

Interpretable Question Answering on Knowledge Bases and Text

Alona Sydorova

iteratec GmbH

Munich, Germany

alona.sydorova@iteratec.com

Nina Poerner & Benjamin Roth

Center for Information and

Language Processing

LMU Munich, Germany

{poerner,beroth}@cis.lmu.de

Abstract

Interpretability of machine learning (ML) models becomes more relevant with their increasing adoption. In this work, we address the interpretability of ML based question answering (QA) models on a combination of knowledge bases (KB) and text documents. We adapt post hoc explanation methods such as LIME and input perturbation (IP) and compare them with the self-explanatory attention mechanism of the model. For this purpose, we propose an automatic evaluation paradigm for explanation methods in the context of QA. We also conduct a study with human annotators to evaluate whether explanations help them identify better QA models. Our results suggest that IP provides better explanations than LIME or attention, according to both automatic and human evaluation. We obtain the same ranking of methods in both experiments, which supports the validity of our automatic evaluation paradigm.

1 Introduction

Question answering (QA) is an important task in natural language processing and machine learning with a wide range of applications. QA systems typically use either structured information in the form of knowledge bases (KBs), or raw text. Recent systems have successfully combined both types of knowledge (Das et al., 2017).

Nowadays, due to the changing legal situation and growing application in critical domains, ML based systems are increasingly required to provide explanations of their output. Lipton (2018), Poursabzi-Sangdeh et al. (2018) and Doshi-Velez and Kim (2017) point out that there is no complete agreement on the definition, measurability and evaluation of interpretability in ML models. Nevertheless, a number of explanation methods have been proposed in the recent literature, with

the aim of making ML models more transparent for humans.

To the best of our knowledge, the problem of explanations for deep learning based QA models working on a combination of structured and unstructured data has not yet been researched. Also, there is a lack of evaluation paradigms to compare different explanation methods in the context of QA.

Contributions

- We explore interpretability in the context of QA on a combination of KB and text. In particular, we apply attention, LIME and input perturbation (IP).
- In order to compare these methods, we propose a novel automatic evaluation scheme based on “fake facts”.
- We evaluate whether explanations help humans identify the better out of two QA models.
- We show that the results of automatic and human evaluation agree.
- Our results suggest that IP performs better than attention and LIME in this context.

2 Question Answering on Knowledge Bases and Text

The combination of knowledge bases and text data is of particular interest in the context of QA. While knowledge bases provide a collection of facts with a rigid structure, the semantic information contained in text documents has the potential to enrich the knowledge base. In order to exploit different information sources within one QA system, Das et al. (2017) introduce the TextKBQA model, which works on a universal schema representation

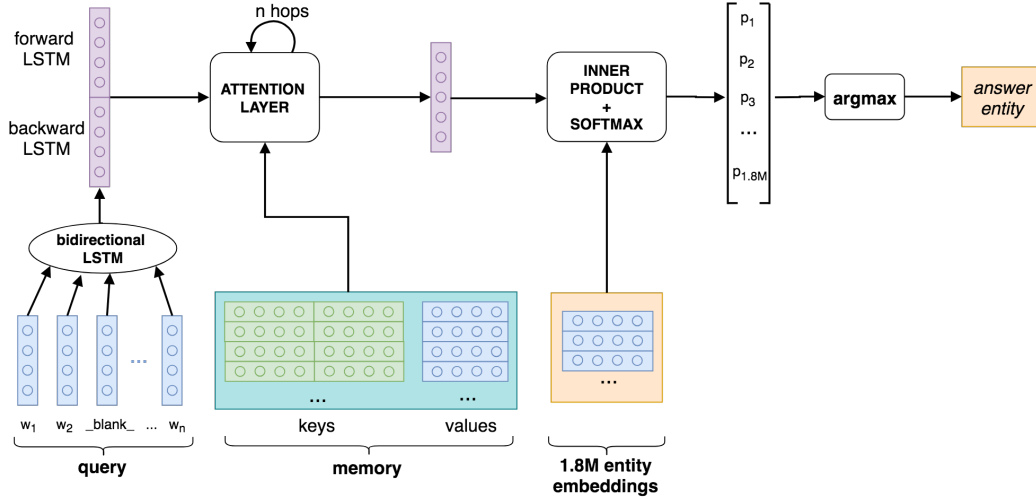


Figure 1: Overview of the TextKBQA model architecture.

(Riedel et al., 2013) of a KB and text documents. They state that “individual data sources help fill the weakness of the other, thereby improving overall performance” and conclude that “the amalgam of both text and KB is superior than KB alone.” Their model solves the so-called *cloze questions* task, i.e., filling in blanks in sentences. For example, the answer to “*Chicago is the third most populous city in blank.*” would be the entity *the USA*. The model has a KB and a number of raw text sentences at its disposal. Das et al. (2017) use Freebase (Bollacker et al., 2008) as KB (8.0M facts) and ClueWeb (Gabrilovich et al., 2013) as raw text source (0.3M sentences). They test on question-answer pairs from SPADES (Bisk et al., 2016) (93K queries).

The TextKBQA model (Figure 1) is a key-value memory network that uses distributed representations for all entities, relations, textual facts and input questions. Every memory cell corresponds to one KB fact or one textual fact, which are encoded as key-value pairs (Miller et al., 2016).

Every **KB fact** is a triple consisting of a subject s , an object o and the relation r between these entities. s, r, o are embedded into real-valued vectors $\mathbf{s}, \mathbf{r}, \mathbf{o}$. The memory key is the concatenation of subject and relation embedding: $\mathbf{k} = [\mathbf{s}; \mathbf{r}] \in \mathbb{R}^{2d}$. The memory value is the embedding of the object: $\mathbf{v} = \mathbf{o} \in \mathbb{R}^d$. **Textual facts** are sentences that contain at least two entities. They are also represented as triples, where the relation is a token sequence: $(s, [w_1, \dots, \text{arg1}, \dots, \text{arg2}, \dots, w_n], o)$. To convert the sentence into a vector, arg1 and arg2 are replaced by s and blank_- re-

spectively. Then, the sequence is processed by a bidirectional LSTM. Its last states are concatenated to form the memory key $\mathbf{k} = [\overrightarrow{LSTM}([w_1, \dots, w_n]); \overleftarrow{LSTM}([w_1, \dots, w_n])] \in \mathbb{R}^{2d}$. The memory value is $\mathbf{v} = \mathbf{o}$, as before.

A question $q = [w_1, \dots, e, \dots, \text{blank}_-, \dots, w_n]$ is transformed into a distributed representation $\mathbf{q} \in \mathbb{R}^{2d}$ using the same bidirectional LSTM as before. In this way, KB and textual facts as well as queries are in the same \mathbb{R}^{2d} space.

Given \mathbf{q} and a set of relevant facts, represented by key-value pairs (\mathbf{k}, \mathbf{v}) , TextKBQA performs multi-hop attention. More specifically, the context vector \mathbf{c}_0 is set to \mathbf{q} . In every iteration (*hop*) t , a new context vector \mathbf{c}_t is computed as:

$$\mathbf{c}_t = W_t(\mathbf{c}_{t-1} + W_p \sum_{(k,v) \in \mathcal{M}} \text{softmax}(\mathbf{c}_{t-1} \cdot \mathbf{k})\mathbf{v}) \quad (1)$$

where W_p, W_t are weight matrices. In practice, \mathcal{M} contains only facts that share an entity with the query. The result of the last hop is fed into a fully-connected layer to produce a vector $\mathbf{b} \in \mathbb{R}^d$. Then, the inner product between \mathbf{b} and all entity embeddings is taken. The entity with the highest inner product is chosen as the model’s answer a_q .

We train the TextKBQA model using the datasets described above. We limit the number of textual facts per query to 500, since only 35 out of 1.8M entities in the dataset have more than 500 textual facts. Apart from this modification, we use the exact same implementation and training setup as in Das et al. (2017). Our final model achieves an F_1 score of 41.59 on the dev dataset and 40.27

on the test dataset, which is slightly better than the original paper (41.1 and 39.9, respectively).

3 Explanation methods

We first present some important notation and give a working definition of an explanation method.

Formally, let \mathbb{F} be a database consisting of all KB and textual facts: $\mathbb{F} = \mathbb{F}_{KB} \cup \mathbb{F}_{text}$. Furthermore, let \mathcal{E} be a set of **entities** that are objects and subjects in \mathbb{F} , and let \mathcal{R} be a set of **relations** from \mathbb{F}_{KB} . In the following we will use a general notation f for a fact from \mathbb{F} , distinguishing between KB and textual facts only when necessary.

Let q be a **query**, and $\mathcal{F} \subseteq \mathbb{F}$ the corresponding set of facts, such that for $\forall f \in \mathcal{F}$ holds: $subject_f \in q$. Let $TextKBQA$ be a function computed by the TextKBQA model and $a_q = TextKBQA(q, \mathcal{F})$, $a_q \in \mathcal{E}$, the predicted **answer** to the query q . Note that a_q is not necessarily the ground truth answer for q .

Analogously to Poerner et al. (2018), we give the following definition: an *explanation method* is a function $\phi(f, a_q, q, \mathcal{F})$ that assigns real-valued relevance scores to facts f from \mathcal{F} given an input query q and a target entity a_q . If $\phi(f_1, a_q, q, \mathcal{F}) > \phi(f_2, a_q, q, \mathcal{F})$ then fact f_1 is of a higher relevance for a_q given q and \mathcal{F} than fact f_2 .

3.1 Attention Weights

The attention mechanism provides an explanation method which is an integral part of the TextKBQA architecture.

We formally define the explanation method **attention weights** as:

$$\phi_{aw}(f, a_q, q, \mathcal{F}) = \text{softmax}(K_{\mathcal{F}} \cdot \mathbf{q})_f \quad (2)$$

where $K_{\mathcal{F}}$ is a matrix whose rows are key vectors of facts in \mathcal{F} .

Since the TextKBQA model takes three attention hops per query, ϕ_{aw} can be extended as follows: On the one hand, we can take attention weights from the first, second or third (=last) hops. Intuitively, attention weights from the first hop reflect the similarity of fact keys with the original query, while attention weights from the last hop reflect the similarity of fact keys with the summarized context from all previous iterations. On the other hand, some aggregation of attention weights could also be a plausible explanation method. For every fact, we take the mean attention weight over

hops to be its average relevance in the reasoning process.

Taking into account the above considerations we redefine ϕ_{aw} :

- **attention weights at hop j :**

$$\phi_{aw_j}(f, a_q, q, \mathcal{F}) = \text{softmax}(K_{\mathcal{F}} \cdot \mathbf{c}_{j-1})_f \quad (3)$$

- **average attention weights:**

$$\phi_{aw_{avg}}(f, a_q, q, \mathcal{F}) = \frac{1}{h} \sum_{j=1}^h \text{softmax}(K_{\mathcal{F}} \cdot \mathbf{c}_{j-1})_f \quad (4)$$

where h is the number of hops.

3.2 LIME

LIME (Local Interpretable Model-Agnostic Explanations) is a model-agnostic explanation method (Ribeiro et al., 2016). It approximates behavior of the model in the vicinity of an input sample with the help of a less complex, interpretable model.

LIME requires a mapping from original features (used by TextKBQA) to an interpretable representation (used by LIME). For this purpose we use binary “bag of facts” vectors, analogously to the idea of bag of words: a vector $z \in \{0, 1\}^{|\mathcal{F}|}$ indicates presence or absence of a fact f from \mathcal{F} . The reverse mapping is straightforward.

We first turn the original fact set \mathcal{F} into an interpretable representation z . Every entry of this vector represents a fact from \mathcal{F} . Then we sample vectors z' of the same length $|\mathcal{F}|$ by drawing facts from \mathcal{F} using the Bernoulli distribution with $p = 0.5$. In every z' vector, the presence or absence of facts is encoded as 1 or 0, respectively. We set the number of samples to 1000 in our experiments.

For every z' , we obtain the corresponding original representation \mathcal{F}' and give this reduced input to the TextKBQA model. Note that the query q remains unchanged. We are interested in the probability that a_q is still the predicted answer to the query q , given facts \mathcal{F}' instead of \mathcal{F} . In the TextKBQA model, this probability is obtained from the inner product of \mathbf{b} and the entity embedding matrix E at position a_q . We define this step as a function $logit(q, \mathcal{F}, a_q) = (E \cdot \mathbf{b})_{a_q}$.

We gather the outputs of $logit(q, \mathcal{F}', a_q)$ for all sampled instances, together with the corresponding binary vectors, into a dataset \mathcal{Z} . Then, we train

a linear model on \mathcal{Z} by optimizing the following equation:

$$\xi(q, \mathcal{F}) = \operatorname{argmin}_{g \in G} \mathcal{L}(\operatorname{logit}, g) \quad (5)$$

where \mathcal{L} is ordinary least squares and G is the class of linear models, such that $g(z') = w_g \cdot z'$.¹

From the linear model g , we extract a weight vector $w_g \in \mathbb{R}^{|\mathcal{F}|}$. This vector contains LIME relevance scores for facts in \mathcal{F} given a_q and q . We formally define the **LIME** explanation method for the TextKBQA model as follows:

$$\phi_{\text{lime}}(f, a_q, q, \mathcal{F}) = w_{g,f} \quad (6)$$

3.3 Input Perturbation Method

Another explanation method is input perturbation (IP), originally proposed by Li et al. (2016), who apply it on a sentiment analysis task. They compute relevance scores for every word in a dictionary as the average relative log-likelihood difference that arises when the word is replaced with a baseline value. This method cannot be directly applied to QA, because the same fact can be highly relevant for one query and irrelevant for another. Therefore, we constrain the computation of log-likelihood differences to a single data sample (i.e., a single query).

We formally define the **input perturbation** (IP) explanation method as follows:

$$\phi_{\text{ip}}(f, a_q, q, \mathcal{F}) = \frac{\operatorname{logit}(q, \mathcal{F}, a_q) - \operatorname{logit}(q, \mathcal{F} \setminus \{f\}, a_q)}{\operatorname{logit}(q, \mathcal{F}, a_q)} \quad (7)$$

where logit is the same logit function that we used for LIME. A positive difference means that if we remove fact f when processing query q , the model’s hidden vector \mathbf{b} is less similar to the entity a_q , suggesting that the fact is relevant.

4 Automatic evaluation using fake facts

This section presents our automatic evaluation approach, which is an extension of the hybrid document paradigm (Poerner et al., 2018). The major advantage of automatic evaluation in the context of explanation methods is that it does not require manual annotation.

¹We do not use a proximity measure, because, unlike the original LIME, we only sample from the facts currently present in \mathcal{F} , and not from the whole \mathbb{F} set.

4.1 Definition of automatic evaluation

Poerner et al. (2018) create hybrid documents by randomly concatenating fragments of different documents. We adapt this paradigm to our use case in the following way:

Let q be a query and \mathcal{F} the corresponding set of facts. We define the corresponding **hybrid fact set** $\hat{\mathcal{F}}$ as the union of \mathcal{F} with another disjoint fact set \mathcal{F}' :

$$\hat{\mathcal{F}} = \mathcal{F} \cup \mathcal{F}', \text{ where } \mathcal{F} \cap \mathcal{F}' = \emptyset. \quad (8)$$

Conceptually, \mathcal{F}' are “fake facts”. We discuss how they are created below; for now, just assume that TextKBQA is unable to correctly answer q using only \mathcal{F}' . Note that we only consider queries that are correctly answered by the model based on their hybrid fact set $\hat{\mathcal{F}} = \mathcal{F} \cup \mathcal{F}'$.

The next step is to obtain predictions a_q for the hybrid instances and to explain them with the help of an explanation method ϕ . Recall that ϕ produces one relevance score per fact. The fact with the highest relevance score, $\operatorname{rmax}(\hat{\mathcal{F}}, q, \phi)$, is taken to be the most relevant fact given query q , answer a_q and facts $\hat{\mathcal{F}}$, according to ϕ . We assume that ϕ made a reasonable choice if $\operatorname{rmax}(\hat{\mathcal{F}}, q, \phi)$ stems from the original fact set \mathcal{F} and not from the set of fake facts \mathcal{F}' .

Formally, a “hit point” is assigned to ϕ if:

$$\operatorname{hit}(\phi, q, \hat{\mathcal{F}}) = \begin{cases} 1, & \text{if } \operatorname{rmax}(\hat{\mathcal{F}}, q, \phi) \in \mathcal{F}, \\ 0, & \text{if } \operatorname{rmax}(\hat{\mathcal{F}}, q, \phi) \in \mathcal{F}'. \end{cases} \quad (9)$$

The **pointing game accuracy** of explanation method ϕ is simply its number of hit points divided by the maximally possible number of hit points.

4.2 Creating Fake Facts

To create fake facts for query q , we randomly sample a different query q' that has the same number of entities and gather its fact set \mathcal{F}' . We then replace subject entities in facts from \mathcal{F}' with subject entities from \mathcal{F} . We call these “fake facts” because they do not exist in \mathbb{F} , unless by coincidence. For example, let q be “*blank_ was chosen to portray Patrick Bateman, a Wall Street serial killer.*” and q' be “*This year Philip and blank_ divided Judea into four kingdoms.*” Then replace subject entities Philip and Judea in facts of \mathcal{F}' by subject entities Patrick Bateman and Wall Street, respectively. E.g., the KB

	Facts in hybrid fact set $\hat{\mathcal{F}}$
Facts from \mathcal{F} (real facts)	Disney award.award_honor.award_winner_award.award_honor.honored_for Bambi Disney’s Steamboat Willie premiered on November 18th 1928 at the Broadway. <u>Disney film.performance.actor_film.performance.character Mickey</u> Disney film.film.directed_by.2_film.film.directed_by.1 The Opry House
Facts from \mathcal{F}' (fake facts)	But in the summer of 2007, Apple rocked Disney by launching the iPhone. <i>Disney fashion.clothing_size.region_fashion.clothing_size.person Frankie Rayder</i> The Libertarian is a Disney political party created in 1971. eBay is the largest marketplace in the Disney.

Table 1: An example of a hybrid instance. Query: “Walt Disney himself was the original voice of *blank*...”. Answer: Mickey. Green underlined: fact with the maximal relevance score assigned by IP. Red italics: fact with the maximal relevance score assigned by average attention weights.

fact Philip people.person.gender.1 Males is turned into Patrick Bateman people.person.gender.1 Males, the textual fact "This year Herod divided Judea into four kingdoms." becomes "This year Herod divided Wall Street into four kingdoms."

Our assumption is as follows: If the model is still able to predict the correct answer despite these fake facts, then this should be due to a fact contained in \mathcal{F} and not in \mathcal{F}' . This assumption fails when we accidentally sample a fact that supports the correct answer. Therefore, we validate \mathcal{F}' by testing whether the model is able to predict the correct answer to q using just \mathcal{F}' . If this is the case, a different query q' and a different fake fact set \mathcal{F}' are sampled and the validation step is applied again. This procedure goes on until a valid \mathcal{F}' is found.

Table 1 contains an example of a query with real and fake facts for which explanations were obtained by average attention weights and IP. IP assigns maximal relevance to a real fact from \mathcal{F} , which means that ϕ_{ip} receives one hit point for this instance. The average attention weight method considers a fake fact from \mathcal{F}' to be the most important fact and thus does not get a hit point.

4.3 Experiments and results

We perform the automatic evaluation experiment on the test set, which contains 9309 question-answer pairs in total. Recall that we discard queries that cannot be answered correctly, which leaves us with 2661 question-answer pairs. We evaluate the following explanation methods:

- ϕ_{aw_1} - attention weights at first hop
- ϕ_{aw_3} - attention weights at third (last) hop

- $\phi_{aw_{avg}}$ - average attention weights
- ϕ_{lime} - LIME with 1000 samples per instance
- ϕ_{ip} - input perturbation (IP)

A baseline that samples a random fact for $rmax(\dots)$ is used for reference.

Table 2 shows pointing game accuracies and the absolute number of hit points achieved by all five explanation methods and the baseline. All methods beat the random baseline.

IP is the most successful explanation method with a pointing game accuracy of 0.97, and **LIME** comes second. Note that we did not tune the number of samples per query drawn by LIME, but set it to 1000. It is possible that as a consequence, queries with large fact sets are not sufficiently explored by LIME. On the other hand, a high number of samples is computationally prohibitive, as TextKBQA has to perform one inference step per sample.

Attention weights at hop 3 performs best among the attention-based methods, but worse than LIME and IP. We suspect that the last hop is especially relevant for selecting the answer entity. The poor performance of attention is in line with recent work by Jain and Wallace (2019), who also question the validity of attention as an explanation method.

We perform significance tests by means of binomial tests (with $\alpha = 0.05$). Our null hypothesis is that there is no significant difference in hit scores between a given method and the next-highest method in the ranking in Table 2. Differences are statistically significant in all cases, except for the difference between attention weights at hop 3 and average attention weights ($p = 0.06$).

Explanation method	Hit points	Pointing game acc.
attention weights at hop 1	1849	0.69
attention weights at hop 3	2116	0.80
average attention weights	2081	0.78
LIME	2271	0.85
IP	2570	0.97
random baseline	1458	0.55

Table 2: Hit points and pointing game accuracy. 2661 out of 9309 test set questions were used.

5 Evaluation with human annotators

The main goal of explanation methods is to make machine learning models more transparent for humans. That is why we conduct a study with human annotators.

Our experiment is based on the trust evaluation study conducted by Selvaraju et al. (2017) which, in turn, is motivated by the following idea: An important goal of interpretability is increasing users’ trust in ML models, and trust is directly impacted by how much a model is understood (Ribeiro et al., 2016). Selvaraju et al. (2017) develop a method to visualize explanations for convolutional neural networks on an image classification task, and evaluate this method in different ways.

One of their experiments is conducted as follows: Given two models, one of which is known to be better (e.g., to have higher accuracy), instances are chosen that are classified correctly by both models. Visual explanations for these instances are produced by the evaluated explanation methods, and human annotators are given the task of rating the reliability of the models relative to each other, based on the predicted label and the visualizations. Since the annotators see only instances where the classifiers agree, judgments are based purely on the visualizations. An explanation method is assumed to be successful if it helps annotators identify the better model. The study confirmed that humans are able to identify the better classifier with the help of good explanations.

We perform a similar study for our use case, but modify it as described below.

5.1 Experimental setup

We use two TextKBQA Models, which are trained differently:

- *model A* is the model used above, with a test set F_1 of 40

- *model B* is a TextKBQA model with a test set F_1 of 23. The lower score was obtained by training the model for fewer epochs and without pre-training in ONLYKB mode (see (Das et al., 2017)).

We only present annotators with query instances for which both models output the same answer. However, we do not restrict these answers to be the ground truth. We perform the study with three explanation methods: average attention weights, LIME and IP. We apply each of them to the same question-answer pairs, so that the explanation methods are equally distributed among tasks.

Every task contains one query and its predicted answer (which is the same for both models), and explanations for both models by the same explanation method. In contrast to image classification, it would not be human-friendly to show participants all input components (i.e., all facts), since their number can be up to 5500. Hence, we show the top5 facts with the highest relevance score. The order in which model A and model B appear on the screen (i.e., which is “left” and which is “right” in Figure 2) is random to avoid biasing annotators.

Annotators are asked to compare both lists of top5 facts and decide which of them explains the answer better. This decision is not binary, but five options are given: *definitely left*, *rather left*, *difficult to say*, *rather right* and *definitely right*. The interface is presented in Figure 2.

25 computer science students, researchers and IT professionals took part in our study and annotated 600 tasks in total.

5.2 Results

As shown in Table 3, the answer *difficult to say* is the most frequent one for all explanation methods. For attention weights and LIME there is a clear trend that, against expectations, users found fact lists coming from model B to be a better explanation. The total share of votes for *definitely*

Question:

Answer:

Which list of facts explains the answer to the query better: facts on the **left** or facts on the **right**?

<p>Left</p> <ul style="list-style-type: none"> • Now Jacob awaked out from Egypt. • So Jacob went down to Egypt. • Then Jacob went into Egypt. • Jacob had to serve through Esau. • And Jacob went into Egypt. 	<p>Right</p> <ul style="list-style-type: none"> • Jacob people.deceased Ibrahim Mosque • Jacob people.deceased Egypt • Jacob people.marriage.spouse Bilhah • Jacob people.marriage.spouse Leah • Jacob people.marriage.spouse Rachel
--	--

Definitely **left**
 Rather **left**
 Difficult to say
 Rather **right**
 Definitely **right**

Figure 2: Interface for the human annotation study.

model B and *rather model B* makes up 49.5% for attention weights and 29% for LIME, while *definitely model A* and *rather model A* gain 19.5% and 23.5%, respectively. In contrast to that, IP achieves a higher share of votes for model A than for model B: 16.5% vs. 10.5%.

Analogously to Selvaraju et al. (2017), we compute an aggregate score that expresses how much an explanation method helps users to identify the better model. Votes are weighted in the following way: *definitely model A* +1, *definitely model A* +0.75, *difficult to say* +0.5, *rather model B* +0.25 and *definitely model B* +0. We then compute a weighted average of votes for all tasks per explanation method. In this way, scores are bounded in $[0, 1]$ like the values of the hit score function used for the automatic evaluation. Values smaller than 0.5 indicate that the less accurate model B was trusted more, while values larger than 0.5 represent a higher level of trust in the more accurate model A. According to this schema, attention weights achieve a score of 0.386 and LIME achieves a score of 0.476. The score of the IP method is 0.524, which means that participants were able to identify the better model A when explanations were given by IP.

Significance tests show that while attention weights perform significantly worse than other methods, the difference between LIME and IP is insignificant, with $p = 0.07$. A larger sample of data and/or more human participants may be necessary in this case.

We also collected feedback from participants and performed qualitative analysis on the evaluated fact lists. The preference for the *difficult to*

say option can be explained by the fact that in many cases, both models were explained with the same or very similar fact lists. In particular, we found that IP provided identical top five fact lists in 120 out of 200 tasks. In the case of attention weights and LIME, this occurs only in 9 and 10 cases out of 200 tasks.

Another problem mentioned by annotators was that KB facts are not intuitive or easy to read for humans that have not dealt with such representations before. It would be interesting to explore if some additional preprocessing of facts would lead to different results. For example, KB facts could be converted into natural language sentences, while textual facts could be presented with additional context like the previous and the next sentences from the original document. We leave such preprocessing to future work.

6 Related work

Rychalska et al. (2018) estimate relevance of words in queries with LIME to test the robustness of QA models. However, they do not analyze the importance of the facts used by these QA systems.

Abujabal et al. (2017) present a QA system called QUINT that provides a visualization of how a natural language query is transformed into formal language and how the answer is derived. However, this system works only with knowledge bases and the explanatory system is its integral part, i.e., it cannot be reused for other models. Zhou et al. (2018) propose an out-of-the-box interpretable QA model that is able to answer multi-relation questions. This model is explicitly designed to work only with KBs. Another approach

	avg. attention weights	LIME	IP
definitely model A	6.0%	6.5%	5.0%
rather model A	13.5%	17.0%	11.5%
difficult to say	31.0%	47.5%	73.0%
rather model B	28.0%	18.5%	9.0%
definitely model B	21.5%	10.5%	1.5%
aggregate score	0.386	0.476	0.524

Table 3: Percentage distribution of votes, and aggregate score, from the human annotation study.

for interpretable QA with multi-hop reasoning on knowledge bases is introduced by Murugan et al. (2018). They claim that the transparent nature of attention distributions across reasoning steps allows humans to understand the model’s behavior.

To the best of our knowledge, the interpretability of QA models that combine structured and unstructured data has not been addressed yet. Even in the context of KB-only QA models, no comprehensive evaluation of different explanation methods has been performed. The above-mentioned approaches also lack empirical evaluation with human annotators, to estimate how useful the explanations are to non-experts.

7 Conclusions

We performed the first evaluation of different explanation methods for a QA model working on a combination of KB and text. The evaluated methods are attention, LIME and input perturbation. To compare their performance, we introduced an automatic evaluation paradigm with fake facts, which does not require manual annotations. We validated the ranking obtained with this paradigm through an experiment with human participants, where we observed the same ranking. Based on the outcomes of our experiments, we recommend the IP method for the TextKBQA model, rather than the model’s self-explanatory attention mechanism or LIME.

References

Abdalghani Abujabal, Rishiraj Saha Roy, Mohamed Yahya, and Gerhard Weikum. 2017. [Quint: Interpretable question answering over knowledge bases](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 61–66. Association for Computational Linguistics.

Yonatan Bisk, Siva Reddy, John Blitzer, Julia Hockenmaier, and Mark Steedman. 2016. [Evaluating in-](#)

[duced ccg parsers on grounded semantic parsing](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. [Freebase: a collaboratively created graph database for structuring human knowledge](#). In *SIGMOD Conference*.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017. [Question answering on knowledge bases and text using universal schema and memory networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–365. Association for Computational Linguistics.

Finale Doshi-Velez and Been Kim. 2017. [Towards a rigorous science of interpretable machine learning](#). *arXiv preprint*. ArXiv:1702.08608v2.

Evgeniy Gabrilovich, Michael Ringgaard, and Amar-nag Subramanya. 2013. [Facc1: Freebase annotation of cluweb corpora, version 1 \(release date 2013-06-26, format version 1, correction level 0\)](#).

Sarthak Jain and Byron C. Wallace. 2019. [Attention is not explanation](#). *arXiv preprint*. ArXiv:1902.10186.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. [Understanding neural networks through representation erasure](#). *arXiv preprint*. ArXiv:1612.08220.

Zachary Chase Lipton. 2018. [The mythos of model interpretability](#). *Queue*, 16(3):30:31–30:57.

Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. [Key-value memory networks for directly reading documents](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409. Association for Computational Linguistics.

Selvakumar Murugan, Suriyadeepan Ramamoorthy, Vaidheeswaran Archana, and Malaikannan Sankar-subbu. 2018. [Compositional attention networks for interpretability in natural language question answering](#). *arXiv preprint*. ArXiv:1810.12698.

- Nina Poerner, Benjamin Roth, and Hinrich Schütze. 2018. [Evaluating neural network explanation methods using hybrid documents and morphological agreement](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers*, pages 340–350, Melbourne, Australia.
- Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Vaughan, and Hanna M. Wallach. 2018. [Manipulating and measuring model interpretability](#). *arXiv preprint*. ArXiv:1802.07810.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. ["why should i trust you?": Explaining the predictions of any classifier](#). In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pages 1135–1144, New York, NY, USA. ACM.
- Sebastian Riedel, Limin Yao, Andrew Mccallum, and Benjamin M Marlin. 2013. [Relation extraction with matrix factorization and universal schemas](#). *Proceedings of NAACL-HLT 2013*, pages 74–84.
- Barbara Rychalska, Dominika Basaj, and Przemyslaw Biecek. 2018. [Are you tough enough? framework for robustness validation of machine comprehension systems](#). In *Interpretability and Robustness for Audio, Speech and Language Workshop*, Montreal, Canada.
- Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. [Grad-cam: Visual explanations from deep networks via gradient-based localization](#). In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, Venice, Italy.
- Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. 2018. [An interpretable reasoning network for multi-relation question answering](#). In *COLING*, pages 2010–2022, Sante Fe, USA.