10417/10617
Intermediate Deep Learning: Fall2020

Russ Salakhutdinov

Machine Learning Department
rsalakhu@cs.cmu.edu

Representation Learning for Reading Comprehension

Reading Comprehension

• **Disclaimer**: Some of the material and slides for this lecture were borrowed from Bhuwan Dhingra's talk.

Reading Comprehension

Document:
"…arrested Illinois governor **Rod Blagojevich** and his chief of staff John Harris on corruption charges … included **Blogojevich** allegedly conspiring to sell or trade the senate seat left vacant by President-elect Barack Obama…"

Query:
"President-elect Barack Obama said Tuesday he was not aware of alleged corruption by **X** who was arrested on charges of trying to sell Obama's senate seat."
Find **X**.

Answer:
Rod Blagojevich

Who-Did-What Dataset (Onishi, Wang, Bansal, Gimpel, McAllester, EMNLP, 2016)

Reading Comprehension

TASK:
Given a document query pair (d, q) find $a \in A$ which answers q.

- d is a document
- q is a question over the contents of that document
- a is the answer to this query

- The answer comes from a fixed vocabulary A.
  - A might consist of all tokens / spans of tokens in the document d
    (Extractive Question Answering)

- Question Answering / Information Extraction
- Test for text representation models
  - A better representation can help answer more questions

Approach -- Supervised Learning

$$\mathcal{D} = \{(d, q, a)\}_{i=1}^{N}$$ **Dataset**

$$Pr(c|d, q) = f_\theta(d, q, c) \quad \forall \quad c \in A$$ **Model**

(A neural network)

$$\mathcal{L}(\theta) = \sum_{(d,q,a)\in\mathcal{D}} -\log Pr(a|d, q)$$ **Loss**

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}(\theta)$$ **Training**

What architectural biases can we build into the model?

Architectural Bias

Designing the connectivity pattern of a Neural Network to reflect the nature of the problem being solved.

- CNNs, RNNs have architectural biases towards images / sequences
- For reading comprehension what biases can we build to reflect **linguistic phenomena?**
  - Alignment, Paraphrasing, Aggregation **(Part 1)**
  - Coreference, Syntactic and Semantic Dependencies **(Part 2)**

Text Phenomena

Document:
"…arrested Illinois governor **Rod Blagojevich** and his chief of staff John Harris on corruption charges … included **Blogojevich** allegedly conspiring to sell or trade the senate seat left vacant by President-elect Barack Obama…"

Query:
"President-elect Barack Obama said Tuesday he was not aware of alleged corruption by **X** who was arrested on charges of trying to sell Obama's senate seat."
Find **X**.

Alignment

Who-Did-What Dataset (Onishi, Wang, Bansal, Gimpel, McAllester, EMNLP, 2016)

Text Phenomena

Document:
"…arrested Illinois governor **Rod Blagojevich** and his chief of staff John Harris on corruption charges… included **Blogojevich** allegedly conspiring to sell or trade the senate seat left vacant by President-elect Barack Obama…"

Query:
"President-elect Barack Obama said Tuesday he was not aware of alleged corruption by **X** who was arrested on charges of trying to sell Obama's senate seat."
Find **X**.

Alignment

Paraphrasing

Who-Did-What Dataset (Onishi, Wang, Bansal, Gimpel, McAllester, EMNLP, 2016)

Text Phenomena

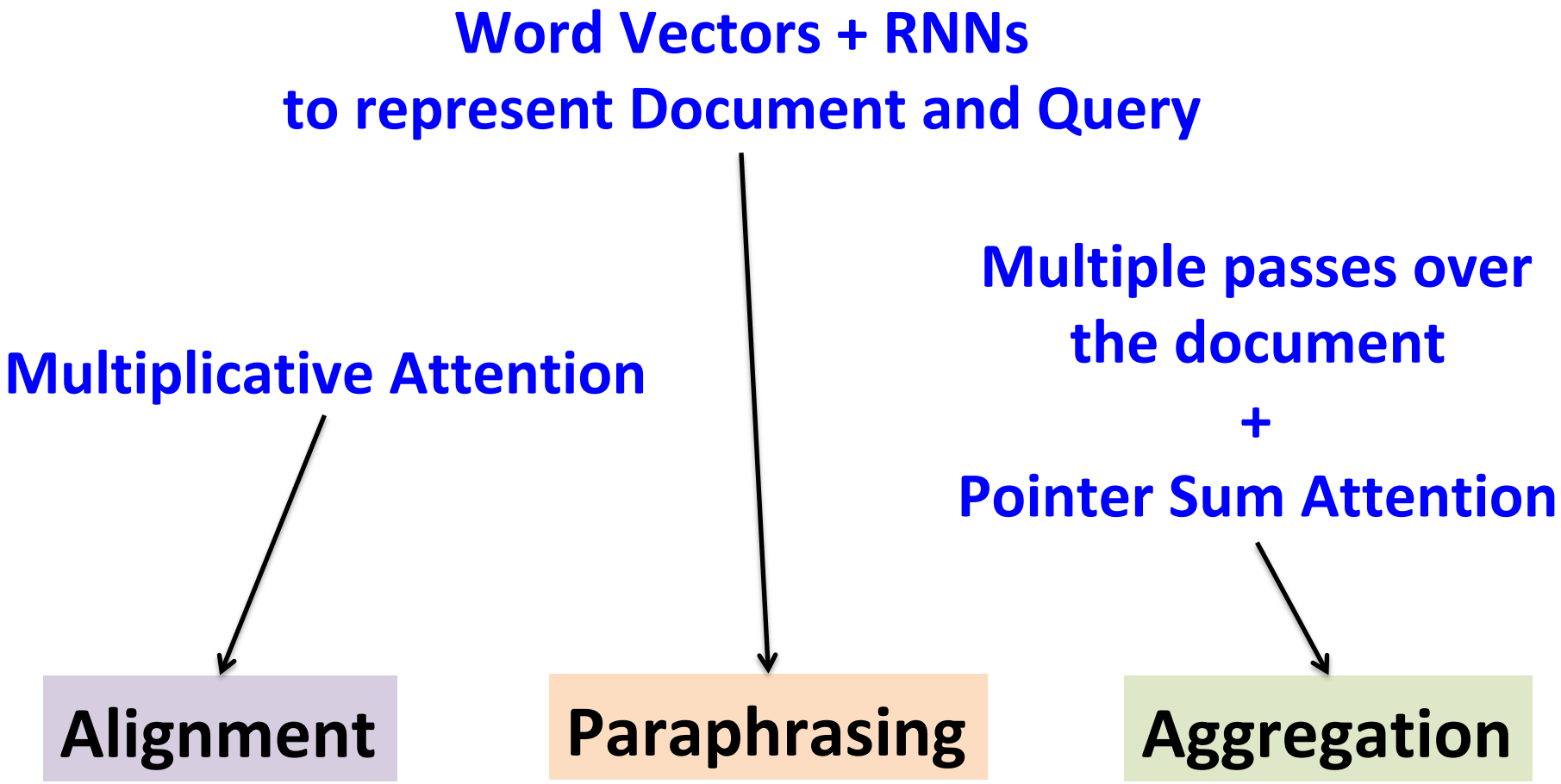
Document:
"…arrested Illinois governor **Rod Blagojevich** and his chief of staff John Harris on corruption charges … included **Blogojevich** allegedly conspiring to sell or trade the senate seat left vacant by President-elect Barack Obama…"

Query:
"President-elect Barack Obama said Tuesday he was not aware of alleged corruption by **X** who was arrested on charges of trying to sell Obama's senate seat."
Find **X**.

Alignment **Paraphrasing** **Aggregation**

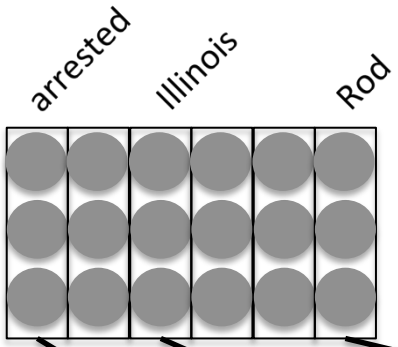
Who-Did-What Dataset (Onishi, Wang, Bansal, Gimpel, McAllester, EMNLP, 2016)

Biases


Word Vectors + RNNs
to represent Document and Query
Multiplicative Attention
Multiple passes over
the document
+
Pointer Sum Attention
Alignment
Paraphrasing
Aggregation

Representing Document / Query

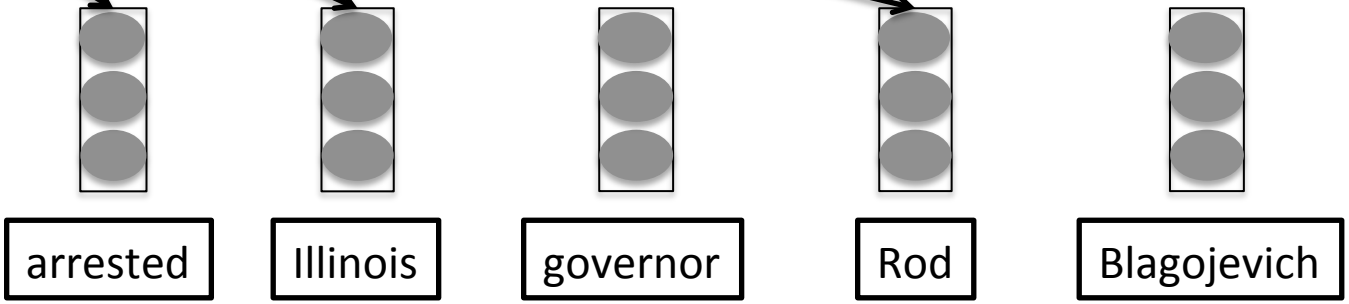
- As compositions of word vectors



Things which help:

- Pretrained **Glove embeddings**
- **Random vectors for OOV** tokens at test time.
  - Better than trained "UNK" embedding.
- **Character embeddings**

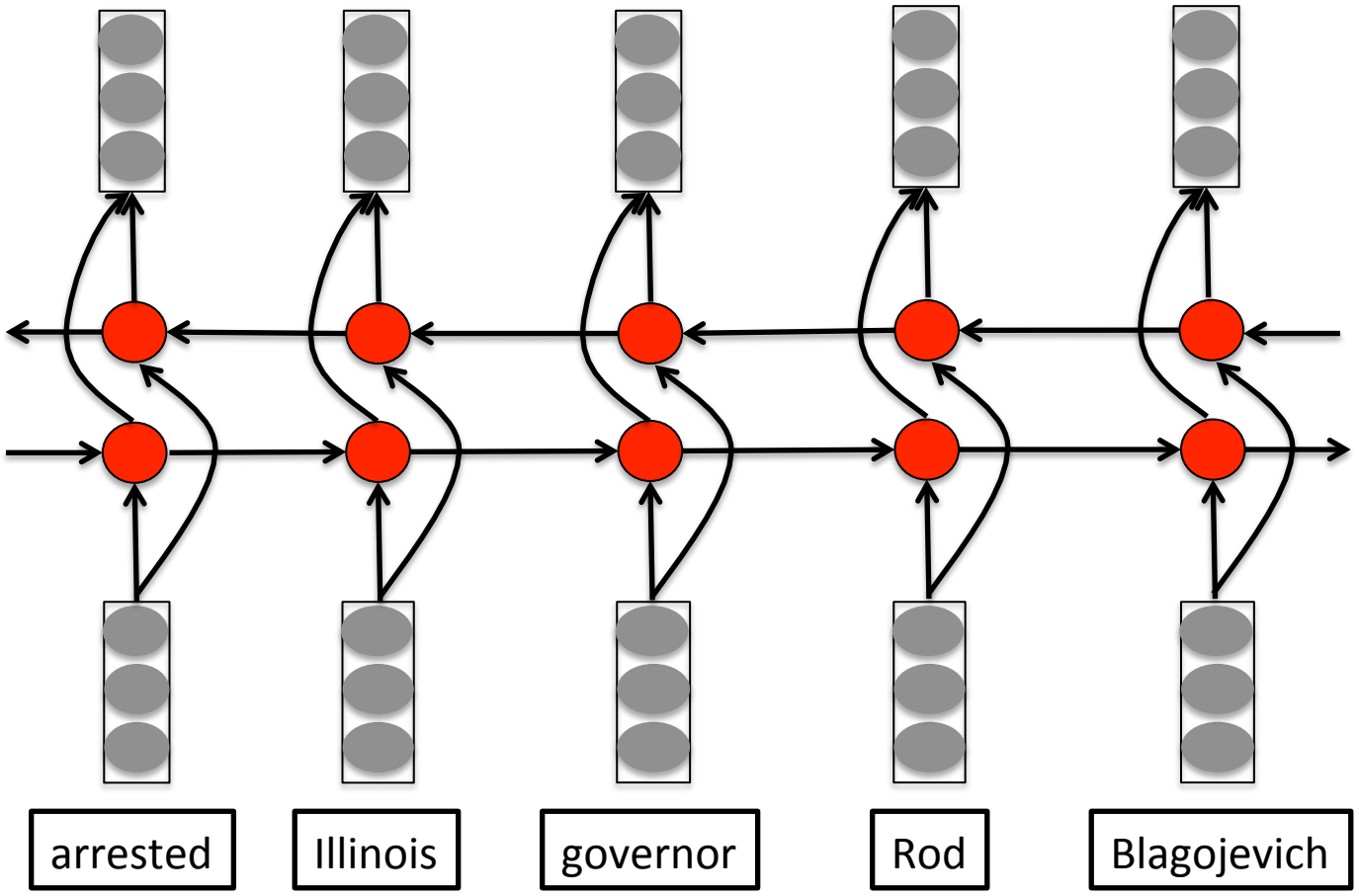
**Dhingra, Liu, Salakhutdinov, Cohen (preprint, 2017)*
***Yang, Dhingra, Yuan, Hu, Cohen, Salakhutdinov (ICLR, 2017)*



Representing Document / Query

- **Bidirectional Gated Recurrent Units** process the tokens from left to right and right to left

$$d_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$$

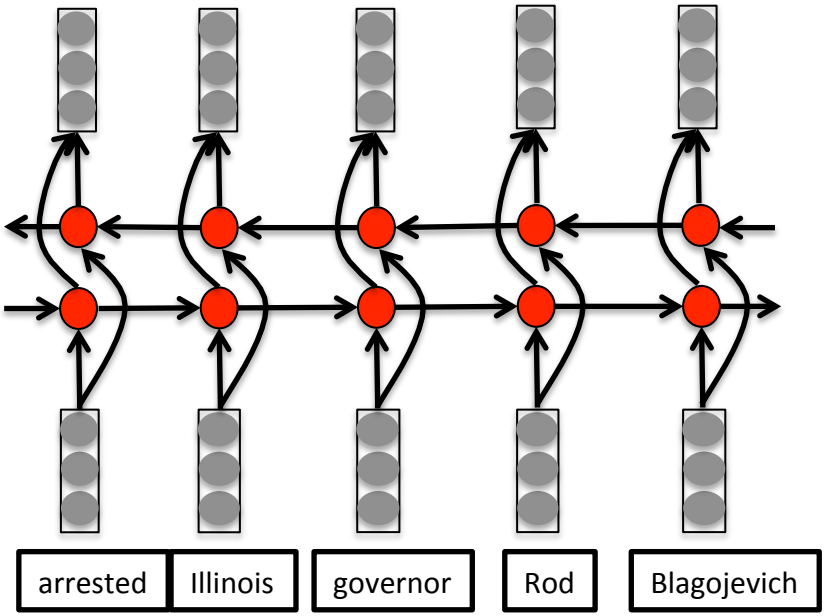


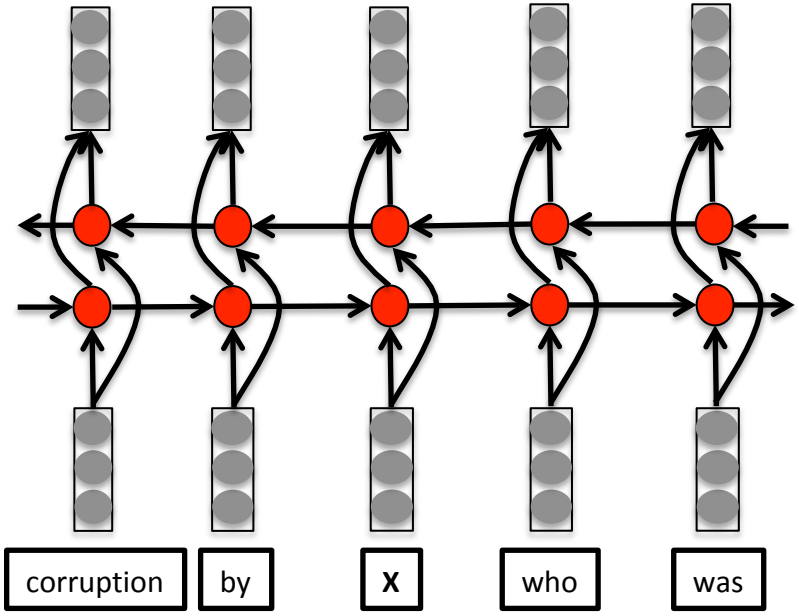
Representing Document / Query

- Both document and query are represented as matrices

$$D \in \mathbb{R}^{2h \times |D|}$$

$$Q \in \mathbb{R}^{2h \times |Q|}$$

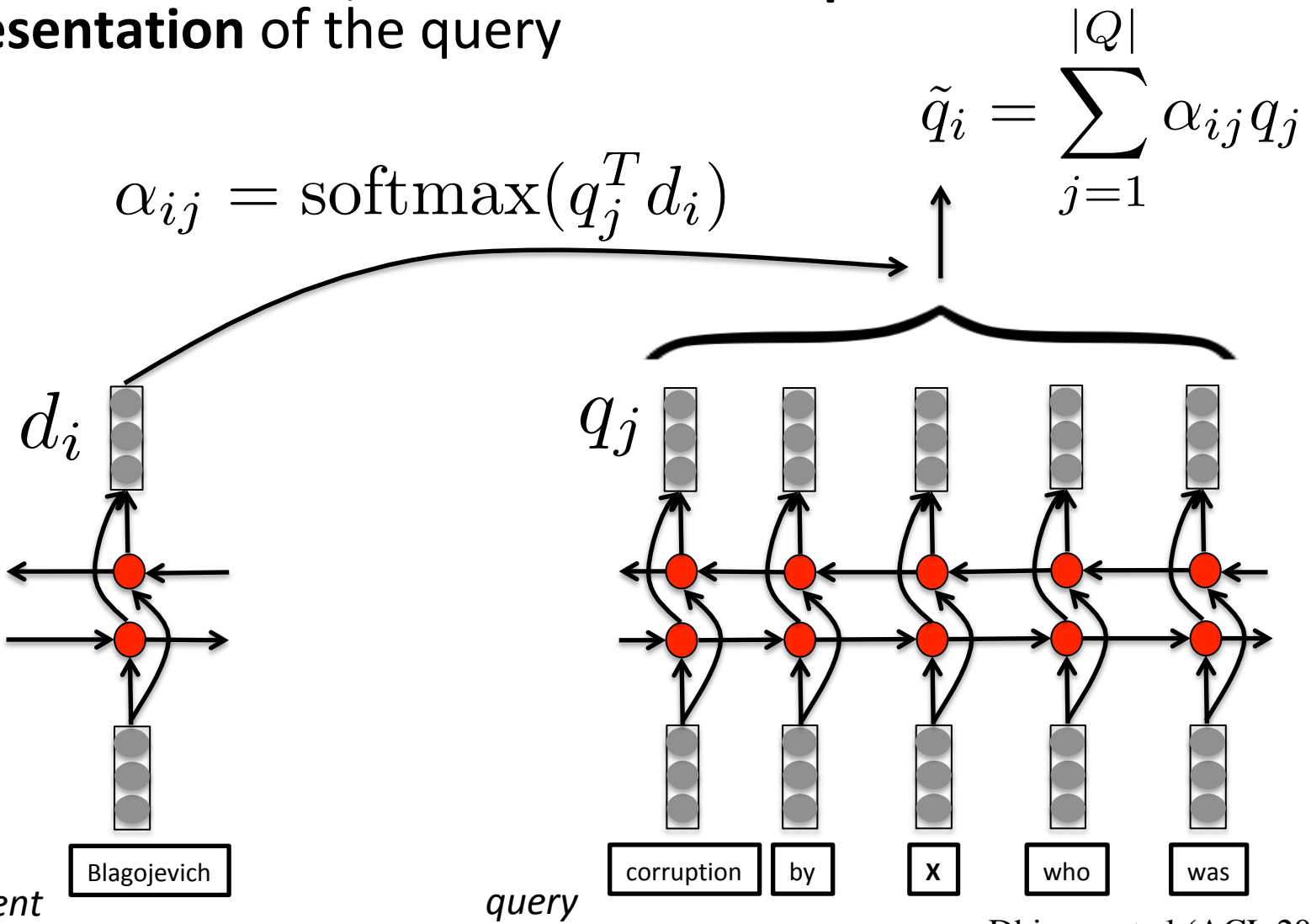




h – State size of each GRU

Gated Attention Mechanism

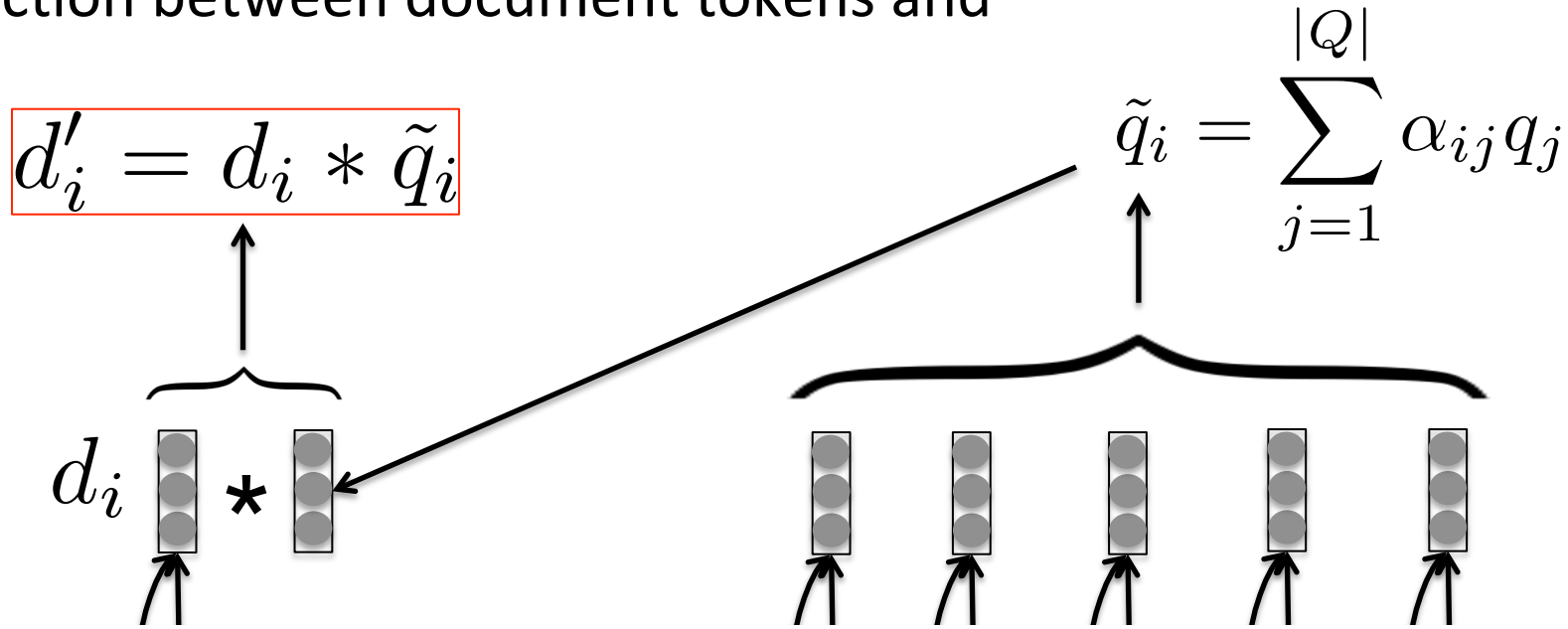
- For each token in D, we form a **token specific representation** of the query



Dhingra et al (ACL 2017)

Gated Attention Mechanism

- Use **element-wise multiplication** to gate the interaction between document tokens and query

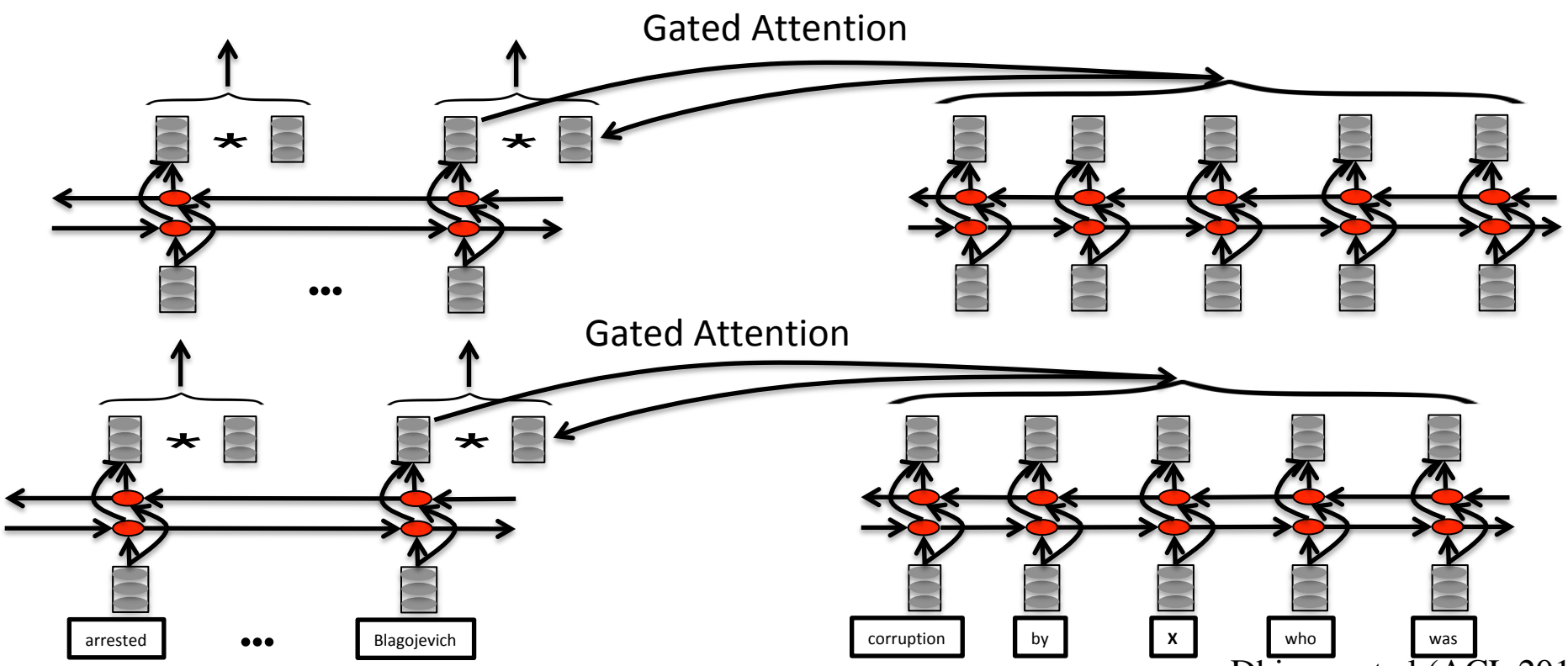


1. Find features in the query which match the contextual representation of the document
2. Gate the document representation by multiplying with these features

Dhingra et al (ACL 2017)

Multi Hop Architecture

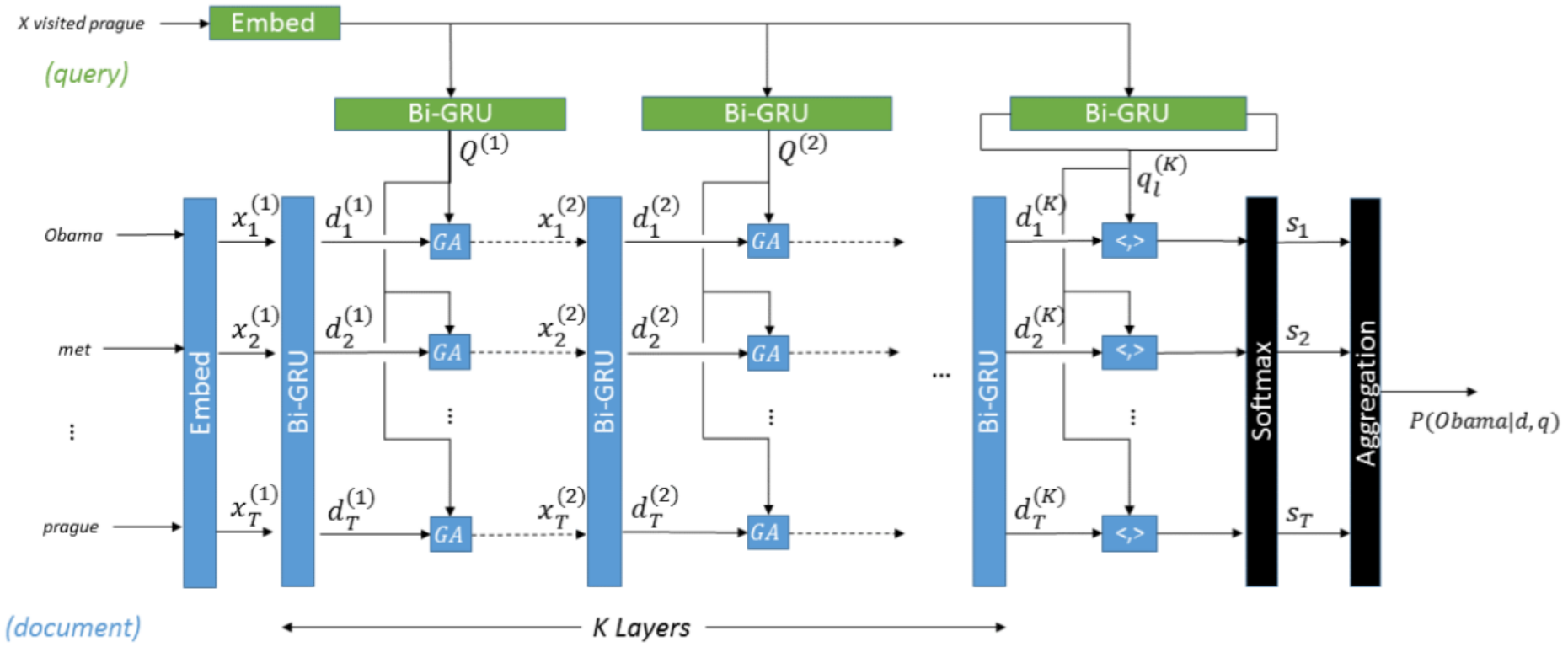
- Perform several passes over the document
  - Allow model to combine evidence from multiple sentences



Dhingra et al (ACL 2017)

Multi-hop Architecture

- Perform several passes over the document
  - Allow model to combine evidence from multiple sentences



Dhingra et al (ACL 2017)

Output Model

• Probability that a particular token in the document answers the query:

- Take an **inner product** between the query embedding and the output of the last layer:

$$s_i = \frac{\exp\left(\langle q^{(K)}, d_i^{(K)} \rangle\right)}{\sum_{i'} \exp\left(\langle q^{(K)}, d_{i'}^{(K)} \rangle\right)}, \quad i = 1, \ldots, |D|$$

• The probability of a particular candidate $c \in \mathcal{A}$ is then aggregated over all document tokens which appear in c:

$$P(c|d,q) \propto \sum_{i \in \mathbb{I}(c,d)} s_i$$

set of positions where a token in c appears in the document d.

Pointer Sum Attention of Kadlec et al., 2016

Output Model

• The probability of a particular candidate $c \in \mathcal{A}$ is then aggregated over all document tokens which appear in c:

$$P(c|d,q) \propto \sum_{i \in \mathbb{I}(c,d)} s_i$$

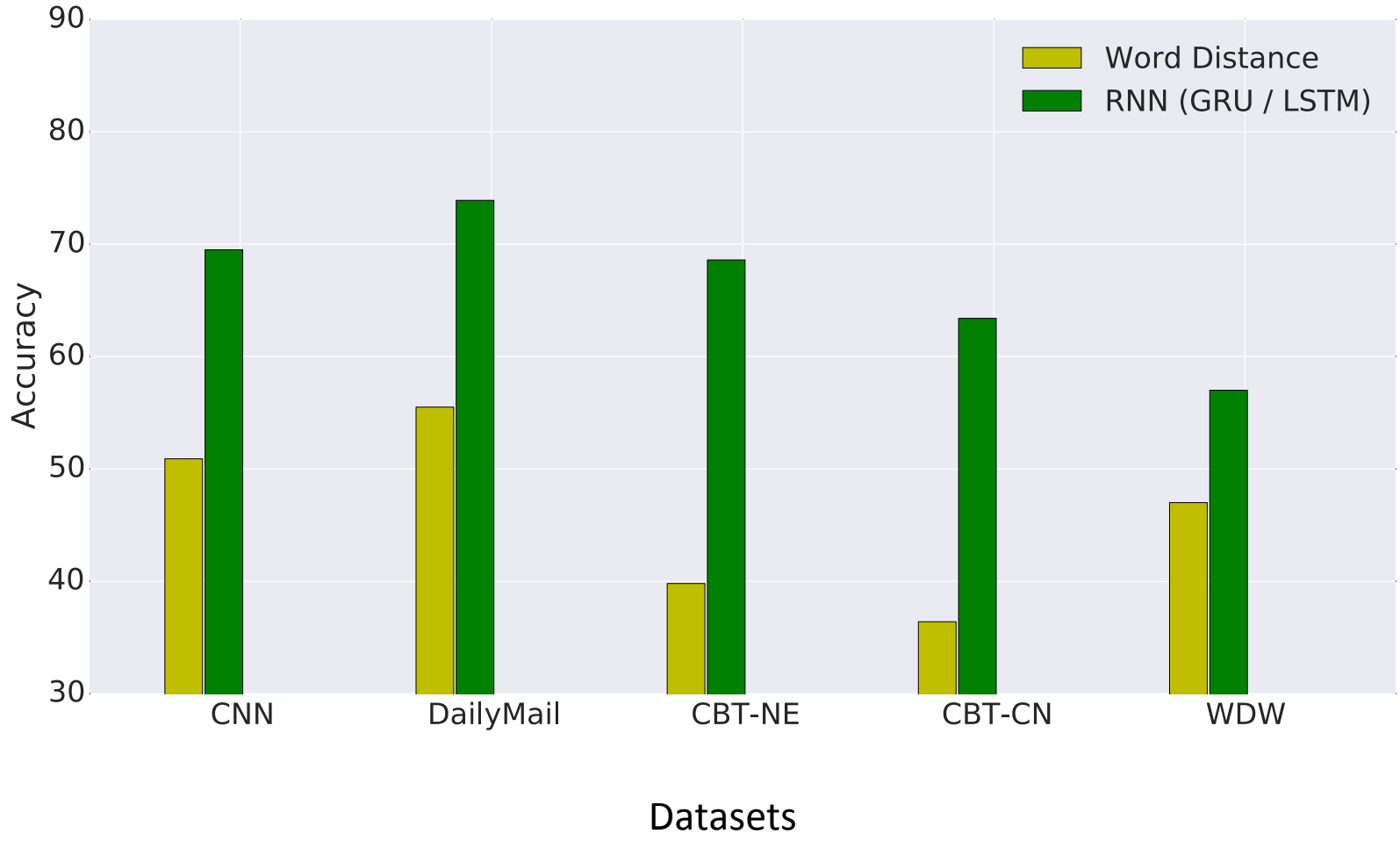
• The candidate with **maximum probability** is selected as the predicted answer:

$$a^* = \text{argmax}_{c \in \mathcal{C}} \; P(c|d,q)$$

• Use cross-entropy loss between the predicted probabilities and the true answers.

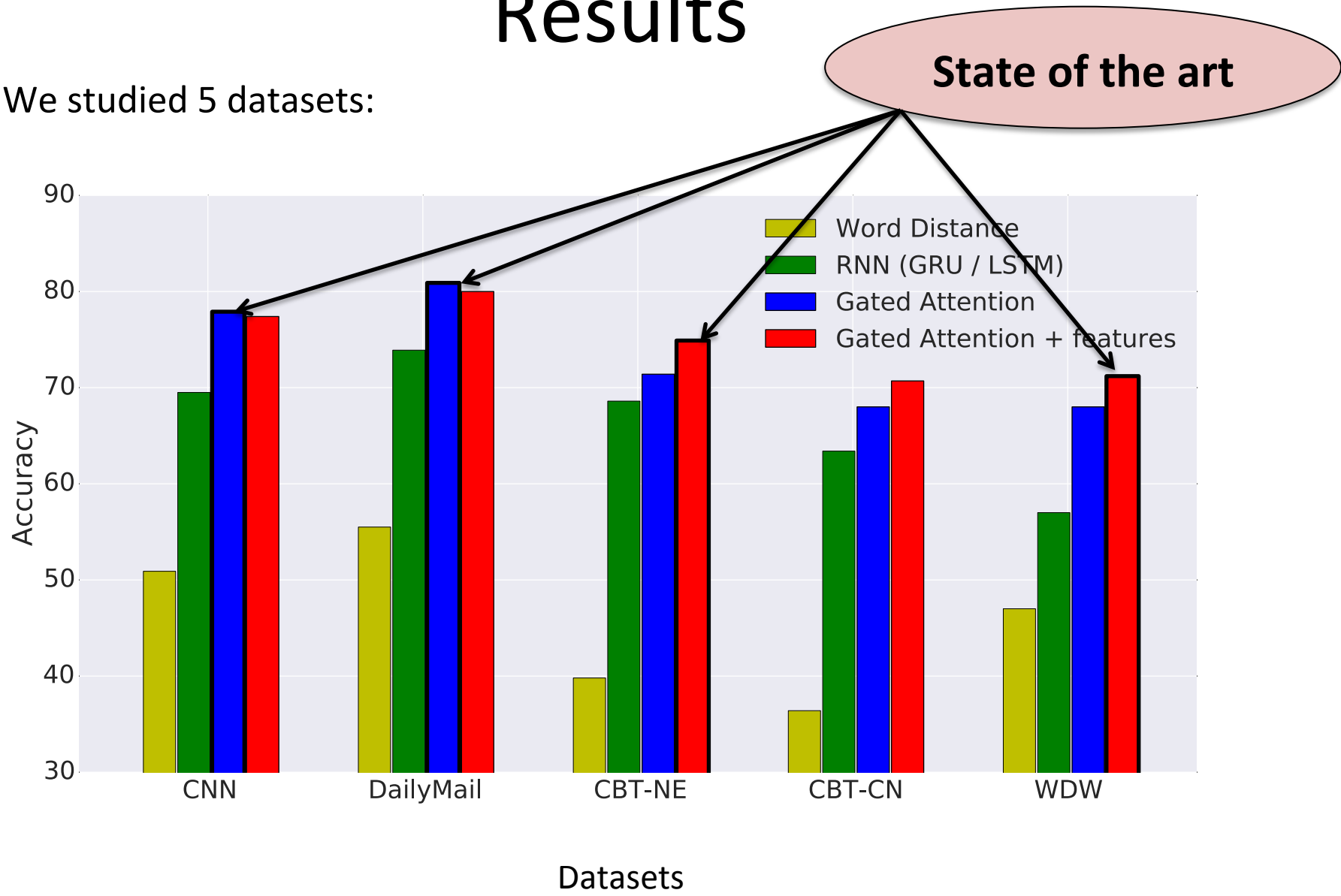
Results

We studied 5 datasets:



Results

We studied 5 datasets:



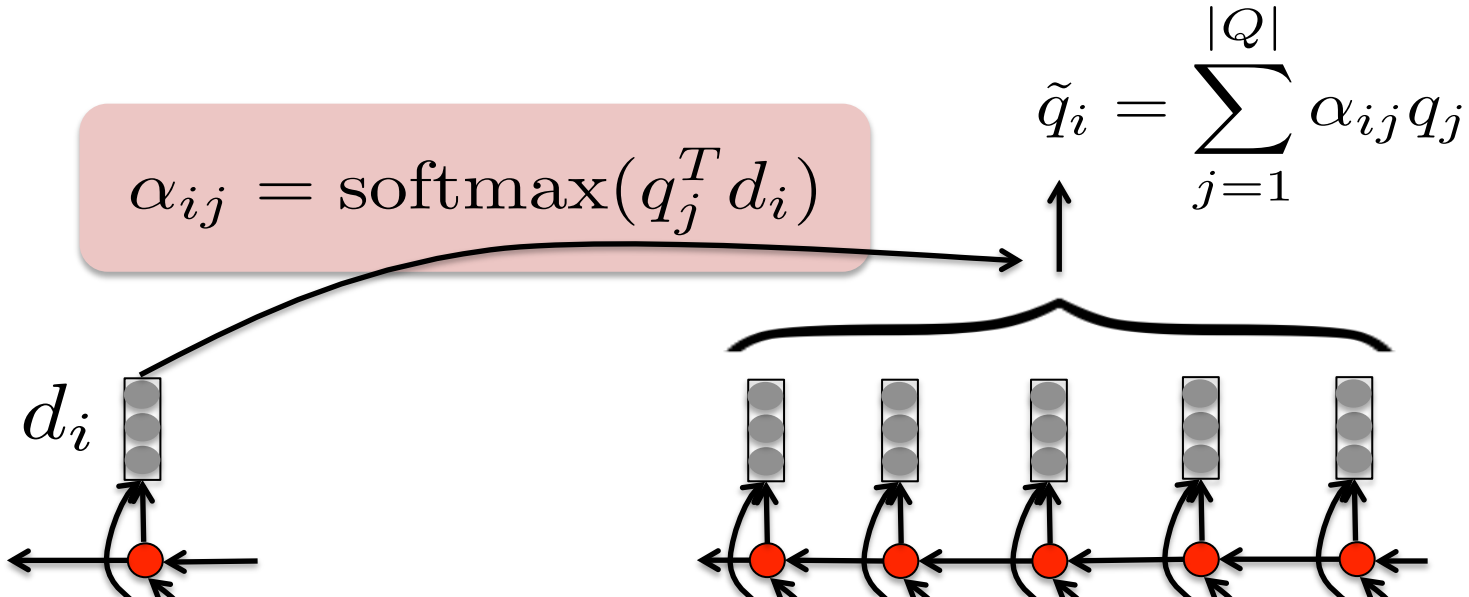
Dhingra et al (ACL 2017)

Model	CNN		Daily Mail		CBT-NE		CBT-CN	
	Val	Test	Val	Test	Val	Test	Val	Test
Humans (query) †	–	–	–	–	–	52.0	–	64.4
Humans (context + query) †	–	–	–	–	–	81.6	–	81.6
LSTMs (context + query) †	–	–	–	–	51.2	41.8	62.6	56.0
Deep LSTM Reader †	55.0	57.0	63.3	62.2	–	–	–	–
Attentive Reader †	61.6	63.0	70.5	69.0	–	–	–	–
Impatient Reader †	61.8	63.8	69.0	68.0	–	–	–	–
MemNets †	63.4	66.8	–	–	70.4	66.6	64.2	63.0
AS Reader †	68.6	69.5	75.0	73.9	73.8	68.6	68.8	63.4
DER Network †	71.3	72.9	–	–	–	–	–	–
Stanford AR (relabeling) †	73.8	73.6	77.6	76.6	–	–	–	–
Iterative Attentive Reader †	72.6	73.3	–	–	75.2	68.6	72.1	69.2
EpiReader †	73.4	74.0	–	–	75.3	69.7	71.5	67.4
AoA Reader †	73.1	74.4	–	–	77.8	72.0	72.2	69.4
ReasoNet †	72.9	74.7	77.6	76.6	–	–	–	–
NSE †	–	–	–	–	78.2	73.2	74.3	**71.9**
MemNets (ensemble) †	66.2	69.4	–	–	–	–	–	–
AS Reader (ensemble) †	73.9	75.4	78.7	77.7	76.2	71.0	71.1	68.9
Stanford AR (relabeling,ensemble) †	77.2	77.6	80.2	79.2	–	–	–	–
Iterative Attentive Reader (ensemble) †	75.2	76.1	–	–	76.9	72.0	74.1	71.0
EpiReader (ensemble) †	–	–	–	–	76.6	71.8	73.6	70.6
AS Reader (+BookTest) † ‡	–	–	–	–	80.5	76.2	83.2	80.8
AS Reader (+BookTest,ensemble) † ‡	–	–	–	–	*82.3*	*78.4*	*85.7*	*83.7*
GA--	73.0	73.8	76.7	75.7	74.9	69.0	69.0	63.9
GA (update $L(w)$)	**77.9**	**77.9**	**81.5**	**80.9**	76.7	70.1	69.8	67.3
GA (fix $L(w)$)	77.9	77.8	80.4	79.6	77.2	71.4	71.6	68.0
GA Reader (+feature, update $L(w)$)	77.3	76.9	80.7	80.0	77.2	73.3	73.0	69.8
GA Reader (+feature, fix $L(w)$)	76.7	77.4	80.0	79.3	**78.5**	**74.9**	**74.4**	70.7

Analysis of Attention

Document:
"…arrested Illinois governor **Rod Blagojevich** and his chief of staff John Harris on corruption charges … included **Blogojevich** allegedly conspiring to sell or trade the senate seat left vacant by President-elect Barack Obama…"

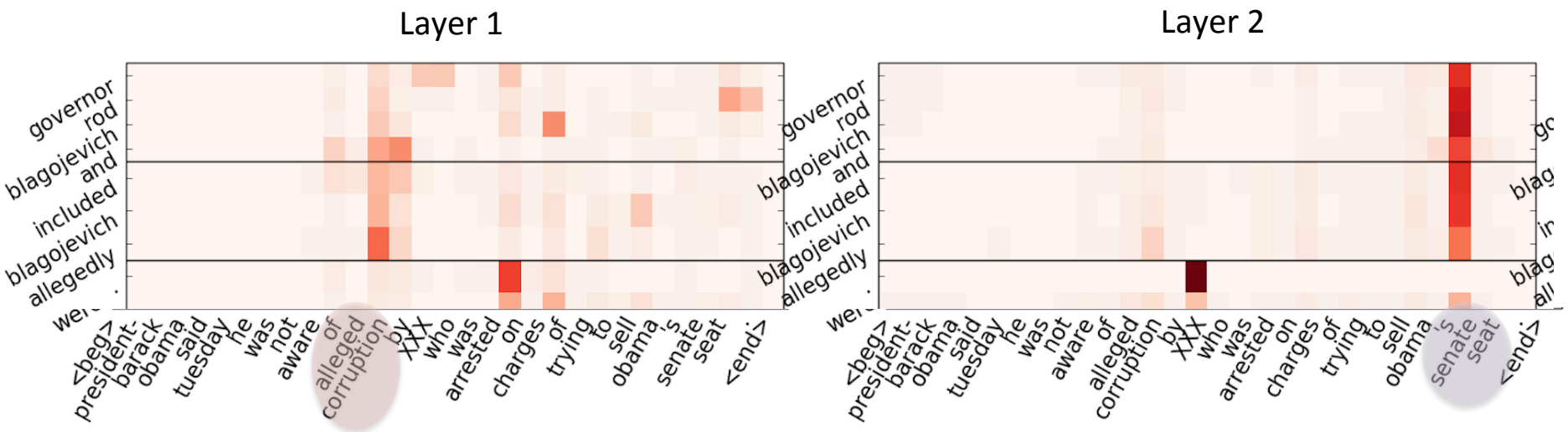
Query:
"President-elect Barack Obama said Tuesday he was not aware of alleged corruption by **X** who was arrested on charges of trying to sell Obama's senate seat."
Find **X**.



Analysis of Attention

Document:
"…arrested Illinois governor **Rod Blagojevich** and his chief of staff John Harris on corruption charges … included **Blogojevich** allegedly conspiring to sell or trade the senate seat left vacant by President-elect Barack Obama…"

Query:
"President-elect Barack Obama said Tuesday he was not aware of alleged corruption by **X** who was arrested on charges of trying to sell Obama's senate seat."
Find **X**.



Dhingra et al (ACL 2017)

Summary so far

- Multiplicative attention for document and query alignment

- Multiple layers allow model to focus on different salient aspects of the query

Code + Data: **https://github.com/bdhingra/ga-reader**

Words vs. Characters

- Word-level representations are good at learning the semantics of the tokens

- Character-level representations are more suitable for modeling sub-word morphologies ("cat" vs. "cats")

  - Word-level representations are obtained from a learned lookup table
  - Character-level representations are usually obtained by applying RNN or CNN

- Hybrid word-character models have been shown to be successful in various NLP tasks (Yang et al., 2016a, Miyamoto & Cho (2016), Ling et al., 2015)

- Commonly used method is to concatenate these two representations

Fine-Grained Gating

- Fine-grained gating mechanism:

$$\mathbf{h} = \mathbf{g} \odot \mathbf{c} + (1 - \mathbf{g}) \odot (\mathbf{E}\mathbf{w})$$

Character - level representation

Gating

Word- level representation

$$\mathbf{g} = \sigma(\mathbf{W}_g \mathbf{v} + \mathbf{b}_g)$$

Additional features: named entity tags, part- of-speech tags, document frequency vectors, word look-up representations

Yang et al, ICLR 2017

Children's Book Test (CBC) Dataset

Model	CN dev	CN test	NE dev	NE test
GA word char concat	0.731	0.696	0.768	0.725
GA word char feat concat	0.7250	0.6928	0.7815	0.7256
GA scalar gate	0.7240	0.6908	0.7810	0.7260
GA fine-grained gate	0.7425	0.7084	0.7890	0.7464
FG fine-grained gate	**0.7530**	**0.7204**	**0.7910**	**0.7496**
Sordoni et al. (2016)	0.721	0.692	0.752	0.686
Trischler et al. (2016)	0.715	0.674	0.753	0.697
Cui et al. (2016)	0.722	0.694	0.778	0.720
Munkhdalai & Yu (2016)	0.743	0.719	0.782	0.732
Kadlec et al. (2016) ensemble	0.711	0.689	0.762	0.710
Sordoni et al. (2016) ensemble	0.741	0.710	0.769	0.720
Trischler et al. (2016) ensemble	0.736	0.706	0.766	0.718

Words vs. Characters

- High gate values: character-level representations
- Low gate values: word-level representations.

Gate values	Word tokens
Lowest	or but But These these However however among Among that when When although Although because Because until many Many than though Though this This Since since date where Where have That and And Such such number so which by By how before Before with With between Between even Even if
Highest	Sweetgum Untersee Jianlong Floresta Chlorella Obersee PhT Doctorin Jumonville WFTS WTSP Boven Pharm Nederrijn Otrar Rhin Magicicada WBKB Tanzler KMBC WPLG Mainau Merwede RMJM Kleitman Scheur Bodensee Kromme Horenbout Vorderrhein Chlamydomonas Scantlebury Qingshui Funchess

Talk Roadmap

- Multiplicative and Fine-grained Attention
- Linguistic Knowledge as Explicit Memory for RNNs

Broad-Context Language Modeling

Her plain face broke into a huge smile when she saw Terry.
“Terry!” she called out.
She rushed to meet him and they embraced.
“Hon, I want you to meet an old friend, Owen McKenna.
Owen, please meet Emily."
She gave me a quick nod and turned back to **X**

Task: Find **X**.

*(**X** can be any word in the vocabulary)*

Passages are filtered such that humans can correctly answer when given the whole context but not when given only the last sentence.

LAMBADA dataset, Paperno et al., ACL 2016

Broad-Context Language Modeling (recast as Reading Comprehension)

Document:
Her plain face broke into a huge smile when she saw Terry.
“Terry!” she called out.
She rushed to meet him and they embraced.
“Hon, I want you to meet an old friend, Owen McKenna.
Owen, please meet Emily."

Query:
She gave me a quick nod and turned back to **X**

Find **X**.
*(Now **X** is assumed to be in the document)*

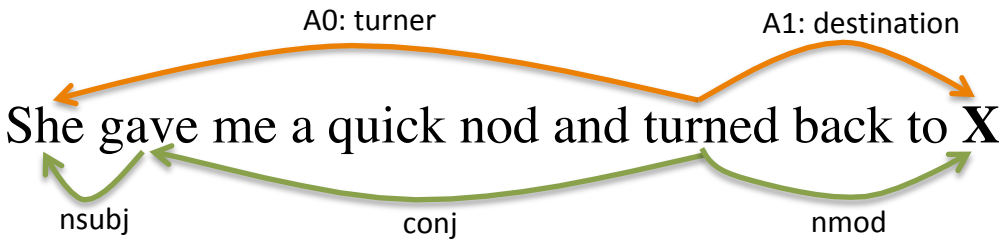
- Reading Comprehension approaches perform best on this task

Chu et al, EACL 2017

Text Phenomena

Document:
Her plain face broke into a huge smile when she saw Terry.
“Terry!” she called out.
She rushed to meet him and they embraced.
“Hon, I want you to meet an old friend, Owen McKenna.
Owen, please meet Emily."

Query:



Find **X**.

Syntactic Dependency

Semantic Dependency

Text Phenomena

Document:
Her plain face broke into a huge smile when she saw Terry.
"Terry!" she called out.
She rushed to meet him and they embraced.
"Hon, I want you to meet an old friend, Owen McKenna.
Owen, please meet Emily."

Query:

She gave me a quick nod and turned back to **X**

X = Terry

Find **X**.

Syntactic Dependency

Semantic Dependency

Coreference

Architectural Bias

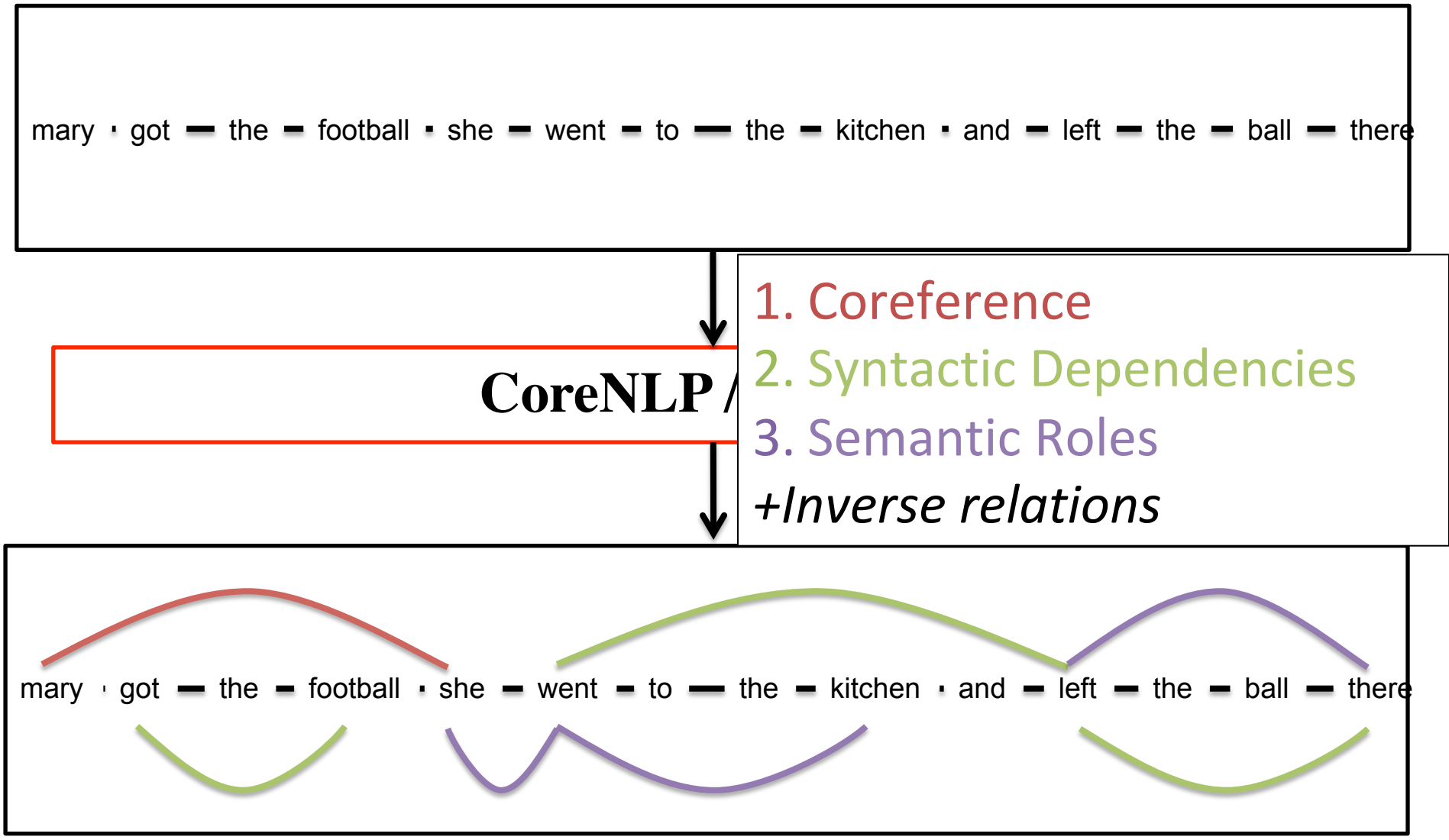
1. Dependent / Coreferent mentions may be far apart in the document
   - RNNs (including GRU / LSTM) tend to "forget" long-term interactions

*Use **memory-augmented** architecture.*

2. Off-the-shelf NLP tools available which extract these dependencies
   - Example: Stanford CoreNLP

*Use **linguistic annotations** to guide memory propagation in the network.*

Linguistic Knowledge

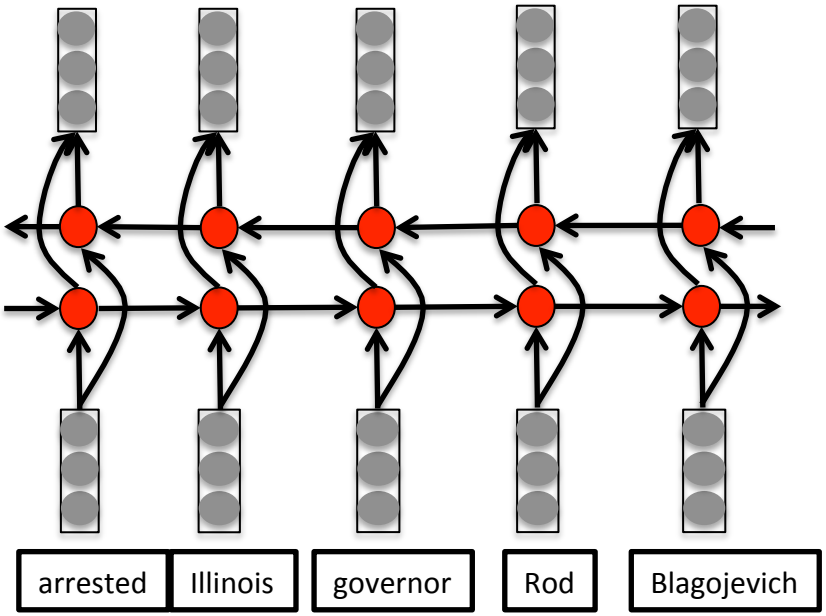


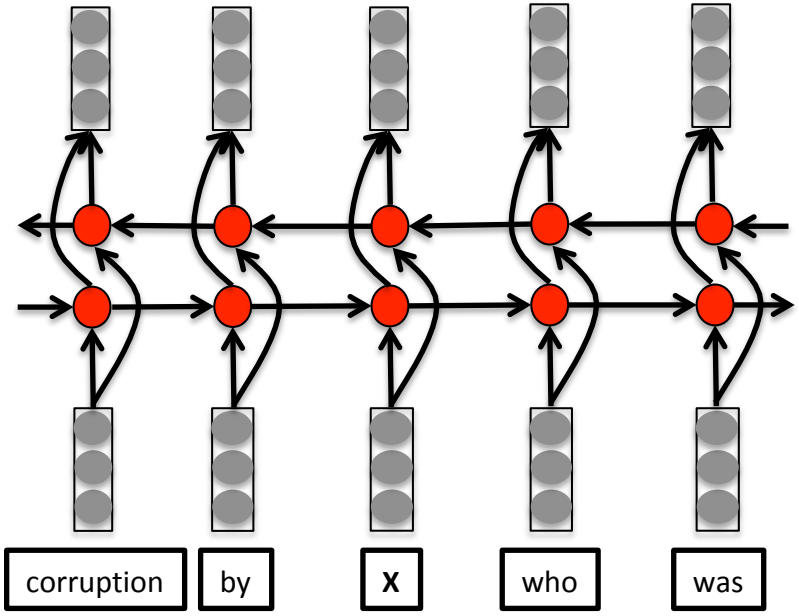
Representing Document / Query

- Both document and query are represented as matrices

$$D \in \mathbb{R}^{2h \times |D|}$$

$$Q \in \mathbb{R}^{2h \times |Q|}$$





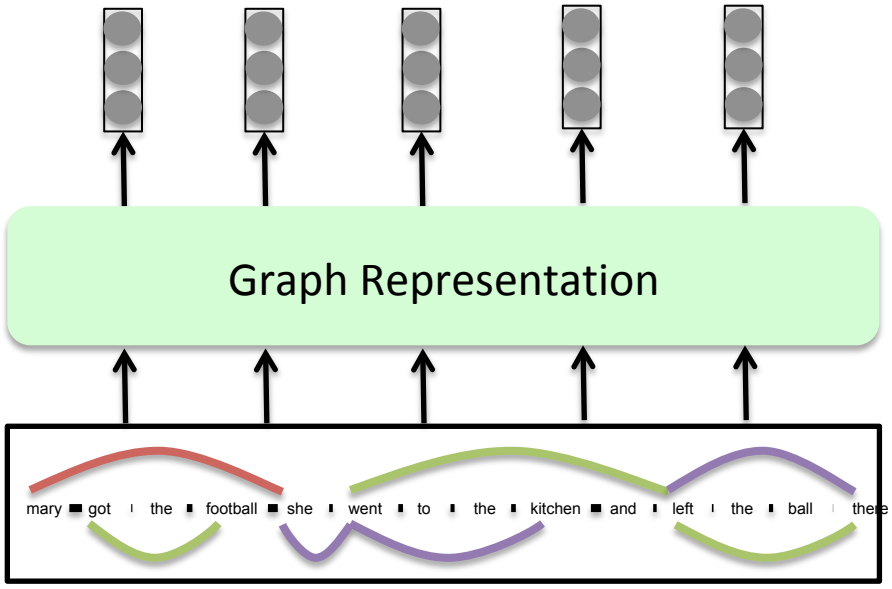
h – State size of each GRU

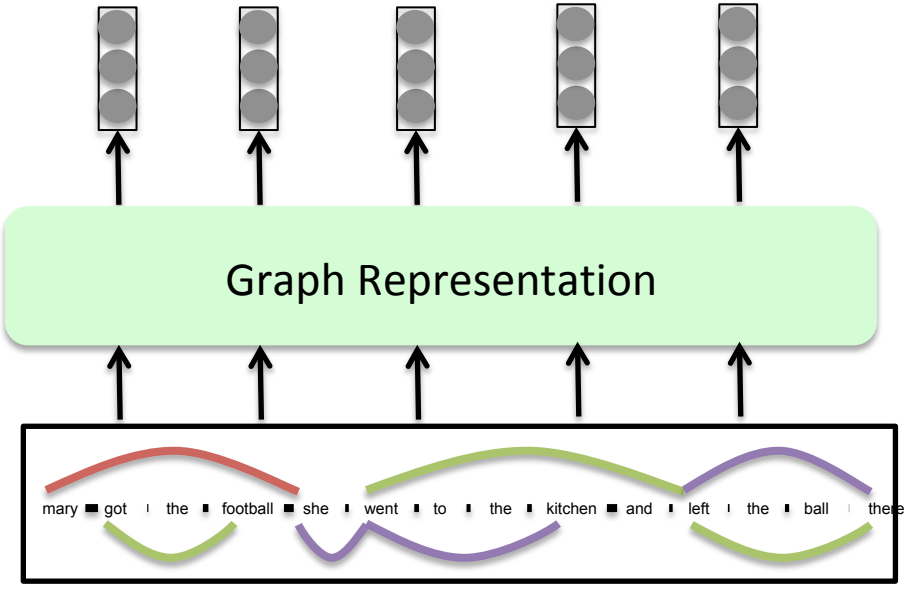
Representing Document / Query **Graphs**

- Both document and query are represented as matrices

$$D \in \mathbb{R}^{2h \times |D|}$$

$$Q \in \mathbb{R}^{2h \times |Q|}$$

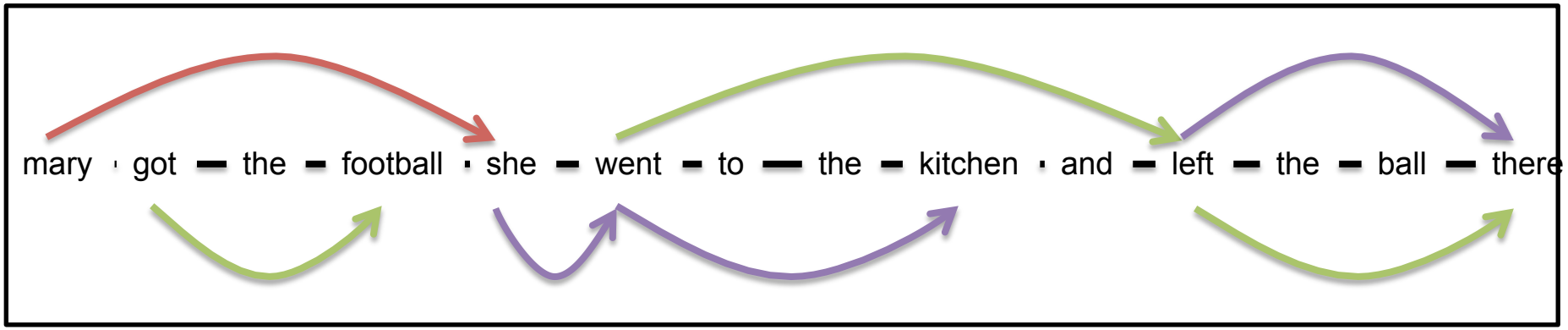




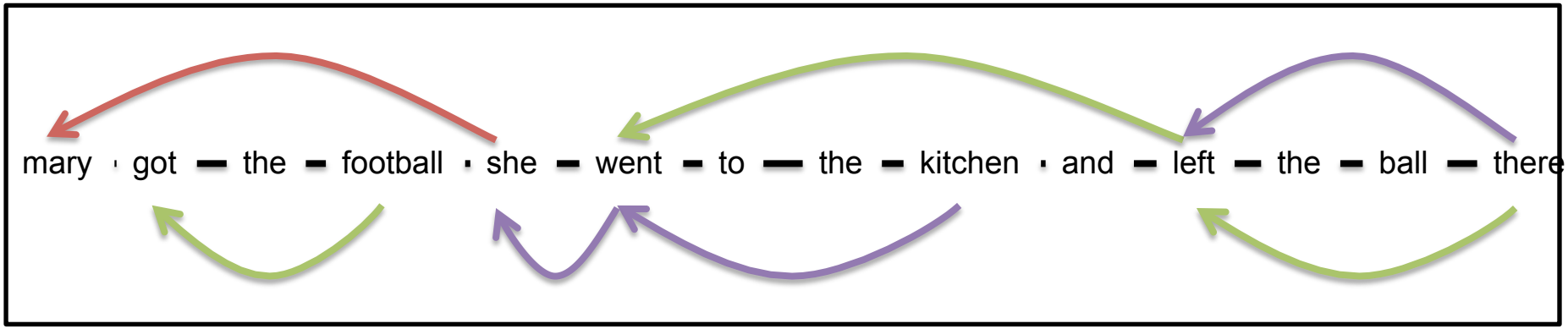
Forward / Backward DAGs

- Topological order given by the sequence itself

Forward DAG

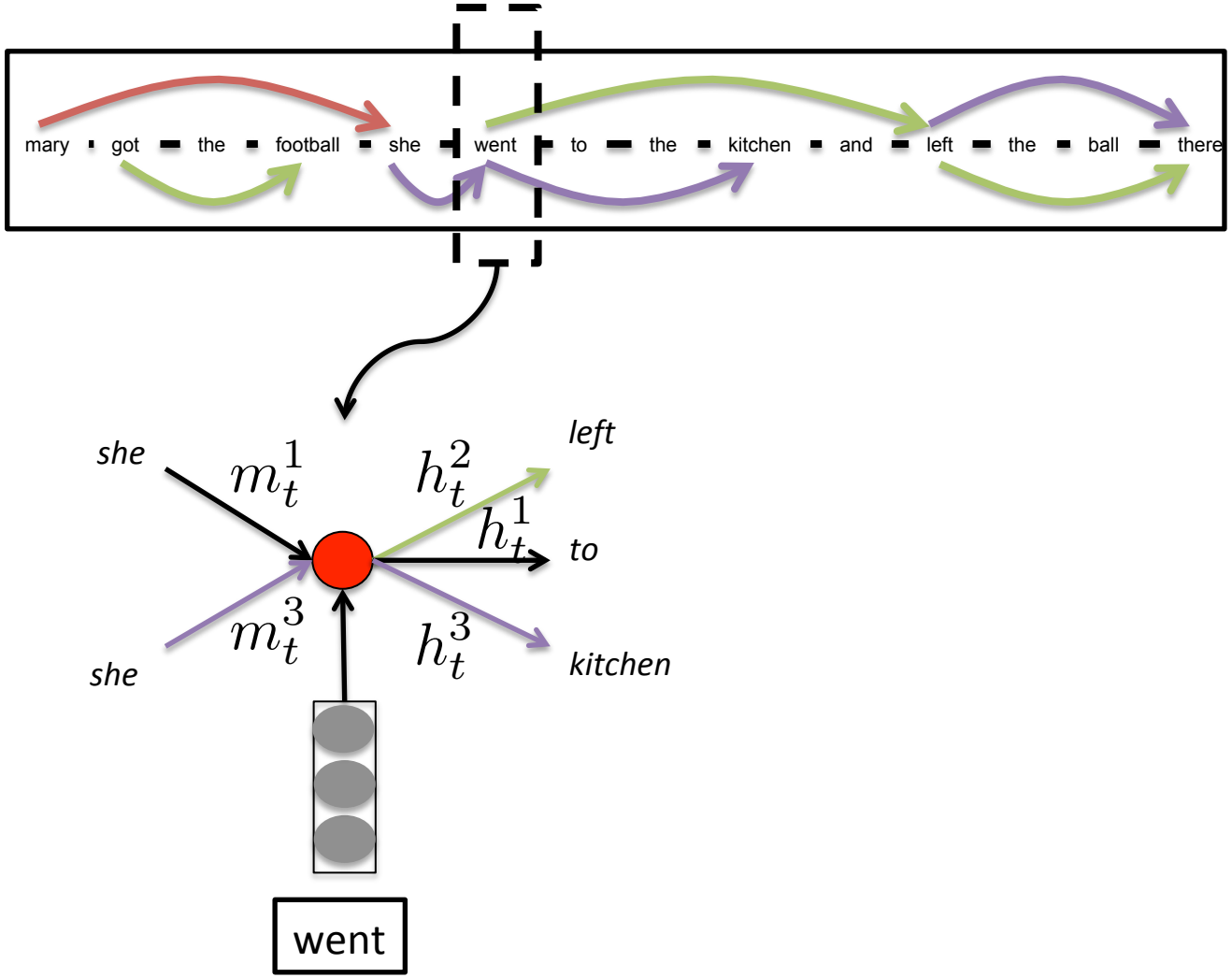


Backward DAG

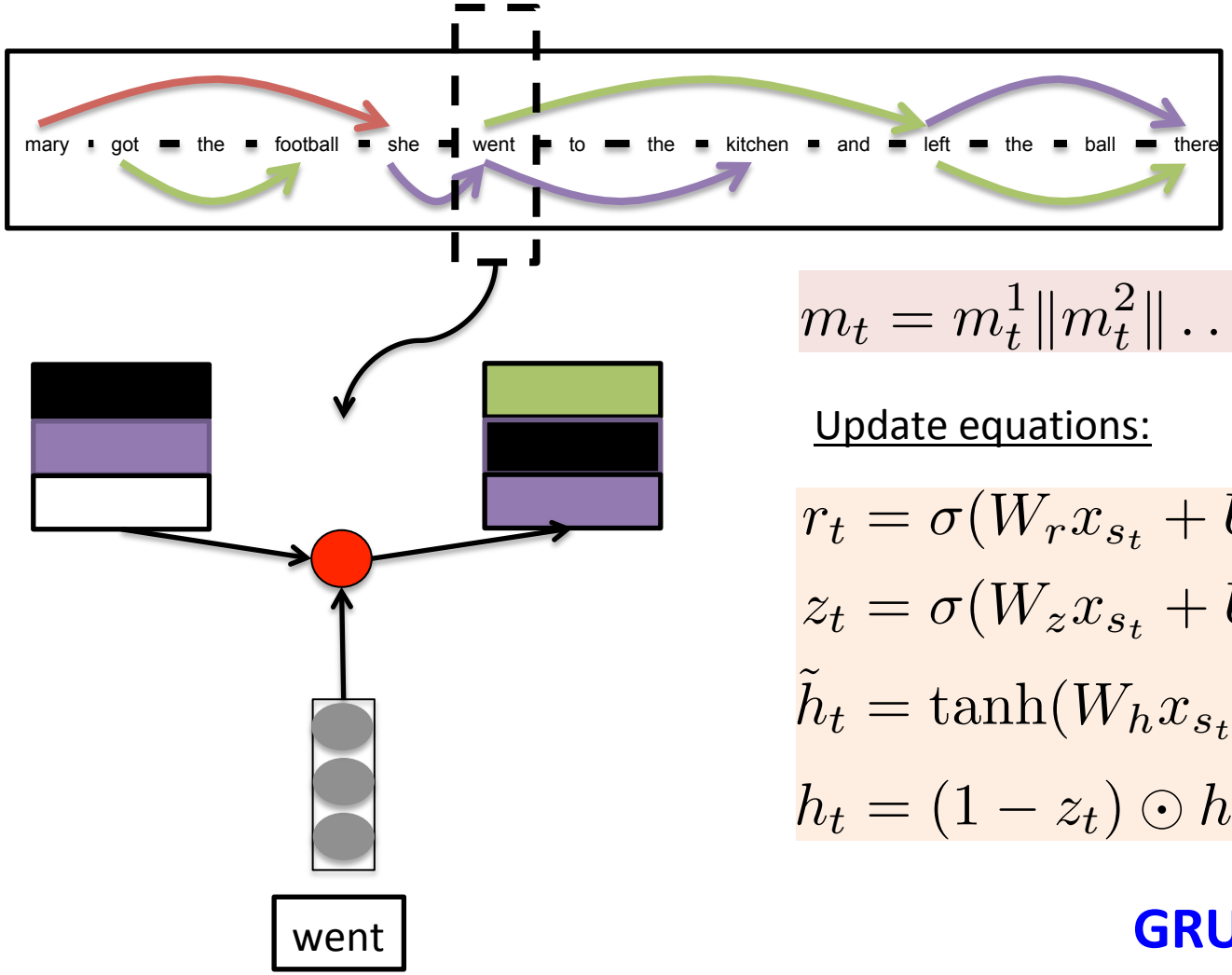


*Peng et al (TACL, 2017) use similar decomposition for Relation Extraction

Memory as Acyclic Graph Encoding (MAGE) RNN



Memory as Acyclic Graph Encoding (MAGE) RNN



$$m_t = m_t^1 \| m_t^2 \| \ldots \| m_t^{|E_f|},$$ **Memory**

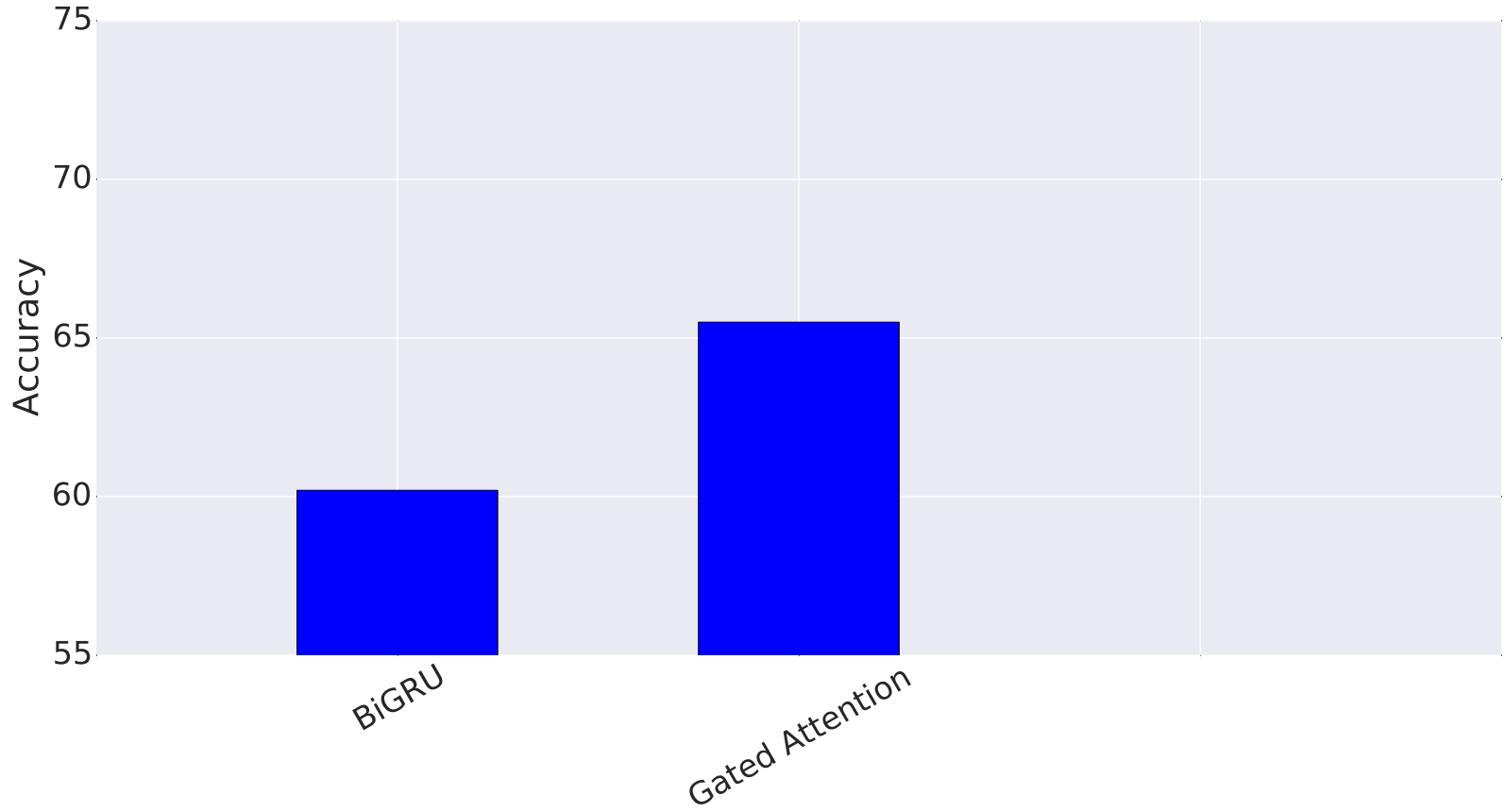
Update equations:

$$r_t = \sigma(W_r x_{s_t} + U_r m_t + b_r),$$
$$z_t = \sigma(W_z x_{s_t} + U_z m_t + b_z),$$
$$\tilde{h}_t = \tanh(W_h x_{s_t} + r_t \odot U_h m_t + b_h),$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t,$$

GRU update

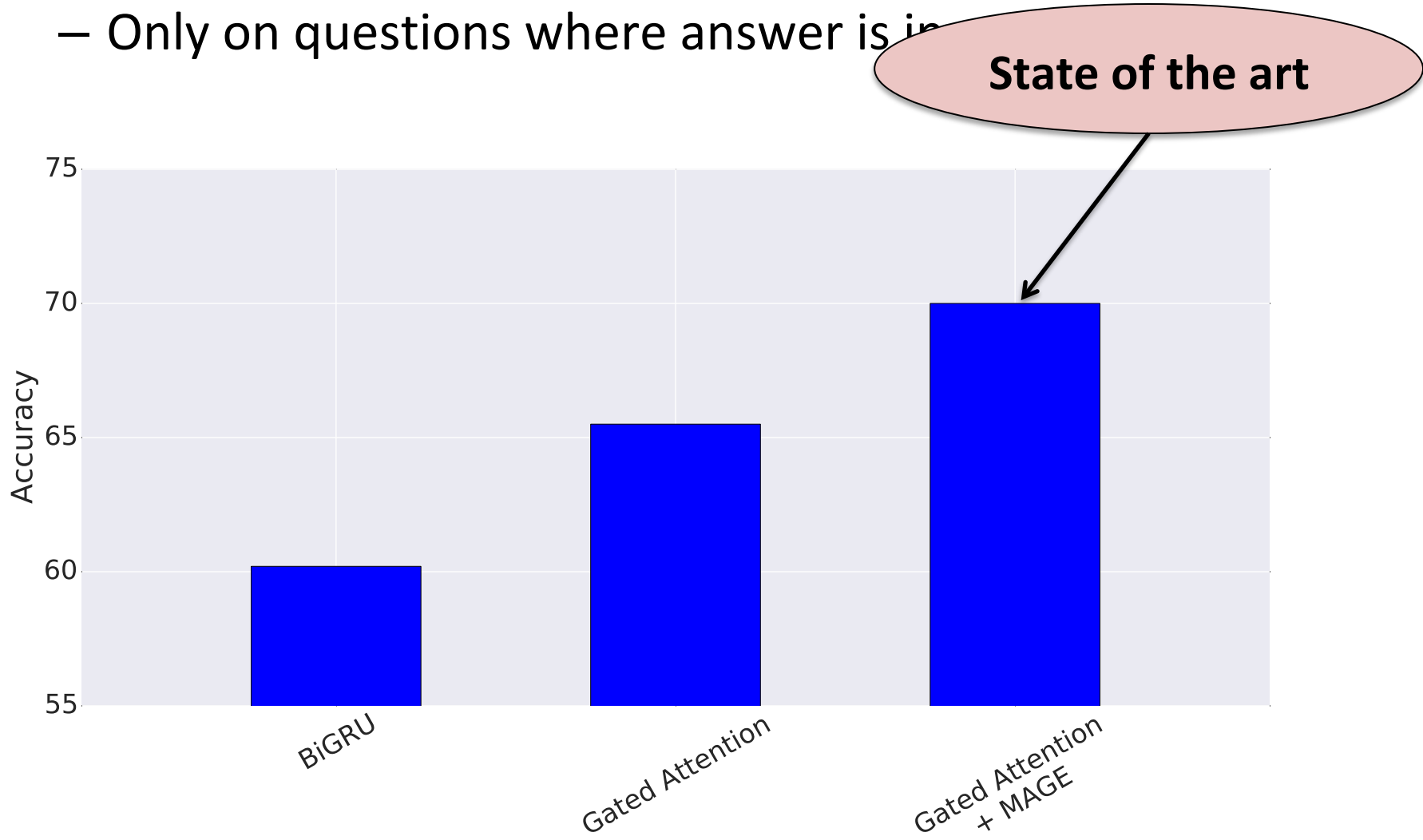
Performance

- LAMBADA dataset
  - Only on questions where answer is in context



Performance

- LAMBADA dataset
  - Only on questions where answer is in



Performance

- Facebook bAbI dataset
  - Task 3 (1000 training examples)

Document:
mary travelled to the kitchen . daniel got the milk . he went to the kitchen **. john picked up the apple** . mary went to the hallway . **john moved to the bedroom .** sandra journeyed to the kitchen . daniel went back to the bedroom . **john went to the office .** daniel put down the milk . he picked up the milk . sandra went to the bedroom . john dropped the apple . mary went back to the bathroom .

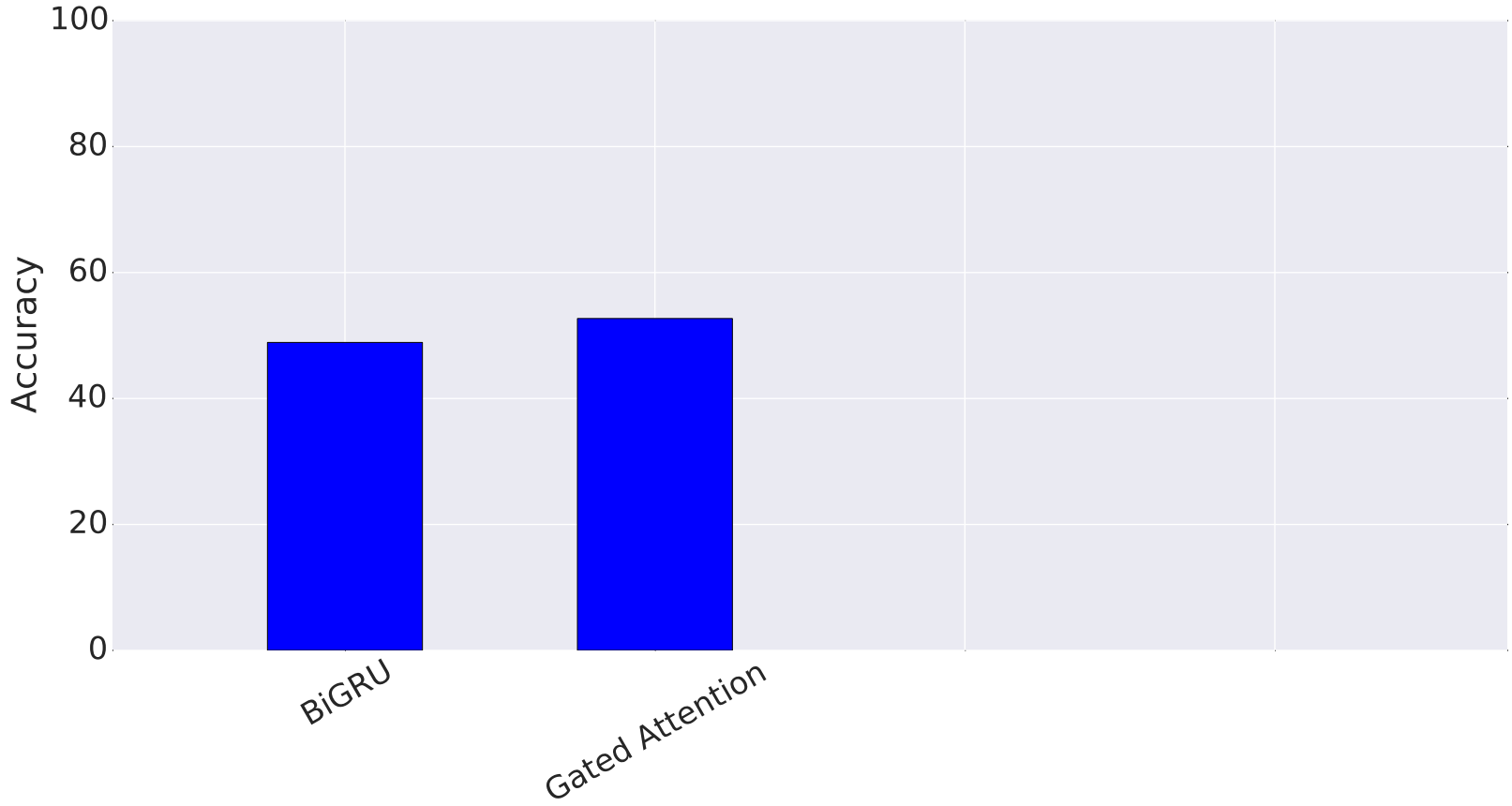
Query:
where was the apple before the office ?

Answer:
bedroom

Only coreference annotations extracted for this dataset

Performance

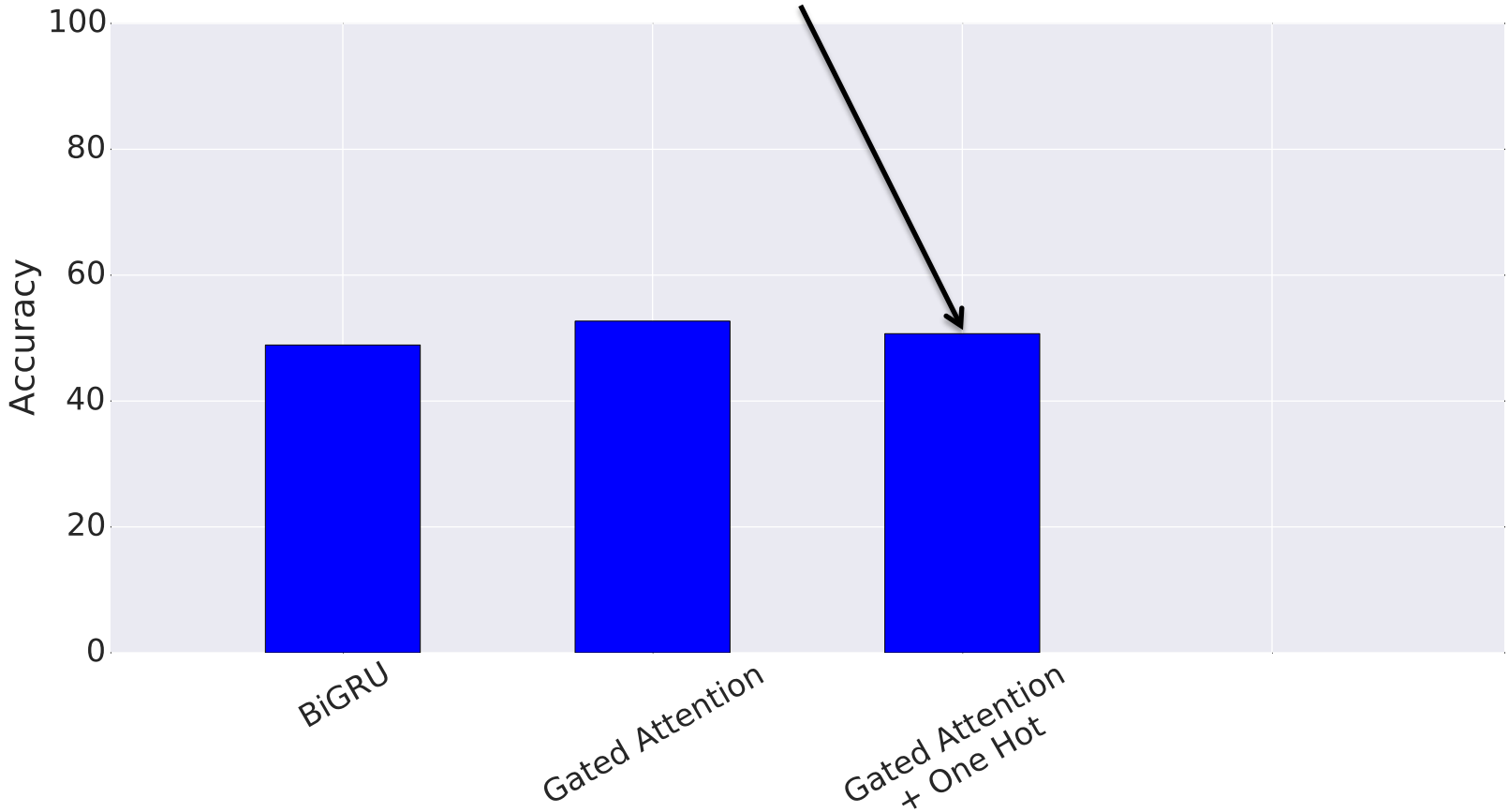
- Facebook bAbI dataset
  - Task 3



Performance

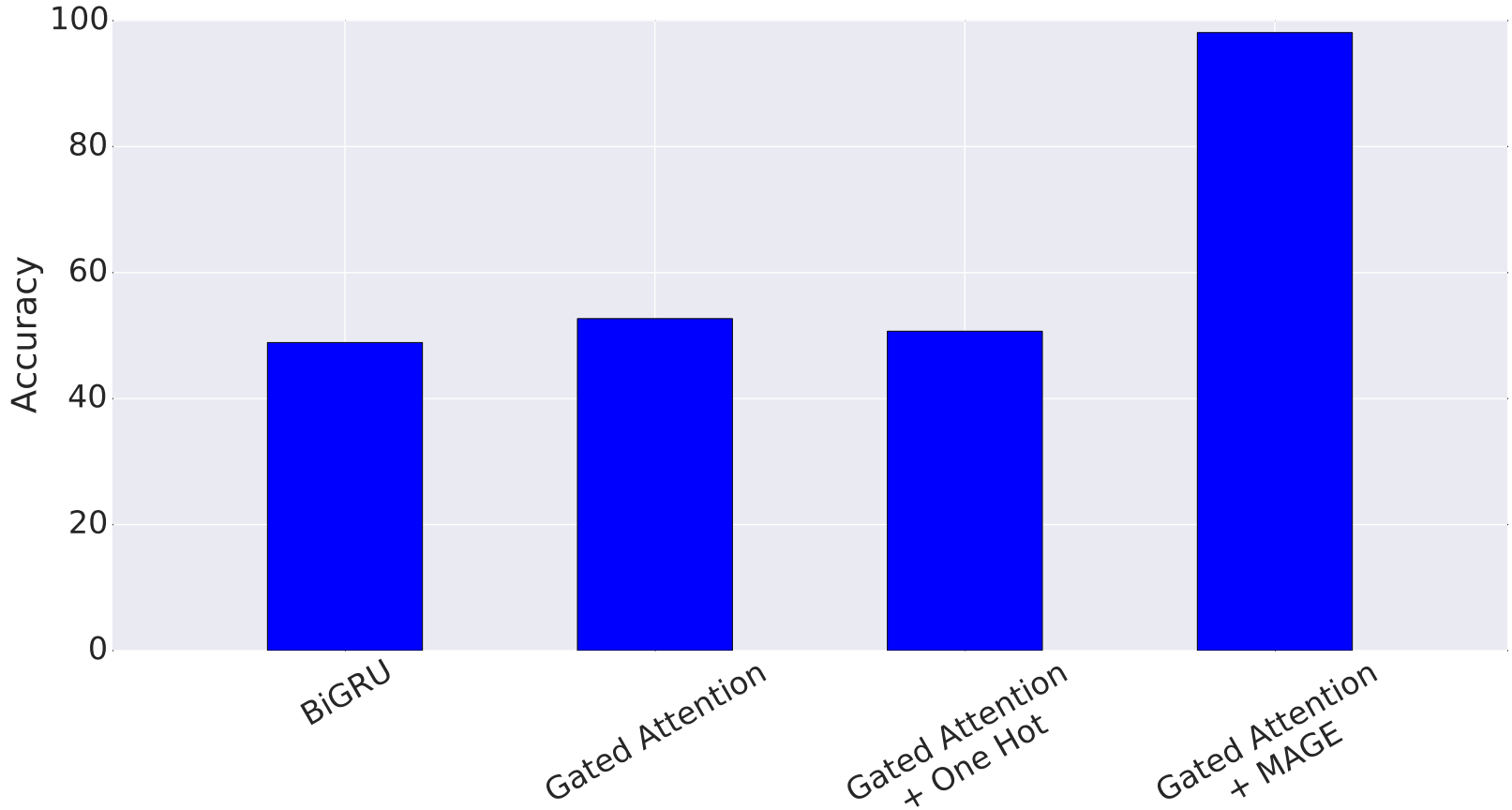
- Facebook bAbI dataset
  - Task 3

One hot indicator feature for entity mention appended to input



Performance

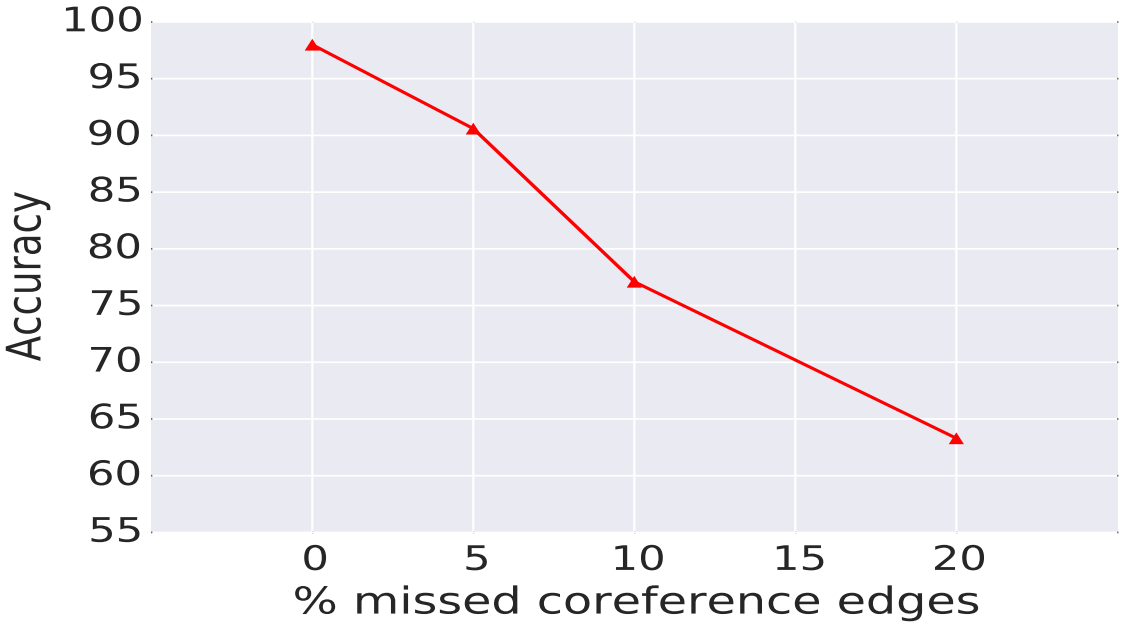
- Facebook bAbI dataset
  - Task 3



Summary so far

- Linguistic annotations can help guide memory propagation in RNNs
  - Better modeling of long term dependencies

Annotator Noise



- Performance drops sharply with added noise in linguistic annotations
- Can we **jointly train** annotation and reading comprehension models?
  - Can we learn task-specific knowledge?

“Common” Sense?

Document:
“Eurythmics were a British music duo consisting of members **Annie Lennox** and David A. Stewart”

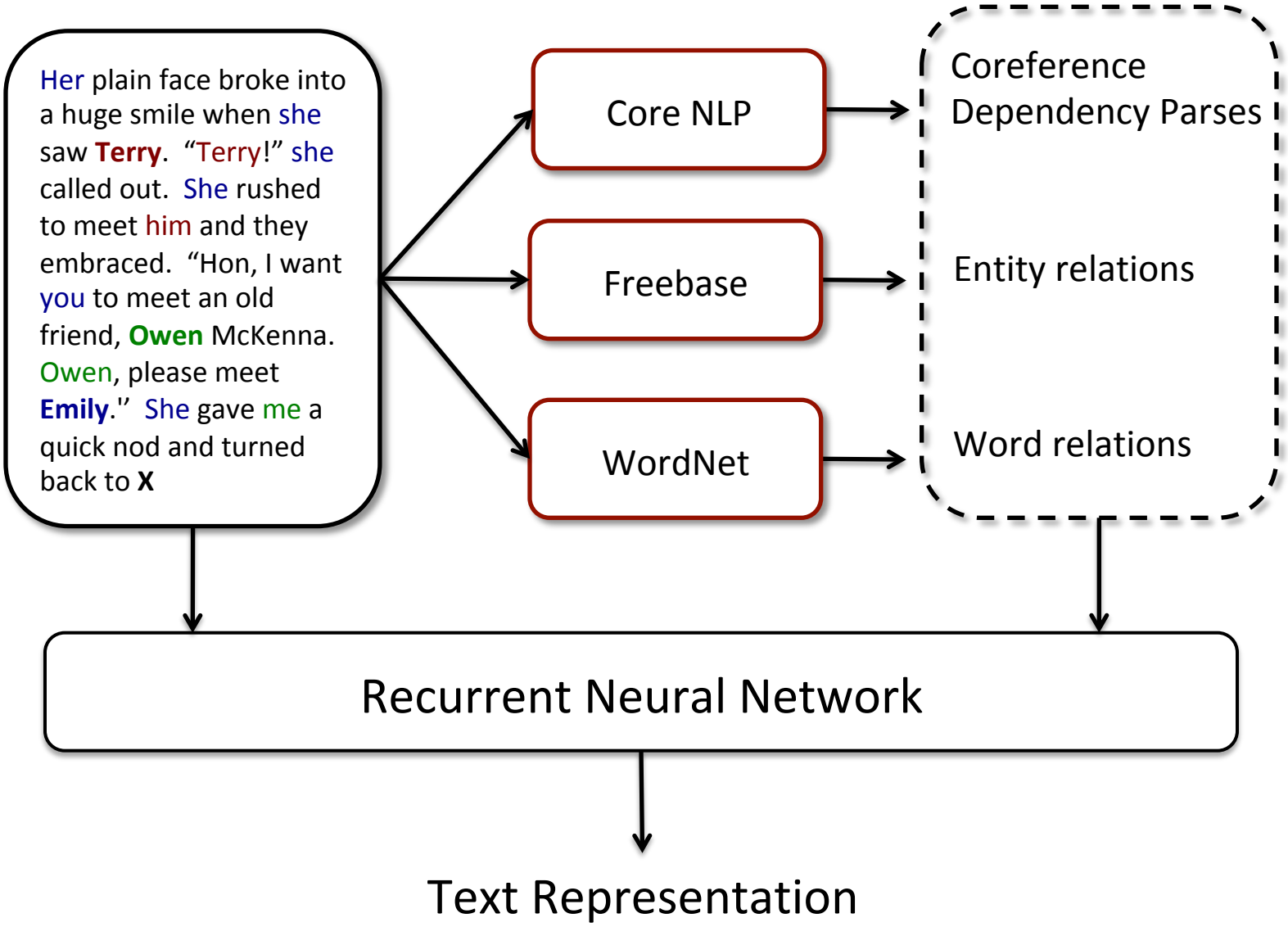
Query:
“Who was the female member of the 1980’s pop music duo Eurythmics?”

Answer:
Annie Lennox

- Where can we find such knowledge?
- Need more datasets for testing this aspect too.

TriviaQA Dataset (Joshi et al, ACL 2017)

Incorporating Prior Knowledge



Search-and-Read for open domain QA

- RC assumes passage containing answer is already known
- For real QA we need to search for the passage as well

7-Eleven stores were temporarily converted into Kwik E-marts to promote the release of what movie?

Corpus

Retrieval

*In July 2007 , 7-Eleven redesigned some stores to look like Kwik-E-Marts in select cities to promote **The Simpsons Movie.***

Tie-in promotions were made with several companies, including 7-Eleven , which transformed selected stores into Kwik-E-Marts .

*7-Eleven Becomes Kwik-E-Mart for **Simpsons Movie** Promotion*

Excerpts

QUASAR Dataset (Dhingra, Mazaitis, Cohen, *in submission*)

Thank you