Employing External Rich Knowledge for Machine Comprehension

IJCAI-16

Bingning Wang, Shangmin Guo , Kang Liu , Shizhu He , Jun Zhao

National Laboratory of Pattern Recognition , Institute of Automation, Chinese Academy of Sciences

Presented By : Dushyanta Dhyani

Outline

Outline

Problem Definition

Machine Comprehension

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

[1]

Outline

Challenges

- The Nature of this task requires a supervised learning approach.
- Availability of labeled data thus serves as a major bottleneck.
- Deep architectures which have proven to contain rich semantic understanding of text require large data.

Outline

Existing Datasets

- **Machine Comprehension Test (MCTest)** [2]
- **Children's Book Test (CBT)** - Part of Facebook's *bAbi Project* [3]
- **CNN/DailyMail Dataset** [4]

Dataset

MCTest

- Collection of 660 stories and associated questions.
- Collected using Amazon Mechanical Turk
- Each questions is labeled as 'one' or 'multiple' to indicate the number of sentences in the document that are related to this question.
- Each question has four candidate answers which may span single or multiple words.
- Questions maybe factoid or non-factoid

Outline

Proposed Approach

External Supervision!!

- Given the small amount of data, deep architectures might not perform well.
- Use additional data to train an additional model that provides external supervision.
- Use a traditional Recurrent Neural Network with attention and incorporate the above model.

Proposed Approach

- Transform the problem of Machine Comprehension into the standard question answering task which is subdivided into
  - Answer Selection (AS)
  - Answer Generation
- For Answer Selection , an attention based RNN
- For Answer Generation, the question is combined with each of the candidate answers and transformed into a sentence, and then each of these answers are ranked according to the semantic similarity to the answer selected in the previous stage.
- External supervision is utilized in both the steps.

Approach

Notations

- **Document** is denoted as D.
- **Document Sentences** are denoted as $\{s_0, s_1, ..., s_n\}$
- **Document Questions** are denoted as $Q = \{q_0, q_1, ..., q_m\}$
- Each q_i consists of 4 candidate answers $A_i = \{a_{i0}, ..., a_{i3}\}$

Approach

Mathematical Formulation

- The task of selecting relevant answer to the given question can be expressed as:

$$p(a|q, D) = p(S|q, D)p(a|q, S)$$

- Thus , the task can be divided into two components
  - Answer Selection - Select an answer statement given the question and the document.
  - Answer Generation - Given the question and the and the answer statement , select the best candidate answer.

Approach

Mathematical Formulation

- Objective Function - Regularized log likelihood

$$L_1(\theta; D_{train}) = log \sum_{i=1}^{|D_{train}|} \sum_{j=1}^{|Q|} P(a_{ij}^* | q_{ij}, D_i) - \lambda g(\theta)$$

Approach

External Answer Selection (AS)

- If we have an external AS model with parameter θ_{AS} , then the AS process can be represented as

$$s^*_{AS} = argmax_{s \in D} P(s|q; \theta_{AS})$$

- Thus the External AS component can first be trained on external AS resource and then re-fit on MCTest.
- To balance the trade-off between the external AS Model and the domain specific AS model, we introduce a hyper-parameter η , and the objective function to maximize is :

$$L_2(\theta_{+AS}; D_{train}) = log \sum_{i=1}^{|D_{train}|} \sum_{j=1}^{|Q|} [P(a^*_{ij}|q_{ij}) - \eta L_{AS}(q_{ij}, D_i)] - \lambda g(\theta_{+AS})$$

Approach

External Answer Selection (AS) - Model

(Quoted from the paper)

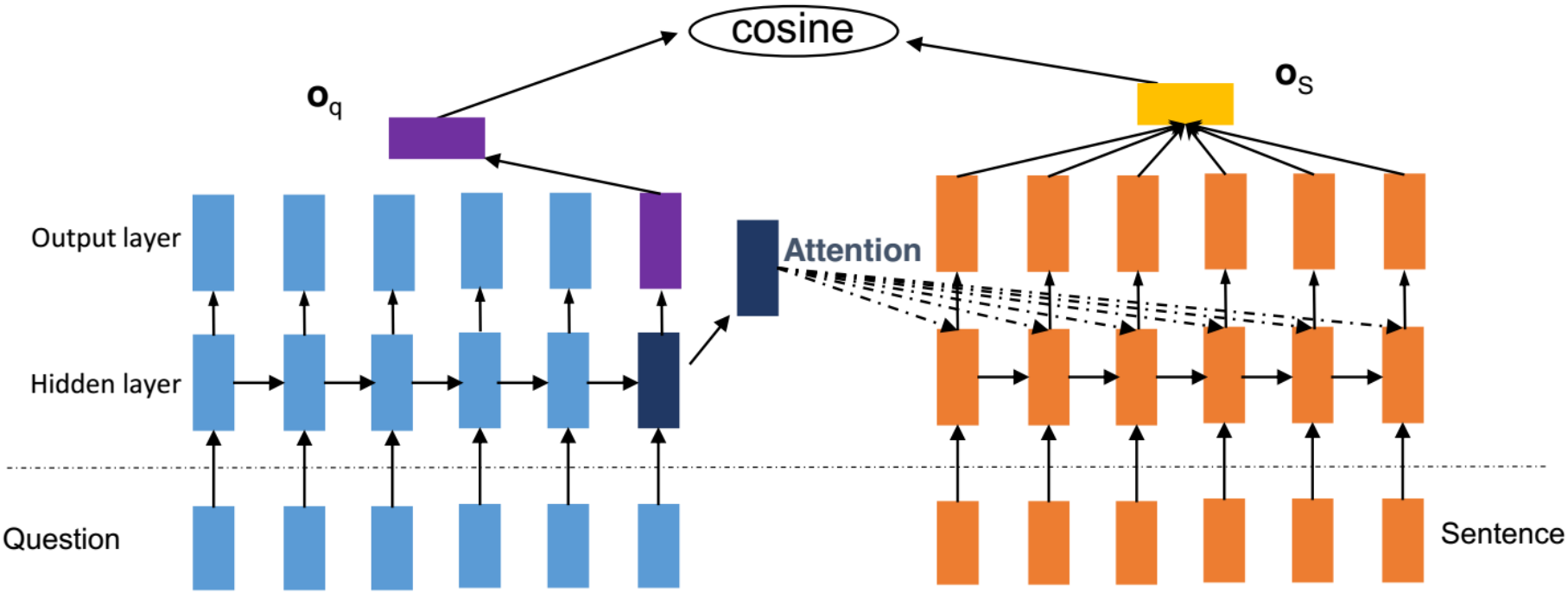
- *... We adopt a smaller neural network architecture that uses semantically expressive recurrent neural network (RNN) to model the question and candidate support sentences.*
- *... In MCTest, the length of most sentences and questions are no more than 10 tokens, the gradient*
- *... We add attention information from question to the candidate sentence output representation as follows* :

$$s_t \propto h_t^T W_{qo} h_n^q$$

$$\widetilde{h}_t = s_t h_t$$

Approach

External Answer Selection (AS) - Model



Approach

External Answer Selection (AS) - Model

- (Quoting from the paper) ... *For the question, we use the last output vector as its representation, for the candidate supporting sentence, we average each time-step output variable* $\widetilde{y}_t$ *to get the final sentence representation* $\mathbf{o}_s$
- The question-sentence pair score is obtained as :

$$\text{SCORE(q,s)} = cosine\{\mathbf{o}_q, \mathbf{o}_s\}$$

- To obtain the distribution of question-sentence pairs i.e. $P(s|q, D; \theta_{RNN})$, the similarity score of all q-s pairs are softmaxed.

Approach

External Answer Selection (AS) - Model

- For training, cross - entropy loss function is used

$$L_{AS}(q, D) = \sum_{s \in D} P(s|q, D; \theta_{RNN}) log Q(s|q, D)$$

where $Q(s|q, D)$ is the supporting sentence probability that the external LSTM AS model predicts

Approach

External Answer Selection (AS) - Model

- WIKIQA was selected as the training corpus because:
  - It matches the MCTest narrative style.
  - It contains not only factoid questions but also non-factoid questions.
  - Relatively Large dataset (more than 20K sentences)
- All named entites in question or answer are replaced with their types (i.e. PERSON, ORGANIZATION, LOCATION)
- An attention based LSTM model is used (similar to that explained previous in Answer Selection Model)
- Instead of Cosine similarity , Geometric mean of Euclidean and Sigmoid Dot (GESD) is used to measure similarity between two representations

$$GESD(x, y) = \frac{1}{1+||x-y||} \cdot \frac{1}{1+exp(-\gamma(xy^T+c))}$$

Approach

External Answer Generation Knowledge

- At this stage, we have the supporting sentence probability and consequently the most confident supporting sentence s.
- This sentence must be combined with the question q_i to get the final answer.
- The problem is transformed into an RTE problem.
- **RTE : Recognizing Textual Entailment** - Determining the truth of one text fragment given another (true) text fragment.
- Thus, each question-answer pair is first transformed into a statement and then an external RTE-enhanced method is used to measure the relationship between the sentence and the candidate statement.

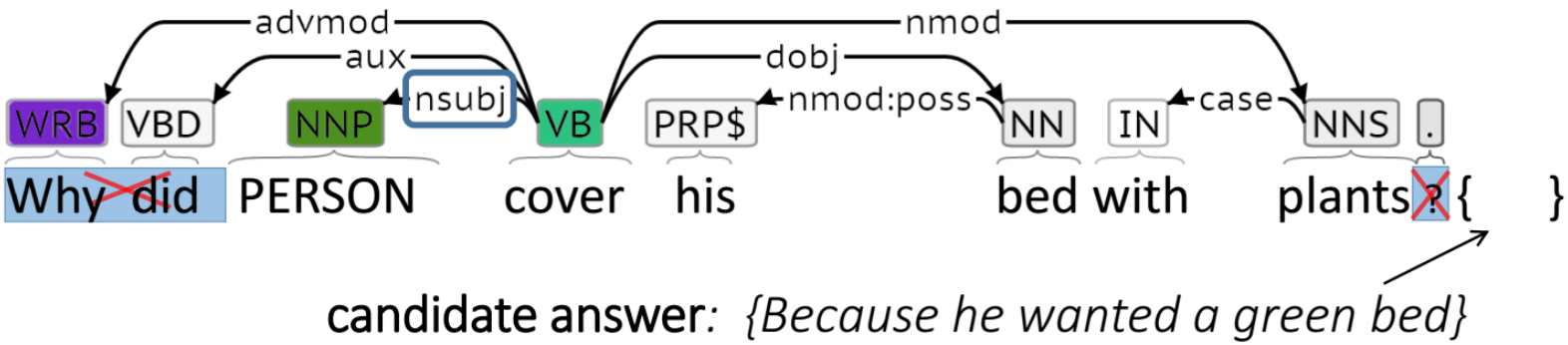
Approach

External Answer Generation Knowledge

Question-Answer Pair to Sentence Transformation

- A rule based system is designed to perform the above task
- StanfordCoreNLP is used to get the constituency tree and Named entities of the question.
- If there exists a NNP with child nodes DT+NN in constituency parsing tree, or a named entity with type PERSON, we transform these words to a special symbol PERSON.
- Some additional rules are designed to convert each question based on the POS of a constituents or dependency relation between two words.

Question-Answer Pair to Sentence Transformation



For e.g. If the question type is why , the POS of the root in dependency tree is VB , the root is located between the question word why and the named entity 'PERSON', then all the words before the 'PERSON' should be deleted and add **'because' + answer**

Recognizing Textual Entailment

- The premise and hypothesis may have no words in common or the linguistic representation might be very different. Thus we use two models, a linguistic feature based model and an external RTE model.
- Let the parameters learn from the external RTE resource be θ_{RTE} and those learned from the linguistic features be θ_1
- The inference from the two models can be combined as follows :

$$P(a|s, D) = [\beta P(s_q|s; \theta_1) + (1 - \beta)P(s_q|s; \theta_{RTE})]$$

- When we cannot infer the entailment from simple linguistic features, we resort to external RTE to judge the entailment probability.
- β is not a hyper-parameter, but is computed as follows:

$$\beta = similarity(s_q^-, s)$$

- Two types of similarity features are used:
  - Constituency Match
  - Dependency Match

Approach

External Answer Generation Knowledge

- Finally , the combined objective function to be maximized looks as follows:

$$L_3(\theta_{+AS+RTE}; D_{train}) =$$

$$log \sum_{i=1}^{|D_{train}|} \sum_{j=1}^{|Q|} [P(a_{ij}^*|q_{ij}) + \eta L_{AS}(q_{ij}, D_i)] - \lambda g(\theta_{+AS+RTE})$$

where

$$P(a_{ij}^*|q_{ij}) = P(s|q, D; \theta_{RNN}) * [\beta P(s_q|s; \theta_1) + (1 - \beta) P(s_q|s; \theta_{RTE})]$$

Approach

Designing External RTE Model

- The Stanford Natural Language Inference (SNLI) dataset is used to train the RTE model.
- The RTE model is similar in design to the Answer Selection Model (AS) as discussed earlier

Outline

Experiments

Evaluation - Measure

- Evaluation Measure
  - For Answer Selection, **MAP**(Mean Average Precision) and **MRR**(Mean Reciprocal Rank) is used.
  - For Answer Generation / RTE simple accuracy is used as evaluation measure.
- Data
  - MCTest is inherently divided into two parts MC160 and MC500 (with the total number of stories being 660)

Experiments

Baselines

1. Sliding Window - Uses a window over document to get bag of words similarity between question+hypothesized answer and document.
2. Sliding Window + Word Distance - Word Distance simply subtracted from the sliding-window score
3. Sliding Window + Word Distance + RTE - Uses off the shelf RTE system in addition to the above
4. Dynamic Memory Networks
5. Discourse Parser to model the relationship between two selected sentences.
6. Extensive features with frames arugments matching and syntax matching as similarity scores.
7. Enhancement of the Sliding Window Method
8. Structural SVM that model the alignment between document sentences and statement as hidden variable.

Outline

Results

MCTest

System	MC160			MC500		
	One	Multiple	All	One	Multiple	All
Sliding Window	64.73	56.64	60.41	58.21	56.17	57.09
Sliding Window+Word Distance	76.78	62.50	67.50	64.00	57.46	60.43
Sliding Window+Word Distance+RTE	76.78	62.50	69.16	68.01	59.45	63.33
[Kapashi and Shah, 2015]	-	-	36.0	-	-	34.2
[Narasimhan and Barzilay, 2015]	82.36	65.23	73.23	68.38	59.90	63.75
[Wang and McAllester, 2015]	84.22	67.85	75.27	72.05	67.94	69.94
[Smith *et al.*, 2015]	78.79	**70.31**	75.77	69.12	63.34	65.96
[Sachan *et al.*, 2015]	-	-	-	67.65	**67.99**	67.83
without External Knowledge ($\beta = 1, \eta = 0$)	40.39	37.94	39.08	38.40	33.13	31.33
without External AS knowledge ($\eta = 0$)	41.07	40.63	40.83	49.63	28.05	32.83
without External RTE knowledge ($\beta = 1$)	74.11	64.06	68.75	57.72	50.91	53.00
Final Model	**88.39**	64.84	**75.83**	**79.04**	63.51	**70.96**

Table 1: Result on MCTest test data

From the equation

$$L_2(\theta_{+AS}; D_{train}) = log \sum_{i=1}^{|D_{train}|} \sum_{j=1}^{|Q|} [P(a_{ij}^*|q_{ij}) - \eta L_{AS}(q_{ij}, D_i)] - \lambda g(\theta_{+AS})$$

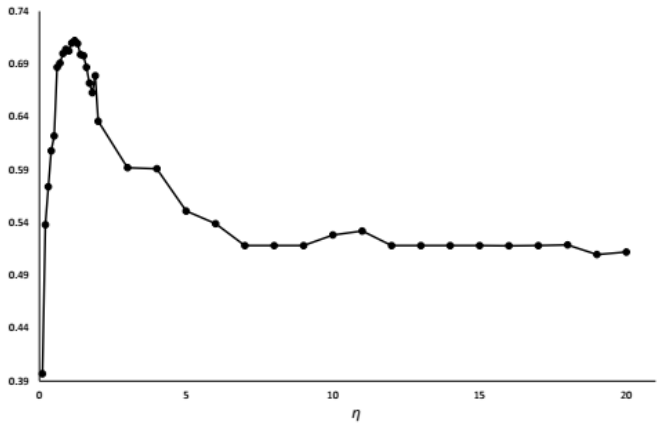


Figure 3: The result of different η value in MC500

References I


Phil Blunsom.
Teaching Machines to Read and Comprehend - Lisbon Summer School, 2015.
http://lxmls.it.pt/2015/lxmls15.pdf, 2015.


Matthew Richardson, Christopher JC Burges, and Erin Renshaw.
Mctest: A challenge dataset for the open-domain machine comprehension of text.
In *EMNLP*, volume 3, page 4, 2013.


Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston.
The goldilocks principle: Reading children's books with explicit memory representations.
arXiv preprint arXiv:1511.02301, 2015.

References II

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom.
Teaching machines to read and comprehend.
In *Advances in Neural Information Processing Systems (NIPS)*, 2015.