d(t) + W_{rm} r(i-1) + W_{qm} y_q(i)\right), 1 \le i \le |q|,$$
$$s(i,t) \propto \exp\left(\mathrm{w}_{ms}^{\mathsf{T}} m(i,t)\right),$$
$$r(0) = \mathbf{r_0}, \quad r(i) = y_d^{\mathsf{T}} s(i), \quad 1 \le i \le |q|.$$

The joint document query representation for prediction is,

$$g^{\mathrm{IR}}(d,q) = \tanh\left(W_{rg}\, r(|q|) + W_{qg}\, u\right).$$

The Impatient Reader

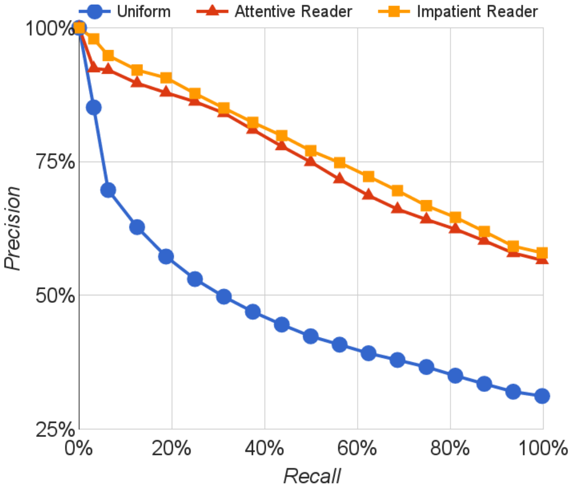


The Impatient Reader

	CNN		Daily Mail	
	valid	test	valid	test
Maximum frequency	26.3	27.9	22.5	22.7
Exclusive frequency	30.8	32.6	27.3	27.7
Frame-semantic model	32.2	33.0	30.7	31.1
Word distance model	46.2	46.9	55.6	54.8
Deep LSTM Reader	49.0	49.9	57.1	57.3
Uniform attention	31.1	33.6	31.0	31.7
Attentive Reader	56.5	58.9	64.5	63.7
Impatient Reader	**57.0**	**60.6**	**64.8**	**63.9**

The Impatient Reader comes out on top, but only marginally.

Attention Models Precision@Recall



Precision@Recall for the attention models on the CNN validation data.

Conclusion

Summary

- supervised machine reading is a viable research direction with the available data,
- LSTM based recurrent networks constantly surprise with their ability to encode dependencies in sequences,
- attention is a very effective and flexible modelling technique.

Future directions

- more and better data, corpus querying, and cross document queries,
- recurrent networks incorporating long term and working memory are well suited to NLU task.

Google DeepMind and Oxford University

Google DeepMind

UNIVERSITY OF OXFORD

DEPARTMENT OF COMPUTER SCIENCE