QA Is the New KR: Question-Answer Pairs as Knowledge Bases

William W. Cohen, Wenhu Chen, Michiel De Jong, Nitish Gupta, Alessandro Presta, Pat Verga, John Wieting*

Google AI
{wcohen,wenhuchen,msdejong,guptanitish,apresta,patverga,jwieting}@google.com

Abstract

We propose a new knowledge representation (KR) based on knowledge bases (KBs) derived from text, based on question generation and entity linking. We argue that the proposed type of KB has many of the key advantages of a traditional symbolic KB: in particular, it consists of small modular components, which can be combined *compositionally* to answer complex queries, including relational queries and queries involving "multi-hop" inferences. However, unlike a traditional KB, this information store is well-aligned with common user information needs. We present one such KB, called a QEDB, and give qualitative evidence that the atomic components are high-quality and meaningful, and that atomic components can be combined in ways similar to the triples in a symbolic KB. We also show experimentally that questions reflective of typical user questions are more easily answered with a QEDB than a symbolic KB.

Introduction: QA Pairs as a KB

Since the very beginnings of Artificial Intelligence (AI) research, symbolic knowledge bases (KBs) have been used for knowledge-intensive tasks. There are many kinds of KBs, but they are based on one central principle: information is stored in small modular components (e.g., entities, KG triples, WikiData statements) that can be combined compositionally to answer complex queries.

Broad-coverage KBs like FreeBase (Bollacker et al. 2008) and WikiData (Vrandečić and Krötzsch 2014) continue to be widely used in AI and Natural Language Processing (NLP). However, many of the facts in these KBs are of little interest to most users, and conversely, there are many facts of interest that are either not represented in the KB, or are difficult to extract from the KB without complex queries. This is a natural consequence of how broad-coverage KBs are created: historically, most broad-coverage KBs were populated in a "information-driven" way, by first identifying available information sources, and then integrating these sources such that they could be jointly queried. In this position paper, we argue that *the availability of large question-answering (QA) datasets now enables a "user-driven" KB development process*, and discuss the implications of this statement. Concretely, we propose that training data for extractive QA systems can be be used to produce a KB-like structure that we call a Question-answer Explanation Database (QEDB). By construction, the QEDB is *unlikely* to contain facts that are true but irrelevant to user's needs because nobody will ask about them. However, a QEDB is not simply a collection of questions and answers: we will demonstrate that elements of QEDB can be combined compositionally with a query language, just like as knowledge graph (KG) triples. We argue that such a KB is better-matched to users' information needs, and support this claim with experimental evidence.

Our approach to generating a KB from raw text exploits recent progress in question generation (QG) (Yang et al. 2017; Lewis et al. 2021) and explainable question-answering (Lamm et al. 2021), and is illustrated in Figure 1.

(1) Given a document d, a large number of question-answer pairs (q, a) are generated, where a is a span from d. In Figure 1, the questions were generated with a model trained on a subset of the NQ dataset (Kwiatkowski et al. 2019). (2) For each pair (q, a), we align spans r_q in the question with *referentially equivalent* spans r_d in the document—e.g., the two light blue spans "*the tv series tipping the velvet*" in q and "Tipping the Velvet" in d are aligned. (3) A graph is formed by constructing a node for each answer span a and each question-aligned reference span r_d in the document. Nodes associated with the same question q are then connected by an edge that is labeled with a "relation" associated with q's surface form. For instance, the question "*what is the tv series tipping the velvet based on*" is used to produce the edge label q_l ="what is \$1 based on". These triples are analogous to a $(subject, relation, object)$ triple in a KB. An entity linker is run on the corpus, and referential spans are linked to contained entities. Figure 1 shows a larger QEDB derived from information in three documents, with dashed lines indicating passage references mentioning a common entity. (For compactness, the generated questions are shown next to the document nodes to which they align.)

The QEDB is strongly influenced by the initial sample of questions; grounded in the corpus; relational, containing an open set of relations; and (since it contains traditional KB entities) it can be easily combined with a symbolic KB over the same entities.

*Authors listed alphabetically by last name.

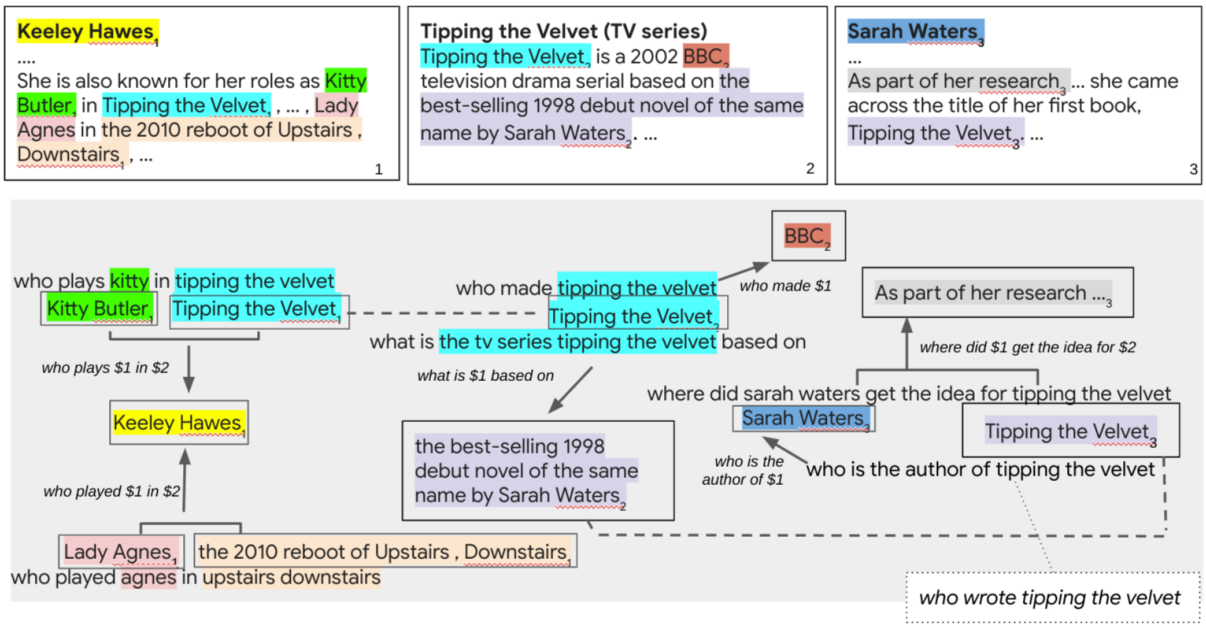

Figure 1: A QEDB derived from three documents. Generated questions are placed near the document spans aligned with them, and the subscript on a document span indicates its containing document. Dashed lines (and shared colors) indicate cross-document co-reference, and dotted lines indicate semantically equivalent questions. Note that the extractions from documents 1 and 3 include ternary relations.

Notice that the formalism naturally models relationships with threee or more arguments, and that the edges in the graph arise from both entity linking, and because multiple questions can contain the same span. The graph edges make it possible to answer complex questions not explicitly in the graph, like "where can i watch that tv series based on a 1998 novel by sarah waters".

Like open IE (Etzioni et al. 2008; Mausam 2016), producing a QEDB does not require a fixed vocabulary of relations, but unlike open IE, a QEDB is driven by an initial GQ stage, which ensures the data is aligned to user's information needs. As a concrete example, we note that the first question in the NQ dev test is *who got the first nobel prize in physics?* , which would lead to a complex KB query, while a QEDB based the document that contains the answer (i.e., "*The first Nobel Prize in Physics was awarded in 1901 to Wilhelm Conrad Röntgen . . .* ") would contain the edge

The first Nobel Prize in Physics $\xrightarrow{\text{who got \$1}}$ Wilhelm Conrad Röntgen

Methods

The experiments of this paper are conducted with a particular QEDB. Below we outline how it was constructed, and the computational methods used.

The questions in the QEDB are derived from the Probably-Asked Questions (PAQ) dataset introduced by Lewis et al. (2021). PAQ contains 64.9 million questions generated from Wikipedia passages. To generate PAQ a four-stage pipeline was used, where passages were selected; potential answers were extracted from each passage, using named entity recognition and neural models trained on NQ short answers; questions were generated conditioned on each answer span and the containing passage; and questions were heuristically filtered for quality. For more details consult (Lewis et al. 2021).

	Passage Reference		Question Reference	
	EM	F1	EM	F1
T5-XXL	67.7	64.9	75.7	82.2
T5-Large	56.5	54.5	72.7	79.7

Table 1: Results on QED dev set for identifying passage and question references.

To align the question references with passage references, we trained a T5 model fine-tuned using annotations for a subset of NQ (Lamm et al. 2021) which were originally used for explainable question-answering. The performance of this model is shown in Table 1.

An off-the-shelf entity linker was run on all the *passages* (not questions) and the entity-linked passage references were then matched heuristically[1] to question references. Entity-linking passages and then aligning linked passages with question is important: preliminary experiments showed that linking entities in the questions directly was very inaccurate, since questions have limited contextual information.

Following Lewis et al. (2021), we implemented a method

[1] An entity e was associated with a question reference r_q if (1) e was linked to the passage p from which the question was generated and (2) no other entity e' linked from p is more similar to r_q than e, using Jaccard similarity on tokens.

that answers user questions q' by retrieving similar question-answer pairs $(q_1, a_1), \ldots, (q_k, a_k)$, concatenating them with the original question q', and then using a Transformer to fuse this information and generate a final combined answer a'. The model we used, called QAMAT (for Question-Answer Memory Augmented Transformer), is described in detail elsewhere (Chen et al. 2022).

For comparison purposes, we also implemented a variation of QAMAT which replaces the memory question-answer pairs with a memory of Wikidata facts, which we will call FAMAT. Pretraining QAMAT/FAMAT requires a corpus of documents that have been heuristically aligned with the items to be retrieved, and for FAMAT we adopted the pre-training corpus used for FILM, the Fact Injected Language Model (Verga et al. 2021). Besides FAMAT, we also consider a strong baselines for QA using a traditional KB, FILM (Verga et al. 2021).

Experiments and Observations

The central claims made here are the following:

- (P1) Like a traditional symbolic KB, *a QEDB is composed of modular components that can be combined in compositionally in many different ways.*
- (P2) *The information in a QEDB will be more useful for answering user information needs* than the information in a traditionally constructed broad-coverage KB.

Quality of Atomic Elements of the QEDB

To visualize the types and quality of relationships in QEDB, we picked two entities e that one author was at least slightly familiar with, and for each entity e we found the top six entities e', ranked by the number of questions $q_{e,e'}$ that have an answer linked to e and a question reference linked to e' (with confidence at least 0.25). Table 2 gives the query entity e, the related entities e', and one associated question $q_{e',e}$ for each e'. The relationships are diverse and generally accurate. These two entities were selected to reflect the error rate of a larger sample[2].

Combining Elements of the QEDB

The primitive components stored in the QEDB are questions annotated with entity references. From these, we constructed more complex queries by combining together pairs of questions q_1, q_2 where the answer to q_1 appears as a question reference in q_2. For example for q_1="*who was the roman proponent of hedonism*" the answer is *Lucretius*, which appears in q_2="*what is the name of lucretius's book on atomism*". Joining these together yields a query like a HotPotQA "bridge question" (Yang et al. 2018): expressed in natural language, the combined query might be written *what is the name of the book on atomism written by a roman proponent of hedonism*.

Many hundreds of millions of plausible bridge questions can be constructed this way. We present a sample of them in Table 3, with the bridging entity's mention in q_2 replaced with a variable *$1*.[3] The sample was selected from question pairs that where (1) q_2 contains a single question reference (2) each question has a single answer (3) the reference alignments were made with high confidence (4) the bridging entity is not too common and (5) the answer to q_2 does not appear in q_1. Subject to these constraints the examples shown are not cherry-picked. For comparison, we also include five bridging questions from HotpotQA, manually converted to a similar format. The QEDB-produced queries are similar in quality to the HotpotQA queries, hence the QEDB also *implicitly contains answers to many plausible multi-hop information needs*, just as a symbolic KB does.

Even with all the constraints employed above (many of which could be easily relaxed), there are still more than 6 times as many QEDB-generated bridging questions as there are in HotpotQA.

Taken together, these results support claim P1: Like a traditional symbolic KB, a QEDB is composed of modular components that can be combined compositionally in many different ways. In particular, the individual relationships in the QEDB are meaningful and correct, and relationships can chained together to construct paths through the QEDB that are semantically similar to HotpotQA multi-hop questions.

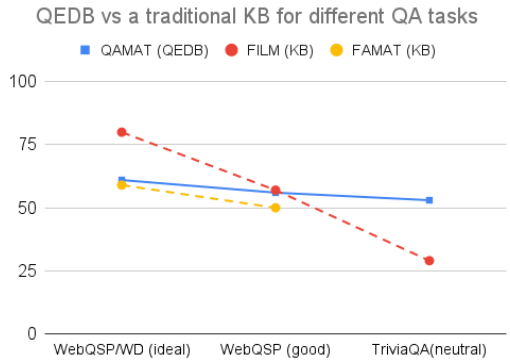

Figure 2: Dashed lines are systems using a KB, and the solid line for QAMAT, which uses a QEDB. The blue lines are associated with QAMAT and its KBQA variant FAMAT.

Utility of QA Pairs versus KB Triples

To argue for P1, we evaluate performance on three question-answering datasets, which vary in how well-aligned one would expect them to be to a symbolic KB.

TriviaQA (Joshi et al. 2017) is a widely-used question-answering dataset composed of questions authored by trivia enthusiasts. We select this as a QA dataset that is plausibly reflective of user information needs, while still being approachable for KBQA methods.[4] **WebQSP** is (Yih et al.

[2]In the larger sample 3 of 50 relationships were errors.

[3]This formalism loosely follows the question decomposition meaning representation (QDMR) language proposed in Wolfson et al. (2020). The lack of case, and the occasional disfluencies in the question, is similar to the NQ questions used to train the question generator for QEDB.

[4]Other QA datasets, notably NQ, are arguably more representa-

Query Entity	Related Entity	Linking Question
Jerry Garcia	Grateful Dead	*which member of the grateful dead died in 1995*
	Guitar	*who played guitar on three tracks by ornette coleman in 1988*
	Lead guitar	*who was the lead guitarist for the dead in europe in 1972*
	Pedal steel guitar	*who plays guitar on teach your children by graham nash*
	Today (Song)	*who plays the lead guitar on today by jefferson airplane*
	*Roseanne (TV show character)	*what's the name of roseanne's fourth child* *answer is Jerry Garcia Conner
Pittsburgh	Pennsylvania	*where is kraft heinz located in pennsylvania*
	United States	*where was the first legion of the united states raised*
	Pittsburgh Steelers	*where did the steelers play in the 1964 super bowl*
	University of Pittsburgh	*where does the university of pittsburgh bus service*
	Fort Duquesne	*which american city was originally called fort duquesne after it was captured by the*
	Monongahela River	*which american city was founded on the river monongahela in 1837*
NC University System	Kuklos Adelphon	*where did kappa alpha and kuklos adelphon meet*
	†Geoff Crompton	*where did geoff crompton play college basketball* †UNC/CH not the system
	Margaret Spellings	*margaret spellsings is the current president of which university*
	President	*margaret spellsings is the president of which university*
	†Bones McKinney	*where did bones mckinney play in college* †UNC/CH not the system
	North Carolina Central U	*what system did nccu join*
	Winston-Salem State Univ	*winston salem state university is a constituent institution of what university*
	UNC/Greensboro	*what system is the university of north carolina at greenboro part of*
	North Carolina State Uni	*what is nc state part of*
	East Carolina University	*who approved the east carolina university dental school*

Table 2: Pairs of entities considered "related" by the QEDB, and the questions that gave rise to those relationships. These pairs are obtained by finding the top k questions touching a query entity, and then finding all entities touching one of these questions. We discard related entities that are years. †Here entity-linking incorrectly links the UNC system to the flagship UNC/CH campus.

2016) is the Web Questions Semantic Parse dataset. We select this as a benchmark that is favorable to QA methods that use traditional broad-coverage KBs.[5] Finally, **WebQSP/WD** has questions from WebQSP answerable with WikiData statements involving a subset of "head" WikiData entities. Together with the appropriate WikiData subset, this is selected as an *ideal* case for QA using a KB.

Table 4 presents results on each of these datasets for QAMAT, which uses the QEDB, and FILM, which uses a KB. On WebQSP/WD, which we take to be the ideal case for QA using a KB, FILM performs much better than QAMAT. However, FILM performs quite similarly for the full WebQSP, which contains KB-oriented questions posed against an imperfect KB, and FILM performs much worse on TriviaQA, trailing QAMAT by more than 20 points. The last line of the table presents performance for the same model used in QAMAT, but populated with a memory of facts instead of question-answer pairs (FAMAT). In this comparison, *where only the information store is varied*, we see that FAMAT is actually slightly worse than QAMAT, even on the "ideal" and "good" cases of WebQSP/WD and WebQSP.

The results of this section thus support claim P2: The information in a QEDB is indeed more useful for answering typical user information needs than the information in a traditionally constructed broad-coverage KB, as witnessed by the 24 point improvement on TriviaQA. In fact, even on collections of questions like WebQSP that have been aggressively filtered to be KB-relevant, the QEDB is competitive with the traditional KB—only when one ensures a nearly perfect alignment between questions and KB (WebQSP/WD) does the traditional KB perform better.

tive of common user questions, but TriviaQA is much more favorable to KBQA approaches than NQ: in particular, around 85% of the answers in TriviaQA are KB entities (Févry et al. 2020), while only about 40% of the answers in NQ are.

[5]WebQSP is a subset of the WebQuestions dataset (Berant et al. 2013), containing entity-oriented questions answerable using FreeBase and a SPARQL query. Less than 10% of the original question pool satisfied these constrants.

Related Work

Extracting and Querying Structured KBs

Large-scale, broad-coverage KBs such as DBPedia (Auer et al. 2007), FreeBase (Bollacker et al. 2008), YAGO (Rebele et al. 2016), NELL (Mitchell et al. 2018), and WikiData (Vrandečić and Krötzsch 2014) make extensive use of semi-structured sources for data, and little data about user needs existed at the time of their design. The contents of the KGs were thus heavily influenced by what information was most accessible—which in turn was influenced by the contents of existing structured databases. NLP research in answering questions using KBs (Berant et al. 2013; Yih et al. 2016; Dubey et al. 2019), or KBQA largely ignores the fact that actual user questions are generally not answerable by KBQA: in one study (Balachandran et al. 2021), less than 7,000 of the 307,000 training questions in the NQ dataset were answerable using WikiData. In contrast, QEDBs are designed to use datasets of representative user questions in determining what information resides in a KB. Although QEDBs will contain different content than existing symbolic KBs, we note that they are very compatible: in fact, it is straightfor-

Question 1	Bridge Entity	Question 2	Answer
where did the cincinnati reds last game	Joe's North ...	*what nfl team used to play at $1*	Cincinnati Bengals
who sings the song please leave the grates	Jebediah	*when was $1 formed and by whom*	1994
who is the main character of the tombs of atuan	Tenar	*who raised $1 in wizard of earthsea lore*	Aihal
*what is the sequel to wild fire by nelson de mille**	Night Fall	*when did the plane crash in $1*	1996
who directed the opening act w	Steve Byrne	*$1 is the lead actor in which us tv series*	Sullivan & Son
who was the roman proponent of hedonism	Lucretius	*what is the name of $1's book ...*	On the Nature of Things
what is the main export of tutuila	Canned fish	*what is the name of the process used to preserve $1*	canning
who replaced randy jackson ... for this season	Scott Borchetta	*who did $1 try to recruit to his record label*	Taylor Swift
where is laurence harbor located in new jersey	Raritan Bay	*what do you do in $1*	recreational fishing
*which novel features ... puttenham, surrey**	Brave N. World	*how did george orwell describe $1*	negative utopia
who was the creator of "the rocky horror show"	Richard O'Brien	*when was $1 born*	1942
which nickelodeon show was bill long credited in	Blues Clues	*who originally hosted $1*	Steve Burns
what college was ... balaton educated at after eton	New College	*when was $1 founded*	1379
which actor composed the song "smile" ...	Charlie Chaplin	*what nationality is $1*	English
what casino is formerly known as vegas world	Stratosphere Las Vegas	*what company is headquartered at $1*	American Casino & Entertainment

Table 3: Above the line, the result of joining question pairs, where the answer of question 1 is a question reference in question 2. Below the line, five bridge questions from HotpotQA, manually converted to the same format. Questions marked with an asterisk have potentially erroneous bridging entities (e.g., linking the novel "Brave New World" with a game of the same name)
.

	WebQSP/WD *(ideal for KBQA)*		WebQSP *(good)*		TriviaQA *(neutral)*	
models using a QEDB						
QAMAT (ours)	61		56		**53**	
models using a traditional KB						
FILM	**80**	(+19)	**57**	(+1)	29	(-24)
FAMAT (ours)	59	(-2)	50	(-6)		

Table 4: Exact match performance of QAMAT, a system using a QEDB as a KB, against two strong baselines designed to answer questions using a traditional symbolic KB.

ward to cross-link a QEDB with an existing KB, by simply running an entity linker on the underlying corpus. , so QEDBs are best thought of as a complement to existing KBs than a replacement.

Open information systems (Etzioni et al. 2008; Mausam 2016) produce KBs syntactically similar to QEDB, in that both are graphs of textual spans. However, for a QEDB, the contents of the graph are not determined by the intuitions of the system designer but by concrete data about likely user queries.

RePAQ, QAMAT, and Other Uses of QA Data

The Probably Asked Questions (PAQ) resource and the accompanying RePAQ QA system (Lewis et al. 2021), as well as the closely-related QAMAT system (Chen et al. 2022), heavily influenced this work. PAQ is a resource containing 65 million question-answer pairs, and RePAQ is a QA system that answers a user question q by finding a matching questions q' in PAQ. QAMAT is a similar system that outperforms RePAQ, and can also answer multi-hop questions. As noted above we made use of the PAQ resource in producing QEDB. Unlike PAQ, which is a collection of question-answer pairs, a QEDB is structured as a KB, with free-text analogs of entities and relations. As a KB, QEDB includes relational information, making it possible to answer more complex multi-hop questions, and can also be easily cross-linked to an existing KB.

Phrase-indexed QA and virtual KBs are also related to QEDB. QA systems like RePAQ and QAMAT are similar to phrase-indexed QA (PIQA) systems (Seo et al. 2018; Lee

et al. 2020), and can be viewed as variants of PIQA, in which rather than encoding and indexing a possible answer span a' directly, one encodes and indexes a question q' generated from a' by a QG system. Relative to PIQA, one advantage of RePAQ is that both steps of the encoding are semantically meaningful and verifiable: QG is correct if it produces a meaningful question that is answerable using the document, and question retrieval is correct if it finds a q that is semantically equivalent to the user query q. Unlike REPAQ and QAMAT, a QEDB is naturally relational, which makes it more closely related to relational extensions to PIQA, like DrKIT(Dhingra et al. 2020) or OPQL (Sun et al. 2021), however, it shares a similar advantage in the interpretability of the intermediate encoding steps (similar to the advantage held by RePAQ and QAMAT relative to PIQA).

Concurrently with this work, a second project made use of the PAQ data as a memory for knowledge-intensive generation tasks (Wu et al. 2022). That system, called EMAT, differs in many respects from QAMAT, notably in being much more faster at inference time (at a small cost in accuracy). EMAT was also evaluated on more tasks, including long-form knowledge-intensive generation tasks, further supporting the claim of this paper that question-answer memories are effective for knowledge-intensive NLP tasks. The arguments made above concerning the relative utility compositionality of QA pairs, and their similarities and differences to conventional KB triples, are not explored in (Wu et al. 2022).

QAMAT uses a novel approach to learning retrieval-augmented generation, in which retrieval and generation are initial learned end-to-end in-batch in a special "retrieval learning phase", followed by a second phase in which the entire memory is encoded and indexed for faster retrieval. This approach was first used by (de Jong et al. 2021), and subsequently by (Zhong, Lei, and Chen 2022).

Explainable QA

Explainable QA annotations (Lamm et al. 2021) were used as training data to model the alignment of question references to document references. The primary difference between a QEDB and explainable QA systems is that explainable QA models are applied at query time, i.e., in a "lazy" way, while in building a QEDB, models are applied once to an entire corpus, i.e., in an "eager" way, to create a KB-like structure which is explicitly stored and indexed. As a result a QEDB has advantages for tasks that require "reasoning", in the broad sense of aggregation of information across many documents. Such questions are often expensive to answer with iterative retrieve-and-read systems. Examples of such questions include multihop QA; finding multiple answers for an ambiguous question (e.g., "*who played jamie lannister in GoT*"); and finding many answers to a question (e.g., "*what movies did william shatner play kirk in*").

Question-based Document Annotation

Question-based document annotations have been proposed as a scheme for annotating text for numerous NLP tasks, e.g., for relation extraction (Levy et al. 2017), slot filling (Du et al. 2021) semantic role labeling (FitzGerald et al. 2018; Klein et al. 2020) and discourse relations (Pyatkin et al. 2020). Most related to this work, (Michael et al. 2017) proposed question-answer pairs as a general meaning representation for text. Specifically, this work introduced *question-answer meaning representation* (QAMR) as a representation of the meaning of documents.Our work differs from QAMR in two important ways: (1) QAMR is not intended to model what is salient in a document, unlike a QEDB; (2) QAMR is not intended to model information in a corpus, and focuses on information in a single sentence. QA-Align (Weiss et al. 2021) makes use of QA-SRL annotations (FitzGerald et al. 2018) as features for cross-document co-reference. Because cross-document co-reference is one of several components in a QEDB, our proposal is broader in scope, but the results of (Weiss et al. 2021) support our general claim that there are synergies between the various subtasks involved in building (or using) a QEDB. Question data has also been proposed as a resource for encoding generally useful language models (Jia, Lewis, and Zettlemoyer 2021; He, Ning, and Roth 2019) as well. While this goal is different from the one pursued here, it shares the intuition that the factual content of documents is summarized well by questions and answers.

Conclusion

Symbolic KBs organize information into small modular components (e.g., entities, KG triples, WikiData statements) that can be combined compositionally to answer complex queries. While many recent papers have focused on tasks like open QA, where questions are answered from text without using a KB, broad-coverage symbolic KBs continue to be widely used in practice, and, despite recent progress in methods for "multi-hop" QA, are still the only computationally efficient way of answering questions that combine information for multiple documents. However, the broad-coverage KBs that are currently in wide use are largely collections of information that *easily collected and integrated*, and need not reflect the actual information needs of users. In this position paper, we advocate for a new approach to constructing KBs, and in particular, an approach to collecting modular, compositionally-combinable knowledge components from text, driven by a sample of user's questions and answers.

Our approach begins with data on likely questions and answers, and extrapolates this data, using neural question generation, to a larger set of QA pairs. A rich relational structure is then created by aligning the entities mentioned in these questions with traditional KB entities. This alignment is done using data for explainable QA systems (Kwiatkowski et al. 2019) and standard entity-linking methods, and leads to a QEDB structure which is relational; grounded in a corpus; aligned with the original sample of questions; and easily combined with existing symbolic KBs.

The central claims made are that like a traditional symbolic KB, a QEDB is composed of modular components that can be combined in compositionally in many different ways; and that the information in a QEDB will be more useful for answering user information needs than the information in a traditionally constructed broad-coverage KB. The former claim is supported qualitatively, by presenting examples of

the component elements of the QEDB, and examples of how they can be recombined. The latter claim is supported quantatively: using available benchmarks, we show that QEDB is a better substrate for question-answering than symbolic KBs, unless the set of question is carefully filtered to contain only questions supported by the KB (as is the case for WebQSP). However, even for questions chosen to be conducive to use of a KB, a QEDB is extremely competitive when similar QA system architectures are used: the only cases in which the traditional KB is more than 3 point better than QEDB is when the match between the questions and KB is ideal (WebQSP/WD) and when QA architectures are optimized for KBQA.

If these positions are generally true, this would suggest a number of new directions and emphases for research. For example, it suggests new focuses for question generation, such as generation of high-recall sets of questions, and also suggests more work on direct evaluation of the correctness of generated questions. (Most current work evaluates QG indirectly, by its contribution to tasks like pre-training for QA, or evaluating factuality of generated text). More notably, it suggests that the target for information extraction systems should generally be not a traditional KB, but a QEDB.

Acknowledgments

The authors are grateful to many colleagues at Google for valuable discussions and comments on earlier versions of this work, and technical help in implementing the ideas of this paper. We especially thank Daniel Andor, Michael Collins, Tom Kwaitkowski, Eva Schlinger, and Livio Baldini Soares for their contributions.

References

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, 722–735. Springer.

Balachandran, V.; Dhingra, B.; Sun, H.; Collins, M.; and Cohen, W. 2021. Investigating the Effect of Background Knowledge on Natural Questions. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 25–30. Online: Association for Computational Linguistics.

Berant, J.; Chou, A.; Frostig, R.; and Liang, P. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1533–1544.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250.

Chen, W.; Verga, P.; de Jong, M.; Wieting, J.; and Cohen, W. 2022. Augmenting Pre-trained Language Models with QA-Memory for Open-Domain Question Answering. *arXiv preprint arXiv:2204.04581*.

de Jong, M.; Zemlyanskiy, Y.; FitzGerald, N.; Sha, F.; and Cohen, W. 2021. Mention Memory: incorporating textual knowledge into Transformers through entity mention attention. *arXiv preprint arXiv:2110.06176*.

Dhingra, B.; Zaheer, M.; Balachandran, V.; Neubig, G.; Salakhutdinov, R.; and Cohen, W. W. 2020. Differentiable reasoning over a virtual knowledge base. *arXiv preprint arXiv:2002.10640*.

Du, X.; He, L.; Li, Q.; Yu, D.; Pasupat, P.; and Zhang, Y. 2021. QA-Driven Zero-shot Slot Filling with Weak Supervision Pretraining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 654–664. Online: Association for Computational Linguistics.

Dubey, M.; Banerjee, D.; Abdelkawi, A.; and Lehmann, J. 2019. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In *International semantic web conference*, 69–78. Springer.

Etzioni, O.; Banko, M.; Soderland, S.; and Weld, D. S. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12): 68–74.

Févry, T.; Soares, L. B.; FitzGerald, N.; Choi, E.; and Kwiatkowski, T. 2020. Entities as experts: Sparse memory access with entity supervision. *arXiv preprint arXiv:2004.07202*.

FitzGerald, N.; Michael, J.; He, L.; and Zettlemoyer, L. 2018. Large-scale QA-SRL parsing. *arXiv preprint arXiv:1805.05377*.

He, H.; Ning, Q.; and Roth, D. 2019. Quase: Question-answer driven sentence encoding. *arXiv preprint arXiv:1909.00333*.

Jia, R.; Lewis, M.; and Zettlemoyer, L. 2021. Question Answering Infused Pre-training of General-Purpose Contextualized Representations. *arXiv preprint arXiv:2106.08190*.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Klein, A.; Mamou, J.; Pyatkin, V.; Stepanov, D.; He, H.; Roth, D.; Zettlemoyer, L.; and Dagan, I. 2020. QANom: Question-Answer driven SRL for Nominalizations. In *Proceedings of the 28th International Conference on Computational Linguistics*, 3069–3083. Barcelona, Spain (Online): International Committee on Computational Linguistics.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 453–466.

Lamm, M.; Palomaki, J.; Alberti, C.; Andor, D.; Choi, E.; Soares, L. B.; and Collins, M. 2021. QED: A Framework and Dataset for Explanations in Question Answering. *Transactions of the Association for Computational Linguistics*, 9: 790–806.

Lee, J.; Sung, M.; Kang, J.; and Chen, D. 2020. Learning dense representations of phrases at scale. *arXiv preprint arXiv:2012.12624*.

Levy, O.; Seo, M.; Choi, E.; and Zettlemoyer, L. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

Lewis, P.; Wu, Y.; Liu, L.; Minervini, P.; Küttler, H.; Piktus, A.; Stenetorp, P.; and Riedel, S. 2021. Paq: 65 million probably-asked questions and what you can do with them. *arXiv preprint arXiv:2102.07033*.

Mausam, M. 2016. Open information extraction systems and downstream applications. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence*, 4074–4077.

Michael, J.; Stanovsky, G.; He, L.; Dagan, I.; and Zettlemoyer, L. 2017. Crowdsourcing question-answer meaning representations. *arXiv preprint arXiv:1711.05885*.

Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Yang, B.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; et al. 2018. Never-ending learning. *Communications of the ACM*, 61(5): 103–115.

Pyatkin, V.; Klein, A.; Tsarfaty, R.; and Dagan, I. 2020. QADiscourse - Discourse Relations as QA Pairs: Representation, Crowdsourcing and Baselines. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2804–2819. Online: Association for Computational Linguistics.

Rebele, T.; Suchanek, F.; Hoffart, J.; Biega, J.; Kuzey, E.; and Weikum, G. 2016. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *International semantic web conference*, 177–185. Springer.

Seo, M.; Kwiatkowski, T.; Parikh, A. P.; Farhadi, A.; and Hajishirzi, H. 2018. Phrase-indexed question answering: A new challenge for scalable document comprehension. *arXiv preprint arXiv:1804.07726*.

Sun, H.; Verga, P.; Dhingra, B.; Salakhutdinov, R.; and Cohen, W. W. 2021. Reasoning over virtual knowledge bases with open predicate relations. *arXiv preprint arXiv:2102.07043*.

Verga, P.; Sun, H.; Soares, L. B.; and Cohen, W. 2021. Adaptable and Interpretable Neural MemoryOver Symbolic Knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3678–3691.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10): 78–85.

Weiss, D. B.; Roit, P.; Klein, A.; Ernst, O.; and Dagan, I. 2021. QA-Align: Representing Cross-Text Content Overlap by Aligning Question-Answer Propositions. *CoRR*, abs/2109.12655.

Wolfson, T.; Geva, M.; Gupta, A.; Gardner, M.; Goldberg, Y.; Deutch, D.; and Berant, J. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8: 183–198.

Wu, Y.; Zhao, Y.; Hu, B.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2022. An Efficient Memory-Augmented Transformer for Knowledge-Intensive NLP Tasks. *arXiv preprint arXiv:2210.16773*.

Yang, Z.; Hu, J.; Salakhutdinov, R.; and Cohen, W. W. 2017. Semi-supervised qa with generative domain-adaptive nets. *arXiv preprint arXiv:1702.02206*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Yih, W.-t.; Richardson, M.; Meek, C.; Chang, M.-W.; and Suh, J. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 201–206.

Zhong, Z.; Lei, T.; and Chen, D. 2022. Training Language Models with Memory Augmentation. *arXiv preprint arXiv:2205.12674*.