Paraphrase for Open Question Answering: New Dataset and Methods

Ying Xu[†], **Pascual Martínez-Gómez**[‡],
Yusuke Miyao[§], **Randy Goebel**[†]
[†]Department of Computing Science, University of Alberta
[‡]National Institute of Advanced Industrial Science and Technology (AIST)
[§]National Institute of Informatics / JST, PRESTO
[†]{yx2,rgoebel}@ualberta.ca
[‡]pascual.mg@aist.go.jp
[§]yusuke@nii.ac.jp

Abstract

We propose a new open question answering framework for question answering over a knowledge base (KB). Our system uses both a curated KB, Freebase, and one that is extracted automatically by an open information extraction model, IE KB. Our system consists of only one layer of paraphrase, compared to the three layers used in a previous open question answering system (Fader et al., 2014). However, because of the more accurately extracted relation triples in IE KB, combined with linked entities from IE KB to Freebase, our system achieves a 7% absolute gain in F_1 score over the previous system.

1 introduction

There are two broad classes of systems that provide question answering (QA) from a knowledge base (KB). One uses semantic parsing (Berant and Liang, 2014; Reddy et al., 2014; Yih et al., 2015), and the other uses information extraction (IE). Semantic parsing systems depend on highly accurate knowledge bases such as Freebase, which are accurate but incomplete (Riedel et al., 2013; Fader et al., 2014). However, although semantic parsing systems currently achieve higher performance than IE-based systems, we think it is desirable to continue to develop the latter as an alternative or in combination with the former.

One major challenge for question answering from a KB is the many ways in which relations can be expressed. On the one hand, we need to deal with language variability, for example, acknowledging that the following two questions have the same meaning: "*What character did Natalie Portman play in Star Wars?*" and "*What is the role of Natalie Portman in Star Wars?*". We call this NL-NL paraphrasing, since it requires the identification of a map between two natural language expressions. On the other hand, we need to bridge the gap between the expression of relations in a curated knowledge base, such as Freebase, and relations conveyed in natural language sentences. We refer to this as NL-KB paraphrasing. For instance, general QA will require a mapping between the natural language relation *brother* and the Freebase relation "/people/person/sibling_s."

Our contribution to Open IE question answering is three-fold. First, we provide a new wide-coverage store of automatically extracted relation triples, which has a higher precision than that of previous work (Fader et al., 2014). This Open IE triple store allows us to tackle the NL-NL paraphrasing problem. Second, the entities in our dataset is linked to Freebase, which allows us to tackle the NL-KB paraphrasing problem. Third, we propose a simple and effective open QA framework that consists of a single paraphrase layer. In this framework, we perform searches on both Open IE KB and Freebase. This single-layered paraphrase model allows us to test and compare a variety of paraphrasing models. Experiments on the WebQuestion set (Berant et al., 2013) shows that our system exhibits better performance than the two IE-based systems (Fader et al. (2014), Yao and Van Durme (2014)), and is comparable to (Berant et al., 2013), where their average F-score is 35.7%, and ours is 37.9%.

Proceedings of 2016 NAACL Human-Computer Question Answering Workshop, pages 53–61,
San Diego, California, June 12-17, 2016.

2 Related Work

A current major research thread in QA is to cast it as a semantic parsing problem, where the objective is to map a natural language question into a formal language, i.e., a database query. This query is then run on a database, and results are returned to the user.

SCISSOR (Ge and Mooney, 2005) was one of the first successful attempts to create a robust semantic parser. It worked by first syntactically parsing a question, augmenting the result with semantic information, and then transforming the result into a logical language. However, this process requires a large volume of training supervision, namely "gold standard" annotations of semantically-augmented syntactic trees paired with their logical representations. Its demonstration was limited to GeoQuery (Zelle and Mooney, 1996), which is a very restricted domain and database. A more recent approach to achieve robust, open-domain semantic parsing is that of Berant et al. (2013), where the training supervision is limited to pairs of questions and answers. In their approach, they use latent λ-DCS (Liang, 2013) logical formulas for their meaning representations, which can be converted deterministically into Sparql queries on Freebase. Yih et al. (2015) developed the best performing semantic parser on the WebQuestion set (Yih et al., 2015). Inspired by (Yao and Van Durme, 2014), instead of searching on the whole knowledge base, they defined a query graph which is more closely related to the target entity in the questions.

Regarding IE QA systems, there are two representatives. Yao and Van Durme (2014) constructs a graph structure to represent each KB topic, which is searched to retrieve answers to questions. Fader et al. (2014)'s system constructs the KB as a triple database, where each triple consists of two entities and one relation phrase. In the base of the KB approach, we follow the design of the latter, by querying on the relation triple database. Moreover, our knowledge base contains both curated data, i.e., Freebase, and our own automatically extracted IE KB. Our relation triples were also extracted from the ClueWeb09 corpus, but we used the Open IE system of Xu et al. (2015), which is based on dependency parsing. There are two main differences between our triples and Fader's: 1. we achieve higher precision of triple extraction, partly because of the dependency parser; 2. our entities are linked to Freebase by an entity linking approach, instead of pure string match. These two differences provide both higher precision and higher recall for question answering.

Fader et al. (2014)'s system exploits several levels of paraphrase. One is the process of paraphrasing from one question to another question, for example, from "How does _ affect your body?" to "What body system does _ affect ?" One is *parsing*, which converts natural language questions into a small number of high-precision templates. For example, from "Who/What is NP_{arg}" to (arg, is-a, ?x). Another is query-rewriting, which is similar to our paraphrase operator. For example, its paraphrase operator re-writes *children* to *was born to*. Our system, however, only uses a single layer paraphrasing, which makes our approach simpler and more intuitive, while achieving higher performance.

3 Our DataSet

To augment our QA system, we create a new Open IE relation triple dataset that contains sentences and document IDs from which the triples are extracted. The relation triples are extracted from ClueWeb09 by the Open IE system of Xu et al. (2015). Each triple contains two arguments, which are entities from Freebase, and one relation phrase. The arguments' Freebase IDs are provided by FACC1 corpus[1]. The relations are lemmatized and the sentences that contain the triples are provided. As a result, the dataset contains more than 300 million relation triples[2].

Table 1 shows one relation triple example, including the originating parsed sentence. The relation triple is from the 485_{th} sentence of the document *clueweb12-0700tw-51-00204*. The relation word is *director*, which is the 4th word in the sentence. The two arguments *Raul Gonzalez* and *National Council of La Raza* have corresponding Freebase IDs: /m/02rmsx3 and /m/085f3n.

Many NLP tasks can potentially benefit from such data. For example, for question answering task, there are at least three advantages. One is that the

[1] http://lemurproject.org/clueweb09/

[2] We use only half of the ClueWeb09 because of resource limits.

Relation Triple Example
<doc>clueweb12-0700tw-51-00204 <relation>485, Raul Gonzalez /m/02rmsx3, National Council of La Raza /m/085f3n, <E1> →(appos) director →(prep_at) <E2> director 4 (ROOT (S (NP (NP (NNP Raul) (NNP Gonzalez)) (, ,) (NP (NP (JJ legislative) (NN director)) (PP (IN at) (NP (NP (DT the) (NNP National) (NNP Council)) (PP (IN of)(NP (NP (NNP La) (NNP Raza)) ...)

Table 1: Example of a relation triple extracted from ClueWeb09, with its source sentence and document ID.

entities are linked to Freebase, which will identify entities that represent one object but with different instances. Secondly, the triple is associated with the parsed sentences and the document ID, which can provide better evidence for questions with n-ary relations, such as "What character did Natalie Portman play in Star Wars?" Finally, we can provide explanations, i.e., by identifying sentences that are evidence in support of our answers. We believe that this large volume of linked triples may not only improve the mapping between the natural language and Freebase relations, but also improve the recall of questions, as we can also search based on entities' Freebase IDs. We intend to make this data publicly available.

4 System Structure

Figure 1 presents our general framework for open question answering.

The first component of our system is *query preprocessing*. We use the Stanford CoreNLP tool for entity extraction and sentence parsing. Entities such as persons and organizations have higher priority as target entities than those such as numbers and dates. We then extract the dependency path between the chosen target entity and the question phrase. The question phrase can be a single word such as *where* or multi-word phrases such as *which character*. The words on the dependency path are considered to be the relation words, i.e., the predicates.

The second component is our *paraphrase recognition*, i.e., the identification of a mapping between relations in the questions and relations in the two knowledge bases. Further details of our paraphrase models are in Section 5.

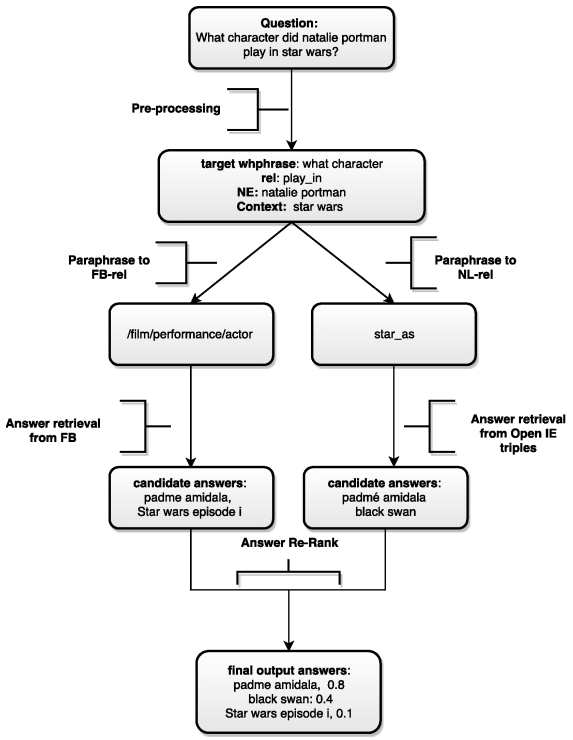

Figure 1: Our open question answering system structure.

The third component is *answer retrieval*. Our knowledge base contains triples from Open IE extraction and Freebase. We retrieve answers based on the target entity, the relation words in the question, the corresponding paraphrase relations, and context words.

The final component is *answer re-ranking*. Since we expect the system to retrieve multiple possible answers, those answers are re-ranked with an SVM-Rank system (Tsochantaridis et al., 2004). Further details of the model and its features are described in Section 7.

5 Paraphrase

Here we concentrate on word/phrase level paraphrase instead of sentence level paraphrase. We noticed that literal paraphrase may not be sufficient for accurate question answering. For example, that a person *lives* in a location, might not be a good paraphrase for a person *is buried* at a location. But the

two have such high correlation that identifying the correlation will help answering questions such as "Where did a person live." The question answering community has currently chosen to adapt either a machine translation model or a simple pointwise mutual information (PMI) model. Here we present the three models we adapted for our question answering system.

The first one is a frequency based model. Paraphrase rules are learned based on the question answering training set. Given a set of questions, the relation words in the questions are extracted as mentioned in Section 4. Then we retrieve the relation triples in the KB that contain both the target entities and the answers. The score of a rule, query_Rel $\rightarrow$ KB_Rel, is the frequency of the entity pairs that the query relation and KB relation share.

The second is a PMI based model, which is adapted by (Fader et al., 2014)'s QA system. The general function of the PMI is:

$$log\frac{P(x,y)}{P(x)P(y)} \quad (1)$$

where $P(x) = count(x)/n$, n is the number of triples. For paraphrase, the frequency of unigrams represents the number of entity pairs associated with one relation. The frequency of bigrams represents the number of entity pairs shared by two relations.

The third is the highly-cited DIRT algorithm (Lin and Pantel, 2001). In this model, relations are represented by two vectors. Each vector represents one argument slot. Their similarity score function is defined as follows:

$$\sqrt{sim(v_l^x, v_l^y) * sim(v_r^x, v_r^y)}. \quad (2)$$

where v_l^x is a word vector representing the relation x's left argument slot. The value of the vectors is the PMI between the relation and the argument.

Our paraphrase model uses a subset of our highly accurate Open IE triples. The restriction is that the relation words should appear between the parsing dependency path of two entities, and that the maximum path length is 3. This restriction will increase the precision of the triples, and improve the performance of the paraphrase.

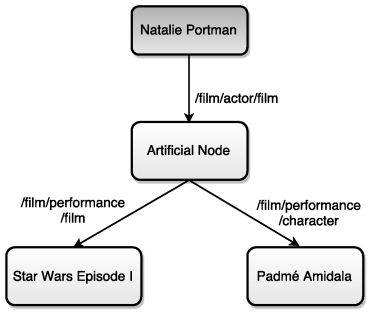

Figure 2: An example of compound artificial nodes.

6 Answer Retrieval

We designed our system so that we can search by both entity names and entity freebase IDs. In addition, the system also considers other context words of a question besides the target entity and relation words.

Our knowledge base contains triples from Open IE extractions and Freebase. The Open IE triples are extracted from ClueWeb documents using the models of (Xu et al., 2015). The Freebase triples contain entity pairs that are directly linked by a relation or by a compound relation, e.g., "/film/actor/film /film/performance/character," (Figure 2) which we hereafter abbreviate as "/film /character." Compound relations are represented by links that connect two entities with distance two. The middle node in between is an artificial node, instead of an entity. It is used to represent relations or events with multiple attributes, such as date, location etc. As pointed out by (Yao and Van Durme, 2014) compound relations are important for the question answering task.

We use Lucene, an information retrieval framework tool, to store the triples. Every triple is constructed as one document that contains 6 fields: left and right argument's surface forms, left and right argument's Freebase IDs, the relation phrase, and the context. The OpenIE triples' context are the sentences containing that triple. The Freebase triples' context are the compound node text for compound relations. For instance, the context of the triple (natalie portman, /film /character, padmé amidala) is "padmé amidala - Star Wars episode ii: attack of the clones - freebase data team - film performance."

To search for the answers, we identify the target named entity of the query and its Freebase IDs. The Freebase ID candidates are extracted by Freebase

API. We also identify the relation word set, which includes both the relations in the question and their paraphrases, as mentioned in Section 5. The original relations are assigned a weight of 2, while others are assigned with a normalized paraphrase score between $[0, 1]$. The normalization assigns 1 to the best paraphrase relations and 0 to the n_{th}, where n is the maximum number of paraphrases.

For explanation purposes, consider again the question "What character did Natalie Portman play in Star Wars?" We extract the top three Freebase ID candidates for *natalie portman*: {/m/09l3p, /m/0dncctc, /m/05ngby1}. With the relation *play*, we build a relation set {*play*} ∪ {*/actor, /written_by, /character*} ∪ {*voice, set, quarterback, star, lead*}. The terms {*/actor, /written_by, /character*} are Freebase relations that have a high paraphrase score with *play*. The context is the word set of the sentence. We then search for the target entity on either the left argument slot or the right, the relation, and the context. The score of a retrieved triple is the weight of the hit relation's paraphrase score. Different relations can lead to the same answer; we currently sum the scores of all retrieved triples for one answer.

7 Supervised Re-Ranking

Answers retrieved by the previous step are based only on word match. Because of the noise retained in paraphrase and relation extraction, we need to add more information to filter incorrect answers. Here we incorporate some extra information as features of an SVM-Rank model (Tsochantaridis et al., 2004).

Table 2 shows the features we proposed for the re-ranking system. Freebase types are extracted according the candidate answer's Freebase IDs, and the value of the Freebase relation /type. As there are thousands of types in Freebase, we further focus by reducing the types to around 100 clusters, as proposed by (Ling and Weld, 2012). The type feature will consider the types of answers, and diminishes those retrieved answers with an inconsistent type. Another source of type information is taken from our Open IE triples. We observe that most types can be considered as a relation between entities. For example, in the triple (Shakespeare, a playwright, England) *Playwright* is a relation between *Shakespeare* and *England*, and can be considered as one type of *Shakespeare*. This leads to our Feature 4: for example, for the question "which country invades Poland" with the candidate answer *Germany*, the feature is the frequency of (*country, Germany*) in the Open IE triple set.

For extra information from the paraphrase rules, we use the rule itself and the scores, which can be determined from the three different paraphrase models.

The *context words hit rate* feature is determined by the percentage of context words in the question that are also in the triple's context words. The target entity, the relation words and question words are excluded from context words. This can improve the answering of questions with n-ary relations. For example, for the question *What character did Natalie Portman play in Star Wars?* we check how many words in the set {Star, Wars} are in the candidate answer's triple context. Suppose the two candidate answers are: *padmé amidala*, with context "padmé amidala - Star Wars episode ii: attack of the clones - freebase data team - film performance;" *black swan*, with context "academy award for actress in a leading role - 83rd academy awards - black swan - role: nina sayers - 2010 - nanette - award honor." The first answer's *context words hit rate* is 1, as the context has {Star, Wars}. The second has hit rate 0.

In training, we use the top n answers from both the Open IE and Freebase. However if a correct answer is ranked below n, we still add it as a training instance. This increases the number of positive training instances. In testing, we only re-rank the top n answers, where n is set according to the performance in the development set.

To combine Freebase and the Open IE KB, we have considered two alternatives. One combines the two sets of answers and then re-ranks the combined answer set. The other is to re-rank the two sets independently, and combine them by assigning weights to different KBs.

8 Experiments

We evaluate our question answering system based on the question set provided by Berant et al. (2013). The questions are generated by the Google Suggest API. The answers are created by Amazon Mechanical Turk. One property of this dataset is that the

Features ID	Feature description
1	Namespace: whether the answer is extracted from Open IE KB, Freebase KB, or both.
2	Candidate answer Freebase types + whphrase, e.g. /location/country + what country
3	Candidate answer Freebase types + relation in the question, e.g. /location/country + invade
4	Frequency of (the type word in the question, answer) in the ClueWeb
5	Shape of the answer (*e.g.* has numbers, multiwords) + the question phrase
6	Paraphrase rules
7	Whether the paraphrase rule from the 3000 training sentences
8	Paraphrase rule with the max score, its score, and which paraphrase approach leads to it
9	Context words hit rate
10	Score of the answer with answer retrieval

Table 2: Features for the Supervised System.

answers are guaranteed to be found in Freebase.

Our experiments attempt to answer several questions:

- Is our dataset useful for the paraphrase tasks?
- Which popular paraphrase approach is more suitable for question answering?
- Is our system better than other Information Extraction-based systems?

There are at least two metrics used in the literature:

1. Top 1 F_1 score, as used by (Fader et al., 2014). Every system outputs only one answer. The system's answer is the entity with highest score (randomly pick one if there is a tie). No answer is produced if the highest score is below a certain threshold. An answer is considered correct if the entity in the system's answer appears in the gold answer. The precision, recall and F_1 score are calculated globally:

$$\text{Precision} = \frac{\#\text{ questions with correct answers}}{\#\text{ questions with answers}}$$
$$\text{Recall} = \frac{\#\text{ questions with correct answers}}{\#\text{ questions}}$$

2. Average F_1 score (accuracy). This is used by semantic parsing question answering systems such as (Berant and Liang, 2014; Yih et al., 2015). For every question, the precision, recall and F_1 score are computed between the systems' answer set and the gold answer set. Then, the F_1 scores are averaged across all questions in the test set. This metric is used to reward partially-complete system answers.

In the following experiments, we will compare our system with Fader's with respect to the first metric, and the rest with the second metric.

8.1 The Effect of Dataset size

We demonstrate the effect of different dataset sizes by estimating a paraphrase PMI model from a smaller subset of our data, and then comparing QA systems' performance with these alternative paraphrase sets. Initially, we use the recall as the comparative metric. Recall is calculated as the percentage of questions that can be answered by the top 30 candidate answers retrieved. To filter the effect of features and supervised models, results are based on answer retrieval on Freebase, and no re-ranking is given.

For every question we extract the top 100 Freebase relations as paraphrase of one question relation (100 is set on a development set). The paraphrase score is used as weight on the answer retrieval phase. We use a PMI model as the paraphrase model metric.

Our baseline consists of 800k triples, which is larger than the size of one existing relation triple dataset from Riedel et al. (2013) (200k). The whole paraphrase set has 26 million triples. Note that this is a smaller number of triples but with higher precision, compared with the whole Open IE knowledge base (300 million). When using 800k triples, the recall is 10.7%, whereas we obtain a recall of 34.5% when using 26 million triples. We notice that the performance difference is dramatic.

8.2 Paraphrase

Here we show the paraphrase effect on the Freebase-based and IE KB based question answering systems.

As mentioned, for the paraphrase between question relations and Freebase relations, we extract the top 100 Freebase relations for one question relation.

models	recall on the top 30
Freq3000	63.2%
Freq3000+PMI	64.5%

Table 3: Comparing different paraphrase models. Recall on the top 30, based on Freebase.

models	recall on the top 40
Freq3000	40.5%
Freq3000 + PMI	40.8%
Freq3000 + DIRT	40.7%

Table 4: Comparing different paraphrase models. Recall on the top 40, based on Open IE KB.

The paraphrase score is used as a weight on the answer retrieval.

Table 3 shows the recall measure on the top 30 answers with alternative paraphrase models based on Freebase. Freq3000 is the case where we use only the 3000 training sentences and the Open IE triple set. Freq3000+PMI is the supervised paraphrase, Freq3000, plus the unsupervised paraphrase with the PMI measure. The results show that the unsupervised paraphrase, which maps between Open IE triples and FB triples, does improve recall. It is not meaningful to use DIRT here because the frequency of every triple is 1 in Freebase, and the frequency of (named entity 1, Freebase relation) is based on the number of values of this slot.

For the paraphrase between question relations and Open IE KB relations, we extract the top 10 Open IE relations for one question relation (6-10 do not display much difference on the development set). Table 4 shows the top 40 answer recall values, with alternative paraphrase models based on Open IE knowledge base. There is no obvious difference among the alternatives. The paraphrase between natural language relations is more difficult than paraphrase between natural language relations and Freebase relations. One reason might be that Open IE relation extraction noise is amplified within the overall process.

8.3 State-of-the-Art

As mentioned previously, our system is an Information Extraction (IE) based system. Table 5 shows the results of our system comparing with the other two IE based systems. Yao14 (Yao and Van Durme, 2014)'s system use FB knowledge base solely; while Fader14 (Fader et al., 2014) use both FB KB and Open IE KB. Our one layer system based on Freebase is much better than the one based on the IE triples. This makes the normal combination method failed, i.e. the two methods mentioned in Section 7. Instead, for the combination system we use OneLayer_IE 's results only when OneLayer_FB returns no answers. Our system is better than both the previous IE-based systems. It is better than Fader14 with an absolute F1 gain of 7% although both systems use FB KB and IE KB. Our system based solely on FB KB is already better than Yao14, which also is based on FB KB only. Notice instead of measured manually (as in Fader14), our system is automatically measured on the WebQuestion answer set, which means the performance is under-estimated, as we will show in Section 8.4.

We also compare our system with several semantic parsing-based systems: Berant *et al.* 's systems Berant13, Berant14; and Microsoft system MS15 (Yih et al., 2015), which is a semantic parsing-based system that achieves the current best performance on the WebQuestion set. Table 6 shows the results. Our system is the first information extraction based system that performs better than Berant13 on the Freebase data.

models	avg P	avg R	avg F_1
OneLayer_IE	29.0	24.2	20.4
Berant13	48.0	41.3	35.7
OneLayer_FB	48.3	45.0	37.2
OneLayer_combine	40.3	45.7	37.9
Berant14	40.5	46.6	39.9
MS15(FB search API)	49.8	55.7	48.4

Table 6: Results that compare our system with semantic parsing-based question answering systems.

8.4 Error Analysis

One problem of using the WebQuestion set as evaluation data is that the gold standard set is incomplete. This is caused both by incompleteness of Freebase and by human error. To show the effect of incompleteness on the test result, we annotated 400 development set questions, to determine whether the top answers from our systems are correct. Table 7 compares the results on the original answer set and the

models	top1_P	top1_R	top1_F_1	avg P	avg R	avg F_1
Fader14	-	-	35	-	-	-
Yao14(FB search API)	-	-	-	51.7	45.8	33.0
OneLayer_FB	44.7	39.5	**41.9**	48.3	45.0	37.2
OneLayer_IE	28.5	26.6	27.5	29.0	24.2	20.4
OneLayer_combine	41.4	40.3	40.8	40.3	45.7	**37.9**

Table 5: Results that compare our system with other IE-based question answering systems.

expanded answer set. With the top1 measure, the absolute difference is 10%. To avoid the manual annotation for every system, our future work will expand the answers for all the questions. We will also do a more complete analysis of the effect of the incomplete answer set on the training process.

dataset	top1_P	top1_R	top1_F_1
original	30.9	29.2	30.0
expanded	42.0	39.7	40.8

Table 7: The results of the open question answering system on the original development set and the one with expanded answers.

When we look more closely at our systems' errors, we notice that one problem of IE KB-based systems is that they can not find numbers or common nouns such as "writer" as an answer. This is because of a weakness in the data extraction process, which is designed to extract relations between named entities that are identified by the entity linking systems. Consider these two examples:

Question1 "what kinda music does john mayer sing?"
Gold standard "Rock music".
Our answer "your body is a wonderland", a song by john mayer.

Question2 "what does jennifer lopez do?"
Gold standard "Actor"
Our answer "american idol", a TV show where jennifer lopez was a judge.

This naturally creates a thread for future work: expand the Open IE triple set to include arguments with common nouns or numbers.

9 Conclusion

We have designed and tested a new open question answering (Open QA) framework for question answering over a knowledge base (KB). Our system consists of only one layer of paraphrase, compared to the three layers used in a previous open question answering system (Fader et al., 2014). However, because of the more accurately extracted relation triples, and use of linked entities from IE KB to Freebase, our system achieves a 7% absolute gain in F_1 score over the previous Open QA system.

An acknowledged problem with our data is that both arguments are named entities, which make our Open IE KB based-system unable to answer questions with answers that are common nouns or numbers. That will be addressed in future work.

References

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Association for Computational Linguistics (ACL)*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of EMNLP*.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1156–1165, New York, NY, USA. ACM.

Ruifang Ge and Raymond J. Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, CONLL '05, pages 9–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

Percy Liang. 2013. Lambda dependency-based compositional semantics. *CoRR*, abs/1309.4408.

Dekang Lin and Patrick Pantel. 2001. DIRT @SBT@discovery of inference rules from text. In

Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01, pages 323–328, New York, NY, USA. ACM.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *In Proc. of the 26th AAAI Conference on Artificial Intelligence*.

Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392.

Sebastian Riedel, Limin Yao, Benjamin M. Marlin, and Andrew McCallum. 2013. Relation extraction with matrix factorization and universal schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '13)*, June.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 104–, New York, NY, USA. ACM.

Ying Xu, Christoph Ringlstetter, Mi-Young Kim, Randy Goebel, Grzegorz Kondrak, and Yusuke Miyao. 2015. A lexicalized tree kernel for open information extraction. In *Proceedings of the 2015 Conference of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, July.

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 956–966. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the Joint Conference of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the AFNLP*. ACL Association for Computational Linguistics, July.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, pages 1050–1055.