Saarland Informatics Campus

UNIVERSITÄT DES SAARLANDES

max planck institut informatik

Question Answering Systems

Reading Comprehension and Open-domain QA

Rishiraj Saha Roy
Max Planck Institute for Informatics, Germany

8

Saarland University, Summer Semester 2020

Question of the day

How do we (generally) design question answering sytems over text (today)?

You’ll find this covered in

- 1 Reading Wikipedia to Answer Open-Domain Questions
  - Chen et al. DrQA
  - ACL 2017
  - https://www.aclweb.org/anthology/P17-1171.pdf
- 2 Simple and Effective Multi-Paragraph Reading Comprehension
  - Clark and Garner Document QA
  - ACL 2018
  - https://www.aclweb.org/anthology/P18-1078.pdf

Research paper 1

Reading Wikipedia to Answer Open-Domain Questions

Reading wikipedia to answer open-domain questions [PDF] arxiv.org
D Chen, A Fisch, J Weston, A Bordes - arXiv preprint arXiv:1704.00051, 2017 - arxiv.org

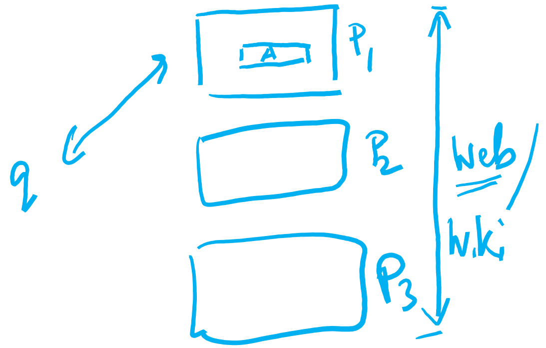
This paper proposes to tackle open-domain question answering using Wikipedia as the unique knowledge source: the answer to any factoid question is a text span in a Wikipedia article. This task of machine reading at scale combines the challenges of document retrieval (finding the relevant articles) with that of machine comprehension of text (identifying the answer spans from those articles). Our approach combines a search component based on bigram hashing and TF-IDF matching with a multi-layer recurrent neural network model …

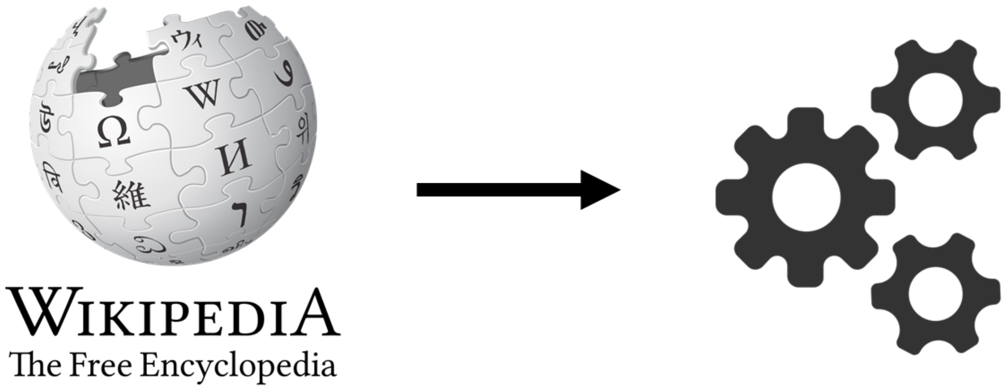
Cited by 615 Related articles All 20 versions

Open-domain question answering

Before DrQA*

Can we **scale up** neural reading comprehension systems and **generalize** them to open-domain?





from where do we get these P_i's?

AI/NLP → IR

DrQA

(Chen et al, 2017)

An open-domain QA system which uses the full English Wikipedia as the knowledge source

Thanks to Danqi Chen for the slide

DrQA: An open-domain QA system

MRC

P

Q: How many of Warsaw's inhabitants spoke Polish in 1933?

1

Document Retriever

Article Talk Read Edit View history

Warsaw

From Wikipedia, the free encyclopedia

This article is about the Polish capital. For other uses, see Warsaw (disambiguation).
"Warszawa" redirects here. For other uses, see Warszawa (disambiguation).
"City of Warsaw" redirects here. For the Second World War fighter squadron, see No. 316 Polish Fighter Squadron. F
1934, see Adamowicz brothers.

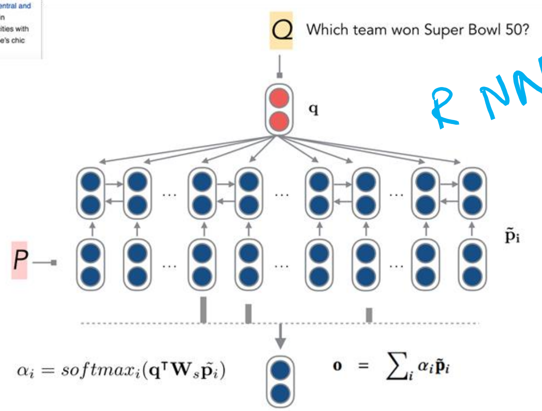
Warsaw (Polish: *Warszawa* [varˈʂava] (listen); see also other names) is the capital and largest city of Poland. It stands on the Vistula River in east-central Poland, roughly 260 kilometres (160 mi) from the Baltic Sea and 300 kilometres (190 mi) from the Carpathian Mountains. Its population is estimated at 1.750 million residents within a greater metropolitan area of 3.105 million residents, which makes Warsaw the 9th most-populous capital city in the European Union.[2][3][4] The city limits cover 516.9 square kilometres (199.6 sq mi), while the metropolitan area covers 6,100.43 square kilometres (2,355.39 sq mi).[5]

In 2012 the Economist Intelligence Unit ranked Warsaw as the 32nd most liveable city in the world.[6] It was also ranked as one of the most liveable cities in Central Europe. Today Warsaw is considered an "Alpha–" global city, a major international tourist destination and a significant cultural, political and economic hub.[7][8][9] Warsaw's economy, by a wide variety of industries, is characterised by FMCG manufacturing, metal processing, steel and electronic manufacturing and food processing. The city is a significant centre of research and development, BPO, ITO, as well as of the Polish media industry. The Warsaw Stock Exchange is one of the largest and most important in Central and Eastern Europe.[10] Frontex, the European Union agency for external border security, has its headquarters in Warsaw. It has been said that Warsaw, together with Frankfurt, London, Paris and Barcelona is one of the cities with the highest number of skyscrapers in the European Union.[11] Warsaw has also been called "Eastern Europe's chic cultural capital with thriving art and club scenes and serious restaurants".[12]

2

Document Reader

833,500



RNN

DrQA = Information Retrieval +

Thanks to Danqi Chen for the slide

DrQA: What's new?

Machine Reading Comprehension

MRC/MC/RC

- Before DrQA*: MRC ← Text QA
- MRC is cool ☻ AI/NLU
  - But how effective is it?
  - Can we make the goal more realistic / general?

oc dried

machine reading@scale

(in practice)

- DrQA: IR (DR) + MRC = MRS ➔ Open domain QA

retrieval ranking over ans spans

Web

redundancy of evidence

aggregate

- Previous open systems: Watson, YodaQA, QuASE, AskMSR

Web DeepMind

stricter

- Learning from Wikipedia: Low-redundancy (analogous to KGs)

– QA

DrQA: In a nutshell

- Document retriever IR
  - Bigram hashing and TF-IDF scoring
- Document reader
  - Multi-layer RNN

Main

- Retriever outperforms Wikipedia search (ElasticSearch)
- Reader comparable to state-of-the-art on SQuAD RC
- Why was it so successful? popular Visit: https://github.com/facebookresearch/DrQA

Document retriever

- Unsupervised approach
- Inverted index lookup
- Local word order with n-gram features
- Use bigram counts
- Hashing to map bigrams to 2^{24} bins with murmur3 hash [Recall Bloom filters? ☺]

doc

Inv. t_i

d_1

$q = t_1 \; t_3 \rightarrow t_3$

2-gram

q

doc 1

doc 2

who is the wife of donald trump?

bigram

hash

hit docs

10^{10}

2^{24}

index at bigram level

unique bigram : too high!!

100000

$2^{24} \ll 10^{10}$

– # unigrams $\simeq 10^5$

– # bigrams $\Rightarrow 10^{10} \gg 10^5$

Document reader

- Document retriever returns top-5 (articles ~~paragraphs~~) from corpus (Wikipedia, docs → passages)
1 - Paragraph encoding
2 - Question encoding
3 - Prediction of answer

ans

W

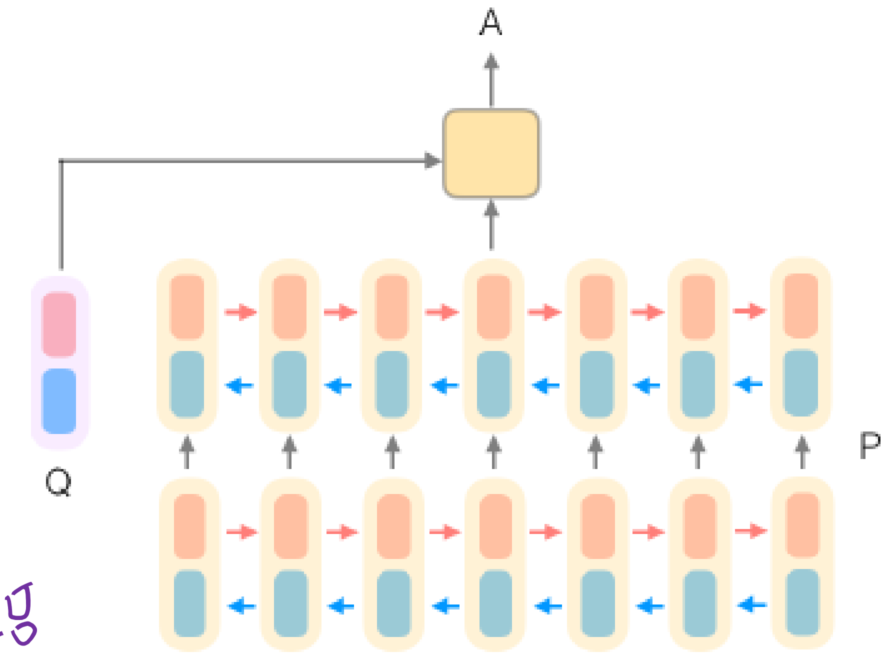
DocRet →

doc1

doc2

doc3

5



Reader: Paragraph encoding

- Uses sequence models
- Recurrent neural networks

  RNN family

- Specifically, bi-LSTMs

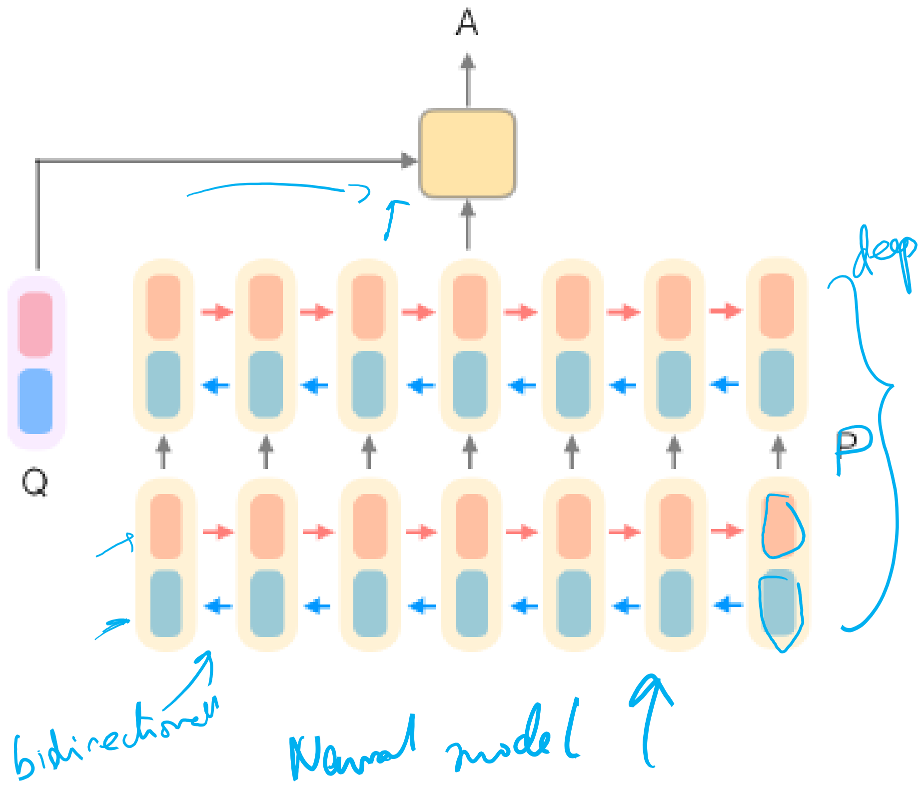
encodings

$$\{\mathbf{p}_1, \ldots, \mathbf{p}_m\} = \text{RNN}(\{\tilde{\mathbf{p}}_1, \ldots, \tilde{\mathbf{p}}_m\})$$

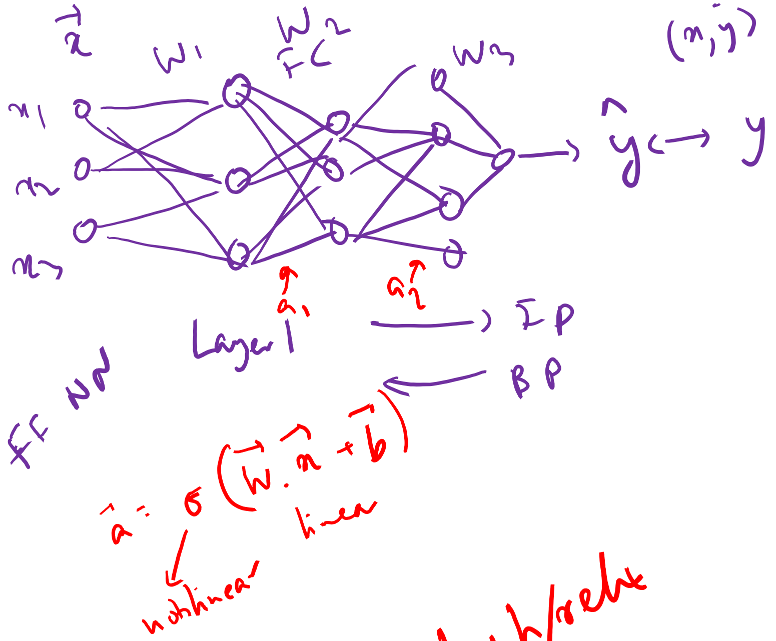
feature vectors of words

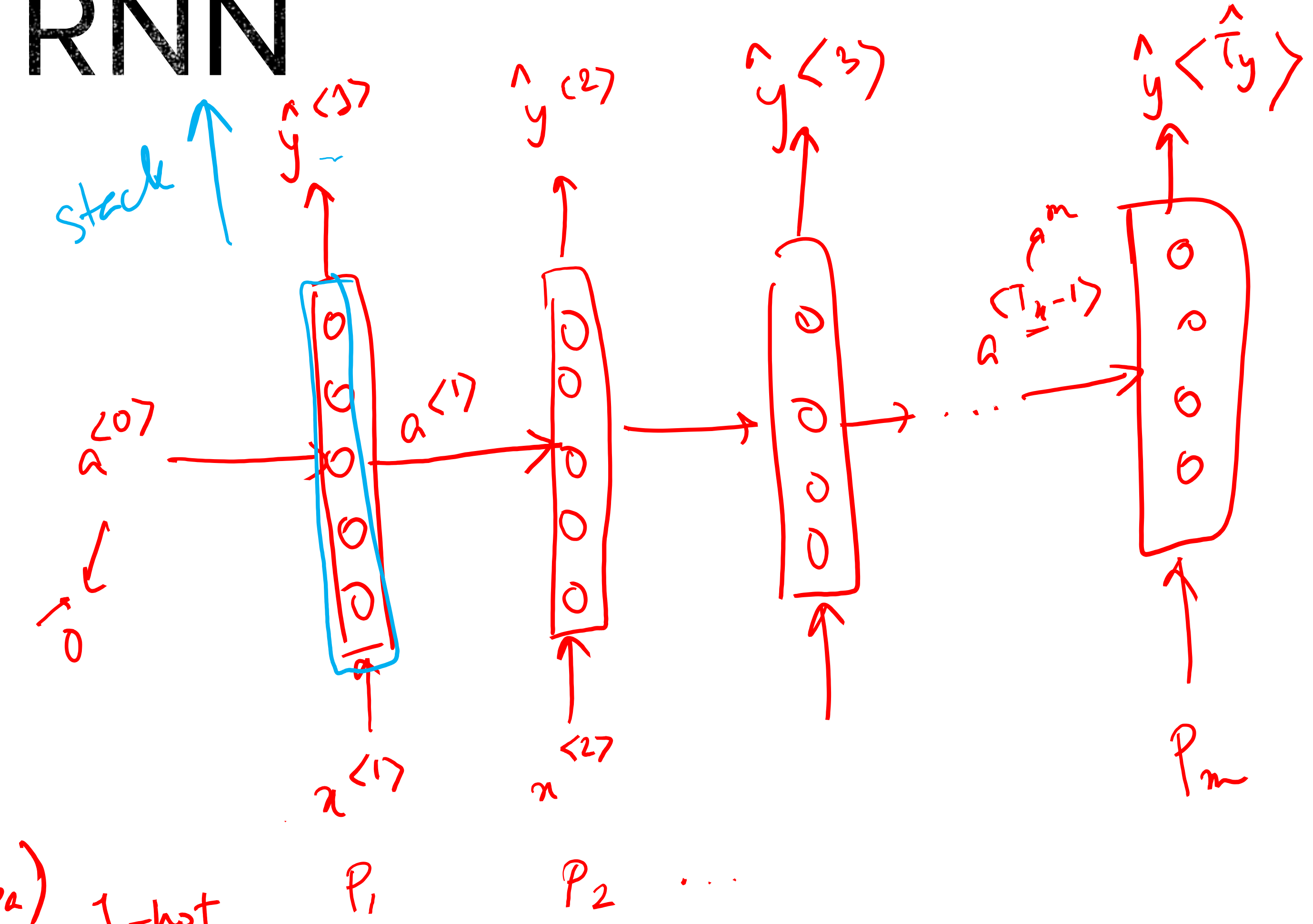
$q = \{q_1, q_2, q_3 \ldots q_l\}$

$p = \{p_1, p_2, p_3, \ldots, p_m\}$



Digression: RNN





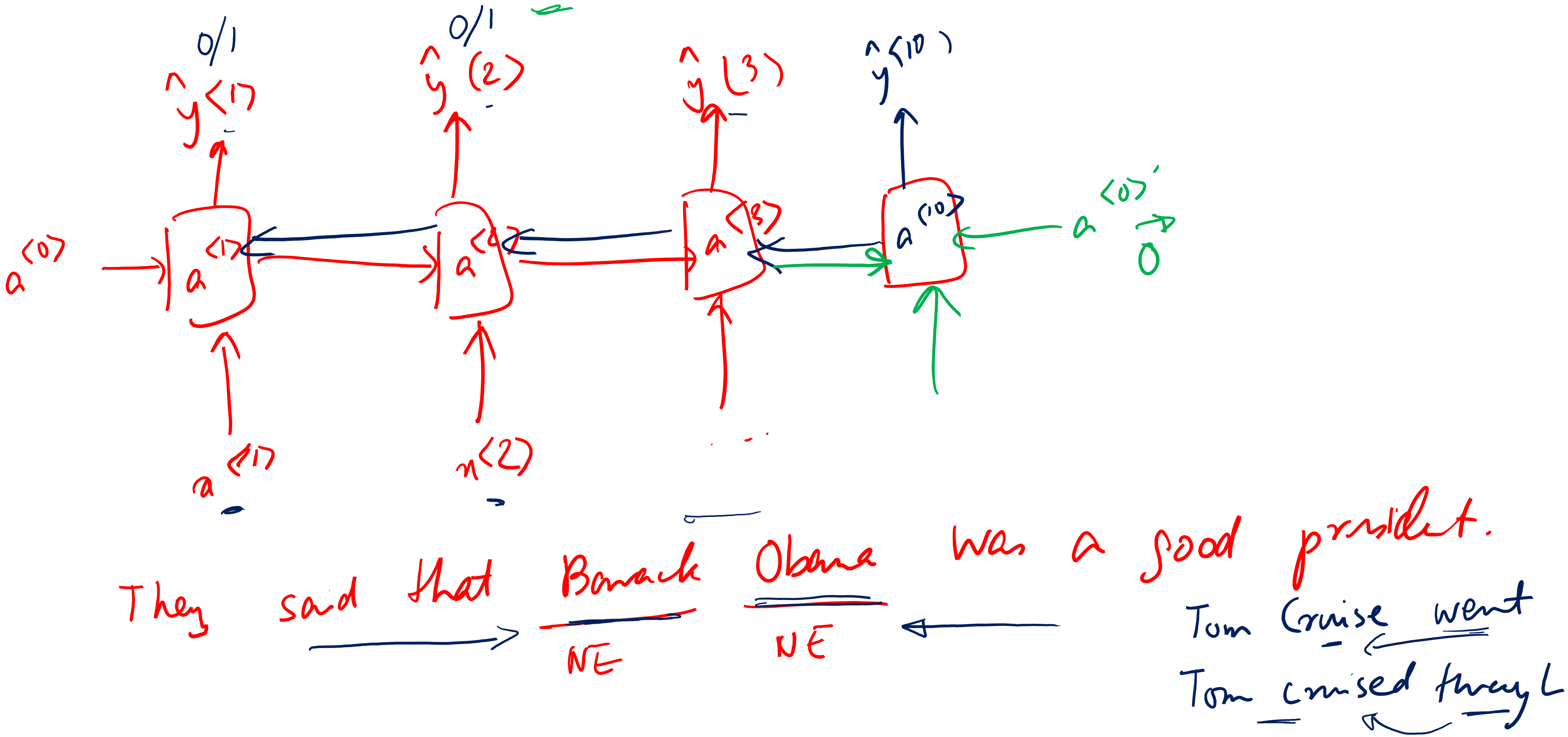
tanh/relu

$a^{\langle 1 \rangle} = g\left(W_{aa}\, a^{\langle 0 \rangle} + W_{ax}\, x^{\langle 1 \rangle} + b_a\right)$ 1-hot or word2vec glove

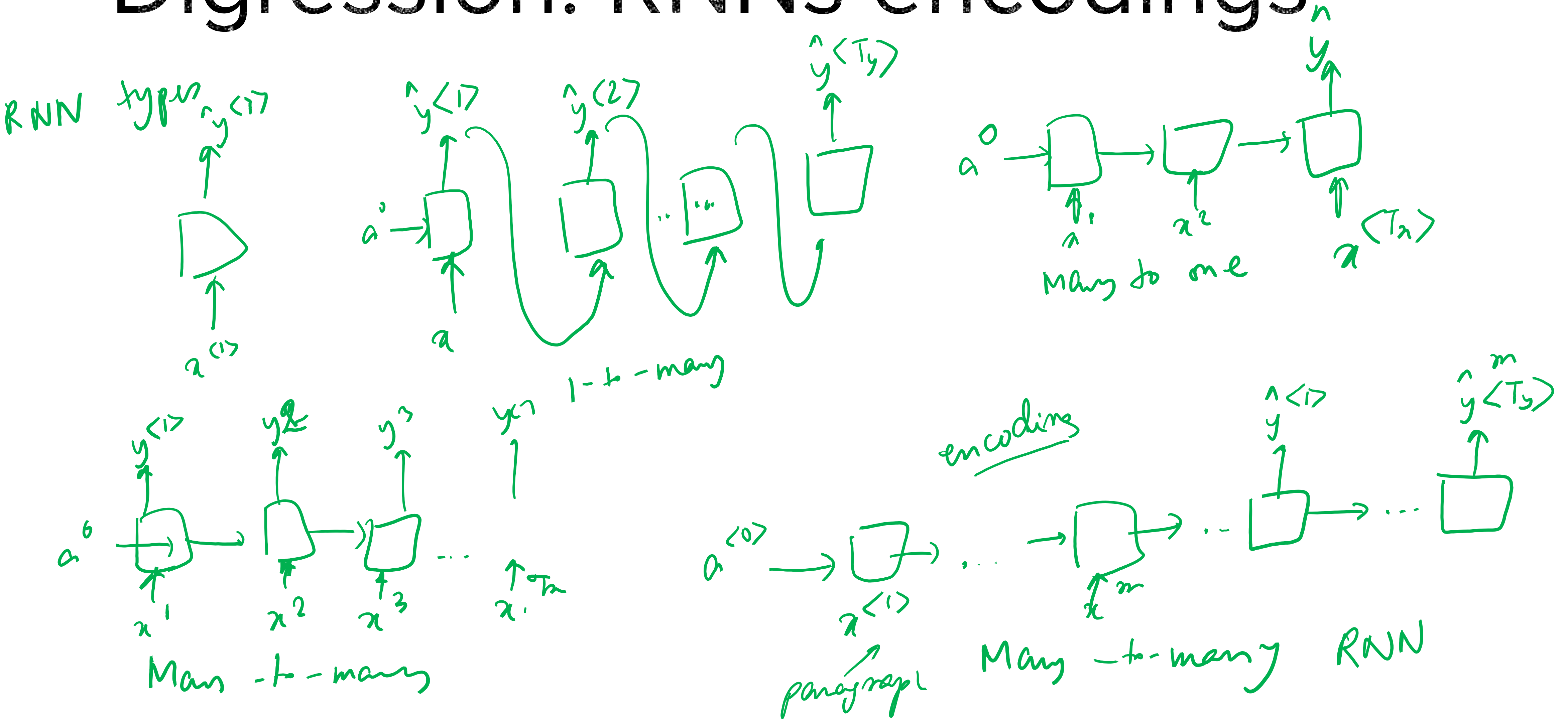
sigmoid → $\hat{y}^{\langle 1 \rangle} = g_2\left(W_{ya}\, a^{\langle 1 \rangle} + b_y\right)$

Andrew Ng

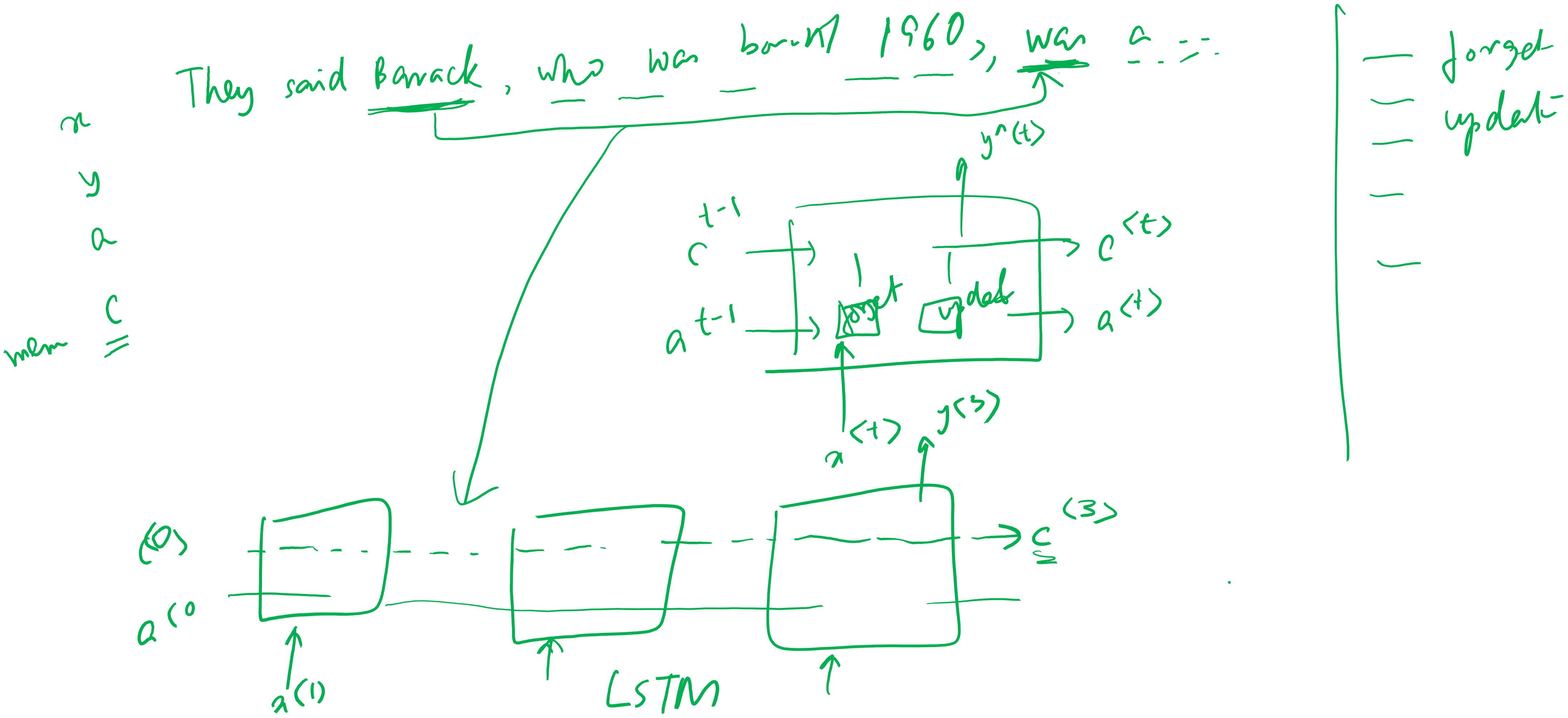
Digression: Bidirectional RNN



Digression: RNNs encodings



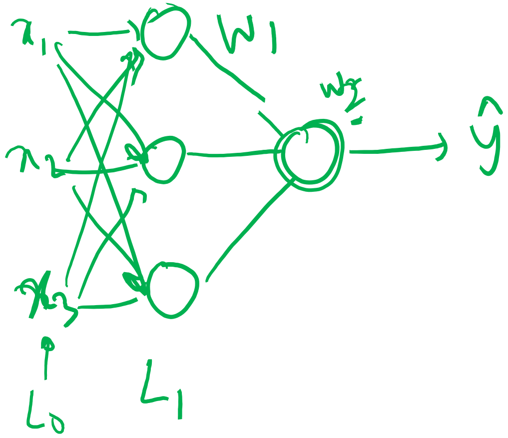
Digression: LSTM



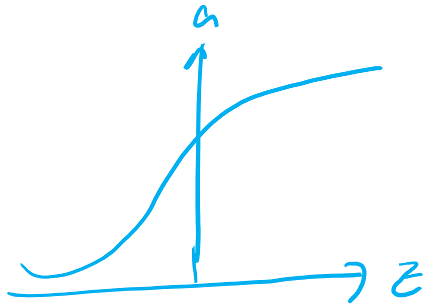
Digression: ReLU

Activation functions

differentiable

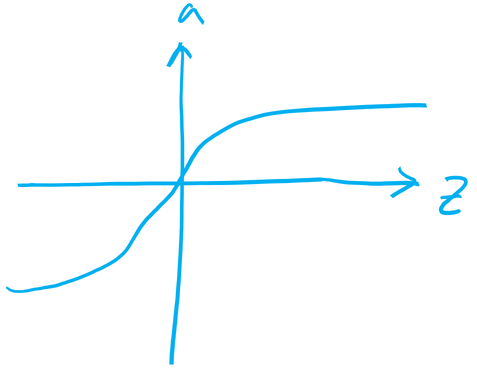


sigmoid $a = \dfrac{1}{1+e^{-z}}$



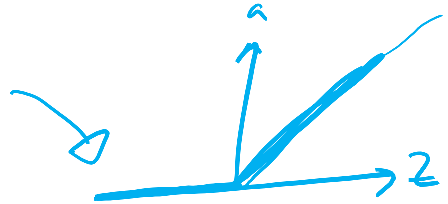
$\sigma'(x) = \sigma(x)(1-\sigma(x))$

tanh : $a = \dfrac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$

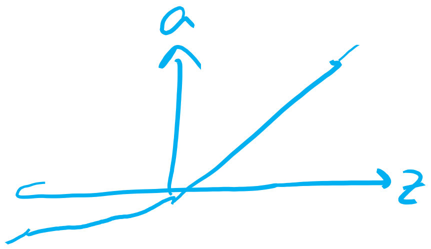


$1 - \tanh^2(x)$

ReLU : $a = \max(0, z)$



Leaky ReLU : $a = \max(0.01z, z)$



$z^{[1]} = W^{[1]} x + b^{[1]}$ ← linear

$a^{[1]} = g(z^{[1]})$

$z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$ / $a^{[2]} = g(z^{[2]})$...

Paragraph encoding: Features

- 1 Word embeddings GloVe
- 2 Exact match
- 3 Token features
- 4 Aligned question features

Attention

capture synonymy

word, embedding, 300D

$$f_{emb}(p_i) = \mathbf{E}(p_i)$$

0/1

$$f_{exact\_match}(p_i) = \mathbb{I}(p_i \in q)$$

1-hot

$$f_{token}(p_i) = (\text{POS}(p_i), \text{NER}(p_i), \text{TF}(p_i))$$

attention

$$f_{align}(p_i) = \sum_j a_{i,j} E(q_j)$$

ReLU

$$a_{i,j} = \frac{\hat{e}\ \exp\left(\alpha(\mathbf{E}(p_i)) \cdot \alpha(\mathbf{E}(q_j))\right)}{\sum_{j'} \exp\left(\alpha(\mathbf{E}(p_i)) \cdot \alpha(\mathbf{E}(q_{j'}))\right)}$$

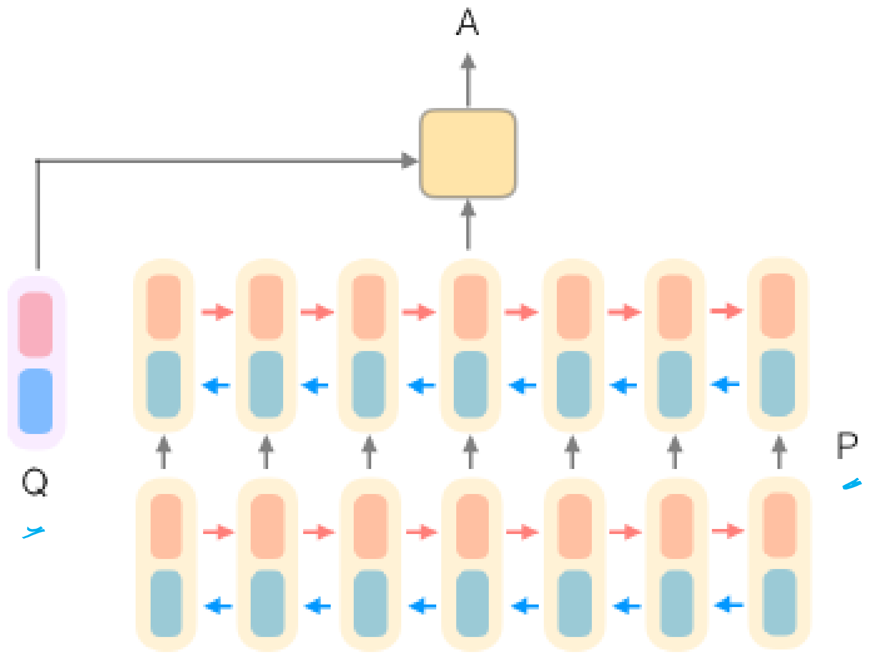
Question encoding

- Another RNN
- Over word embeddings of question tokens

$\{\vec{q}_1, \vec{q}_2, \ldots\} \longrightarrow \vec{q}$ | $\vec{q} = \sum_j b_j \vec{q}_j$

learnt

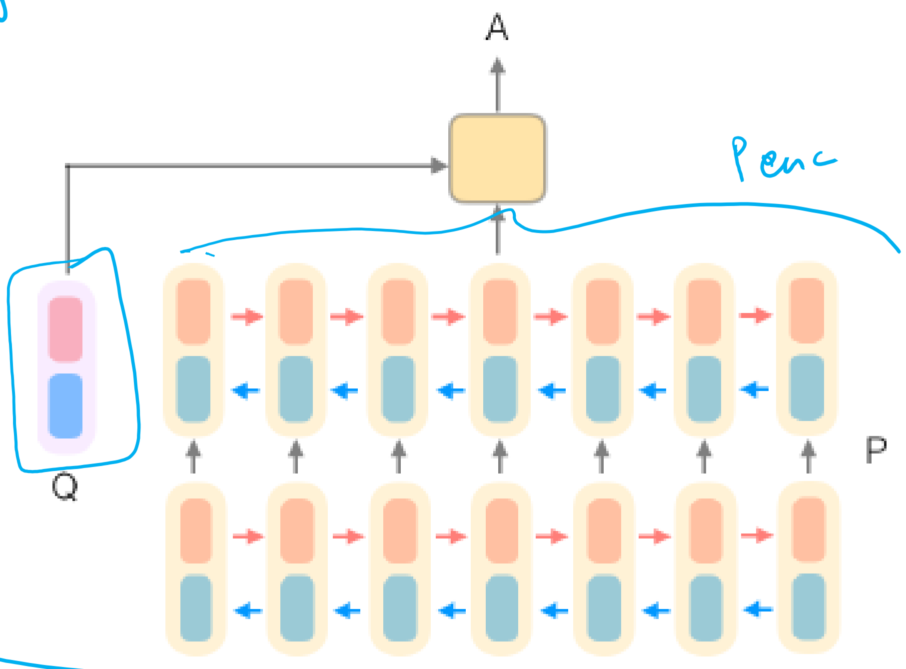
$$b_j = \frac{\exp(\mathbf{w} \cdot \mathbf{q}_j)}{\sum_{j'} \exp(\mathbf{w} \cdot \mathbf{q}_{j'})}$$



Answer prediction

Noting.

- Predict answer span beg / end } of span of text
- Two independent classifiers
- Use bilinear term
- Choose span from indices i to i' (ans)
- i <= i' <= i + 15 (start, end) max 15 words.
- Maximize $P_{start}(i) \times P_{end}(i')$ arg max over all



unnormalized exponential score q enc

$$P_{start}(i) \propto \exp(\mathbf{p}_i \mathbf{W}_s \mathbf{q})$$
$$P_{end}(i) \propto \exp(\mathbf{p}_i \mathbf{W}_e \mathbf{q})$$

Representative examples

Training

Dataset	Example	Article / Paragraph
SQuAD	**Q**: How many provinces did the Ottoman empire contain in the 17th century? **A**: 32	**Article**: Ottoman Empire **Paragraph**: ... At the beginning of the 17th century the empire contained 32 provinces and numerous vassal states. Some of these were later absorbed into the Ottoman Empire, while others were granted various types of autonomy during the course of centuries.
CuratedTREC	**Q**: What U.S. state's motto is "Live free or Die"? **A**: New Hampshire	**Article**: Live Free or Die **Paragraph**: "Live Free or Die" is the official motto of the U.S. state of New Hampshire, adopted by the state in 1945. It is possibly the best-known of all state mottos, partly because it conveys an assertive independence historically found in American political philosophy and partly because of its contrast to the milder sentiments found in other state mottos.
WebQuestions	**Q**: What part of the atom did Chadwick discover?† **A**: neutron	**Article**: Atom **Paragraph**: ... The atomic mass of these isotopes varied by integer amounts, called the whole number rule. The explanation for these different isotopes awaited the discovery of the neutron, an uncharged particle with a mass similar to the proton, by the physicist James Chadwick in 1932. ...
WikiMovies	**Q**: Who wrote the film Gigli? **A**: Martin Brest	**Article**: Gigli **Paragraph**: Gigli is a 2003 American romantic comedy film written and directed by Martin Brest and starring Ben Affleck, Jennifer Lopez, Justin Bartha, Al Pacino, Christopher Walken, and Lainie Kazan.

DrQA
vs
YodaQA

SQuAD
WebQuestion
:

Research paper 2

Simple and Effective Multi-Paragraph

Reading Comprehension

Simple and effective multi-paragraph reading comprehension [PDF] arxiv.org

C Clark, M Gardner - arXiv preprint arXiv:1710.10723, 2017 - arxiv.org

We consider the problem of adapting neural paragraph-level question answering models to the case where entire documents are given as input. Our proposed solution trains models to produce well calibrated confidence scores for their results on individual paragraphs. We sample multiple paragraphs from the documents during training, and use a shared-normalization training objective that encourages the model to produce globally correct output. We combine this method with a state-of-the-art pipeline for training models on …

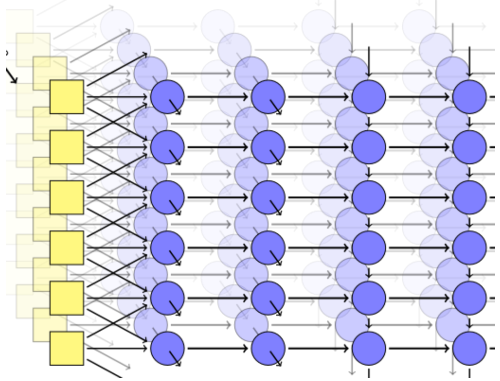
Cited by 198 Related articles All 5 versions

Neural question answering

Question: “What color is the sky?”

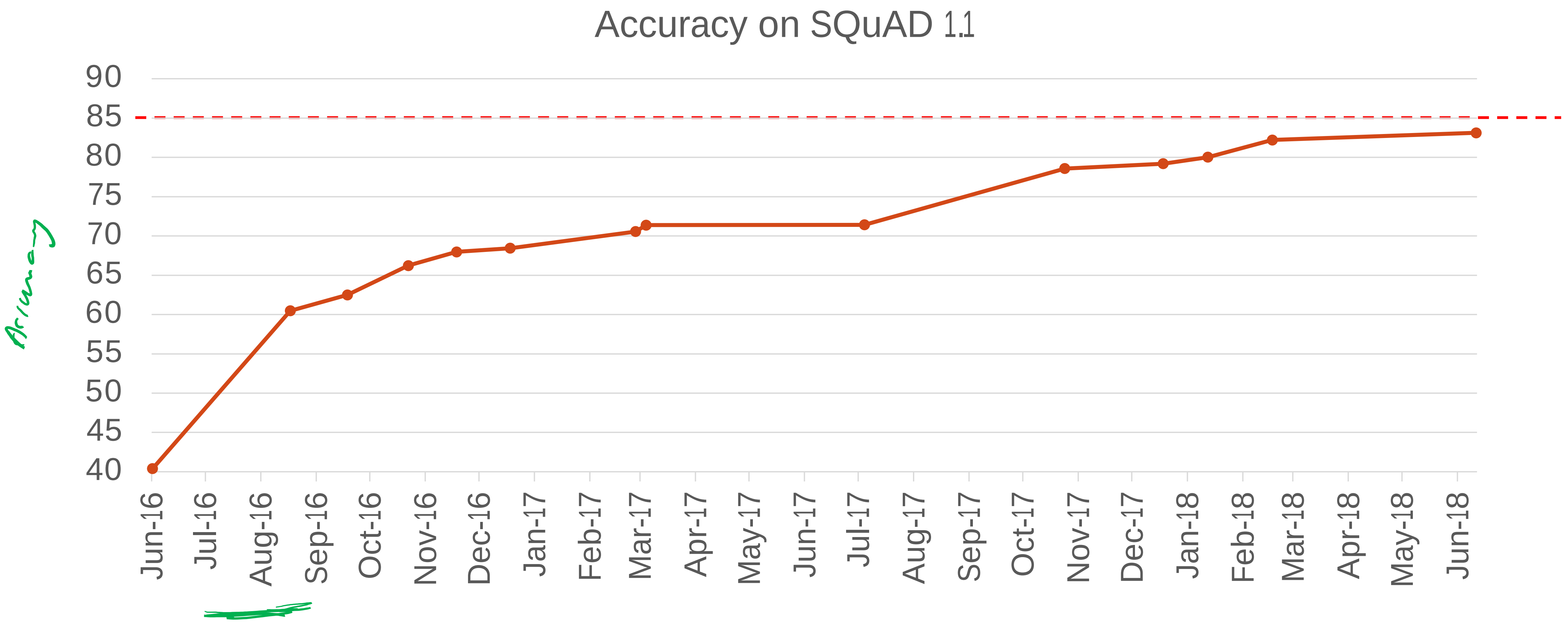
Passage: “Air is made mainly from molecules of nitrogen and oxygen. These molecules scatter the blue colors of sunlight more effectively than the green and red colors. Therefore, a clean sky appears blue.”

deep



Thanks to Christopher Clark for the slides

Fast progress on paragraph datasets



Challenge: Scaling Models to Documents

2018 July

- Modern reading comprehension models have many layers and parameters
- The trend is continuing in this direction, for example with the use of large language models
- Reduced efficiency as the paragraph length increases due to long RNN chains or transformers/self-attention modules
- Limits the model to processing short paragraphs

vanishing/exploding gradients

① Training's (q, doc) pairs

q span

q doc ans

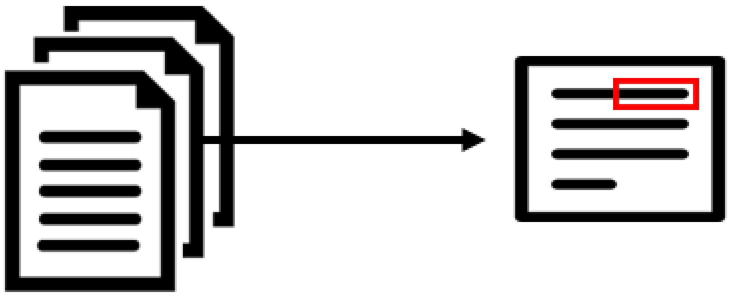
Distant supervision

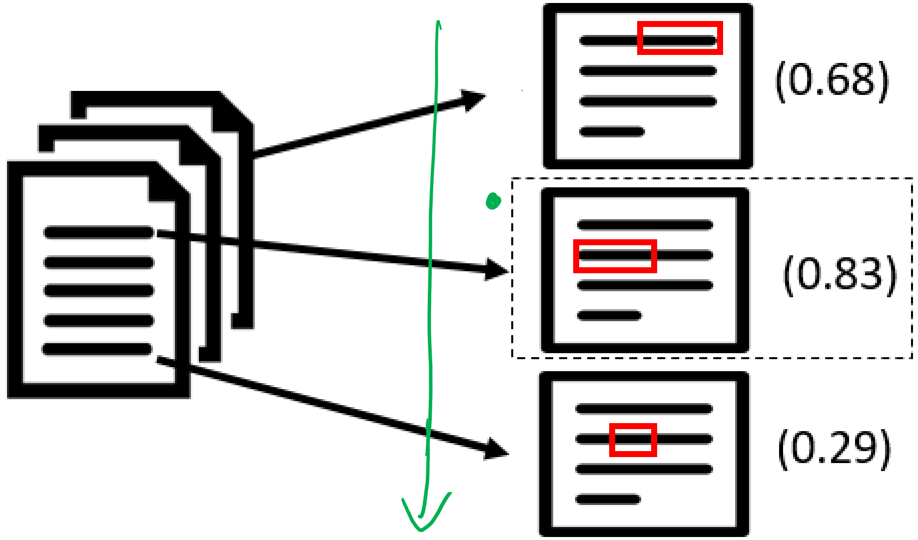
multipassage

② Shared normalization

Challenge: Scaling Models to Documents

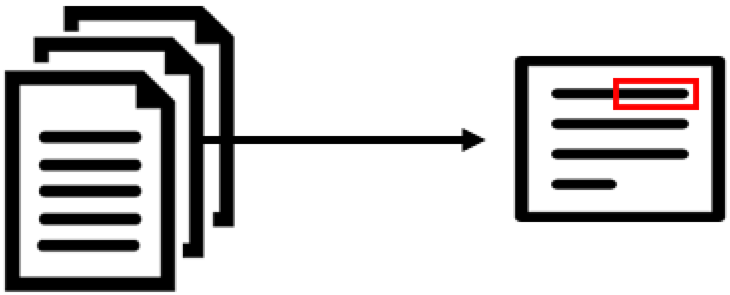
1. - Pipelined Systems
    - Select a single paragraph from the input, and run the model on that paragraph
2. - Confidence Systems DrQA
    - Run the model on many paragraphs from the input, and have it assign a confidence score to its results on each paragraph



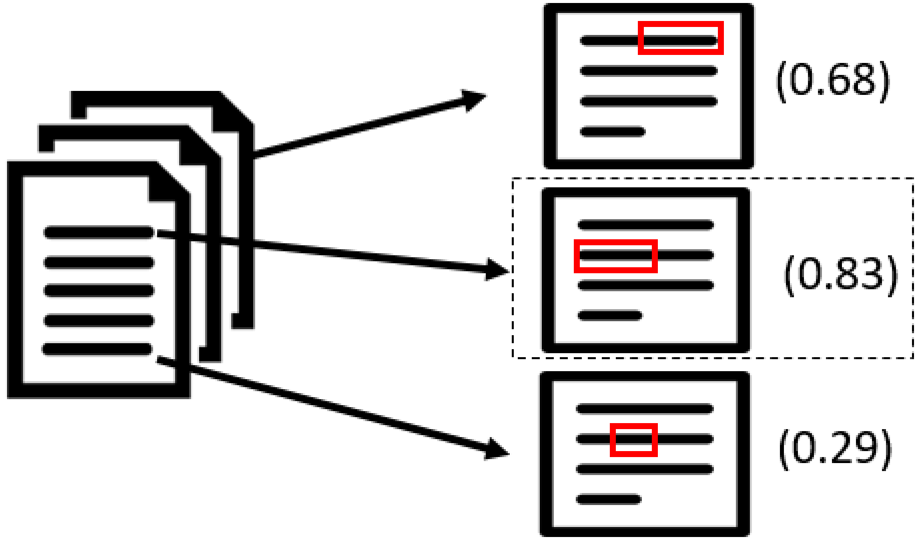


DocumentQA

- Improved Pipeline Method
  - Improve several of key design decision that arise when training on document-level data



- Improved Confidence Method
  - Study ways to train models to produce correct confidence scores



Pipeline Method: Paragraph Selection

- Train a shallow linear model to select the best paragraphs from corpus
  - Features include TF-IDF, word occurrences, and its position within document
- If there is just one document TF-IDF alone is effective
- Improves change of selecting an answer-containing paragraph from 83.0 to 85.1 on TriviaQA Web

benchmark

Pipeline Method: Noisy Supervision

Document level data can be expected to be distantly supervised:

Q

Question: Which British general was killed at Khartoum in 1885?

A

Gordon

Passage:

In February 1884 Gordon returned to the Sudan to evacuate Egyptian forces. Rebels broke into the city , killing Gordon and the other defenders. The British public reacted to his death by acclaiming ' Gordon of Khartoum , a saint. However, historians have since suggested that Gordon defied orders and....

Pipeline Method: Noisy Supervision

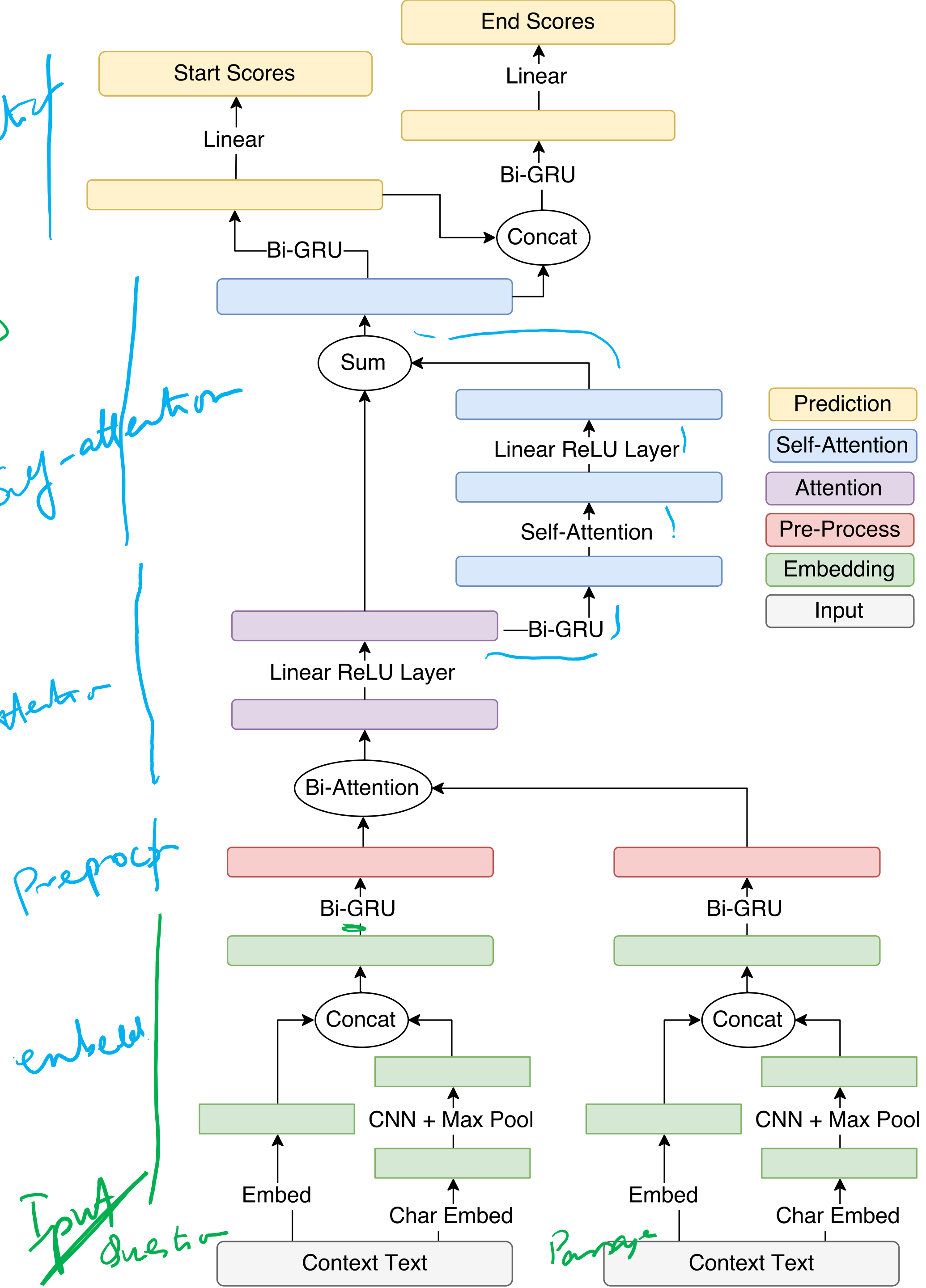
- Need a training objective that can handle multiple (noisy) answer spans
- Use the summed objective from Kadlec et al (2016), that optimizes the log sum of the probability of all answer spans
- Remains agnostic to how probability mass is distributed among the answer spans

Pipeline Method: Model

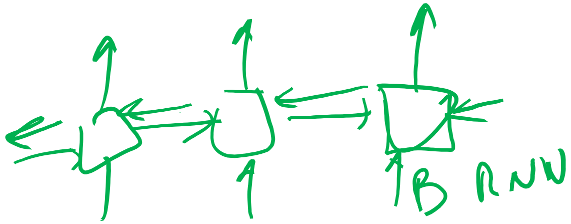
- Construct a fast, competitive model
- Use some keys ideas from prior work, bidirectional-attention, self-attention, character-embeddings, variational dropout
- Also added learned tokens for document and paragraphs starts

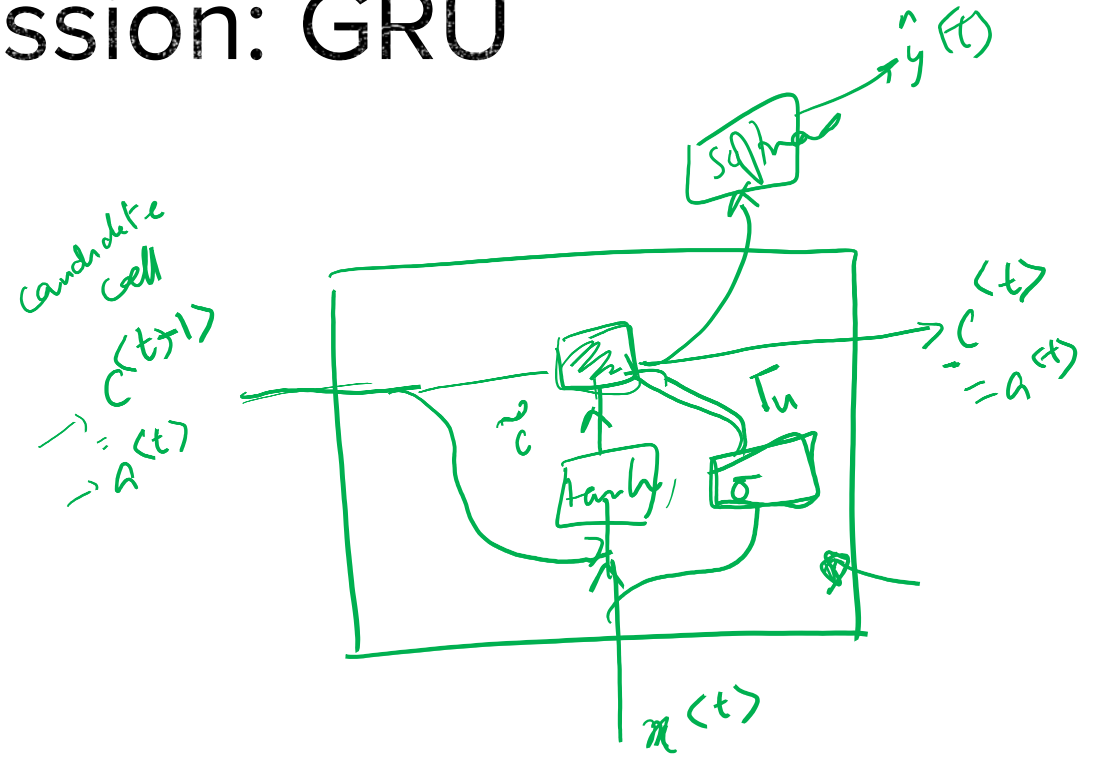
start <w,> <wc>

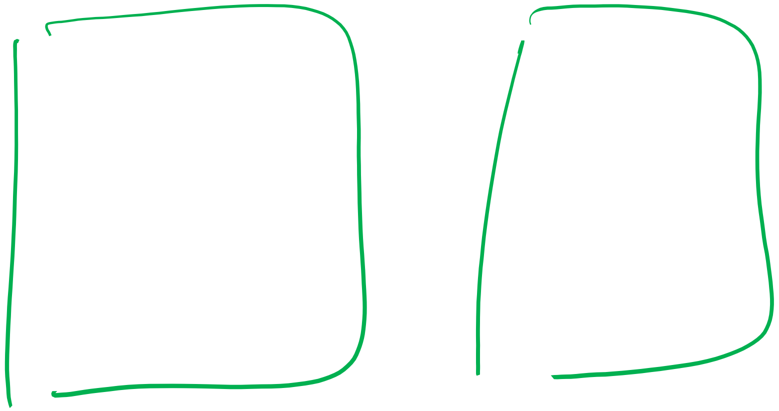
- < 5 hours to train for 26 epochs on SQuAD



Digression: GRU







GRU ~ 2014

LSTMs ~ 1997

They said that Obama, ..., was a great president.

RNN

1-to-1, 1-to-many, many-to-1, many-to-many

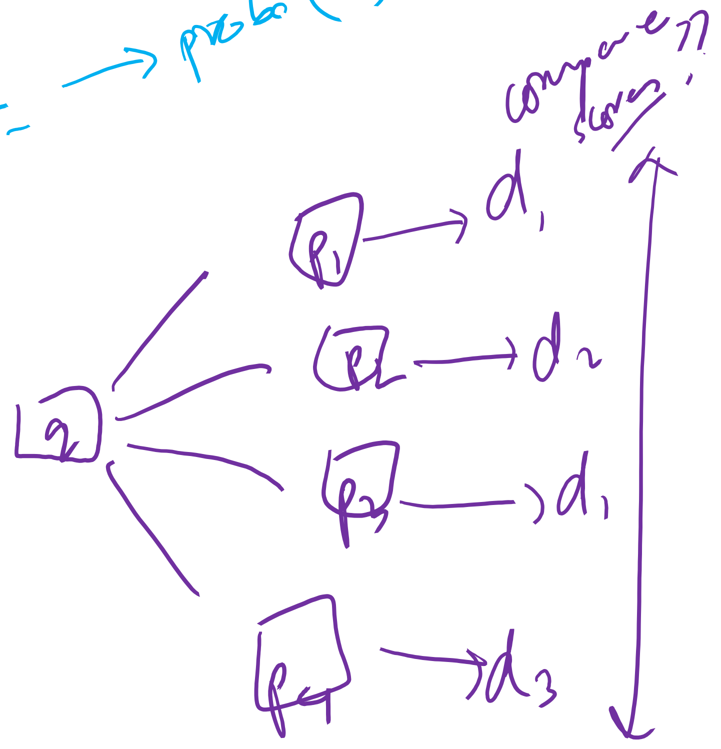
Confidence methods

- ~~We~~ Then can derive confidence scores from the logit scores given to each span by the model, i.e., the scores given before the softmax operator is applied

  logistic σ( )

  $\frac{e}{\sum e}$ → prob (...)

- Without re-training this can work poorly



Example from SQuAD

Question: “When is the Members Debate held?”

Model Extraction: “..majority of the Scottish electorate voted for it in a referendum to be held on 1 March 1979 that represented at least... ”

Correct Answer: “Immediately after Decision Time a “Members Debate” is held, which lasts for 45 minutes... ”

Learning well-calibrated confidence scores for span

- Train the model on both answer-containing and non-answer-containing paragraph and use a modified objective function
- 1 Merge: Concatenate sampled paragraphs together
- 2 No-Answer: Process paragraphs independently, and allow the model to place probability mass on a “no-answer” output
- 3 Sigmoid: Assign an independent probability on each span using the sigmoid operator

$$\frac{e^{s_{ap}}}{\sum_{j \in P} \sum_{i=1}^{n_j} e^{s_{ij}}}$$

- 4 Shared-Norm: Process paragraphs independently, but compute the span probability across spans in all paragraphs

over all passages with same context

Conclusions

MRC

- Neural machine reading comprehension systems now form the bulk of text-QA today

IR/DR

- MRC systems coupled with retrieval pipeline result in so-called “open domain QA systems” IR+MRC

- Retrieval is still primarily TF-IDF based not restricted to entities

Reader arbitrary

- Answers from text are predicted as free-form spans

- Often designed as classifiers for start and end positions in passage text

strong supervision

q a passage 32 34

Tim B Lee

distant supervision

q a d

span anno