

# Farewell Freebase: Migrating the SimpleQuestions Dataset to DBpedia

Michael Azmy\*, Peng Shi\*, Jimmy Lin, and Ihab F. Ilyas

David R. Cheriton School of Computer Science

University of Waterloo

Waterloo, Ontario, Canada

{mwazmy, p8shi, jimmylin, ilyas}@uwaterloo.ca

## Abstract

Question answering over knowledge graphs is an important problem of interest both commercially and academically. There is substantial interest in the class of natural language questions that can be answered via the lookup of a single fact, driven by the availability of the popular SIMPLEQUESTIONS dataset. The problem with this dataset, however, is that answer triples are provided from Freebase, which has been defunct for several years. As a result, it is difficult to build “real-world” question answering systems that are operationally deployable. Furthermore, a defunct knowledge graph means that much of the infrastructure for querying, browsing, and manipulating triples no longer exists. To address this problem, we present SIMPLEDBPEDIAQA, a new benchmark dataset for simple question answering over knowledge graphs that was created by mapping SIMPLEQUESTIONS entities and predicates from Freebase to DBpedia. Although this mapping is conceptually straightforward, there are a number of nuances that make the task non-trivial, owing to the different conceptual organizations of the two knowledge graphs. To lay the foundation for future research using this dataset, we leverage recent work to provide simple yet strong baselines with and without neural networks.

## 1 Introduction

Question answering over knowledge graphs is an important problem at the intersection of multiple research communities, with many commercial deployments. To ensure continued progress, it is important that open and relevant benchmarks are available to support the comparison of various techniques. In this paper, we focus on the class of questions that can be answered by a single triple (i.e., fact) from a knowledge graph. For example, the question “What type of music is on the album Phenomenon?” can be answered via the lookup of a simple fact—in this case, the “genre” property of the entity “Phenomenon”. Analysis of an existing benchmark dataset (Yao, 2015) and real-world user questions (Dai et al., 2016; Ture and Jojic, 2017) show that such questions cover a broad range of users’ needs.

The SIMPLEQUESTIONS dataset (Bordes et al., 2015) has emerged as the de facto benchmark for evaluating these simple questions over knowledge graphs. However, there is one major deficiency with this resource: the answers draw from Freebase. Unfortunately, Freebase is defunct and no longer maintained. This creates a number of insurmountable challenges: First, because the knowledge graph is stale, it is no longer possible to build a “real-world” operational QA system using models trained on SIMPLEQUESTIONS. Second, a defunct knowledge graph means that researchers must develop custom infrastructure for querying, browsing, and manipulating the graph. Thus, we are not able to leverage multiple cooperative and interchangeable service APIs that are deployed and maintained by different parties—which is the strength of the broader “open linked data” ecosystem. While it may be the case that one can apply transfer learning so that models trained on SIMPLEQUESTIONS can be re-targeted to another “live” knowledge graph, we are not aware of research along these lines.

---

\*These authors contributed equally.

This work is licensed under a Creative Commons Attribution 4.0 International License.  
License details: <http://creativecommons.org/licenses/by/4.0/>

Dataset	Training	Validation	Test	Total
SIMPLEQUESTIONS	75,910	10,845	21,687	<b>108,442</b>
SIMPLEDBPEDIAQA	30,186	4,305	8,595	<b>43,086</b>

Table 1: Statistics of SIMPLEQUESTIONS and SIMPLEDBPEDIAQA.

To address these issues, we present SIMPLEDBPEDIAQA, a new dataset that we have created by mapping entities and predicates that comprise the answers to SIMPLEQUESTIONS from Freebase to DBpedia. Unlike Freebase, DBpedia is actively maintained by a dedicated community. We describe how this dataset migration is accomplished via high-quality alignments between entities in the two different knowledge graphs, and explain many of the nuances that make the creation of this dataset non-trivial. Our new dataset includes a total of 43,086 questions and corresponding answers that cover 40% of the original dataset. Summary statistics of SIMPLEDBPEDIAQA and SIMPLEQUESTIONS are shown in Table 1. The complete dataset is available at <https://github.com/castorini/SimpleDBpediaQA>.

In addition to the contribution of providing the community with a new evaluation resource, we provide a series of simple yet strong baselines to lay the foundation for future work. These baselines include neural network models and other techniques that do not take advantage of neural networks, building on recently-published work (Mohammed et al., 2018). An additional contribution of this paper is that having two parallel datasets allows us to examine the effects of different conceptual organizations and knowledge graph structures: For example, we notice that many single-fact triples in Freebase require two-hop traversals in the DBpedia knowledge graph, which makes them no longer “simple” questions. Finally, evaluation resources targeting different conceptual organizations of knowledge help “keep researchers honest” in guarding against model overfitting on a single dataset.

## 2 Background and Related Work

The development and continual advance of question answering techniques over knowledge graphs require benchmark datasets that cover different aspects of the task. Quite obviously, each dataset has to target one (or more) knowledge graphs, which means that the structure of the answers are dictated by the conceptual organization of the particular knowledge graph.

Over the years, researchers have built a number of datasets based on Freebase (Bollacker et al., 2008). For instance, FREE917 (Cai and Yates, 2013) contains 917 questions involving 635 distinct Freebase predicates. WEBQUESTIONS (Berant et al., 2013) contains 5,810 question-answer pairs collected using the Google Suggest API and manually answered using Amazon Mechanical Turk (AMT). Both contain answers that require complex, multi-hop traversals of the knowledge graph. In contrast, the SIMPLEQUESTIONS dataset focuses on questions that can be answered via the lookup of a single fact (i.e., triple). Due to its much larger size and thus support for data-hungry machine learning techniques, this dataset has gained great popularity with researchers. Unfortunately, Google shut down Freebase in 2015; a final snapshot of the knowledge graph is still available online for download, but the associated APIs are no longer available.

Like Freebase, DBpedia (Bizer et al., 2009) has also been used as the target knowledge graph for multiple question answering datasets. For example, QALD<sup>1</sup> (Question Answering over Linked Data) is a series of evaluation campaigns focused on question answering over linked data. LC-QUAD (Trivedi et al., 2017) is another recent dataset that comprises 5,000 questions with answers in the form of SPARQL queries over DBpedia. These questions are relatively complex and require the integration of evidence from multiple triples. However, a more recent analysis by Singh et al. (2018) found that only 3,252 of the questions returned answers using the provided queries.

We are not the first to attempt to migrate SIMPLEQUESTIONS to another knowledge graph. Diefenbach et al. (2017) mapped the dataset from Freebase to Wikidata.<sup>2</sup> However, our migrated SIMPLE-

<sup>1</sup><https://qald.sebastianwalter.org/>

<sup>2</sup>[https://www.wikidata.org/wiki/Wikidata:Main\\_Page](https://www.wikidata.org/wiki/Wikidata:Main_Page)

DBPEDIAQA dataset has roughly twice the number of mapped questions. DBpedia is generally considered to be more mature than Wikidata due to its longer history, and thus we believe targeting DBpedia will ultimately yield higher-impact applications.

### 3 Problem Definition

We begin with a formal definition of our problem. Some preliminaries: Let  $E = \{e_1, e_2, \dots, e_r\}$  be a set of entities, where  $e_i$  is a Uniform Resource Identifier (URI) uniquely identifying each entity. Let  $P = \{p_1, p_2, \dots, p_s\}$  be a set of predicates. Let  $S \subseteq E$  be a set of subjects and  $O \subseteq (L \cup E)$  be a set of objects, where  $L$  is a set of literals. In this context,  $t = (s, p, o)$  denotes a Resource Description Framework (RDF) triple, comprised of a subject  $s \in S$ , a predicate  $p \in P$ , and an object  $o \in O$ .

Given this formalism, Freebase (Bollacker et al., 2008) represents a specific knowledge graph  $T^b$ , where  $T^b = \{t_1^b, \dots, t_m^b\}$  (i.e., a set of Freebase triples). Each Freebase entity is uniquely identified by a MID (Machine ID). Similarly, DBpedia (Bizer et al., 2009) represents another knowledge graph  $T^d$ , where  $T^d = \{t_1^d, \dots, t_n^d\}$ .

The SIMPLEQUESTIONS dataset is a collection of natural language questions and answers based on Freebase. Formally,  $Q^b = \{q_1^b, \dots, q_l^b\}$ , where  $q_i^b = (\mathcal{Q}_i, t_j^b)$ ;  $\mathcal{Q}_i$  is a natural language question and  $t_j^b \in T^b$  is a Freebase triple that supplies the answer to that question.

For example, the question “Who wrote The New Canada?” has the following answer triple:

(fb:m/02qtvzv, fb:book/written\_work/author, fb:m/01hxz2)

where fb stands for the prefix <http://www.freebase.com/>. The subject of the above answer triple is referred to as the topic entity, and the object of the triple is referred to as the answer entity. To answer the natural language question, a system must correctly identify the topic entity and the predicate, and then consult the knowledge graph to look up the answer entity.

Given Freebase  $T^b$ , DBpedia  $T^d$ , and SIMPLEQUESTIONS  $Q^b$ , our problem can be formally defined as follows: for each  $q_i^b = (\mathcal{Q}_i, t_j^b) \in Q^b$ , find  $q_i^d = (\mathcal{Q}_i, t_k^d)$ , where  $t_k^d \in T^d$  is a DBpedia triple, such that  $t_j^b$  is semantically equivalent to  $t_k^d$ . The result  $Q^d = \{q_1^d, \dots, q_l^d\}$  is our SIMPLEDBPEDIAQA dataset. Although this characterizes the basic structure of the problem, there are a number of nuances that deviate from this formalism, which we describe in the following sections.

## 4 Dataset Migration

Our overall strategy for dataset migration breaks down into the following steps: entity mapping, predicate mapping, and candidate refinement. At a high level, we begin by first mapping the topic and answer entities from Freebase to DBpedia; these then serve as anchors from which we can project the Freebase predicates to DBpedia. To assist in the process, we ingest the knowledge graphs into an RDF store to facilitate querying via SPARQL. For this effort, we use the latest version of DBpedia released in 2017.<sup>3</sup>

### 4.1 Entity Mapping

The first step is to map Freebase entities from SIMPLEQUESTIONS to entities in DBpedia. Freebase MIDs and DBpedia URIs are linked through the predicate <http://www.w3.org/2002/07/owl#sameAs>; these official mappings are released as part of DBpedia.<sup>4</sup> For each Freebase entity MID (topic entity or answer entity), we issue a SPARQL query to retrieve the corresponding DBpedia URI. For example, Justin Trudeau, the current Prime Minister of Canada, is mapped via the triple:

(dbr:Justin\_Trudeau, <http://www.w3.org/2002/07/owl#sameAs>, fb:m/02b5jh).

Here and throughout the paper we use dbr as the DBpedia prefix for <http://dbpedia.org/resource/>. For approximately 56% of questions in SIMPLEQUESTIONS, we can map both the topic entity and the answer entity from Freebase to DBpedia. For the remaining questions, we are only able to map the topic entity, the answer entity, or neither. The detailed breakdowns are shown in Table 2.

<sup>3</sup><https://wiki.dbpedia.org/develop/datasets/dbpedia-version-2016-10>

<sup>4</sup>[http://downloads.dbpedia.org/2016-10/core-i18n/en/freebase\\_links\\_en.ttl.bz2](http://downloads.dbpedia.org/2016-10/core-i18n/en/freebase_links_en.ttl.bz2)

Note that the URI for Justin Trudeau can be used to uniquely identify this entity within the broader open linked data ecosystem. For example, a human-readable version of facts associated with this individual is located at [http://dbpedia.org/page/Justin\\_Trudeau](http://dbpedia.org/page/Justin_Trudeau). This, as well as a variety of other libraries, toolkits, APIs, etc. provide infrastructure that simplifies the development of operational question answering systems. The existence of these resources illustrates one of the major benefits of migrating SIMPLEQUESTIONS over to a knowledge graph that is actively maintained by a dedicated community.

## 4.2 Predicate Mapping: One-Hop Predicates

Let us consider the case where we are able to map both the topic entity and the answer entity from Freebase to DBpedia. We can then issue a SPARQL query over DBpedia to enumerate the paths (sequence of one or more predicates) connecting those entities. In the simplest case, there is a single predicate connecting the topic entity to the answer entity, which yields a straightforward mapping of the triple from Freebase to DBpedia. This occurs for approximately half of the questions with successfully mapped topic and answer entities; see detailed statistics in Table 2.

Consider the question “Which city is McCormick Field in?” The Freebase topic entity `fb:m/05_xgn` is mapped to DBpedia as `dbr:McCormick_Field` and the answer entity is mapped from `fb:m/0ydpd` to `dbr:Asheville,_North_Carolina`. The DBpedia predicate `dbo:location` connects those two entities, which provides a valid and correct mapping for the Freebase predicate `fb:location/location/containedby`. Here and throughout the paper we use `dbo` as the DBpedia prefix for <http://dbpedia.org/ontology/>.

Due to differences in the conceptual organization of the two knowledge graphs, the directionality of equivalent predicates in Freebase and DBpedia may differ. For example, the DBpedia predicate `dbo:birthPlace` takes a person as the subject and a location as the object, whereas the equivalent predicate in Freebase `fb:location/location/people_born_here` inverts the subject and object. Therefore, for a question such as “Who was born in Aguascalientes?”, the subject in the Freebase triple becomes the object in the DBpedia triple.

During the migration from Freebase to DBpedia, if the directionality of the mapped predicate is the same, we refer to the result as a forward predicate; if the directionality is reversed, we refer to the result as a backward predicate. We explicitly keep track of this metadata, which is necessary for the actual question answering task.

## 4.3 Predicate Mapping: Two-Hop Predicates

Next, we consider the more complex case where the topic entity and the answer entity are *not* directly connected by a single predicate in DBpedia. That is, the results of our SPARQL query over DBpedia to enumerate the paths connecting the mapped entities contain multiple hops. In this work, we only consider two-hop traversals, as even longer paths are generally rare and spurious. These two-hop predicates can be categorized into *disambiguation predicates*, *redirection predicates*, *complex predicates*, and *missing predicates*, detailed as follows:

- *Disambiguation Predicates*: DBpedia uses `wikiPageDisambiguates` predicates to disambiguate different entities with the same name. The DBpedia `sameAs` links, however, might map a Freebase MID to an ambiguous URI, thus yielding a two-hop traversal from the topic entity to the answer entity. In these cases, we can “compress” the path back into a single predicate by changing the original topic entity to the disambiguated entity. Note that this disambiguation process can occur with forward predicates, as in Figure 1a, where `dbr:Jack_Carr` is disambiguated to `dbr:Jack_Carr_(footballer,_born.1878)`, as well as backward predicates, as in Figure 1b, where `dbr:QBS` is disambiguated to `dbr:QBS_(band)`.
- *Redirections Predicates*: Similar to disambiguation links, DBpedia uses `wikiPageRedirects` predicates to redirect an entity to another entity (typically, the canonical variant of that entity). As with disambiguation predicates above, these two-hop redirection predicates can also be compressed into a single triple. For example, `dbr:Douglas_Hofstadter` is redirected back to `dbr:Douglas.R.Hofstadter` and `dbr:Midfielder` is redirected back to `dbr:Defensive_Midfielder`, as shown in Figure 2a and Figure 2b, respectively. Once again, this can occur with both forward and backward predicates.

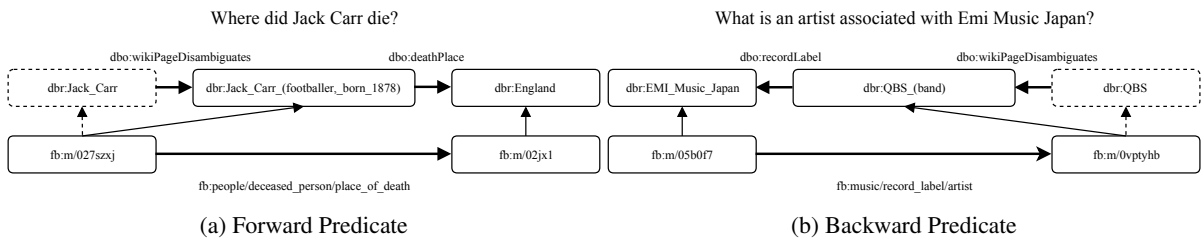


Figure 1: Examples of mapping disambiguation predicates from Freebase to DBpedia.

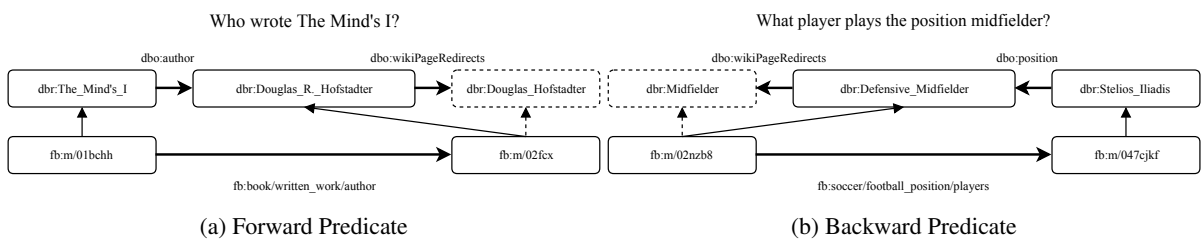


Figure 2: Examples of mapping redirection predicates from Freebase to DBpedia.

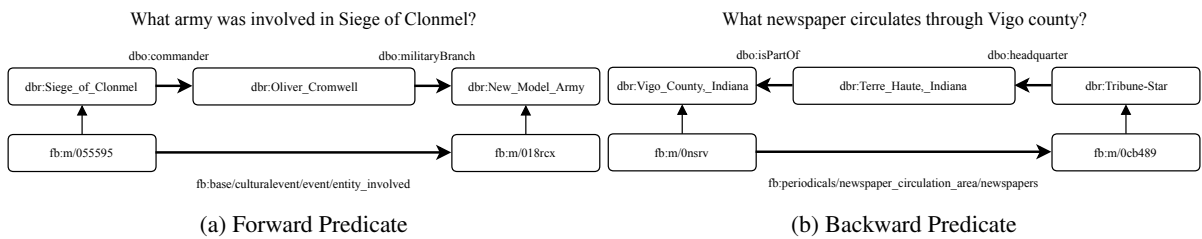


Figure 3: Examples of mapping complex predicates from Freebase to DBpedia.

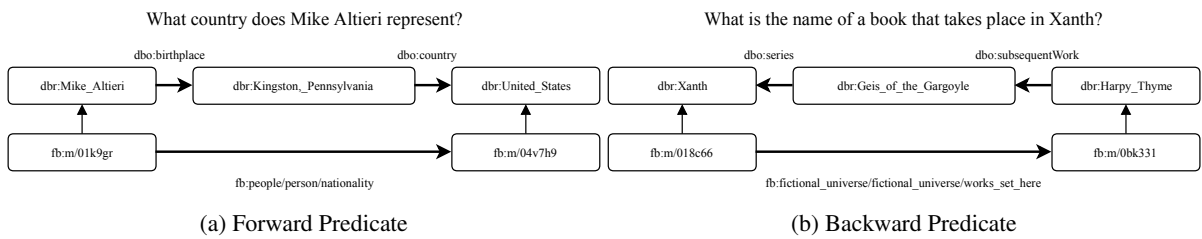


Figure 4: Examples of missing predicates in DBpedia.

- *Complex Predicates*: Due to differences in the conceptual organization of Freebase and DBpedia, there is no direct equivalent in DBpedia for some Freebase predicates. Instead, a chain of two predicates is necessary to capture the relationship between the topic and answer entities. An example is shown in Figure 3a: the question “What army was involved in Siege of Clonmel?” can be answered using the Freebase predicate `fb:base/culturalevent/event/entity_involved`, but in DBpedia the same fact requires a chain of two predicates, `dbo:commander` and `dbo:militaryBranch`. Note that this can also occur with backward predicates, as shown in Figure 3b.
- *Missing Predicates*: Some questions in DBpedia are answered using two-hop predicates even though there exists a one-hop predicate in the knowledge graph that represents a better mapping; this situation arises due to the incompleteness of DBpedia. Note that missing predicates actually represent a special case of complex predicates, which we only discovered by manual examination of the predicate mapping results. Nevertheless, it seems appropriate to separately categorize this particular type of predicate mismatch between Freebase and DBpedia. An example is shown in Figure 4a: The entity `dbr:Mike_Altieri` should have a predicate `dbo:nationality` that directly links to `dbr:United_States`, as is typical of person entities. However, since this predicate is missing, our SPARQL query discovered a roundabout path via `dbo:birthplace` then `dbo:country`. Figure 4b shows a similar case involving a backward predicate, where the entity `dbr:Harpy_Thyme` should have a predicate `dbo:series` that directly links to `dbr:Xanth`; instead, the answer entity is discovered via the extra hop `dbo:subsequentWork`. We believe that DBpedia can be enhanced by inserting these missing links, but augmenting DBpedia is beyond the scope of this work.

Detailed statistics of these two-hop predicate matches are shown in Table 2. As described above, there is no automatic way to differentiate between complex and missing predicates, and thus we provide the sum of the two categories. For questions that have both topic and answer entity mappings, we are not able to find any predicate mappings for approximately 34% of them.

#### 4.4 Candidates Refinement

The output of the initial entity and predicate mapping process (as described above) is then refined to produce the final SIMPLEDBPEDIAQA dataset; detailed statistics are shown in Table 3. In this section, we detail the candidate refinement process.

The need for post-processing candidate results from the output of the processes described above is apparent from manual examination. While the entity mappings are generally of high quality, some of the mapped predicates are invalid, primarily due to two reasons:

- *Semantic drift*: Some candidate predicates are not semantically correct even though the answer may be factually correct. For example, consider the question “From where does Anjali Devi claim nationality?” The predicate mapping produces `dbo:deathPlace` instead of the correct predicate, `dbo:nationality`. This is because the correct predicate is missing for this entity, and by coincidence, this person’s nationality is the same as her death place.
- *Predicate constraints*: In some cases, we observe mismatches between the domains of the subjects or objects of a Freebase predicate and its corresponding DBpedia predicate. For example, the DBpedia predicate `dbo:author` can take as subject books, movies, etc. However, the Freebase predicate `fb:book/author/works_written` can only be mapped to the DBpedia predicate `dbo:author` (in the backward direction) if the DBpedia subject has the type `dbo:WrittenWork`. More generally, a predicate mapping is valid only under certain type constraints.

To tackle these challenges with minimal manual effort, we construct manual rules that map high-frequency Freebase predicates in the initial mappings to all potentially correct (at the semantic level) DBpedia predicates. Each rule includes a Freebase predicate and a list of corresponding DBpedia predicates, an associated directionality (forward or backward), and an optional type constraint. A few examples are shown in Table 4. The interannotator agreement of these rules based on three human annotators

				Training	Validation	Test	Total
		One Hop		22,158	3,193	6,304	<b>31,655</b>
Mapped Entities	Mapped Predicates	Two Hop	Disambiguation	173	24	59	<b>256</b>
			Redirection	1,992	247	553	<b>2,792</b>
			Complex + Missing	3,504	501	985	<b>4,990</b>
		Not Mapped Predicates		14,875	2,059	4,373	<b>21,307</b>
<b>Sub-Total</b>				<b>42,702</b>	<b>6,024</b>	<b>12,274</b>	<b>61,000</b>
Not Mapped Entities	Only Answer Entity Mapped			18,040	2,635	5,174	<b>25,849</b>
	Only Topic Entity Mapped			9,611	1,415	2,648	<b>13,674</b>
	Both Not Mapped			5,557	771	1,591	<b>7,919</b>
<b>Sub-Total</b>				<b>33,208</b>	<b>4,821</b>	<b>9,413</b>	<b>47,442</b>
<b>SIMPLEQUESTIONS Total</b>				<b>75,910</b>	<b>10,845</b>	<b>21,687</b>	<b>108,442</b>

Table 2: Statistics from the initial mapping of entities and predicates in SIMPLEQUESTIONS.

				Training	Validation	Test	Total
		One Hop		19,271	2,773	5,467	<b>27,511</b>
Mapped Entities	Mapped Predicates	Two Hop	Disambiguation	84	17	32	<b>133</b>
			Redirection	1,547	191	429	<b>2,167</b>
			Complex + Missing	1,365	191	377	<b>1,933</b>
		Not Mapped Predicates		3,940	531	1,183	<b>5,654</b>
<b>Sub-Total</b>				<b>26,207</b>	<b>3,703</b>	<b>7,488</b>	<b>37,398</b>
Not Mapped Entities	Only Answer Entity Mapped			0	0	0	<b>0</b>
	Only Topic Entity Mapped			3,979	602	1,107	<b>5,688</b>
	Both Not Mapped			0	0	0	<b>0</b>
<b>Sub-Total</b>				<b>3,979</b>	<b>602</b>	<b>1,107</b>	<b>5,688</b>
<b>SIMPLEDBPEDIAQA Total</b>				<b>30,186</b>	<b>4,305</b>	<b>8,595</b>	<b>43,086</b>

Table 3: Final statistics of SIMPLEDBPEDIAQA following candidates refinement.

is 97%, where agreement is computed as the number of predicates that were identically labeled by all the annotators, divided by the count of all predicates.

Using these mapping rules, we can filter and discard spurious one-hop mappings (including the disambiguation and redirection cases) where both the topic and answer entities are correctly mapped. Furthermore, we can expand the dataset by applying these rules to a few additional cases. Consider the case of complex and missing predicate: since these questions have two-hop predicates, making them no longer “simple questions”, they would have been discarded from our dataset. However, we can issue a SPARQL query using the topic entity and the mapped DBpedia predicates from our mapping rules to search for valid answers (ignoring the answer entity). If the query returns a result, we can add the question to our dataset. Heuristically, this means that the question *does* have an answer in DBpedia, just not the same as the one provided in Freebase.

The same process can be applied to cases where we have successfully mapped the entities but not the predicates, and even to cases where we have only successfully mapped the topic entity. As a concrete example, for the question “What is a song by John Rutter?”, only the topic entity is mapped. Based on our rules, the Freebase predicate fb:music/artist/track is mapped to the DBpedia predicate dbo:artist with a constraint of dbo:MusicalWork in the backward direction. Using the topic entity as an anchor, a SPARQL query returns a valid result.

Detailed statistics from the refinement process are shown in Table 3. The final output of entity mapping, predicate mapping, and candidate refinement is our SIMPLEDBPEDIAQA dataset, which successfully migrates SIMPLEQUESTIONS from Freebase over to DBpedia.

Freebase Predicate	DBpedia Predicate	Directionality	Type Constraint
fb:architecture/structure/architect	dbo:architect	forward	-
fb:location/location/contains	dbo:country	backward	-
fb:baseball/baseball_position/players	dbo:position	backward	dbo:BaseballPlayer
fb:music/album_release_type/albums	dbo:type	backward	dbo:Album
fb:book/author/works_written	dbo:author	backward	dbo:WrittenWork

Table 4: Examples of predicate mapping rules.

## 5 Question Answering Baseline

To lay the foundation for future work on our new dataset, we provide simple yet strong baselines using recent work by Mohammed et al. (2018), who applied techniques with and without neural networks to SIMPLEQUESTIONS. In this paper, we used their open-source code<sup>5</sup> to generate the experimental results reported here. We briefly describe their approach, which decomposes into four tasks:

- **Entity Detection:** Given a question, the task is to identify the topic entity of the question. For this task, we examined bidirectional LSTMs and Conditional Random Fields (CRFs).
- **Entity Linking:** Detected entities (text strings) need to be linked to entities in the knowledge graph (e.g., URI from DBpedia in our case). This is formulated as a string matching problem: Levenshtein Distance is used along with a few heuristics for ranking candidate entities.
- **Predicate Prediction:** Given a question, the task is to identify the predicate being queried. We examined three models: bidirectional GRU, convolutional neural network (CNN), and logistic regression (LR). The first two are standard neural network models; for logistic regression we used as input the average of the word embeddings of each word. BiGRU was selected over BiLSTM based on the experiments of Mohammed et al. (2018), where it was found to be slightly more accurate.
- **Evidence Integration:** With  $m$  candidate entities and  $r$  candidate predicates from the previous components, the evidence integration model selects the best (entity, predicate) pair based on the product of each component score as well as a number of heuristics.

One additional detail is necessary to understand our experimental methodology for entity detection. In SIMPLEQUESTIONS, the topic entity is not explicitly tagged in the natural language question at the token level; as a result, SIMPLEDBPEDIAQA does not have token-level annotations either. This presents a problem, as our formulation of entity detection as sequence labeling requires per-token labels. The solution adopted by Mohammed et al. (2018) with SIMPLEQUESTIONS was to “back-project” the entities onto the natural language questions to automatically derive token labels, either ENTITY or NOTENTITY. We performed exactly the same back-projection in this work. If the entity text can be matched exactly in the question, the corresponding tokens are tagged appropriately. If there is no exact match,  $n$ -grams are generated from the question (from length one up to the length of the question) and the Levenshtein Distances between these  $n$ -grams and the entity text are computed. The  $n$ -gram with the highest score is selected and the corresponding tokens are tagged appropriately. We find that 94.1% of questions have exact matches with entity strings.

## 6 Experiment Results and Error Analysis

We evaluated the quality of our models in the same way as Mohammed et al. (2018): For entity detection, we compute F1 in terms of the entity labels. For both entity linking and predicate prediction, we evaluate recall at  $N$  ( $R@N$ ). For the final end-to-end evaluation, we use accuracy (or equivalently,  $R@1$ ). For evidence integration, our model considers 20 entity candidates and 5 predicate candidates. All hyperparameters and other settings follow the original paper; we have not specifically fine-tuned parameters for this dataset.

<sup>5</sup><http://buboqa.io/>



Entity Linking	R@1	R@5
BiLSTM	78.0	88.2
CRF	75.1	85.4
Predicate Prediction	R@1	R@5
CNN	89.2	99.1
BiGRU	88.1	99.0
LR	84.2	97.6

(a) Component accuracy on validation set.

Entity Detection	Relation Prediction	Accuracy
BiLSTM	CNN	78.5
BiLSTM	BiGRU	78.2
BiLSTM	LR	75.8
CRF	CNN	76.1
CRF	BiGRU	76.0
CRF	LR	73.5

(b) End-to-end accuracy on test set.

Figure 5: Experiment results applying the models of Mohammed et al. (2018) to SIMPLEDBPEDIAQA.

Error Type	# Errors	Prevalence
Hard ambiguity	42	21.0%
Soft ambiguity	21	10.5%
Entity detection error	19	9.5%
Predicate prediction error	28	14.0%
Error in both	90	45.0%
<b>Total</b>	200	100.0%

Figure 6: Results of error analysis.

For entity detection, on the validation set, the BiLSTM achieves 90.3 F1, compared to the CRF at 88.1. The top of Table 5a shows the entity linking results for the BiLSTM and the CRF. These results are consistent with the findings of Mohammed et al. (2018): the BiLSTM achieves a higher F1 score than the CRF, which translates into higher recall in entity linking (both R@1 and R@5). Predicate prediction results are shown on the bottom of Table 5a: the CNN slightly outperforms the BiGRU on R@1, but in terms of R@5 the accuracy of both are quite similar. The neural network models appear to be more effective than logistic regression.

Finally, Table 5b shows end-to-end accuracy on the test set. The best model combination uses the BiLSTM for entity detection and the CNN for predicate prediction, achieving 78.5% accuracy. By swapping the BiLSTM with the CRF for entity detection, we observe a 2.4% absolute decrease in end-to-end accuracy. Results from other combinations are also shown in Table 5b. Note that using the CRF for entity detection and logistic regression (LR) for predicate prediction, which is a baseline that does not use neural networks (with the exception of word embeddings), is also reasonably accurate. This finding is also consistent with Mohammed et al. (2018), who advocate that NLP researchers examine baselines that do not involve neural networks as a sort of “sanity check”.

Following Lukovnikov et al. (2017), we sampled 200 examples of errors on the test set from the BiLSTM + CNN model to analyze their causes. We manually classified them into the following categories, summarized in Table 6 and described below:

- *Hard ambiguity*: The context provided by the question is insufficient, even for a human, to disambiguate between two or more entities with the same name. In these cases, our model correctly identified the entity string, but linked it to an incorrect entity in the knowledge graph. For example, in the question “What is the place of birth of Sam Edwards?”, it is unclear if Sam Edwards refers to the actor `dbr:Sam_Edwards` or the physicist `dbr:Sam_Edwards_(physicist)`.
- *Soft ambiguity*: The context provided by the question is sufficient to disambiguate the entity, but our model fails to identify the correct entity in the knowledge graph. In these cases, our model correctly identified the entity string, so the error is isolated to the entity linking component. For example, in the question “What kind of show is All In?”, the model predicted `dbr:All_In_(song)` instead of `dbr:All_In_(TV_series)` (the correct entity). Note that in this case, it is clear to a human based on context that the question refers to a show and not a song.

- *Entity detection error*: The extracted entity string is incorrect.
- *Predicate prediction error*: The predicted predicate is incorrect.
- *Error in both*: Both the extracted entity and the predicted predicate are incorrect.

From the above analysis, we find that there is still substantial room to improve on the effectiveness of our baselines. However, these results also suggest that there is an upper bound on accuracy that lies substantially below 100%, as the cases of hard ambiguity are difficult to resolve, even for humans. In those cases, correct entity linking is more a matter of luck and other idiosyncratic characteristics of the dataset rather than signals that can be reliably extracted to understand the true question intent.

## 7 Conclusion

This paper presents SIMPLEDBPEDIAQA, a new benchmark dataset for simple question answering over knowledge graphs created by migrating the SIMPLEQUESTIONS dataset from Freebase to DBpedia. Although this mapping process is conceptually straightforward, there are a number of nuances and complexities we had to overcome with a combination of special-case handling and heuristics. The result is a dataset targeting a knowledge graph that is actively maintained by a dedicated community. We hope that our efforts better connect existing research communities, in particular, NLP researchers with the open linked data community, and spur additional work in question answering over knowledge graphs.

## Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

## References

- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question–answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1533–1544, Seattle, Washington.
- Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia — A crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165.
- Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1249, Vancouver, British Columbia, Canada.
- Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv:1506.02075v1*.
- Qingqing Cai and Alexander Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 423–433, Sofia, Bulgaria.
- Zihang Dai, Lei Li, and Wei Xu. 2016. CFO: Conditional focused neural question answering with large-scale knowledge bases. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 800–810, Berlin, Germany.
- Dennis Diefenbach, Thomas Pellissier Tanon, Kamal Deep Singh, and Pierre Maret. 2017. Question answering benchmarks for Wikidata. In *Proceedings of the 2017 International Semantic Web Conference (Posters, Demos & Industry Tracks)*.
- Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1211–1220, Perth, Australia.

- Salman Mohammed, Peng Shi, and Jimmy Lin. 2018. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 291–296, New Orleans, Louisiana.
- Kuldeep Singh, Arun Sethupat Radhakrishna, Andreas Both, Saeedeh Shekarpour, Ioanna Lytra, Ricardo Usbeck, Akhilesh Vyas, Akmal Khikmatullaev, Dharmen Punjani, Christoph Lange, Maria Esther Vidal, Jens Lehmann, and Sören Auer. 2018. Why reinvent the wheel: Let’s build question answering systems together. In *Proceedings of the 2018 World Wide Web Conference*, pages 1247–1256, Lyon, France.
- Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. LC-QuAD: A corpus for complex question answering over knowledge graphs. In *Proceedings of the 16th International Semantic Web Conference*, pages 210–218, Vienna, Austria.
- Ferhan Ture and Oliver Jojic. 2017. No need to pay attention: Simple recurrent neural networks work! (for answering “simple” questions). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 2856–2862, Copenhagen, Denmark.
- Xuchen Yao. 2015. Lean question answering over Freebase from scratch. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations (NAACL/HLT 2015)*, pages 66–70, Denver, Colorado.