Carnegie Mellon University
Master of Computational Data Science

Carnegie Mellon University
Language Technologies Institute

SQUAD

Self-Attention & Trees

Tian Qin, Yifan Yang, Griffin Adams

Advisor: Matthias Grabmair

11-632 (Fall 2017)
MCDS Capstone Seminar

Task: Question Answering

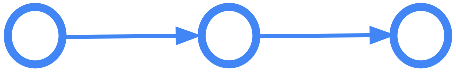
- Given text context and question, find answer in context
- Hypothesis: answer exists as a span of consecutive tokens in context

Question: When people take on debt, it leads potentially to what?

Development Goals

- Spring Semester Goals
  - Dynamic attention modeling
  - Answer extractor improvement
- Fall Semester First Midterm Goals
  - Implement Tree-LSTM BiDAF Model baseline
  - Implement Self-attention Model baseline
- Fall Semester Second Midterm Goals
  - Incorporate Bi-Directional Attention
  - Hyper-parameter tuning
- Final Goals
  - Improve Self-Attention Model's Answer extractor by LSTM
  - Add modeling layer to Tree-LSTM Model
  - Hyper-parameter tuning

Design and Motivation



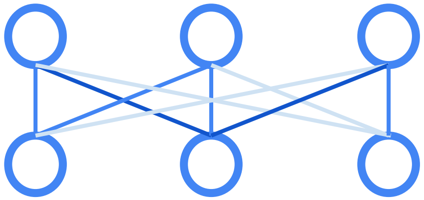
Sequential

Example: RNN (GRU, LSTM)

Pros: Imitate how human reads (sequentially)

- Linear time complexity
- Ideally can resolve lexical and coreference ambiguities

Cons: cell states forced to remember too much → compression loss



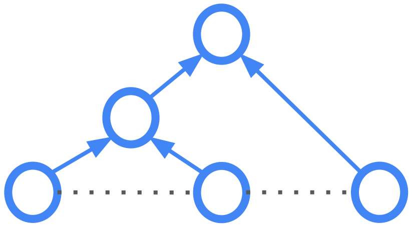
Self Attention

Example: Transformer encoder

Pros: no summarization loss

- Fast, Highly parallelizable
- Selectively pick from full context to disambiguate

Cons: May lose some positional info



Recursive / Tree

Example: Tree-RNN

Pros: Well supported by linguistic theory

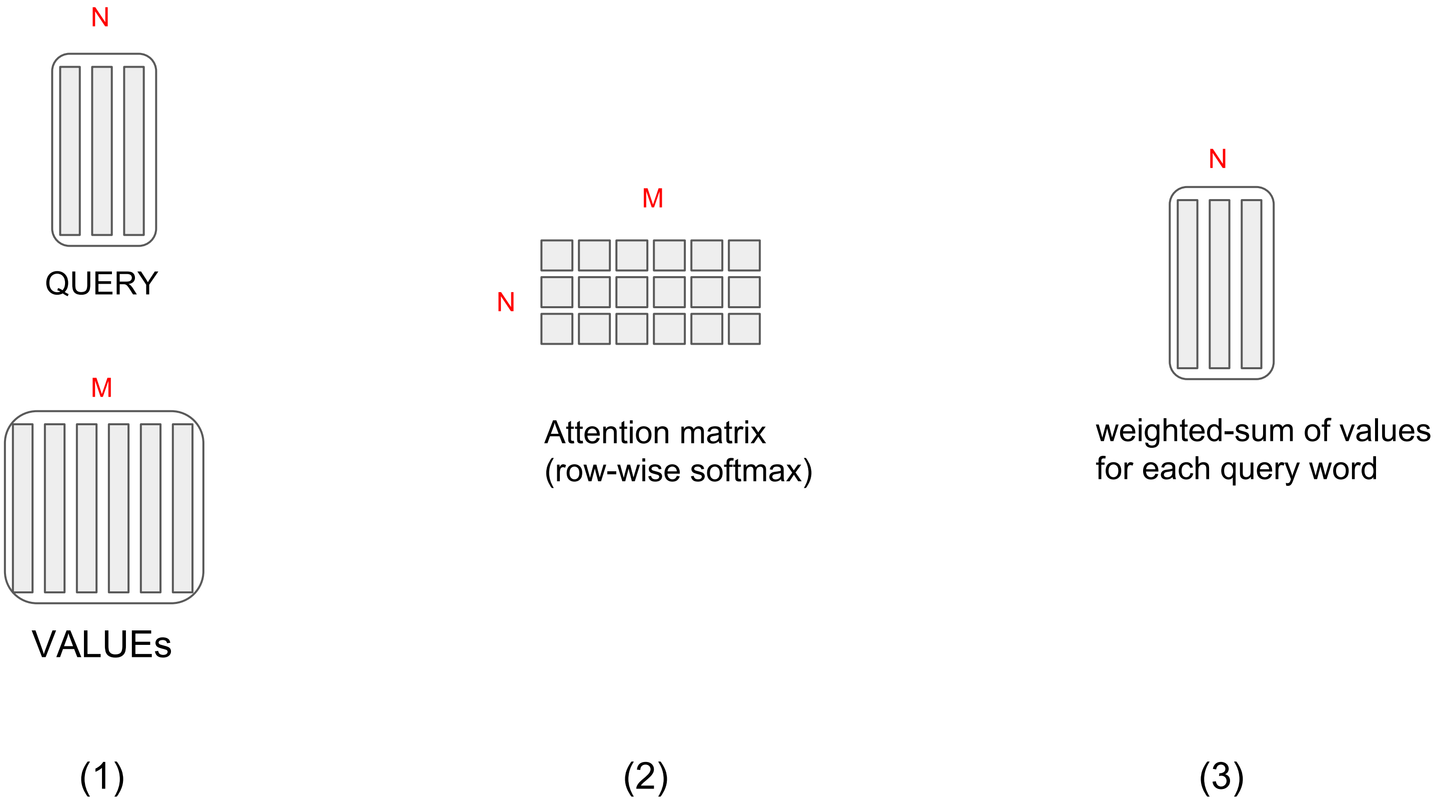
- Output representation for every syntactic unit
- Filters out irrelevant words at higher levels

Cons: Very slow, no standard approach to incorporate sibling context

Model: Outline

- Attention Modeling
  - How to calculate attention matrix
  - Self attention
- Tree-LSTM Model
  - Tree formation
  - BiDAF on top of tree
- Self-Attention Model
  - Multi-layers of self-attention
  - LSTM for answer extraction

Attention Modeling - General Paradigm



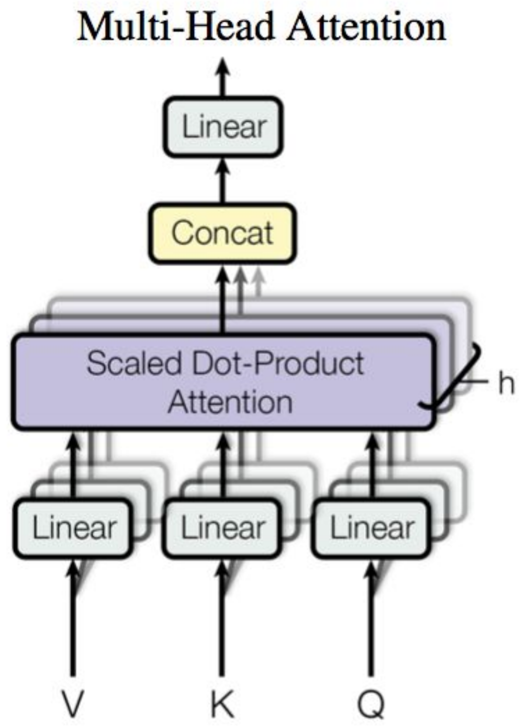
Attention Modeling cont’d

- Tree-LSTM BiDAF Model
  - **Context to Question** attention (QUERY = context, VALUE = question)
  - **Question to Context** attention (QUERY = question, VALUE = context)
  - Concatenate the above two to fuse information from both question and context

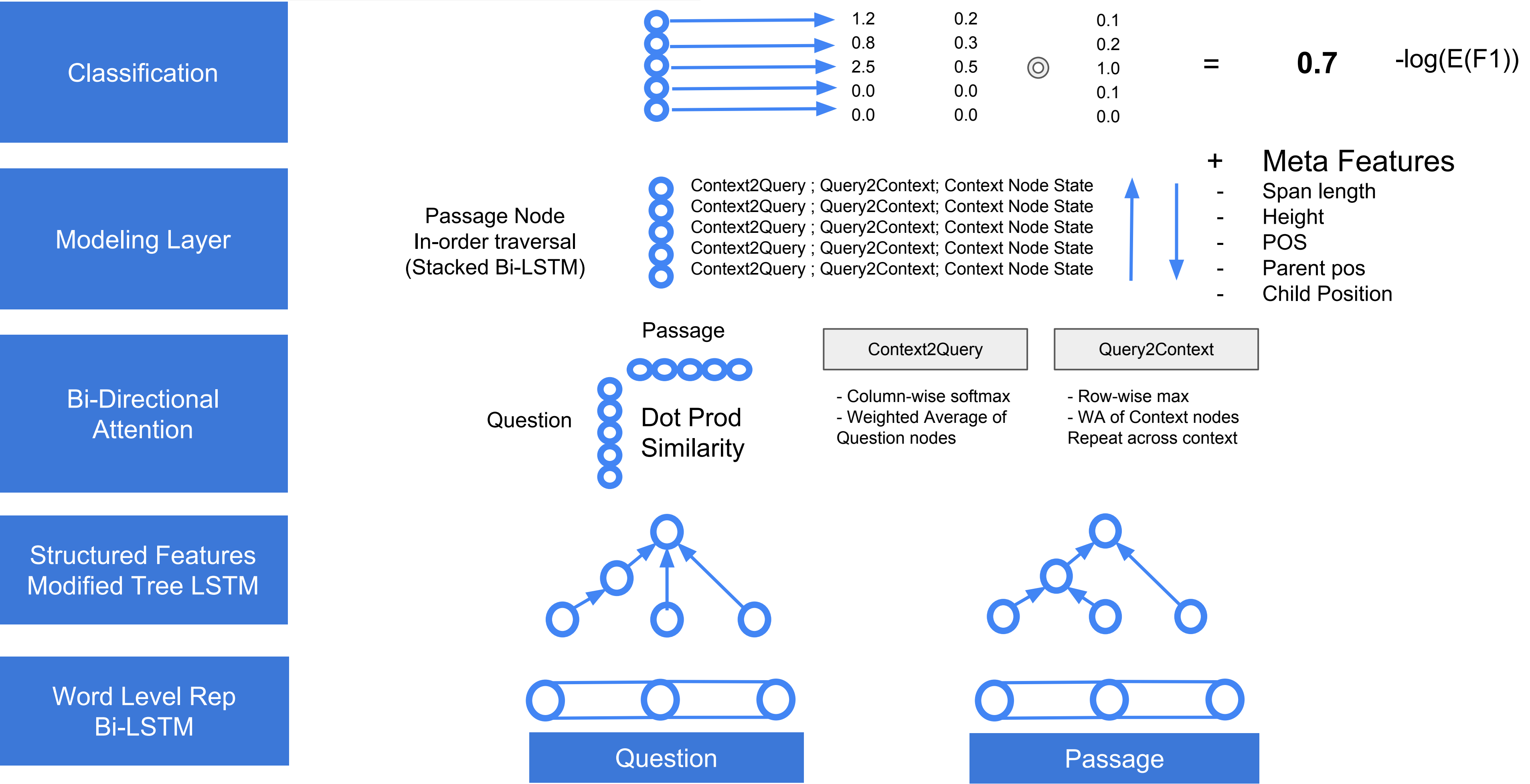
- Self-Attention Model
  - **Context to Question** attention (QUERY = context, VALUE = question)
  - Context **self-attention** (QUERY = context, VALUE = context)
  - Question **self-attention** (QUERY = question, VALUE = question)

Attention Modeling - Common Tricks

- Dot Product Attention
  - Scaled by square root of dimension
- Multi-layers perceptron Attention
  - Memory in-efficient
- Dynamic Attention (spring semester)
  - Iteratively improve attention
- Multi-Head Attention
  - Project into lower dimension
  - Each head focus on a different 'perspective'

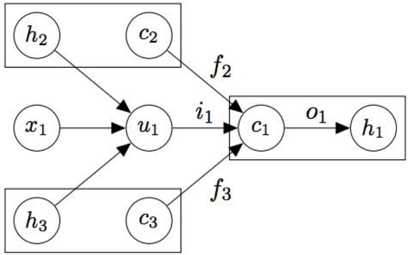


Tree Model Diagram



Tree Model Diagram - Tree LSTM

Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks
Kai Sheng Tai, Richard Socher*, Christopher D. Manning, 2015



$$\tilde{h}_j = \sum_{k \in C(j)} h_k, \tag{2}$$

$$i_j = \sigma\left(W^{(i)} x_j + U^{(i)} \tilde{h}_j + b^{(i)}\right), \tag{3}$$

$$f_{jk} = \sigma\left(W^{(f)} x_j + U^{(f)} h_k + b^{(f)}\right), \tag{4}$$

$$o_j = \sigma\left(W^{(o)} x_j + U^{(o)} \tilde{h}_j + b^{(o)}\right), \tag{5}$$

$$u_j = \tanh\left(W^{(u)} x_j + U^{(u)} \tilde{h}_j + b^{(u)}\right), \tag{6}$$

$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k, \tag{7}$$

$$h_j = o_j \odot \tanh(c_j), \tag{8}$$

Modelling Sentence Pairs with Tree-structured Attentive Encoder
Zhou, Liu, Pan (2016)

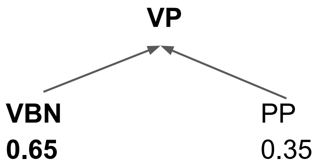
$$m_k = \tanh(W^{(m)} h_k + U^{(m)} s),$$
$$g = \textstyle\sum_{1 \le k \le n} \alpha_k h_k,$$

$$\alpha_k = \frac{\exp(w^\intercal m_k)}{\sum_{j=1}^{n} \exp(w^\intercal m_j)},$$

- Only difference is a learned weighted sum over children
- We experiment & find learned embeddings for meta tree vars works best
- Incorporating question summary representation does **not** add value

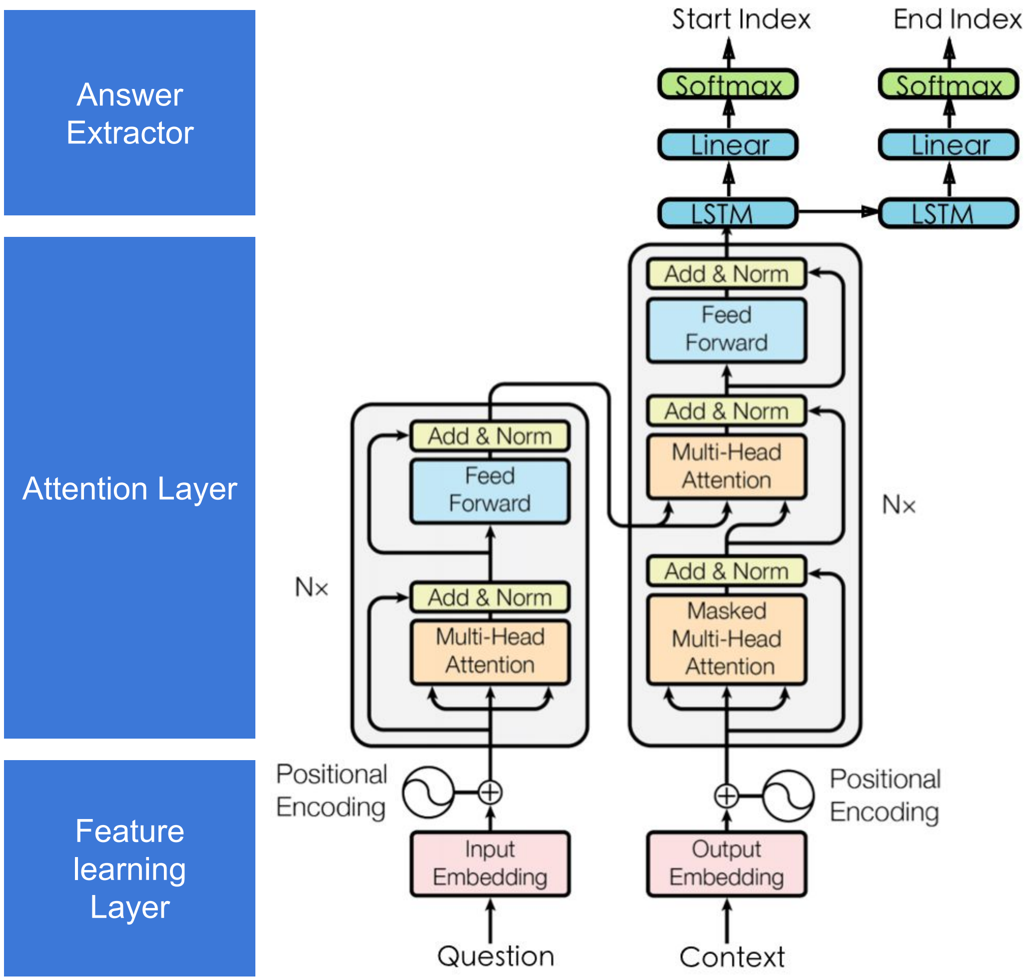
child_1 = W(node_meta_embeds_1 + b)
child_2 = W(node_meta_embeds_2 + b)
α_1, α_2 = softmax(child_1, child_2)

node_meta_embeds
- Span length
- Child position
- POS / parent POS
- Height
- Parent POS



Self-Attention Model

- Adapted from Transformer [1]
    - A network proposed by Google
    - Solely based on attention (w/o CNN/LSTM)
    - Achieve state-of-art result on translation
- Modifications
    - Encoder: Questions
    - Decoder: Context
    - Output layer: Start/End Index



Data and Evaluation Metrics

- Data
  - SQuAD, based on Wikipedia articles
  - Each instance contains a question, a context and answers
  - The train and dev data consists of 90k and 10k instances
- Evaluate on Self-Attention Model and Tree-LSTM model
  - Baseline: BiDAF [2]
- Metrics
  - F1 score
  - Exact Match score

Results

	EM	F1
BiDAF(single) [2]	0.68	0.77
Self-Attention Model	0.56	0.68
Tree LSTM Model	0.27	0.54

* EM and F1 for Tree LSTM Model is a lower bound on official EM and F1 score
Example of understatement (0 F1 assigned)...

Question: where are teachers recruited from ?

Passage: in germany , teachers are mainly civil servants recruited in special university classes , called teaching education studies.......
True: teaching education studies
Guessed: special university classes

Major Improvements - Tree LSTM Model

Improvements	**How do we find out?**	**F1 scores**
From Structured Prediction to choosing Max Node	Structured prediction too much for 2 class problem	~0.05 => 0.23
Speed up	Tree is hard to batch - Create a queue over flattened tree and compose parents greedily over whole batch	One epoch time from a day to under an hour
Incorporate Bi-direction Attention (BiDAF) Improved training time dramatically	Tree model alone does not co-attend	0.35 => 0.42
Adding Modeling Layer (In-order tree traversal)	Few model parameters after BiDAF	0.42 => 0.54

Major Improvements - Self Attention Model

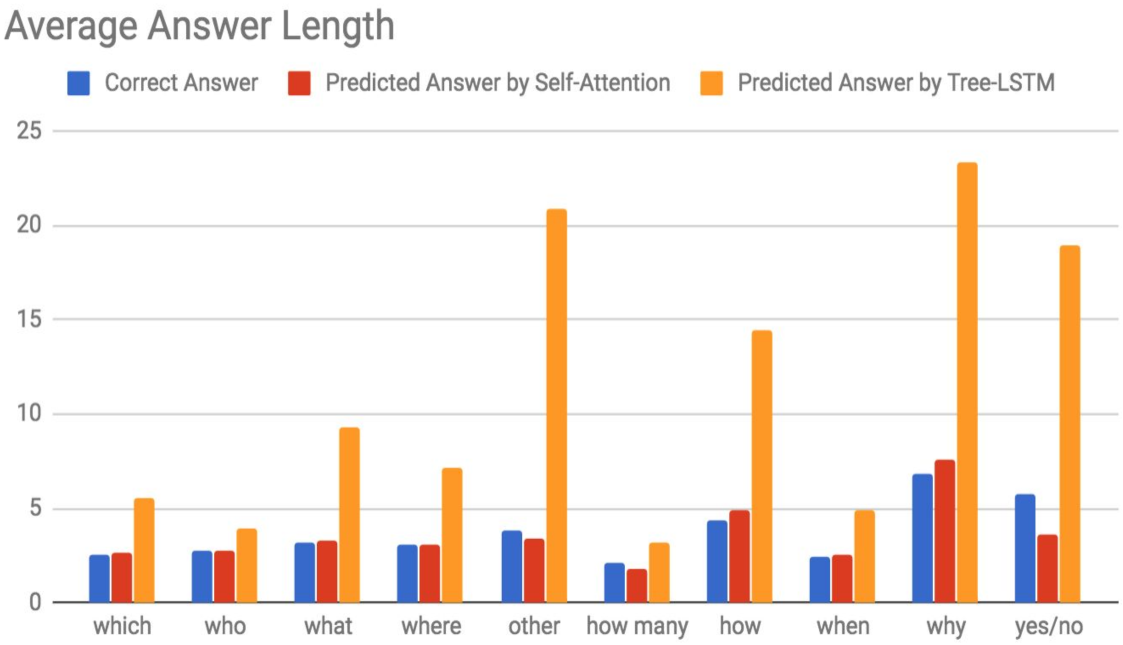
Improvements	**How do we find out?**	**F1 scores**
Freezing word embedding during training solves overfitting problem	The curve of training and validation loss	0.445 => 0.507
Limit the length (15) of predicted answer span	Experience from Spring term	0.507 => 0.523
Adding LSTM	Lack of feature learning	0.523 => 0.582
Fix a bug in evaluation script	The predicted answer spans contain unknown words	0.582 => 0.679

Marginal Improvements

- Tree-LSTM Model
  - In-order / pre-order traversal
  - Tree-LSTM attention
  - Using tree to propose answer spans
  - Additional tree features (height, span length, POS, child position)
  - Weight decay
  - Tree encoding attention mlp
  - Expected F1 loss over logistic best node
- Self-Attention Model
  - Bi-directional Attention
  - Label smoothing
  - Highway connection
  - Condition end index on start index
  - # heads in self-attention
  - Increase word embedding size
  - L2 regularization
  - Gradient clipping

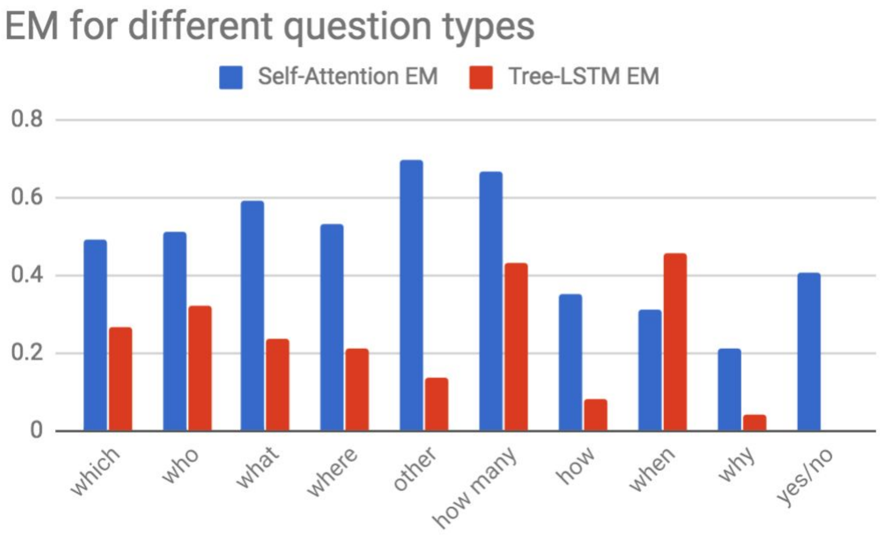
Error Analysis - Question type

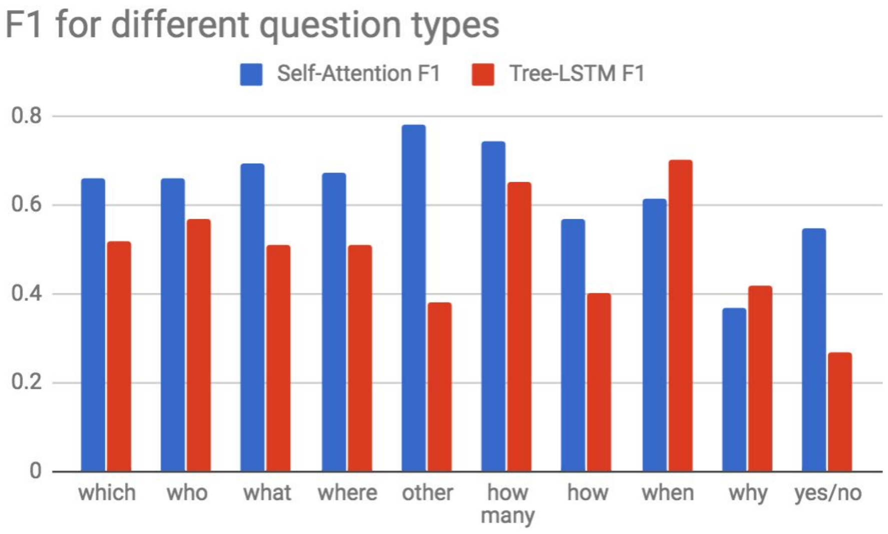
- Average length of predicted answers and true answers
- The Self-Attention model generates reasonable answer length
- The Tree-LSTM BiDAF much longer average length
    - Dominated by few outliers
    - Solved by setting max span



Error Analysis - Question type cont'd

- There is a large gap in EM score





Error Analysis - Attention Analysis

An example from Self-Attention Model

Context: tesla served as a *vice president* of the american institute of electrical engineers , the forerunner (along with the institute of radio engineers) of the modern - day ieee , from 1892 to 1894 .

Query: what position did tesla hold in the american institute of electrical engineers ?

Correct answer: *vice president*

Query2Context Attention

Query words	Context words with top attention values
position	vice=0.04, institute=0.04, 1892=0.04
tesla	tesla=0.05 1894=0.03 american=0.03
hold	a=0.05, as=0.03, ","=0.03, served=0.03

Error Analysis - Attention Analysis cont'd

An example from Self-Attention Model

Context: tesla served as a *vice president* of the american institute of electrical engineers , the forerunner (along with the institute of radio engineers) of the modern - day ieee , from 1892 to 1894 .

Query: what position did tesla hold in the american institute of electrical engineers ?

Correct answer: *vice president*

Context2Query Attention

Context words	**Query words with top attention values**
served	did=0.15, hold,=0.09, ?=0.08
vice	position=0.15, in=0.08, what=0.08
president	position=0.13, ?=0.08, what=0.08

Error Analysis - Attention Analysis cont'd

Multi-Head Attention

CONTEXT: super bowl 50 was an american football game to determine the champion of the national football league (nfl) for the 2015 season . the american football conference (afc) champion denver -UNK- defeated the national football conference (-UNK-) champion carolina panthers 24 – 10 to earn their third super bowl title . the game was played on february 7 , 2016 , at levi -UNK- ' s stadium in the san francisco bay area at santa clara , california . as this was the 50th super bowl , the league emphasized the -UNK- " -UNK- golden anniversary -UNK- " -UNK- with various gold - themed initiatives , as well as temporarily suspending the tradition of naming each super bowl game with roman numerals (under which the game would have been known as -UNK- " -UNK- super bowl l -UNK- " -UNK-) , so that the logo could prominently feature the arabic numerals 50 .
Query: what color was used to emphasize the 50th anniversary of the super bowl ?
Correct Answer: *golden*

Error Analysis - Attention Analysis cont’d

Attention Head	**Top context words for query ‘color’**	**Possible Interpretation**
#1	50=0.0117, ‘,’=0.0116, 50=0.0116	Number
#2	national=0.0085, league=0.0085, football=0.0085	Football
#3	area=0.0125, at=0.0110, arabic=0.0107	Location
#4	UNK=0.0131, UNK=0.0131, UNK=0.0131	Unknown words
#5	**gold=0.0129, santa=0.0123, golden=0.0117**	**color**
#6	UNK=0.0126, UNK=0.0126, UNK=0.0126	Unknown words
#7	themed=0.0124, conference=0.0124, initiatives=0.0123	???
#8	francisco=0.0102, levi=0.0102, american=0.0102	Location

Error Analysis - Attention Analysis cont'd

An example from Tree-LSTM Model

Context: the victorian alps in the northeast are the coldest part of victoria . the alps are part of the great dividing range mountain system extending *east west* through the centre of victoria . average temperatures are less than 9 °c 48 °f in winter and below 0 °c 32 °f in the highest parts of the ranges . the state 's lowest minimum temperature of <unk> °c 10.9 °f was recorded at omeo on 13 june 1965 , and again at falls creek on 3 july 1970 . temperature extremes for the state are listed in the table below
Query: in what direction does the mountain system extend ?
Correct Answer: *east west*

Query words	Context constituents with top attention values
direction	the northeast=0.0152 extending east west through the centre of victoria=0.0142 east west=0.0141
extend	extending=0.0307 extending east west through the centre of victoria=0.0206 system=0.0138
mountain	mountain=0.0328 the great dividing range mountain system=0.0274 again at falls creek on 3=0.0197

Error Analysis - Attention Analysis cont'd

An example from Tree-LSTM Model

Context: the victorian alps in the northeast are the coldest part of victoria . the alps are part of the great dividing range mountain system extending *east west* through the centre of victoria . average temperatures are less than 9 °c 48 °f in winter and below 0 °c 32 °f in the highest parts of the ranges . the state 's lowest minimum temperature of <unk> °c 10.9 °f was recorded at omeo on 13 june 1965 , and again at falls creek on 3 july 1970 . temperature extremes for the state are listed in the table below
Query: in what direction does the mountain system extend ?
Correct Answer: *east west*

Context2Query Attention

Context words/phrases	Query words with top attention values
east west	direction=0.1601, mountain=0.1521, extend=0.1281, system=0.1158
the great dividing range mountain system	system=0.2559, mountain=0.2283, the=0.1321

Error Analysis - Attention Analysis cont’d

- Self-attention Model’s attention is harder to interpret
  - Multi layers && multi heads
  - Seems to prefer word-matching rather than understanding
  - Mistakenly assign high attention value to punctuations
  - No way to force different heads to learn different perspectives
- Attention distribution not sparse enough
  - No sparsity constraint in loss function
- Tree-LSTM model attends to constituents instead of just words
  - Encourage attention over a structural feature space

Error Analysis - Classify errors into 6 error types

Randomly select 50 EM-incorrect answers and classify them into 6 categories

Error type	BiDAF (%)	Self-Attention Model (%)	Tree-LSTM (%)
Imprecise answer boundaries	50	54	45
Syntactic complications and ambiguities	28	34	14
Paraphrase problems	14	4	20
External knowledge	4	0	10
Multi-sentence	2	0	0
Incorrect preprocessing	2	8	8

Error Analysis - Tree LSTM Top k Prediction

- Top prediction F1 is **0.54**
- Max F1 among top 3 list is **0.73**
- Max F1 among top 10 list is **0.88**
- Usually very close because it directly models and predicts syntactic candidates
- Tried re-ranker based on tf-idf agreement but didn’t improve

Question: statues of british artists adorn which part of the tower above the main entrance ?
True: top row of windows
Top 5 Ranked Predictions:
Rank=1 shallow arches
Rank=2 cromwell gardens
Rank=3 the top row of windows
Rank=4 a gothic feature , the top row of windows
Rank=5 the galleries

Question: what is one function that prime numbers have that 1 does not ?
True: the sum of divisors function
Top 5 Ranked Predictions:
Rank=1 different factorizations of 15
Rank=2 euler 's totient function
Rank=3 the statement of that theorem
Rank=4 have several properties that the number…[ctd]
Rank=5 the relationship of the number to its…[ctd]

Error Analysis - NP dominance - Confusion for Exact Match errors

Guessed Part of Speech

True Part of Speech

	S	NP	NNS	NN	VP	PP	<UNK>	OTHER	
S	2	22	0	6	4	2	0	9	(49 examples)
NP	45	735	37	111	97	18	3	170	(1283 examples)
NNS	5	81	17	3	4	2	0	8	(127 examples)
NN	13	323	6	79	17	3	0	56	(519 examples)
VP	16	91	5	12	35	4	0	32	(203 examples)
PP	11	95	6	5	14	2	0	18	(160 examples)
OTHER	30	334	11	23	53	6	0	213	(694 examples)
	122	1682	82	239	224	37	3	506	--> guessed distribution

Error Analysis - Parent Preference - confusion for node height

Predicted answer height

True answer tree height

	1	2	3	4	5	6	7	8	9+	
1	330	**491**	90	108	29	27	27	24	58	(1233 examples)
2	256	353	**104**	102	39	24	18	24	70	(1047 examples)
3	57	105	12	**45**	14	12	2	9	19	(288 examples)
4	38	55	19	13	**8**	13	4	8	14	(180 examples)
5	13	32	5	19	2	**5**	5	2	8	(97 examples)
6	9	29	5	14	3	3	**2**	3	5	(74 examples)
7	5	5	1	7	2	3	0	**3**	3	(32 examples)
8	4	3	2	5	2	0	0	1	**7**	(25 examples)
9+	6	20	2	5	3	2	2	2	15	(60 examples)
	718	1093	240	318	102	89	60	76	199	--> guessed distribution

Error Analysis - Tree Model By True Span Length

Span Length **1** F-Score=**0.49**	Recall = 0.60	Precision = **0.41**	(**1902** examples)
Span Length **2** F-Score=0.58	Recall = 0.62	Precision = 0.55	(895 examples)
Span Length **3** F-Score=0.60	Recall = 0.62	Precision = 0.57	(862 examples)
Span Length **4** F-Score=**0.62**	Recall = **0.64**	Precision = **0.60**	(554 examples)
Span Length **5+** F-Score=0.56	Recall = **0.54**	Precision = 0.59	(779 examples)

- Points to benefit of tree encoding at synthesizing descendents without information loss
- Possibly insufficient word-level modeling.

Discussion

- The results are well below state of the art (high 80s F1)
  - Yet for Tree model, we use overly conservative evaluation script which only considers single top span
- Yet both models show they are learning and generalizing well
- Both models are well-motivated, have enough inductive bias and expressive power to learn fine-grained concepts
  - Likely a great deal of performance to be had in ensembling & hyper-parameter tuning.

Lessons Learned

- Controlling the pace and scope of work
  - Trying too many configurations/features
- Always be flexible about changing directions
  - Most things you try won’t work immediately (or ever)
    - Structured prediction, reranking, metadata modeling, all time consuming & were abandoned
- Hyper-parameters can make or break a model
  - Leave time for tuning
- (Bi)LSTMs can do magic!
- Double check evaluation (ours understates results)

Lessons Learned - Limit number of model ‘free’ parameters

```
parser.add_argument('--batch_size', type=int, default=32)
parser.add_argument('--data_path', default='./fulldata', help='Relative path wh
parser.add_argument('--lr', type=float, default=0.001, help='Initial learning r
parser.add_argument('--mem_dim', type=int, default=150, help='Size of hidden an
parser.add_argument('--embed_size', type=int, default=300)
parser.add_argument('--epochs', type=int, default=15)
parser.add_argument('--eval_freq', type=int, default=1, help='number of epochs
parser.add_argument('--lstm_dropout', type=float, default=0.2)
parser.add_argument('--meta_dropout', type=float, default=0.25)
parser.add_argument('--mlp_att_dropout', default=0.2, type=float)
parser.add_argument('--classifier_dropout', type=float, default=0.25)
parser.add_argument('--classifier_h_size', default=100, type=int)
parser.add_argument('--classifier_h_layers', default=1, type=int)
parser.add_argument('--unary_priority', default='last', help='first or last.  l
parser.add_argument('--clip', default=0.25, type=float)
parser.add_argument('--max_span_len', default=10000, type=int)
parser.add_argument('--tree_reducer', default='sum')
parser.add_argument('-bidaf_cosine', action='store_true', default=False)
parser.add_argument('--mlp_att_hidden_dim', default=50, type=int)
parser.add_argument('--embed_unfreeze_epoch', default=30, type=int)
parser.add_argument('-add_to_param_group', action='store_true', default=False)
parser.add_argument('--embed_lr', default=0.001, type=float)
parser.add_argument('--embed_decay', default=1e-4, type=float)
parser.add_argument('--model_layers', default=2, type=int)
parser.add_argument('--input_layers', default=1, type=int)
parser.add_argument('-no_compress', default=False, action='store_true')

parser.add_argument('-sep_encoders', default=False, action='store_true')

# global boolean flags
parser.add_argument('-no_cuda', action='store_true', default=False)
parser.add_argument('-mini', action='store_true', default=False)
parser.add_argument('-load_saved', action='store_true', default=False)
parser.add_argument('-load_batchers', action='store_true', default=False)
parser.add_argument('--eval_size', default=5000, type=int)
parser.add_argument('-mlp_att', action='store_true', default=False)
parser.add_argument('-model_w_question', action='store_true', default=False)
parser.add_argument('-bidaf_w_meta', action='store_true', default=False)
parser.add_argument('--meta_embed', default=30, type=int)
parser.add_argument('--max_meta_val', default=15, type=int, help='Maximum value
height in tree)')
```

Future Work

- For self-attention model:
  - Increase vocabulary size
  - Force different heads to attend different things
- For tree model:
  - Explore Better Tree-Structured Lexical Features
    - We used span length, pos, height, child position
    - Nothing seemed to improve training time or accuracy
  - Question-aware tree encoding (naive approach didn’t work)
- General:
  - Enforce sparsity on attention
  - Full hyper-parameter tuning
  - Enforce syntactic and heuristic constraints during span extraction

Conclusions

- We develop two distinct models for Question Answering
- Both models seek to learn rich question-aware features before classification
- Results show promise and semantically reasonable errors
- We believe that both self-attentional and tree-based approaches can be applied to Question Answering
    - Very sensitive to model choice and hyperparameters
- Both models have the capacity to reason without early summarization of context or question

References

1. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. arXiv preprint arXiv:1706.03762 .
2. Seo, Minjoon, et al. "Bidirectional attention flow for machine comprehension." *arXiv preprint arXiv:1611.01603* (2016).
3. Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. "Improved semantic representations from tree-structured long short-term memory networks." *arXiv preprint arXiv:1503.00075*(2015).
4. Vinyals, Oriol, Meire Fortunato, and Navdeep Jaitly. "Pointer networks." Advances in Neural Information Processing Systems. 2015.
5. Kumar, Ankit, et al. "Ask me anything: Dynamic memory networks for natural language processing." International Conference on Machine Learning. 2016.
6. Zhou, Yao, Cong Liu, and Yan Pan. "Modelling Sentence Pairs with Tree-structured Attentive Encoder." arXiv preprint arXiv:1610.02806 (2016).