TEXAS
The University of Texas at Austin

REFRAMING OTHER PROBLEMS AS QA

CS 395T: Topics in Natural Language Processing

11/19/2020

Ryo Kamoi and *Yejin Cho*, The University of Texas at Austin

Reframing other problems as QA

- 1) **CorefQA: Coreference Resolution as Query-based Span Prediction** (Wu et al., ACL 2020)
  - Coreference resolution → Span prediction as in QA task

- 2) **Zero-Shot Relation Extraction via Reading Comprehension** (Levy et al, CoNLL 2017)
  - Relation extraction → Reading comprehension QA task

CorefQA: Coreference Resolution as Query-based Span Prediction

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li

CorefQA: Overview

- CorefQA formulates the Coreference Resolution problem as a **span prediction task**, like in **question answering**

Original Passage

In addition , *many people* were poisoned when **toxic gas** was released. *They* were poisoned and did not know how to protect *themselves* against **the poison**.

Our formulation

Q1: Who were poisoned when toxic gas was released?
A1: [*They*, *themselves*]
Q2: What was released when many people were poisoned?
A2: [**the poison**]
Q3: Who were poisoned and did not know how to protect themselves against the poison?
A3: [*many people*, *themselves*]
Q4: Whom did they not know how to protect against the poison?
A4: [*many people*, *They*]
Q5: They were poisoned and did not know how to protect themselves against what?
A5: [**toxic gas**]

Background: Coreference Resolution

Recent Approaches

1. Clustering for mentions from **parsers and hand-engineered mention proposal algorithms**
2. **End-to-end** fashion by jointly detecting mentions and predicting coreferences

Background: Coreference Resolution

Earlier Neural-based Models (e.g. Wiseman et al., 2016)

- Assume that a sequence of mentions are given (e.g. **syntactic parser**)
- Use representations from neural models for clustering

Background: Coreference Resolution

End-to-End method (e.g. Lee et al., 2017)

- **Syntactic parsers are not required**
- Jointly learns which spans are entity mentions and how to best cluster them

1. Computes embedding representations of spans
2. Low-scoring spans are pruned (mention proposal)
3. Compute clustering score

Background: Problems in Prior Work

- Mentions left out at the mention proposal stage can never be recovered
- Only based on mention representations from the output layer and lacks the connection between mentions and their contexts

CorefQA: Overview

- CorefQA formulates the Coreference Resolution problem as a **span prediction task**, like in **question answering**

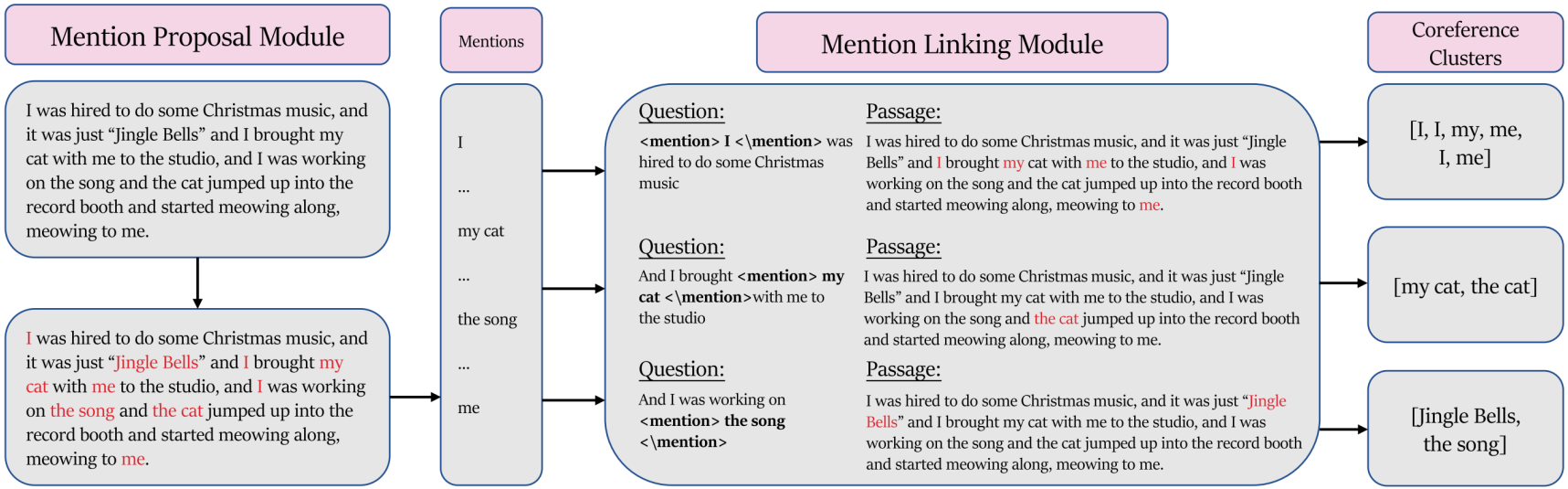
Original Passage

In addition , *many people* were poisoned when **toxic gas** was released. *They* were poisoned and did not know how to protect *themselves* against **the poison**.

Our formulation

Q1: Who were poisoned when toxic gas was released?
A1: [*They*, *themselves*]
Q2: What was released when many people were poisoned?
A2: [**the poison**]
Q3: Who were poisoned and did not know how to protect themselves against the poison?
A3: [*many people*, *themselves*]
Q4: Whom did they not know how to protect against the poison?
A4: [*many people*, *They*]
Q5: They were poisoned and did not know how to protect themselves against what?
A5: [**toxic gas**]

CorefQA


Mention Proposal Module
I was hired to do some Christmas music, and it was just "Jingle Bells" and I brought my cat with me to the studio, and I was working on the song and the cat jumped up into the record booth and started meowing along, meowing to me.
I was hired to do some Christmas music, and it was just "Jingle Bells" and I brought my cat with me to the studio, and I was working on the song and the cat jumped up into the record booth and started meowing along, meowing to me.
Mentions
I
...
my cat
...
the song
...
me
Mention Linking Module
Question:
<mention> I <\mention> was hired to do some Christmas music
Passage:
I was hired to do some Christmas music, and it was just "Jingle Bells" and I brought my cat with me to the studio, and I was working on the song and the cat jumped up into the record booth and started meowing along, meowing to me.
Question:
And I brought <mention> my cat <\mention>with me to the studio
Passage:
I was hired to do some Christmas music, and it was just "Jingle Bells" and I brought my cat with me to the studio, and I was working on the song and the cat jumped up into the record booth and started meowing along, meowing to me.
Question:
And I was working on <mention> the song <\mention>
Passage:
I was hired to do some Christmas music, and it was just "Jingle Bells" and I brought my cat with me to the studio, and I was working on the song and the cat jumped up into the record booth and started meowing along, meowing to me.
Coreference Clusters
[I, I, my, me, I, me]
[my cat, the cat]
[Jingle Bells, the song]

CorefQA: Mention proposal

- Similar to Lee et al. (2017)
- Use the SpanBERT to obtain input representations
- Considers all spans up to a maximum length L as potential mentions
- Prune the candidate spans by using calculated scores

$$s_{\mathrm{m}}(i) = \mathrm{FFNN}_{\mathrm{m}}\left(\left[\boldsymbol{x}_{\mathrm{FIRST}(i)}, \boldsymbol{x}_{\mathrm{LAST}(i)}\right]\right)$$

CorefQA: Span Prediction

- Similar to Li et al. (2019)
- Generates a BIO tag for each token
  - Beginning (B), inside (I) and outside (O) of a coreferent mention

CorefQA: Data Augmentation

- Hypothesis: the reasoning required for QA is also useful for coreference resolution
- Pretrain the mention linking network on
  - Quoref dataset (Dasigi et al., 2019b)
  - SQuAD dataset (Rajpurkar et al., 2016b).

CorefQA: Advantages

- Left-out mentions can still be retrieved at the span prediction stage
- Span prediction requires a more thorough and deeper examination of the lexical
- Allows us to take advantage of existing question answering datasets

CorefQA: Experiments - Metrics

MUC (Vilain et al., 1995)

- A link based metric
- K is the key entity set

key entities. *MUC* recall is defined as:

$$\text{Recall} = \frac{\sum_{k_i \in K}(|k_i| - |p(k_i)|)}{\sum_{k_i \in K}(|k_i| - 1)}$$

where $p(k_i)$ is the set of partitions that is created by intersecting k_i with the corresponding response entities. *MUC* precision is computed by switching the role of the key and response entities.

CorefQA: Experiments - Metrics

B^3 (Bagga and Baldwin, 1998)

- A mention based metric

- K is the key entity set and
  R is the response entity set

recall/precision of the individual mentions. For each mention m in the key entities, B^3 recall considers the fraction of the correct mentions that are included in the response entity of m. B^3 recall is computed as follows:

$$\text{Recall} = \frac{\sum_{k_i \in K} \sum_{r_j \in R} \frac{|k_i \cap r_j|^2}{|k_i|}}{\sum_{k_i \in K} |k_i|}$$

Similar to MUC, B^3 precision is computed by switching the role of the key and response entities.

CorefQA: Experiments - Metrics

$CEAF_{\phi 4}$ (Luo, 2005)

- K is the key entity set and R is the response entity set

vice versa. *CEAF* uses a similarity measure (ϕ) to evaluate the similarity of two entities. It uses the Kuhn-Munkres algorithm to find the best one-to-one mapping of the key to the response entities (g^*) using the given similarity measure. Assuming K^* is the set of key entities that is included in the optimal mapping, recall is computed as:

$$\text{Recall} = \frac{\sum_{k_i \in K^*} \phi(k_i, g^*(k_i))}{\sum_{k_i \in K} \phi(k_i, k_i)} \quad (1)$$

For computing *CEAF* precision, the denominator of Equation 1 is changed to $\sum_{R_i \in R} \phi(r_i, r_i)$.

CorefQA: Experiments

E2E Methods

	MUC			B^3			$CEAF_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
e2e-coref(Lee et al., 2017)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
c2f-coref + ELMo (Lee et al., 2018)	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
EE + BERT-large (Kantor and Globerson, 2019)	82.6	84.1	83.4	73.3	76.2	74.7	72.4	71.1	71.8	76.6
c2f-coref + BERT-large (Joshi et al., 2019b)	84.7	82.4	83.5	76.5	74.0	75.3	74.1	69.8	71.9	76.9
c2f-coref + SpanBERT-large (Joshi et al., 2019a)	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
CorefQA + SpanBERT-base	85.2	87.4	86.3	78.7	76.5	77.6	76.0	75.6	75.8	79.9 (+0.3)
CorefQA + SpanBERT-large	**88.6**	**87.4**	**88.0**	**82.4**	**82.0**	**82.2**	**79.9**	**78.3**	**79.1**	**83.1** (+3.5)

Table 1: Evaluation results on the English CoNLL-2012 shared task. The average F1 of MUC, B^3, and $CEAF_{\phi_4}$ is the main evaluation metric. Ensemble models are not included in the table for a fair comparison. P, R and $F1$ in the first row represent precision, recall and F1 score respectively.

CorefQA: Experiments

BERT

Lee et al. (2018)

	Avg. F1	Δ
CorefQA	83.4	
–– SpanBERT	79.6	-3.8
–– Mention Proposal Pre-train	75.9	-7.5
–– Question Answering	75.0	-8.4
–– Quoref Pre-train	82.7	-0.7
–– SQuAD Pre-train	83.1	-0.3

Table 3: Ablation studies on the CoNLL-2012 development set. SpanBERT token representations, the mention-proposal pre-training, and the question answering pre-training all contribute significantly to the good performance of the full model.

CorefQA: Experiments

Speaker modeling strategies

- This paper: Speaker as input directly concatenates the speaker's name
- Previous work: Speaker as feature converts speaker information into binary features indicating whether two mentions are from the same speaker

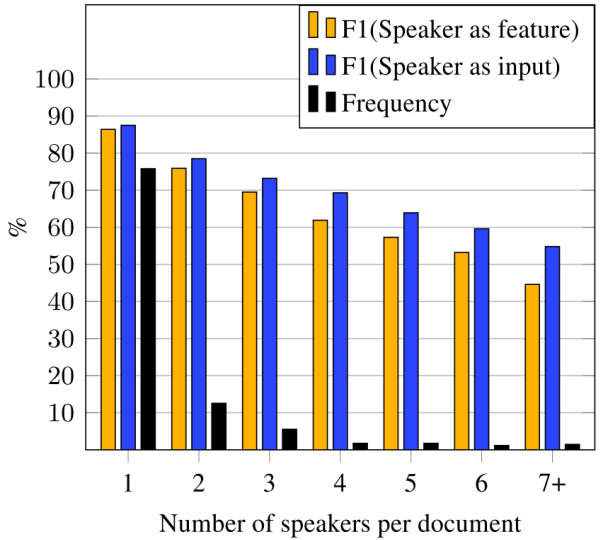


Figure 3: Performance on the development set of the CoNLL-2012 dataset with various number of speakers. F1(Speaker as feature): F1 score for the strategy that treats speaker information as a mention-pair feature. F1(Speaker as input): F1 score for our strategy that treats speaker names as token input. Frequency: percentage of documents with specific number of speakers.

CorefQA: Experiments

- Keep up to λn (where n is the document length) spans with the highest mention scores
- The proposed method is less sensitive to smaller values of λ because missed mentions can still be retrieved later

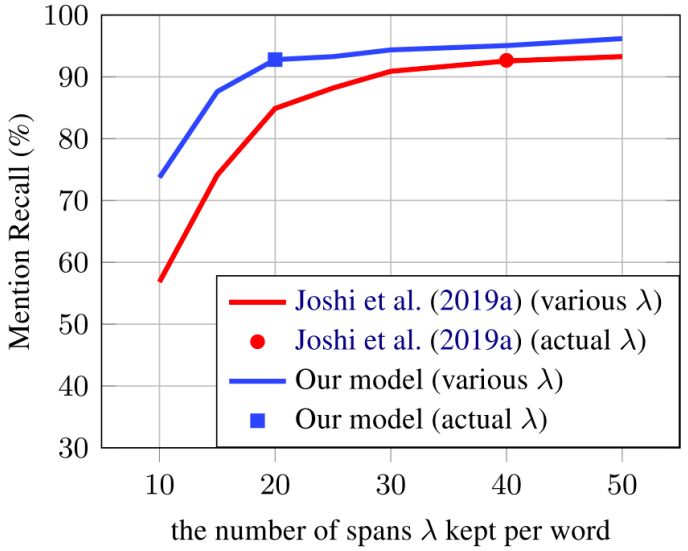


Figure 4: Change of mention recalls as we increase the number of spans λ kept per word.

CorefQA: Experiments

- Successful examples of the proposed method
- 1: The answer from a longer distance
- 3: The use of speaker information

1	[**Freddie Mac**] is giving golden parachutes to two of its ousted executives. . . . Yesterday Federal Prosecutions announced a criminal probe into [**the company**].
2	[**A traveling reporter**] now on leave and joins us to tell [**her**] story. Thank [**you**] for coming in to share this with us.
3	*Paula Zahn:* [**Thelma Gutierrez**] went inside the forensic laboratory where scientists are trying to solve this mystery. *Thelma Gutierrez:* In this laboratory alone [**I**] 'm surrounded by the remains of at least twenty different service members who are in the process of being identified so that they too can go home.

Table 4: Example mention clusters that were correctly predicted by our model, but wrongly predicted by c2f-coref + SpanBERT-large. Bold spans in brackets represent coreferent mentions. Italic spans represent the speaker's name of the utterance.

Discussion

- Error Analysis
  - mentions left out at the mention proposal stage
  - distant mentions
- Are results without ``Speaker as Input" better than baseline methods?
- Evaluation on other datasets

Zero-Shot Relation Extraction via Reading Comprehension

[CoNLL 2017] Omer Levy, Minjoon Seo, Eunsol Choi, Luke Zettlemoyer

Relation Extraction

- Task: Given some **unstructured text**, predict relations between entities
  - Ultimate goal: Fill in the **information gap** (missing links) in a knowledge base (KB)
- Challenge
  - **Intractability**: How many relations exist in language/world? 🧐
    - **Not all relations** can be seen during **training**
    - If we only care about a fixed set of **pre-defined** relation types, data collection and supervised learning for such specific relations are feasible
  - However, we want to go beyond by generalizing to **unseen relations**
    - ➡ ***Zero-shot*** *setting* relation extraction

Proposed Idea

- Relation extraction as **reading comprehension QA**

Relation type *(KB relation)*

Questions

Reading comprehension QA
using generated questions

educated_at (x, y)
R(x, y)

- "Where did x study?"
- "Which university did x graduate from?"
q_x

Entity x: *Turing*
Text: *Turing* obtained his PhD from Princeton.
Questions: {Where did x study?, Which univ. did x...} (q_x)
Answer span from text: ***Princeton*** *(=> Entity* ***y****)*

Relation Extraction as QA

- The **biggest charm** of reducing RE as QA?
  - Enables **zero-shot learning**
    - i.e., Generalizing to new relations unobserved during training
- Specifically, this paper proposes to:
  - **Train** a reading comprehension QA model with labeled data of N relation types (**R_1-R_N**)
  - **Test** with unseen, unspecified (zero-shot) relation types (**R_{N+1}**)
    - No additional data feeding for new relations
    - Instead, simply use the **QA model** trained with R_N to answer adequate **questions** in natural language

Approach

- Task: **Slot-filling** for relation extraction

Given information	KB relation **R**	***occupation*** (**e**, ?)
	Entity **e**	Steve Jobs
	Sentence **s**	*"Steve Jobs was an American businessman, inventor, and industrial designer."* Collected from WikiReading Hewlett et al. (2016)
Querification	Question **q**	Q: What did Steve Jobs do for a living?
Answer prediction	Answer text span set **A**	A: {businessman, inventor, industrial designer} (A=∅, if not answerable from the given sentence s)

Approach

- **Schema Querification**
  - Idea: No fixed schema used as in previous relation extraction studies
    - Instead, any schema (or any relation) can be asked as a **question**
  - Convert a relation **R(e, ?)** to natural language **questions**

occupation(e, ?)	"What did **x** do for a living?"	"What did **Steve Jobs** do for a living?"
Relation R(e, ?)	question template q_x	instantiation with relevant entity e

  - Transforms relation extraction dataset to reading comprehension dataset

Approach

- **Reading comprehension QA** using querified schemas
  - **Train** a reading comprehension model with the transformed dataset
    - Input: sentence **s** and question **q**
    - Output: a set of answer spans in sentence **s** (**A**)
  - **Test** phase: zero-shot scenario
    - Input sentence:
      - “Turing and colleagues came up with a method for efficiently *deciphering* the Enigma.”
    - Input relation: deciphered(e, ?)
    - Question: "Which code did x break?" (x instantiated with 'Turing')
    - Answer: Enigma

Dataset

Schema questions

Slot-filling data

Relation	Question	Sentence & Answers
educated_at	What is **Albert Einstein**'s alma mater?	**Albert Einstein** was awarded a PhD by the **University of Zürich**, with his dissertation titled...
occupation	What did **Steve Jobs** do for a living?	**Steve Jobs** was an American **businessman**, **inventor**, and **industrial designer**.
spouse	Who is **Angela Merkel** married to?	**Angela Merkel**'s second and current husband is quantum chemist and professor **Joachim Sauer**, who has largely...

- Each instance consists of:
  - A **relation**, a **question**, a **sentence**, and a set of **answer spans** (underlined in the figure)
- 1) **Slot-filling data**: collected using distant supervision on existing QA dataset (WikiReading)
- 2) **Schema questions**: crowdsourced data collection and verification

Data Collection (1) **Slot-Filling** Data

- **WikiReading** (Hewlett et al. 2016):
  - Reading comprehension dataset
  - Collected by aligning Wikipedia article to each relation R(e,a)
  - Each instance consists of document D, relation R, entity e, and answer a
- Distant supervision on WikiReading:
  - From each document, select the **first sentence s** that contains the entity **e** and the specified answer **a**
  - Merge all answers for R(e, ?) given s into a set of answer spans A

Data Collection (2) **Schema Querification**

- Collected by **crowdsourced** workers on Amazon Mechanical Turk
- Two phases: Collection + Verification
  - a. **Collection**
    - Given 4 **example sentences**, each annotator should come up with **3 questions** about **X** whose answer is the underlined span, considering each sentence.
    - (1) The wine is produced in the **X** region of **France**.
      (2) **X**, the capital of **Mexico**, is the most populous city in North America.
      (3) **X** is an unincorporated and organized territory of **the United States**.
      (4) The **X** mountain range stretches across **the United States** and **Canada**.

➡ *"In which country is x?"*

Data Collection (2) **Schema Querification**

- b. **Verification**
  - **Quality control** for the collected question templates
  - **Reverse** setting:
    - Given a question (instantiated with entity e), annotators **find the answer** from sentence s
    - If their answer **matches with A**, then the question template is verified as **valid**
  - Discard the template if not matched for less than 6 out of 10 times

- **Collected data size:** 1.2k verified question templates with 120 relations
  - After combining with the slot-filling data and instantiation with entities: >30M examples

Data Collection: **Negative** Examples

- **Negative examples**: Unanswerable question-sentence pairs (A=∅)
  - Additionally collected to help relation extraction (c.f., deviation from RC setting)
  - Idea: **Intentionally mismatch** a question q and a sentence s (Morales et al., 2016)
    - Both of them mention the same **entity e**
    - However, q should be **unanswerable from s**
      - q: "Who is Angela Merkel married to?"
      - s: "Angela Merkel is a German politician who is currently the Chancellor of Germany."
  - Created >2M negative samples
  - All training and testing sets had 1:1 ratio of positive and negative examples

Data Collection: **Discussion**

- **Limitation** of data collection in several previous studies
  - SimpleQA (Bordes et al., 2015), QA-SRL (He et al., 2015), and more
  - **High cost** for data collection: the cost **linearly grows** with the number of **instance**
    - Thus, difficult to build large-scale dataset

- On the other hand, **schema querification** enables **scaling up**
  - Collected 300x larger dataset than SimpleQA
  - Main reason: annotates on the **relation**-level and abstracts each entity as a variable
  - The first approach to robustly collect QA dataset using **schema-level crowdsourcing**

Model: Modified **BiDAF** (Seo et al., 2016)

- **Adapted** a reading comprehension model BiDAF (Seo et al., 2016) to the current task
  - **Difference** between reading comprehension (RC) and current task
    - RC: Always assumes the answer to be some span of a given sentence
    - Current: Model should decide whether the question is answerable or not from the given sentence (i.e., whether the answer exists in the sentence)

Model: Modified BiDAF

- **BiDAF** (Seo et al., 2016)
  - Input: sentence **s**, question **q**
    - Pretrained GloVe word embeddings without finetuning
  - Output: $\mathbf{z}^{\text{start}}, \mathbf{z}^{\text{end}} \in R^N$ (= # of words in the sentence s)
    - **Confidence score** of the start and end positions $\mathbf{y}^{\text{start}}, \mathbf{y}^{\text{end}}$ of the answer span in s
    - Apply softmax to convert to pseudo-probabilities $\mathbf{p}^{\text{start}}, \mathbf{p}^{\text{end}}$
      - ➡ Predicts the most probable answer span in **s**
  - Algorithm: Bi-LSTM with attention encodes and aligns **s** and **q**

Model: Modified BiDAF

- **Modification** of BiDAF
  - Added a **bias** b at the end of each confidence score vectors $\mathbf{z}^{\text{start}}, \mathbf{z}^{\text{end}} \in \mathbf{R}^N$
    - i.e., model's confidence that the answer has no start or end, respectively
  - Again, apply softmax to new score vectors ($\in \mathbf{R}^{N+1}$) to compute pseudo-probability distributions $\tilde{\mathbf{p}}^{\text{start}}, \tilde{\mathbf{p}}^{\text{end}}$
  - Use the probability of the two biases to compute **null answer probability** P(a=∅)

$$P(a = \emptyset) = \tilde{\mathbf{p}}^{start}_{N+1} \tilde{\mathbf{p}}^{end}_{N+1}$$

  - If P(a=∅) > P(the most likely span), then decide the instance as '**not answerable**'
  - Works as a dynamic **per-example threshold** for decision (↔ global threshold)

Experiments

- Experimental settings with **three test subjects**:
  - A. Unseen **entities**
  - B. Unseen **question templates**
  - C. Unseen **relations**

Least challenging

Most challenging

- **Evaluation metrics**
  - Precision: (# true positives) / (# times a model returned non-null answers)
  - Recall: (# true positives) / (# instances which are answerable)

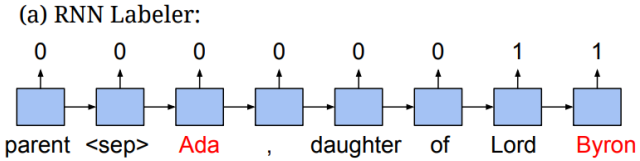
Experiments: **Variations**

- Five **variations** of **our BiDAF systems**
  - Vary on **how a relation is represented/queried** during train [#1-4] and test [#5] time

#	Variation	Question Template	Description	Example	Expectation
1	**KB** Relation	X	Provide **relation indicator** instead of question	R_{17}	Will generalize well on **unseen entities** but will fail on **unseen relations**
2	**NL** Relation	X	Provide **relation name** instead of question	"educated at"	
3	**Single** Template	O	[**Weak** variant of proposed model] Allow **only one question template** for each relation during training	q: "Where did x study?"	
4	**Multiple** Template	O	[**Full** variant of proposed model] Allow **multiple variants of questions** for each relation during training	q: {"Where did x study?", "Which university did x graduate from?", ... }	Will have better paraphrasing skill than single template
5	Question **Ensemble**	O	**Per test instance**, ask three different forms of questions about one relation and choose the answer with the highest sum of scores		

Experiments: **Other baselines**

- 1) **Random NE**
  - Random baseline
  - From the given sentence, simply choose an entity that is not present in the question
- 2) **RNN Labeler**
  - Answer extraction model in WikiReading (Hewlett et al. 2016)
  - At each timestep an RNN cell decides whether the current word is part of the answer or not



- 3) **Miwa and Bansal** (2016)
  - Off-the-shelf end-to-end relation extraction system that worked well on multiple benchmarks
  - Represent each relation as an indicator
  - Unseen relations cannot be extracted (as many other RE models)

Experiment A. Unseen **Entities**

	Precision	Recall	F1
Random NE	11.17%	22.14%	14.85%
RNN Labeler	62.55%	62.25%	62.40%
Miwa & Bansal	96.07%	58.70%	72.87%
KB Relation	89.08%	91.54%	90.29%
NL Relation	88.23%	91.02%	89.60%
Single Template	77.92%	73.88%	75.84%
Multiple Templates	87.66%	91.32%	89.44%
Question Ensemble	88.08%	91.60%	89.80%

Table 1: Performance on unseen entities.

- **All five** of our models generalize well to new entities and texts
  - All outperform both of the off-the-shelf relation extraction systems
  - **Single Template** < all four others
- Error analysis on **Multiple Templates**
  - Only a **small** portion (18%) of the sampled errors are **pure model errors**,
  - while the rest are mostly due to trivial annotation errors.

Experiment B. Unseen **Question Templates**

	Precision	Recall	F1
Seen	86.73%	86.54%	86.63%
Unseen	84.37%	81.88%	83.10%

Table 2: Performance on seen/unseen questions.

- For each relation, **one question template** is **held-out** for evaluation (one for dev, another for test)
  - e.g.,"What did **x** do for a living?" --> in *train* only "What is **x**'s job?" --> in *test* only
- Trained and tested **Multiple Templates** for each of 10-folds of dataset
  - **Seen**: Test performance when unseen templates **replaced with templates seen during training**
  - **Unseen**: Selectively measured performance on **unseen question templates**
- Result:
  - Our approach **generalizes** on **unseen question templates**

Experiment C. Unseen **Relations**

- **Fully zero-shot** environment
  - **None** of the evaluated relations is observed during training
- Results
  - Two RE baselines which represents each **relation** as an **indicator** (and not natural language) clearly **fails** in zero-shot setting
  - **Multiple** Templates show **big improvement** from others, including **Single** Template
    - Thanks to rich exposure to **diverse phrasings** of the **same relation**

	Precision	Recall	F1
Random NE	9.25%	18.06%	12.23%
RNN Labeler	13.28%	5.69%	7.97%
Miwa & Bansal	100.00%	0.00%	0.00%
KB Relation	19.32%	2.54%	4.32%
NL Relation	40.50%	28.56%	33.40%
Single Template	37.18%	31.24%	33.90%
Multiple Templates	43.61%	36.45%	39.61%
Question Ensemble	45.85%	37.44%	41.11%

Table 3: Performance on unseen relations.

Experiment C. Unseen **Relations**

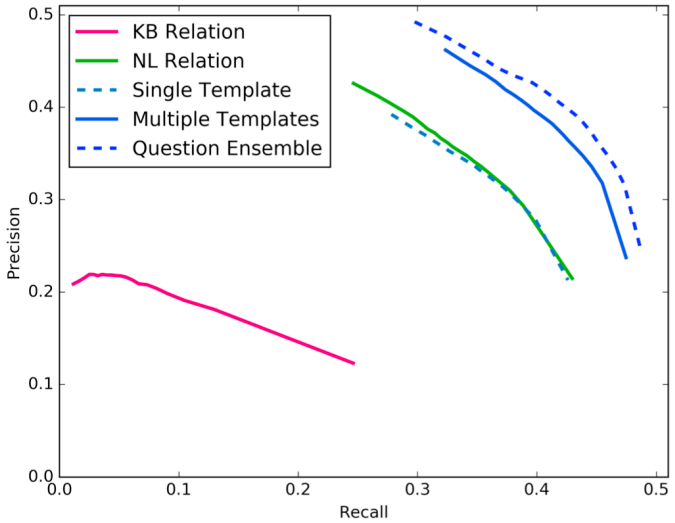


Figure 4: Precision/Recall for unseen relations.

- **Precision-recall curve** when applied a varying range of **global threshold** p_{min} for confidence score
  - Whenever the best answer's score is lower than p_{min}, then decide 'not answerable from text'
- Observation
  - Question Ensemble
    > Multiple Templates
    > Single Templates = NL Relation
    >>>> KB Relation

Qualitative Analysis

- How does the proposed model extract **unseen relations**?
  - For better understanding they analyzed 100 random samples (60 pos, 40 neg)

- 1) **Negative** samples (i.e., not answerable from sentence)
  - 35% of them had a **distractor** in a sentence
    - 'Distractor': an incorrect answer of **correct answer entity type** (e.g., person, time)
  - Most negative samples are easy, but some with a distractor are non-trivial

Qualitative Analysis

- 2) **Positive** samples (i.e., answerable)

Part of the question **literally appears** in the sentence **s**

Takes **typical rephrasing** methods used across different relations

Takes **unique rephrasing** method closely tied to the specific relation

Verbatim	Relation	András Dombai **plays for** what team? András Dombai... ...currently **plays** as a goalkeeper **for** *FC Tatabánya*.
	Type	Which **airport** is most closely associated with Royal Jordanian? Royal Jordanian Airlines... ...from its main base at *Queen Alia International* ***Airport***...
Global	Relation	Who was responsible for **directing** Les petites fugues? Les petites fugues is a 1979 Swiss comedy film **directed by** *Yves Yersin*.
	Type	**When** was The Snow Hawk released? The Snow Hawk is a ***1925*** film...
Specific	Relation	Who **started** Fürstenberg China? The Fürstenberg China Factory **was founded**... ...**by** *Johann Georg von Langen*...
	Type	What **voice type** does Étienne Lainez have? Étienne Lainez... ...was a French operatic ***tenor***...

Figure 5: The different types of discriminating cues we observed among positive examples.

Qualitative Analysis

- **Distribution** of cues (Table 4)
  - Analyzed the **most important** cues for solving each instance
    - **Type** cues > **Relation** cues
    - **Specific** (50%) > Global (33%) > Verbatim (17%)

- **Accuracy** by cue types (Table 5)
  - **Relation** column (left): No marked tendency (agnostic)
  - **Type** column (right): Catches **global cues** much better than others
  - ➡ Thus, the generalizability to new relations could be attributed to **global type cues** and **relation paraphrase detection of all types** (∵ balanced accuracy)

	Relation	Type
Verbatim	12%	5%
Global	8%	25%
Specific	22%	28%

Table 4: The distribution of cues by type, based on a sample of 60.

	Relation	Type
Verbatim	*43%*	*33%*
Global	*60%*	73%
Specific	46%	18%

Table 5: Our method's accuracy on subsets of examples pertaining to different cue types. Results in *italics* are based on a sample of less than 10.

Takeaways

- The contributions of this work are as follows:
  - 1) Reframing as QA
    - Creatively **repurposed** relation extraction as **reading comprehension** (QA) problem using **schema querification** approach
    - Showed neural QA model (BiDAF) can effectively adapted for relation extraction
  - 2) Enabled **zero-shot** relation learning by adopting **span QA** framework
  - 3) Categorized and analyzed three **different types of discriminative cues** (Verbatim, Global, Specific) that can be used for relation extraction

More Recent Works

- **Relation Extraction**
  - Li et al. ACL 2019.
    - A follow-up work heavily motivated by the current paper
    - Entity and relation extraction tasks framed as a problem of **multi-turn QA**
  - Alt et al. AKBC 2019: a pre-trained Transformer based LM fine-tuned on the RE task

- **QA application to other NLP tasks**
  - Gardener et al. Question Answering is a Format; When is it Useful?, ArXiv 2019.
    - Argument: QA should be considered a ***format*** instead of a ***task*** in itself
    - Multiple values of QA: fills information needs in natural language, as a probing tool, and as a storage for transferrable linguistic knowledge

Discussions

- Some **assumptions** in the task
  - The task setup assumes three things: **s** (a sentence), **r** (a KB relation), **e** (an entity).
    - i.e., R(e, ?) and a sentence that mentions the entity e.
  - Is this a realistic assumption for RE? How do we retrieve the right sentence **s**?
  - How can we extend this approach to extract unseen relations from unstructured text alone (without **s**)?

- **Question answering**
  - Do you consider QA as a ***task*** or a ***format***?
  - What are other problems in NLP that could also benefit from QA?


YOU
BELONG
HERE
CNS Diversity and Inclusion • cns.utexas.edu/diversity

Questions?