Team SVM^{rank}: Leveraging Feature-rich Support Vector Machines for Ranking Explanations to Elementary Science Questions

Jennifer D'Souza[1], Isaiah Onando Mulang'[2], Sören Auer[1]
[1]TIB Leibniz Information Centre for Science and Technology, Hannover, Germany
{jennifer.dsouza|auer}@tib.eu
[2]Fraunhofer IAIS, Sank Augustin, Germany
{isaiah.mulang.onando}@iais.fraunhofer.de

Abstract

The TextGraphs 2019 Shared Task on Multi-Hop Inference for Explanation Regeneration (MIER-19) tackles explanation generation for answers to elementary science questions. It builds on the AI2 Reasoning Challenge 2018 (ARC-18) which was organized as an advanced question answering task on a dataset of elementary science questions. The ARC-18 questions were shown to be hard to answer with systems focusing on surface-level cues alone, instead requiring far more powerful knowledge and reasoning.

To address MIER-19, we adopt a hybrid pipelined architecture comprising a feature-rich learning-to-rank (LTR) machine learning model, followed by a rule-based system for reranking the LTR model predictions. Our system was ranked fourth in the official evaluation, scoring close to the second and third ranked teams, achieving 39.4% MAP.

Question Granite is a hard material and forms from cooling magma. Granite is a type of
Answer igneous rock
Explanation
[rank 1] igneous rocks or minerals are formed from magma or lava cooling
[rank 2] igneous is a kind of rock
[rank 3] a type is synonymous with a kind
[not in gold expl] rock is hard
[not in gold expl] to cause the formation of means to form
[not in gold expl] metamorphic rock is a kind of rock
[not in gold expl] cooling or colder means removing or reducing or decreasing heat or temperature

Table 1: Example depicting the Multi-Hop Inference Explanation Regeneration Task. The multi-hop inference task was formulated around the presence of a lexical overlap (shown as underlined words) between explanation sentences with the question or answer or other correct explanation sentences. Since the lexical overlap criteria was not strictly defined around only the correct explanation candidates (as depicted with the last four explanation candidates), the task necessitated use of additional domain and world knowledge to rule out incorrect explanation candidates.

1 Introduction

The TextGraphs 2019 Shared Task on Multi-Hop Inference for Explanation Regeneration (Jansen and Ustalov, 2019) was organized for the semantic evaluation of systems for providing explanations to elementary science question answers. The task itself was formulated as a ranking task, where the goal was to rerank the relevant explanation sentences in a given knowledge base of over 4,000 candidate explanation sentences for a given pair of an elementary science question and its correct answer. The QA part of the MIER-19 dataset, including the questions and their multiple-choice answers, had been released previously as the AI2 Reasoning Challenge (Clark et al., 2018) dataset called ARC-18. Since answering science questions necessitates reasoning over a sophisticated understanding of both language and the world and over commonsense knowledge, ARC-18 specifically encouraged progress on *advanced reasoning QA* where little progress was made as opposed to factoid-based QA. This was highlighted when sophisticated neural approaches for factoid-based QA (Parikh et al., 2016; Seo et al., 2016; Khot et al., 2018) tested on the ARC-18 did not achieve good results. Now with the MIER-19 task, the ARC-18 objective of testing QA systems for advanced reasoning dives deeper into the reasoning aspect by focusing on reranking explanations for questions and their correct answer choice.

In this article, we describe the version of our system that participated in MIER-19. Systems participating in this task assume as input the question, its correct answer, and a knowledge base of over 4,000 candidate explanation sentences. The task then is to return a ranked list of the explanation sentences where facts in the gold explana-

Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13), pages 90–100

tion are expected to be ranked higher than facts not present in the gold explanation (cf. Table 1). Our team was ranked fourth in the official evaluation, scoring within a point gap to the second and third ranked teams, achieving an overall 39.40% MAP.

Going beyond mere Information Retrieval such as $tf-idf$ for ranking relevant sentences to a query, our system addresses the explanation sentence ranking task as follows: we (a) adopt lexical, grammatical, and semantic features for obtaining stronger matches between a question, its correct answer, and candidate explanation sentences for the answer in a pairwise learning-to-rank framework for ranking explanation sentences; and (b) perform a reranking of the returned ranked explanation sentences via a set of soft logical rules to correct for obvious errors made by the learning-based ranking system.[1]

The remainder of the article is organized as follows. We first give a brief overview of the MIER-19 Shared Task and the corpus (Section 2). After that, we describe related work (Section 3). Finally, we present our approach (Section 4), evaluation results (Section 5), and conclusions (Section 6).

2 The MIER-19 Shared Task

2.1 Task Description

The MIER-19 task (Jansen and Ustalov, 2019) focused on computing a ranked list of explanation sentences (as shown in Table 1) for a question and its correct answer (QA) from an unordered collection of explanation sentences. Specifically, given a question, its known correct answer, and a list of n explanation sentences, the goal was to (1) determine whether an explanation sentence is relevant as justification for the QA, and if so, (2) rank the relevant explanation sentences in order by their role in forming a logical discourse fragment.[2]

2.2 The Task Corpus

To facilitate system development, 1,190 Elementary Science (i.e. 3rd through 5th grades) questions were released as part of the training data.

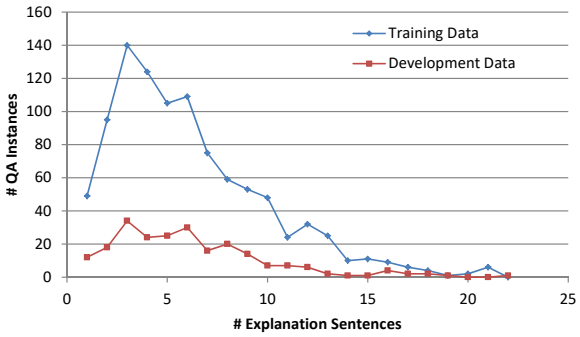

Figure 1: Explanation sentences per question-answer pair in the training and development dataset.

Each question is a 4-way multiple choice question with the correct answer known. Further, every question in the training data is accompanied by up to 21 explanation sentences picked from a human-authored tablestore of candidate explanation sentences (see Section 2.2.1 for more details). Similarly, development data was provided, containing 264 multiple choice questions with a known correct answer and their explanations. The distribution of explanation sentences per QA in the training and development datasets is depicted in Fig. 2.

This dataset of explanations for elementary science QA was originally released as the WorldTree corpus (Jansen et al., 2018).

2.2.1 The Explanations Tablestore

The task corpus separately comprised a tablestore of manually authored 4,789 candidate explanation sentences. Explanations for the QA instances were obtained from one or more tables in the tablestore.

Total unique explanation sentences:	4,789
Seen in training data:	2,694
Seen in development data:	964
Seen in training and development data:	589

The tablestore comprised 62 separate tables each containing explanation sentences around a particular relation predicate such as "kind of" (e.g., an acorn is a kind of seed), "part of" (e.g., bark is a part of a tree), "cause" (e.g., drought may cause wildfires), etc., and a number of tables specified around specific properties such as "actions" of organisms (e.g., some adult animals lay eggs), the "properties of things" (e.g., an acid is acidic), or "if-then" conditions (e.g., when an animal sheds its fur, its fur becomes less dense). Table 2 lists prominent explanation table types used in at least 1% of the training and development explanations.

[1] Our code is released for facilitating future work https://bit.ly/2lZo9eW

[2] Each explanation sentence is also annotated with its explicit discourse role in the explanation fragment for training and development data (i.e. as *central* if it involves core QA concepts, or as *lexical glue* if it simply serves as a connector in the explanation sentence sequence, or as *background* information, etc.). However, we do not consider this annotation as part of the data since it is not available for the test set.

KINDOF	25.22	REQUIRES	2.87	ATTRIBUTE-VALUE-RANGE	1.53
SYNONYMY	14.27	PARTOF	2.74	CHANGE	1.53
ACTION	6.48	COUPLEDRELATIONSHIP	2.67	CHANGE-VEC	1.43
IF-THEN	5.31	SOURCEOF	1.89	EXAMPLES	1.43
CAUSE	4.17	CONTAINS	1.79	PROPERTIES-GENERIC	1.21
USEDFOR	4.17	AFFECT	1.73	TRANSFER	1.11
PROPERTIES-THINGS	3.58	MADEOF	1.69	AFFORDANCES	1.08

Table 2: Explanation table types (21 of 63 in total) sorted by the proportion of their occurrence for their respective sentences participating in at least 1% of the training and development set QA explanations.

3 Background and Related Work

Elementary Science QA requiring diverse text representations. In a study conducted on the New York Regents standardized test QA, Clark et al. (2013) identified at least five QA categories in elementary science, viz. taxonomic questions, definition-based questions, questions based on properties of things (e.g., parts of an object), and questions needing several steps of inference. With the Aristo system (Clark et al., 2016), they demonstrated that a system operating at different levels of textual representation and reasoning was needed to address diverse QA types since it substantially outperformed a singular information retrieval approach.

From these prior insights about the benefit of a heterogeneous system on a dataset with diverse QA types, our approach follows suit in using a set of features over diverse representations of the QA and its explanation.

Commonsense Knowledge for Explanations. The relevance of commonsense knowledge in reasoning task settings was demonstrated by two recent systems (Paul and Frank, 2019; Bauer et al., 2018). Paul and Frank (2019), in a sentiment analysis task, specifically devise a novel method to extract, rank, filter and select multi-hop relation paths from ConceptNet (Liu and Singh, 2004) to interpret the expression of sentiment in terms of their underlying human needs, thereby obtaining boosted task performance for predicting human needs. Bauer et al. (2018) for a narrative QA task, that required the model to reason, gather, and synthesize disjoint pieces of information within the context to generate an answer, employed ConceptNet for multi-step reasoning where they constructed paths starting from concepts appearing in the question to concepts appearing in the context, aiming to emulate multi-hop reasoning.

Relatedly, we employ commonsense knowledge by tracing commonalities between conceptual categories of QA and explanation sentence words.

Explanations for Elementary Science QA. One of the first attempts creating justifications for answers to elementary science exam questions was by Jansen et al. (2017) which jointly addressed answer extraction and justification creation. Since in answering science exam questions, many questions require inferences from external knowledge sources, they return the sentences traversed in inferring the correct answer as a result. After identifying the question's focus words, they generate justifications by aggregating multiple sentences from a number of textual knowledge bases (e.g., study guides, science dictionaries) that preferentially (i.e. based on a number of measures designed to assess how well-integrated, relevant, and on-topic a given justification is) connect sentences together on the focus words, selecting the answer corresponding to the highest-ranked justification. By this method, they obtained a boost in QA performance and, further, the inference mechanism as an additional result justifying the QA process.

4 Our Approach

Unlike Jansen et al. (2017) who jointly perform answer extraction and answer explanation inference, our approach only addresses the task of answer explanation inference assuming we are given the question, its correct answer, and a knowledge base of candidate explanation sentences. [3]

In the methodology for the manual authoring of explanations to create the explanation tablestore for elementary science QA, it was followed that an explanation sentence:

- overlaps lexically with the question or answer, or overlaps lexically with other explanation sentences to the QA which we call *the overlap criteria*; and

[3]The MIER-19 shared task does not evaluate selecting the correct answer, hence the choice was up to the participants whether to model an approach assuming the correct answer or to perform explanation extraction as a function of correct answer selection.

- the sequence of explanation sentences form a logically coherent discourse fragment which we call *the coherency criteria*.

We model both criteria within a pairwise learning-to-rank (LTR) approach.

Let (q, a, e) be a triplet consisting of a question q, its correct answer a, and a candidate explanation sentence e that is a valid or invalid candidate from the given explanations tablestore.

4.1 Features for Learning-to-Rank

First, given a (q, a, e) triplet, we implement *the overlap criteria* by invoking a selected set of feature functions targeting lexical overlap between the triplet elements. For this, each triplet is enriched lexically by lemmatization and affixation to ensure matches with word variant forms. However, more often than not, a QA lexically matches with irrelevant candidate explanation sentences (consider the latter explanation sentences in the example in Table 1) resulting in semantic drift. Therefore, we hypothesize that the semantic drift can be controlled to some extent with matches at different levels of grammatical and semantic abstraction of the q, a, and e units, which we also encode as features.

Specifically, to compute the features, each q, a, and e unit are represented as bags of: words, lemmas, OpenIE (Angeli et al., 2015) relation triples, concepts from ConceptNet (Liu and Singh, 2004), ConceptNet relation triples, Wiktionary categories, Wiktionary content search matched page titles, and Framenet (Fillmore, 1976) predicates and arguments. [4] These representations resulted in 76 feature categories shown in Table 3 which are used to generate (q, a, e) triplet instance one-hot encoded feature vectors.

Second, given as input the (q, a, e) triplet feature vectors, we model the criteria of valid versus invalid explanation sentences and the precedence between explanation sentences, i.e. *the coherency criteria*, within the supervised pairwise LTR framework.

4.2 Pairwise Learning-to-Rank

Pairwise LTR methods are designed to handle ranking tasks as a binary classification problem for pairs of resources by modeling instance ranks as relative pairwise classification decisions.

We employ SVM^{rank} (Joachims, 2006) as our pairwise LTR algorithm, which after transforming the ranking problem into a set of binary classification tasks address the classification through the formalism of Support Vector Machines (SVM). Ranking SVMs in a non-factoid QA ranking problem formulation have showed similar performances to a Neural Perceptron Ranking model (Surdeanu et al., 2011).

4.2.1 Training our MIER-19 Task Model

Next, we describe how an LTR model can be trained using a (q, a, e) triplet feature vector computed according to the 76 feature categories shown in Table 3.

The ranker aims to impose a ranking on the candidate explanation sentences for each QA in the test set, so that (1) the correct explanation sentences are ranked higher than the incorrect ones and (2) the correct explanation sentences are ranked in order of their precedence w.r.t. each other. In LTR, this is modeled as an ordered pair $(x_{q_i,a_i,e_j}, x_{q_i,a_i,e_k})$, where x_{q_i,a_i,e_j} is a feature vector generated between a QA (q_i, a_i) and a correct candidate explanation sentence e_j, and x_{q_i,a_i,e_k} is a feature vector generated between (q_i, a_i) and an incorrect candidate explanation sentence e_k. In addition, another kind of training instance in our dataset can occur between correct explanation sentences as an ordered pair $(x_{q_i,a_i,e_j}, x_{q_i,a_i,e_m})$, where e_j logically precedes e_m in the explanation sentence sequence. The goal of the ranker-learning algorithm, then, is to acquire a ranker that minimizes the number of violations of pairwise rankings provided in the training set.

The ordered pairwise instances are created above based on the labels assigned to each training instance. One detail we left out earlier when discussing our (q, a, e) triplet features for LTR, was that each triplet is also assigned a label indicating a graded relevance between the QA and the candidate explanation sentence. This is done as follows. For each (q, a, e) triplet instance, if e is in the sequence of correct explanation sentences, then it is labeled in a descending rank order starting at 'rank=number of explanation sentences+1' for the first sentence and ending at 'rank= 2' for the last one in the sequence, otherwise, 'rank= 1' for all incorrect explanation sentences.

[4] OpenIE relations and FrameNet structures are extracted only for q and e since they need to be computed on sentences and the answers a are often expressed as phrases.

1. Lexical (31 feature categories)

1. lemmas of $q/a/e$
2. lemmas shared by q and e, a and e, and q, a and e
3. 5-gram, 4-gram, and 3-gram prefixes and suffixes of $q/a/e$
4. 5-gram, 4-gram, and 3-gram prefixes and suffixes shared by q, a, and e
5. e's table type from the provided annotated tablestore data

2. Grammatical (11 feature categories)

1. using OpenIE (Angeli et al., 2015) extracted relation triples from q, a, and e sentences, the features are: the $q/a/e$ lemmas in the relation subject role, shared q, a and e subject lemmas, $q/a/e$ lemmas in the relation object role, shared q, a and e object lemmas, and $q/a/e$ lemma as the relation predicate

3. Semantic (34 feature categories)

1. top 50 conceptualizations of $q/a/e$ words obtained from ConceptNet (Liu and Singh, 2004)
2. top 50 ConceptNet conceptualizations shared by q and e, a and e, and q, a and e words
3. words related to $q/a/e$ words by any ConceptNet relation such as FormOf, IsA, HasContext, etc.
4. words in common related to q, a, and e words
5. Wiktionary[5] categories of $q/a/e$ words
6. Wiktionary categories shared by q, a, and e words
7. Wiktionary page titles for content matched with $q/a/e$ words
8. Wiktionary page titles for content matched with q, a, and e words in common
9. FrameNet v1.7 frames and their frame-elements (Fillmore, 1976) using open-SESAME (Swayamdipta et al., 2017) were extracted from q and e sentences

Table 3: 76 feature categories for explanation ranking. Each training instance corresponds to a triplet (q, a, e), where q, a, and e are bags of question, answer, and explanation words/lemmas, respectively, with stopwords filtered out, where the data was sentence split and tokenized using the Stanford Parser [6].

4.3 Rules Solver

The application of the LTR system on development data revealed 11 classes of errors that we call *obvious* error categories in the sense that they could be easily rectified via a set of logical $if - then - else$ chained rules where the output of one rule is the input of the next. We hypothesize that a rule-based approach can complement a purely learning-based approach, since a human could alternatively encode the commonsense knowledge that may not be accessible to a learning algorithm given our features set. This inclusion of rules as a post-processing step resulted in our hybrid learning-based and rule-based system to MIER-19 explanation sentence ranking.

We list four rules from our complete set of 11 rules as examples next. [7]

[7] We list all the rules in Appendix A.

E.g. Rule 1: Match with uni- or bigram answers.

if answer is a unigram or bigram **then**
rerank all explanation sentences containing the answer to the top
end if

E.g. Rule 2: Match with named entities.

if explanation sentence contains named entities identified by [A-Z][a-z]+([A-Z][a-z]+)+ **then**
rerank the explanation sentence to the bottom if neither the question or answer contain the explanation's named entities
end if

E.g. Rule 3: Rerank explanation sentences with other color words than the answer

if an answer contains a color word **then**
rerank all explanation sentences about other colors in the form "[other color] is a kind of color" to the bottom of the list
end if

E.g. Rule 4: Rerank based on gerund or participle answer words

if answer contains gerund or participle words, i.e. "ing" words **then**
rerank all explanation sentences from the SYNONYMY table type containing gerund or participle words other than the answer "ing" word to the bottom of the list
end if

4.4 Testing the Hybrid System

The trained LTR model and the rules were then applied on the QA instances released as the test dataset. From test data, (q, a, e) triplets were created in the same manner as the development data where each test QA is given all 4,789 candidate explanation sentences for ranking. Unlike development data, however, in testing the valid explanation sentences are unknown.

5 Evaluation

In this section, we evaluate our hybrid approach to explanation sentence ranking for elementary science QA.

5.1 Experimental Setup

Dataset. We used the 1,190 and 264 elementary science QA pairs released as the MIER-19 challenge training and development data, respectively, for developing our system. For testing, we used the 1,247 QA instances released in the evaluation phase of the MIER-19 challenge. For explanation candidate sentences, we used the tablestore of the 4,789 sentences which remained the same in the course of the challenge.

Evaluation Metrics. Evaluation results are obtained using the official MHIER-19 challenge scoring program. Results are expressed in terms of mAP computed by the following formula.

$$mAP = \frac{1}{N}\sum_{n=1}^{N} AP_n$$

where N is the number of QA instances and AP is the average precision for a QA computed as follows.

$$AP@k = \frac{1}{GTP}\sum_{i=1}^{k} \frac{TPseen@i}{i}$$

where $AP@k$, i.e. average precision at k, is the standard formula used in information retrieval tasks. Given the MHIER-19 challenge data, for each QA, GTP is the total ground truth explanation sentences, k is the total number of explanation sentences in the tablestore (i.e. 4,789), and $TPseen@i$ are the total ground truth explanation sentences seen until rank i.

By the above metric, our results are evaluated only for the correct explanation sentences returned as top-ranked, without considering their order.

Parameter Tuning. To optimize ranker performance, we tune the regularization parameter C (which establishes the balance between generalizing and overfitting the ranker model to the training data). However, we noticed that a ranker trained on all provided explanation sentences is not able to learn a meaningful discriminative model at all owing to the large bias in the negative examples outweighing the positive examples (consider that valid explanation sentences range between 1 to 21 whereas there are 4,789 available candidate explanation sentences in the tablestore). To overcome the class imbalance, we tune an additional parameter: the number of negative explanation sentences for training. Every QA training instance is assigned 1000 randomly selected negative explanation sentences. We then test tuning the number of negative training data explanation sentences to range between 500 to 1,000 in increments of 100.

Both the cost factor and the number of negative explanation sentences are tuned to maximize performance on development data. Note, however, that our development data is created to emulate the

	Dev MAP	Test MAP
SVM^{rank}	37.1	34.1
+Rules	44.4	39.4

Table 4: Mean Average Precision (mAP) percentage scores for Elementary Science QA explanation sentence ranking from only pairwise LTR (row 1) and as a hybrid system with rules (row 2) on development and test datasets, respectively.

testing scenario. So every QA instance during development is given all 4,789 candidate explanation sentences to obtain results for the ranking task.[8]

Our best LTR model when evaluated on development data was obtained with C = 0.9 and 700 negative training instances.

5.2 Results and Discussion

Table 4 shows the elementary science QA explanation sentence ranking results from the official MIER-19 scoring program in terms of mAP. The first row corresponds to results from the feature-rich SVM^{rank} LTR system and the second row shows the reranked results using rules. While adding the rules gives us close to a 7 and 5 points improvement on development and test sets, respectively, the LTR system results are nonetheless significantly better than an information retrieval TF-IDF baseline which gives 24.5% and 24.8% mAP on development and test data. Additionally, Figure 2 shows the impact of the features-only LTR system versus the features with rules hybrid system on different length explanation sentences up to 12.[9] It illustrates that the longer explanations are indeed harder to handle by both approaches and on both development and test data.

To provide further insights on the impact of adding different feature groups in our LTR system, we show ablation results in Table 5. We discuss the maximum impact feature groups (viz. affix, concepts, and relations) with examples, next, to demonstrate why they work.

Compared to all other features, adding affixes in the LTR system resulted in the maximum performance gain of 6 points on the development data. In general, affixation enables lexical matches with variant word forms, which for us, facilitated better matches between QA and candidate explanation sentences. Consider the following example showing the top 3 returned explanation sentences.

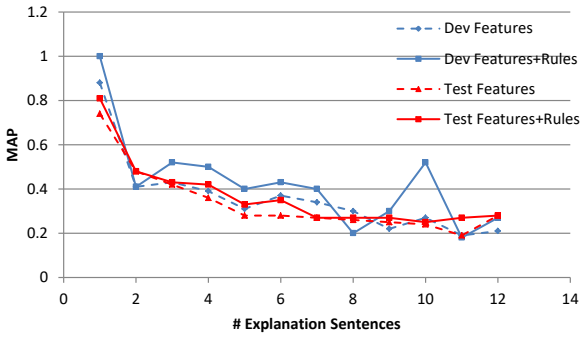

Figure 2: Percentage mAP of the 'Features' versus 'Features+Rules' systems on Development data (in blue) and Test data (in red), respectively, on different length explanation sentences.

	Feature Type	Dev MAP	Test MAP
1	lemma	28.40	24.85
2	+tablestore	28.05	24.95
3	+affix	34.01	31.01
4	+concepts	35.89	32.94
5	+relations (openIE & conceptNet)	36.79	33.69
6	+Wiktionary	37.14	33.65
7	+framenet	37.12	34.14

Table 5: Ablation results of the LTR SVM^{rank} system in terms of percentage mAP with feature groups (from seven feature types considered) incrementally added.

Question What happens when the Sun's energy warms ocean water?
Answer The water evaporates.
Before
[not gold] an ocean is a kind of body of water
[not gold] temperature or heat energy is a property of objects or weather and includes ordered values of cold or cool or warm or hot
[not gold] coral lives in the ocean or warm water
After
[not gold] an ocean is a kind of body of water
[r_p 2 & r_g 1] boiling or evaporation means change from a liquid into a gas by adding heat energy
[r_p 3 & r_g 6] the sun transfers solar energy or light energy or heat energy from itself to the planets or Earth through sunlight

Before affixation, none of the valid explanation sentences are retrieved among the top 3. After affixation, however, two valid explantion sentences get reranked among the top 3 [10] owing to enabled matches with the QA words "Sun's" and "evaporates" based on their trigram prefixes.

As hypothesized earlier, we found ConceptNet's semantic knowledge preventing semantic

[8] For parameter tuning, C is chosen from the set {0.1, 0.9, 1,10,50,100,500,800} and the number of negative training instances is chosen from the set {500,600,700,800,900,1000}.

[9] We only consider explanation length up to 12 for the comparison since the longer explanations are underrepresented in the data with up to 3 QA instances.

[10] r_p and r_g stand for predicted rank and gold rank, respectively.

drift in several instances. This is illustrated in the example below.

Question In which part of a tree does photosynthesis most likely take place?
Answer leaves
Before
[r_p 1 & r_g 1] a leaf performs photosynthesis or gas exchange
[r_p 5 & r_g 2] a leaf is a part of a green plant
[r_p 10 & r_g 3] a tree is a kind of plant
After
[r_p 1 & r_g 1] a leaf performs photosynthesis or gas exchange
[r_p 3 & r_g 2] a leaf is a part of a green plant
[r_p 7 & r_g 3] a tree is a kind of plant

In the example, with additional knowledge such as that "plant" has a ConceptNet conceptual class "photosynthetic organism" enables higher reranking for the second and third explanation sentences since one of the focus concepts in the question is photosynthesis.

We find the ConceptNet relations as features enable making connections between the question and answer. These connections enable a more accurate reranking of those explanation sentences that rely on information from both the question and the correct answer closer to the top. Consider the following example.

Question Cows are farm animals that eat only plants. Which of these kinds of living things is a cow?
Answer Herbivore
Before
[r_p 3 & r_g 6] an animal is a kind of living thing
[r_p 5 & r_g 1] herbivores only eat plants
After
[r_p 2 & r_g 1] herbivores only eat plants
[r_p 5 & r_g 6] an animal is a kind of living thing

For the above example, from ConceptNet we obtain the lexical relations "herbivore IsA animal", "cow RelatedTo animal", and "an animal Desires eat" which lexically links the explanation sentence with the question and the correct answer. We attribute the application of such relations as the reason for the correct reranking of the two sentences in terms of precedence and their proximity to the gold rank.

5.2.1 Negative Results

Apart from the features depicted above, we also considered WordNet (Miller, 1998) for additional lexical expansion to facilitate matches for the (q, a, e) triplets based on linguistic relations such as synonymy, hypernymy, etc., but did not obtain improved system performance. Further, features computed from the word embeddings, viz. Word2vec (Mikolov et al., 2013), Glove (Pennington et al., 2014), and ConceptNet Numberbatch (Speer et al., 2017), as averaged vectors also did not improve our model scores.

Finally, while the hybrid system via the reranking rules addresses lexical ordering between candidate explanation sentences, they still cannot handle eliminating intermediate explanation sentences that may not be semantically meaningful to the QA. This is illustrated in the representative example below.[11]

Question Jeannie put her soccer ball on the ground on the side of a hill. What force acted on the soccer ball to make it roll down the hill?
Answer gravity
[r_f 3 & r_h 1] gravity is a kind of force
[r_f 21 & r_h 3] gravity or gravitational force causes objects that have mass or substances to be pulled down or to fall on a planet
[r_f 4 & r_h 14] a ball is a kind of object
[r_f 7 & r_h 17] to cause means to make

The example shows that the reranked result from the hybrid system follows the order in the gold data, however, not consecutively. Sentences extraneous to the explanation such as "the ground is at the bottom of an area", "a softball is a kind of ball", etc., are still in between gold explanation sentences in the reranked results.

6 Conclusions

We employed a hybrid approach to explanation sentence ranking for Elementary Science QA consisting of a feature-rich LTR system followed by a series of 11 rules. When evaluated on the MIER-19 official test data, our approach achieved an mAP of 39.4%.

An immediate extension to this system would be to encode the dependencies between explanation sentences as features. While the pairwise LTR model tackles this dependency to some extent, we hypothesize that explicit modeling of features between explanation sentences should produce significantly improved scores.

Acknowledgments

We would like to thank the MIER-19 organizers for creating the corpus and organizing the shared task. We would also like to thank the anonymous reviewers for their helpful suggestions and comments.

[11] r_f and r_h stand for ranking by features and the hybrid system, respectively.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *ACL*.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Turney, and Daniel Khashabi. 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Peter Clark, Philip Harrison, and Niranjan Balasubramanian. 2013. A study of the knowledge base requirements for passing an elementary science test. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 37–42. ACM.

Charles J Fillmore. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*, 280(1):20–32.

Peter Jansen, Rebecca Sharp, Mihai Surdeanu, and Peter Clark. 2017. Framing qa as building and ranking intersentence answer justifications. *Computational Linguistics*, 43(2):407–449.

Peter Jansen and Dmitry Ustalov. 2019. TextGraphs 2019 Shared Task on Multi-Hop Inference for Explanation Regeneration. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, Hong Kong. Association for Computational Linguistics.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton T. Morrison. 2018. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*.

Thorsten Joachims. 2006. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *AAAI*.

Hugo Liu and Push Singh. 2004. Conceptneta practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *EMNLP*.

Debjit Paul and Anette Frank. 2019. Ranking and selecting multi-hop knowledge paths to better predict human needs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3671–3681.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *ArXiv*, abs/1611.01603.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. pages 4444–4451.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2011. Learning to rank answers to non-factoid questions from web collections. *Computational linguistics*, 37(2):351–383.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold. *arXiv preprint arXiv:1706.09528*.

A Appendix

Rule 1: Match with uni- or bigram answers.

if answer is a unigram or bigram **then**
rerank all explanation sentences containing the answer to the top
end if

Rule 2: Match with named entities.

if explanation sentence contains named entities identified by [A-Z][a-z]+([A-Z][a-z]+)+ **then**
rerank the explanation sentence to the bottom if neither the question or answer contain the explanation's named entities

end if

Rule 3: Match with energy insulator commonsense knowledge.

if explanation contains wax or rubber or brick or down feathers as "energy insulator" **then**
rerank the explanation to the bottom if neither the question or answer contain "energy insulator"
end if

Rule 4: Match with "increase" or "decrease" commonsense knowledge.

if explanation contains "increase" or "decrease" **then**
rerank the explanation to the bottom if neither the question or answer contains "increase" or "decrease"
end if

Rule 5: Rerank explanation sentences with unrelated entities to the QA.

By this rule, we identify pairs of entities that are unrelated and rerank all explanation sentences containing an unrelated entity mention to the bottom of the list. For instance, if the QA is about "planets", then all explanation sentences about "fern" can be reranked to the bottom as it unlikely for any discussion to exist that relates "planets" and the "fern" plant. Similarly, if a QA is about "puppies", then all explanation sentences about "peas" can be reranked to the bottom.

To form pairs of unrelated entities, first, we create lists of living and nonliving entities using the KINDOF explanation table type (one among 61 explanation tables with a few shown in Table 2), where sentences are of the pattern "[LHS] is a kind of [RHS]". These lists as created in a recursive manner. For instance, to create the list of living entities, we begin with all sentences where the RHS="living thing" and add the LHS value to the list. In the next step, we substitute in the RHS the new found living entities extracted in the previous step. Again, we add the new LHS values to the list of living entities. This process, i.e. substituting new entities in the RHS and extracting the entity in the LHS, continues until the list of living entities no longer changes. As a simple example, given "a plant is a kind of living thing", in step 1, we add plant to the list of living entities. In step 2, given "peas are a kind of plant", we add peas to the list of living entities. The lists for non-living entities are created in a similar manner, with the starting pattern using RHS="nonliving thing".

Once weve obtained these lists, we obtain all pairwise combinations of living, non-living, and living and non-living entities. We identify the pairs of unrelated entities by filtering out all pairs of entities that have appeared in training and development data. We also manually filter out additional entities that we recognize could be related. Using this list, we rerank the explanation sentences as follows.

if question or answer contain an entity in the list of unrelated entity pairs **then**
rerank all explanation sentences containing the second element of the pair to the bottom of the list
end if

Rule 6: Singular form matched in explanation sentences with plural unigram answers.

if a unigram answer is in plural **then**
rerank all explanation sentences containing its singular form to the top of the list following explanation sentences containing the exact plural match
end if

Rule 7: Rerank explanation sentences with other color words than the answer.

if an answer contains a color word **then**
rerank all explanation sentences about other colors in the form "[other color] is a kind of color" to the bottom of the list
end if

Rule 8: Rerank KINDOF explanation sentences with entities not present in the QA.

if QA does not contain generic living entity types such as "plants", "animal", "organism" or "human" **then**
rerank all KINDOF explanation sentences to the bottom relating entities not expressed in the either the question or answer
end if

Rule 9: Rerank explanations from table types not considered in training and development data based on six QA types and overall.

We identify six QA types: "Which", "When", and "What" questions; questions beginning with "Some"; questions beginning with indefinite article "A"; and those beginning with definite article "The". For each of these six QA types, we identify the table types that are never used to generate explanations in training and development data. Additionally, we identify table types never used to

generate explanations in the training and development set overall.

if QA is in one of the six types or overall **then**
rerank all explanation sentences from the never used table types for the particular type of QA to the bottom of the list
end if

Rule 10: Match based on alternative sense of the word "makes"

if QA contains the word "makes" (e.g., "What makes up most of a human skeleton?") **then**
rerank all explanation sentences from the SYNONYMY table type of "make" with alternative word senses to the bottom of the list (e.g., "to make something easier means to help")
end if

Rule 11: Rerank based on gerund or participle answer words.

if answer contains gerund or participle words, i.e. "ing" words **then**
rerank all explanation sentences from the SYNONYMY table type containing gerund or participle words other than the answer "ing" word to the bottom of the list
end if