Conversational Question Answering on Heterogeneous Sources

Philipp Christmann
Max Planck Institute for Informatics
Saarbruecken, Germany
pchristm@mpi-inf.mpg.de

Rishiraj Saha Roy
Max Planck Institute for Informatics
Saarbruecken, Germany
rishiraj@mpi-inf.mpg.de

Gerhard Weikum
Max Planck Institute for Informatics
Saarbruecken, Germany
weikum@mpi-inf.mpg.de

ABSTRACT

Conversational question answering (ConvQA) tackles sequential information needs where contexts in follow-up questions are left implicit. Current ConvQA systems operate over homogeneous sources of information: either a knowledge base (KB), or a text corpus, or a collection of tables. This paper addresses the novel issue of jointly tapping into all of these together, this way boosting answer coverage and confidence. We present Convinse, an end-to-end pipeline for ConvQA over heterogeneous sources, operating in three stages: i) learning an explicit structured representation of an incoming question and its conversational context, ii) harnessing this frame-like representation to uniformly capture relevant evidences from KB, text, and tables, and iii) running a fusion-in-decoder model to generate the answer. We construct and release the first benchmark, ConvMix, for ConvQA over heterogeneous sources, comprising 3000 real-user conversations with 16000 questions, along with entity annotations, completed question utterances, and question paraphrases. Experiments demonstrate the viability and advantages of our method, compared to state-of-the-art baselines.

CCS CONCEPTS

• **Information systems** → *Question answering*.

KEYWORDS

Conversations, Question Answering, Explainability

ACM Reference Format:

Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2022. Conversational Question Answering on Heterogeneous Sources. In *Proceedings of the 45th Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3531815

1 INTRODUCTION

Motivation. Conversational question answering (ConvQA) [6, 32, 37, 40] is is a popular mode of communication with digital personal assistants like Alexa, Cortana, Siri, or the Google Assistant, that are ubiquitous in today's devices. In ConvQA, users pose questions to the system sequentially, over multiple turns. In conversations between two humans, follow-up questions usually contain *implicit* context. The ConvQA system is expected to resolve such implicit information from the conversational history. Consider, for example, a typical ConvQA session on factual knowledge below:

Table 1: Question understanding approaches for ConvQA.

Original q^3	*Release date of first season?*
Question resolution	*Release date of first season? in GoT*
Question rewriting	*What was the release date of the first season of GoT?*
Convinse SR	⟨ `GoT` | `first season` | `release date` | `date` ⟩

q^0: *Who played Jaime Lannister in GoT?*
a^0: `Nikolaj Coster-Waldau`
q^1: *What about the dwarf?*
a^1: `Peter Dinklage`
q^2: *When was he born?*
a^2: `11 June 1969`
q^3: *Release date of first season?*
a^3: `17 April 2011`
q^4: *Duration of an episode?*
a^4: `50-82 minutes`

State-of-the-art works on ConvQA make use of single information sources: either curated knowledge bases (KB) [8, 10, 18, 20, 22, 26, 29, 41, 44], or unstructured text collections [5, 13, 31, 33, 34], or Web tables [14, 27], but only one of these. Questions q^0 and q^2 can be answered more conveniently using KBs like Wikidata [53], YAGO [46], or DBpedia [1], that store factual world knowledge in compact RDF triples. However, answering q^1, q^3 or q^4 via a KB requires complex reasoning efforts. For q^1, even with named entity disambiguation (NED) in conversations [17, 43], it is unlikely that the correct KB entity (`Tyrion Lannister`) can be inferred, which means that the resulting answer search space [7] will not have the answer. For q^3, answering via a KB requires a two-step lookup involving the first season, and then the corresponding release date. For q^4, a KB might have the details for each individual episode, but collecting this information and aggregating for the final answer can be quite cumbersome. In contrast, the answers to these three questions are much more easily spotted in content of text documents, or Web tables. In addition, there are obviously many information needs where answers are present only in text form, as KBs and Web tables have inherently limited coverage. An example would be a question like: *What did Brienne call Jaime?* (*"Kingslayer"*). A smart ConvQA system should, therefore, be able to tap into more than one kind of knowledge repository, to improve answer recall and to boost answer confidence by leveraging multiple kinds of evidence across sources.

Limitations of state-of-the-art. Existing research on ConvQA has considered solely one kind of information source for deriving answers. Further, when specializing on a given source, methods often adopt source-specific design choices that do not generalize well [8, 10, 13]. For example, representations of the conversational context, like KB subgraphs or text passages, are often specifically

SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07.
https://doi.org/10.1145/3477495.3531815

modeled for the knowledge repository at hand, making these heterogeneous sources apparently incompatible. Methods for question rewriting [9, 36, 51] and question resolution [21, 52] convert short user utterances into full-fledged questions where the intent is made completely *explicit*. However, this adds major complexity and may lose valuable cues from the conversation flow. Further, these methods face evidence retrieval problems arising from long and potentially verbose questions [11].

There has been substantial work on single-question QA over heterogeneous sources [3, 12, 28, 47–49, 54, 57], with complete questions as input. Among these, only Oğuz et al. [28] and Ma et al. [25] try to deal with KBs, text, and tables. Their approach is designed for simple questions, though, and cannot easily be extended to the challenging ConvQA setting [40]. Finally, none of the prior works on ConvQA produce human-interpretable structures that could assist end users in case of erroneous system responses.

Approach. To overcome these limitations, we propose Convinse (CONVQA with INtermediate Representations on Heterogeneous Sources for Explainability), an end-to-end framework for conversational QA on a mixture of sources. Convinse consists of three main stages: i) *question understanding (QU)*, ii) *evidence retrieval and scoring (ERS)*, and iii) *heterogeneous answering (HA)*.

The first stage, QU, is our primary contribution in this work. It addresses the challenges of incomplete user utterances introduced by the conversational setting. We derive an *intent-explicit structured representation (SR)* that captures the complete information need. Table 1 shows such an SR for q^3 of our running example. SRs are frame-like structures for a question that contain designated slots for open-vocabulary lexical representations of entities in the conversational context (marked gray) and the current question (red), relational predicates (blue), and expected answer types (cyan). SRs can be viewed as concise gists of user intents, intended to be in a form independent of any specific answering source. They are self-contained interpretable representations of the user's information need, and are inferred using fine-tuned transformer models trained on data generated by distant supervision from plain sequences of QA pairs. We further propose a conversational flow graph (CFG), which can be inferred from the SR, and enhances the explainability of the derivation process.

The second stage, ERS, exploits recent developments in entity-based retrieval [7] to judiciously retrieve question-relevant evidences (KB-facts, text-sentences, table-records, or infobox-entries) from each information source. These heterogeneous evidences are verbalized [16, 28, 30] on-the-fly and run through a scoring model. The top-k pieces of evidence are passed to the answering stage.

The third and final stage, HA, consists of a fusion-in-decoder (FiD) model [15], that is state-of-the-art in the retrieve-and-read paradigm for open-domain QA. FiD acts as a "generative reader", creating a crisp answer from the top-k evidences, that is returned to the end user.

Benchmark. Another novel contribution is the construction of ConvMix, the first benchmark for ConvQA over heterogeneous sources. ConvMix is a crowdsourced dataset that contains questions with answers emanating from the Wikidata KB, the full text of Wikipedia articles, and the collection of Wikipedia tables and infoboxes. ConvMix contains 2800 conversations with five turns (14k utterances), and 200 conversations with ten turns (2k utterances), their gold answers and respective knowledge sources for answering. Conversations are accompanied by metadata like entity annotations, completed questions, and paraphrases. The collected dataset ConvMix, and all our code and data for Convinse can be accessed at **https://convinse.mpi-inf.mpg.de**.

Contributions. Our salient contributions are the following:

- The paper proposes Convinse, the first end-to-end method for ConvQA over heterogeneous sources.
- It introduces structured representations to capture user intents in a structured and explainable manner, a key element for seamless answering over a mixture of heterogeneous sources.
- It presents distant supervision mechanisms to automatically annotate conversations with structured representations.
- It provides ConvMix, the first benchmark for ConvQA over heterogeneous sources.

2 CONCEPTS AND NOTATION

Question. A natural language question q is a sequence of words expressing an interrogative intent. A question can be complete (i.e., self-contained/full-fledged/intent-explicit), or incomplete (i.e., context-dependent/partial/intent-implicit). Incomplete questions require context from previous questions and answers in the conversation to be answered correctly.

Answer. An answer a to q is a crisp phrase (or a list) that satisfies the intent in q. In a heterogeneous scenario, the answer phrase a can be an entity or literal (constant) coming out of the KB or a table or infobox, or any span of short text from the document corpus.

Conversation. A conversation C consists of a sequence of questions $(q^0, q^1, \ldots)$ and corresponding answers $(a^0, a^1, \ldots)$ (see Sec. 1 for an example). The first question q^0 in C is complete, while follow-up questions are usually incomplete.

Turn. A turn in C consists of a specific $\langle q^i, a^i \rangle$ pair. For example, the second turn refers to $\langle q^1, a^1 \rangle$.

Knowledge base. A knowledge base is a set of facts, where each fact is a `<subject, predicate, object>` (SPO) triple, optionally augmented by `<qualifier predicate, qualifier object>` pairs which specify additional information for the main triple (e.g. `<Game of Thrones, cast member, Nikolaj Coster-Waldau; character role, Jaime Lannister>`). Subjects are entities (`Game of Thrones`), while objects can be entities, types (`human`) or literals (constants such as numbers with or without units, dates like `11 June 1969`, etc.). Predicates (`cast member`) denote relationships.

Text collection. A text collection is a set of documents, where each document consists of a sequence of sentences.

Table. A table is a structured relational construct consisting of cells organized into rows and columns, with optional row and column headers. Cell values are typically entities or literals, while headers are often predicates.

Infobox. An infobox is a list of salient attribute-value pairs about an entity. A Wikipedia infobox appears on the top right corner of the entity's Wikipedia page. Infobox entries resemble KB-facts, but they are not necessarily clean in terms of entity linkage (e.g., a birthplace could be given as a string with city, country or other regional variations).

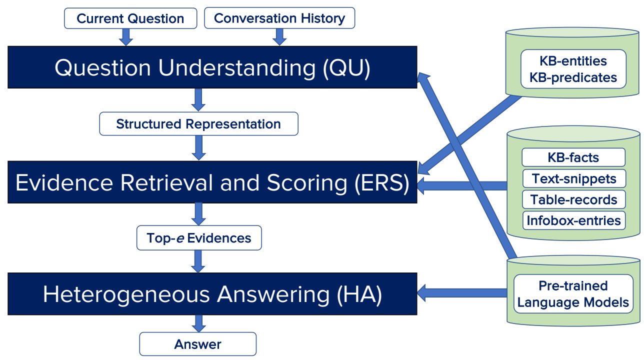

Figure 1: An overview of the CONVINSE system.

Evidence. An evidence is a unit of retrieval that can come from any of the heterogeneous sources above: a KB-fact, a text-sentence, a table-record (row), or an infobox-entry. Evidences form the sources of answer candidates.

Answering evidence. An answering evidence is an evidence which contains at least one correct answer *a*. It can either mention the answer as a string, or include an answer entity (in case of KB-facts).

3 THE CONVINSE METHOD

Fig. 1 shows an overview of the CONVINSE architecture. The following subsections discuss the three steps: question understanding, evidence retrieval and scoring, and heterogeneous answering.

3.1 Question understanding (QU)

Follow-up questions in a conversation ($q^1, q^2, \ldots$) usually contain implicit intent. A key challenge in ConvQA, therefore, is to understand the flow of the conversation, towards deriving intent-explicit structured representations (SR) of the user's information need in q^i. Instead of trying to generate a full-fledged question, we rather aim to capture the semantic structure, using a *syntax-agnostic* approach. This can be perceived as a "logical form" for heterogeneous QA, where no explicit grounding or canonicalization is possible. The representation is purely on the question-level, and thus agnostic to the information sources that are used during the answering process. However, it can readily be matched with different kinds of evidences, which often take the form of keyword phrases (e.g. text snippets or verbalized table records).

Specifically, an SR is a 4-tuple holding a slot for each of:

- Context entities (depicted in gray in Table 1),
- Question entities (in red),
- Question predicates (in blue), and,
- Expected answer types (in cyan).

Context and question entities. As an example, consider the gold SR for q^1 of the running example: ⟨ `GoT | the dwarf | who played | human` ⟩. The context entity (`GoT` in this case) is an entity mention from the conversational context. The question entity is the entity mention targeted in the current question (e.g. `the dwarf`). Here, the context entity makes the question entity explicit, indicating that the question is on the dwarf in Game of Thrones. Inferring the question entity may need to take the history into account (e.g., for q^3 in Table 1). The context entity and question entity can consist of multiple such mentions. This is required for questions such as *"Where did Dany and Jon first meet?"*, with the gold SR being ⟨ `GoT | Dany and Jon Snow | first meet | location` ⟩.

Question predicates. The question predicate is the counterpart to the relation or attribute of interest in a logical form. However, it is merely a surface phrase, without any normalization or mapping to a KB. This way, it is easy to match it against any kind of information source. For example, the question predicates `who played` or `first meet` can be matched with evidences from KB or text-snippets from documents or table headers, alike.

Answer types. Expected answer types assist the answering model in detecting and eliminating spurious answer candidates [40, 58]. In general, multiple types could be inferred here. The question predicate `first meet` alone could imply the answer type to be either *"date"* or *"location"*. Stopwords like *"where"* are often disregarded by downstream QA models [8]; in contrast, the SR answer type retains this information and would infer only the correct *"location"*. Further, this type can help in identifying the expected granularity of the answer. For the question *"When is his birthdate?"*, one would expect a complete date with day, month and year as the answer, but for *"When did they win their last world cup?"* the corresponding year would be enough and actually desired. *"Date"* and *"year"* would respectively populate the fourth slot in these cases.

Specific slots in the SR can be left blank. For q_4, the question entity `GoT` is already explicit, and thus no context entity is required: ⟨ `_ | GoT | duration of an episode | number` ⟩.

The SR generation is implemented by fine-tuning a pre-trained sequence generation model. We tried BART [24] and T5 [35] in preliminary experiments, and found BART to perform better. BART is particularly effective when information is copied and manipulated from the input to generate the output autoregressively [24], which is exactly the setting here. The conversation history and the current question concatenated with a delimiter constitute the input, and the SR is the output. When encoding the history and the current question, the model considers cross-attention between turns, identifying relevant parts from the conversation history.

SRs and explainability. One of our primary goals in CONVINSE was to produce intermediate representations for end users as we proceed through the QA pipeline. Concretely, understanding the flow within the conversation is an essential problem in ConvQA [8, 13]. While the SR itself is human-readable, when presented only with the SR (or some rewritten/resolved question), certain decisions of the ConvQA system might not be immediately obvious to a real user. Here, we propose an intuitive mechanism to infer and present the conversational flow to a user: given the generated SR, we identify the source turn for each word, using exact match in the history, and consider such source turns as relevant for the current question. If there is no source turn, we consider the question at hand as self-sufficient. A conversational flow graph (CFG) is established as follows: questions and answers are nodes, and an edge connects a question to its relevant history. Due to the potential dependence of a turn on multiple preceding ones, the CFG for a conversation may not strictly be a tree, but rather a directed acyclic graph (DAG). The CFG can be presented to the interested end user together with the SR, as depicted for q^4 in Fig. 2, either for gaining confidence in final answers, or for scrutinizing error cases.

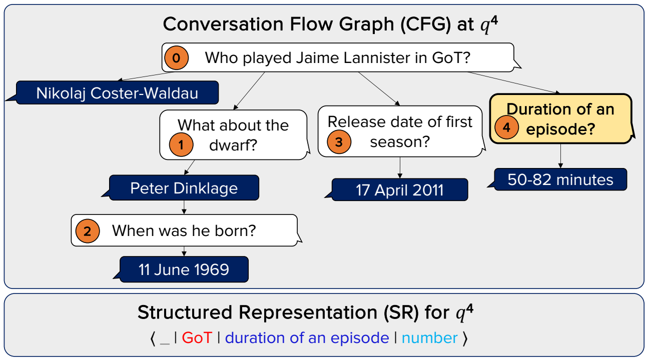

Figure 2: Conversation flow graph for our running example.

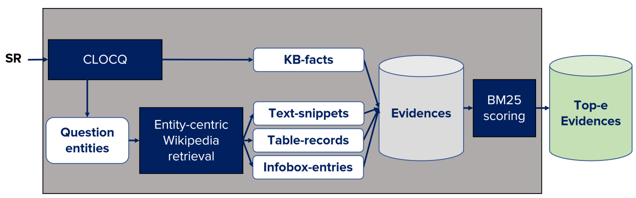

Figure 3: An overview of the ERS stage.

3.2 Evidence retrieval and scoring (ERS)

Evidence retrieval. At this stage, the goal is to retrieve relevant evidences given the generated SR. We convert retrieved evidences on-the-fly to verbalized NL forms [28]. This is done for harnessing the state-of-the-art fusion-in-decoder (FiD) model downstream in our pipeline's final step (Sec. 3.3): FiD generates answers, given the question and a number of NL sentences as input. Table 2 shows example evidences from different sources for the context entity `Game of Thrones`.

For evidences from the **KB**, we make use of CLOCQ [7] (available publicly at https://clocq.mpi-inf.mpg.de/), which is a recent method for retrieving relevant KB-facts and providing top-k disambiguations (mappings to KB-items) for relevant cue words (entity, predicate, and type mentions), given an explicit user question. Since questions are treated as keyword queries, the SR can directly be fed into CLOCQ (removing the separator '|' during retrieval). KB-facts are verbalized [16, 28, 30], separating the individual constituents of a fact by a comma. These mappings between KB-item mentions in verbalized facts and KB-item IDs are retained in-memory to help us later during evaluation (Sec. 5.3).

For **text-corpus** evidences, we take an entity-centric view of Wikipedia: for each of the disambiguated entities from CLOCQ, we retrieve the corresponding Wikipedia page. For example, for the SR ⟨ `GoT | Jaime Lannister | who played | human` ⟩, we would consider the Wikipedia pages for `Jaime Lannister`, `Game of Thrones (TV Series)`, `Game of Thrones (A Song of Ice and Fire)`, and so on (CLOCQ allows for multiple disambiguations per question token). The text within the page is split into single sentences.

We extract **tables and infoboxes** from the retrieved Wikipedia pages. Every table row is transformed to text individually, concatenating cell values with the respective column headers with an *"is"* in between, and separating such ⟨cell, header⟩ pairs with a comma [28]. For each infobox attribute (similar to predicates in a KB), we concatenate the corresponding lines, having again, a comma as separator. The page title is prepended to all evidences from Wikipedia as additional context.

In addition, we exploit anchor links in evidences to other Wikipedia pages to map the corresponding entity mentions (anchor texts) to KB-items. Wikipedia links are transformed to Wikidata IDs using a dictionary. Similarly, dates and years in evidences are identified and normalized to the standard KB format using simple patterns. We keep such ⟨entity mention, KB-item⟩ pairs for all evidences in memory, as this helps us later in grounding answers to the KB during evaluation (Sec. 5.3). For example, for the text evidence in Table 2, *"Tyrion"* might link to the Wikipedia page of `Tyrion Lannister`: so we add the pair ⟨*"Tyrion"*, `Tyrion Lannister`⟩ as metadata to the corresponding evidence.

Evidence scoring. The set of evidences compiled by the previous stage can be quite large (several thousands), which can affect the efficiency and effectiveness of the answering phase. Therefore, we first reduce this set, keeping only the most relevant information. Since all evidences are verbalized at this stage, each individual evidence can be treated as a document, with the SR as the query. We then use the standard IR model BM25 [38] for retrieving the top-e relevant pieces of information.

3.3 Heterogeneous answering (HA)

Given the top-e relevant evidences per question, we make use of a state-of-the-art fusion-in-decoder [15] (FiD) model. Different from the typical span prediction in the retrieve-and-read paradigm [2], FiD *generates* the answer, following a sequence-to-sequence approach. FiD first pairs every evidence with the question, which is the SR in our case, and encodes these pairs, leveraging cross-attention to identify relevant information. The concatenation of these encodings is then fed into a decoder, which generates the answer autoregressively.

3.4 Distantly supervised labeling

Intuition. To train the QU phase of CONVINSE, we need a collection of ⟨history, question⟩-pairs, with the corresponding gold SRs. Annotating conversations with such information is tricky with crowdworkers, and expensive too (c.f. Sec. 4). Even the annotation of conversations with completed questions is harder and much more expensive than collecting plain sequences of question-answer pairs. Therefore, we devise a mechanism to automatically generate the gold SRs from pure conversations. Our technique is based on the following intuition: if a piece of information (e.g. an entity or relation phrase) in previous turns is essential for the understanding of the current incomplete question and this information has been left implicit by the user, then it should be included in a completed version of the question. Consider this example:

> q^0: *Who played Jaime Lannister in GoT?*
> a^0: `Nikolaj Coster-Waldau`
> q^1: *What about the dwarf?*

It is unlikely that proper evidences (on Game of Thrones, given q^0), can be found for the incomplete question q^1. However, once *"GoT"*

Table 2: Verbalized evidences from different input sources.

KB	*Game of Thrones, cast member, Nikolaj Coster-Waldau, character role, Jaime Lannister*
Text	*Game of Thrones, The third and youngest Lannister sibling is the dwarf Tyrion (Peter Dinklage) (...).*
Table	*Game of Thrones, Season is Season 1, (...), First aired is April 17, 2011 (...).*
Infobox	*Game of Thrones, Running time, 50–82 minutes*

is added to q^1 (e.g., *"What about the dwarf in GoT?"*), *answering evidences* (defined in Sec. 2) can be retrieved. This suggests that the phrase *"GoT"* should feature in the SR for q^1.

Implementation. Based on this idea, we create data for training the QU model as follows: starting with the complete question q^0, we retrieve all evidences (Sec. 3.2). Since our retrieval is entity-centric, we can identify entity mentions which bring in *answering evidences*: for each evidence, the retriever returns the text span from the input question that the evidence was retrieved for. Such entity mentions are considered relevant for the respective conversation turn. For the incomplete follow-up questions, we iteratively add such relevant entity mentions from previous turns, and then retrieve evidences for the current question at hand. When adding the entity mention results in answering evidences being retrieved, we consider the entity as relevant for the current turn. Similarly, entity mentions in the current turn are identified as relevant, if answering evidences are retrieved for them.

The gold SR is then constructed heuristically: i) if there are relevant entities from the current turn, then these feature in the question entity slot, and relevant entities (if any) from previous turns become context entities in the SR, and, ii) if there are only relevant entities from previous turns, then these become question entities. The remaining words in the current question (except for stopwords) fill up the question predicate slot. The expected answer type is directly looked up from the KB, using the gold answer. Since the KB may have several types for the same entity, we take the most frequent one to avoid very specific types. For example, `Tyrion Lannister` has types `fictional human` (more frequent) and `GoT character` (less frequent) in Wikidata: so we only consider `fictional human` in our SR.

3.5 Training the Convinse framework

We train the QU model first, using data generated in the manner discussed in Sec. 3.4. After training, we directly apply the QU model on the whole data (including training data), to generate SRs. In the remainder of the approach, we utilize only the SRs generated by our model. When training the FiD model, we skip instances for which the top-e evidences do not have the answer, since the model would not be able to find the answer within the evidences, and could hallucinate at inference time [39]. This way, the model is taught to predict the answer from the input. Since we would like to treat evidences from different sources in a unified way, only one model is trained on the top-e evidences after retrieval from all sources. To demonstrate the robustness of Convinse, this same model is subsequently used for all combinations of input sources, including inputs from single sources.

Table 3: Basic statistics for the ConvMix benchmark.

Title	Generate 5 conversations for question answering
Description	Choose entities of your choice from five domains, generate questions about them, and find answers from Wikidata and Wikipedia
Participants	32 unique Master Turkers
Time allotted (HIT)	4 hours maximum
Time taken (HIT)	1.5 hours on average
Payment per HIT	15 USD
Domains	Books, Movies, Music, TV series, Soccer
Conversations	3000
Questions	16000
Question length	8.78 words (initial), 5.19 (follow-ups), 5.87 (all)
Answer size	1.02 entities/strings on average
Entities covered	5418 (long-tail: 2511, with <50 KB-facts)
Heterogeneity	2626 conversations (>1 source used by Turker)

4 THE CONVMIX BENCHMARK

Limitations of existing ConvQA benchmarks. Notable efforts at ConvQA benchmarking like QuAC [6] (text), CoQA [37] (text), SQA [14] (tables), ConvQuestions [8] (KB), and CSQA [41] (KB) assume a single answering source. Rather than the easier option of artificially augmenting any of these with heterogeneous inputs, we believe that it is much more natural and worthwhile to build a new resource from scratch by users browsing through a mixture of sources, as they would do in a typical information seeking session on the Web.

Initiating conversations. How a user initiates a conversation is a key conceptual challenge that needs to be overcome in creating good ConvQA benchmarks. One could, for example, provide users with passages or documents, and ask them to create a sequence of questions from there [6, 37]. Alternatively, one could also provide annotators with some conversation from a benchmark so far, and request their continuation in some fashion [20]. Large-scale synthetic benchmarks would try to automate this as far as possible using rules and templates [41]. In keeping with our philosophy of natural conversations, we asked users to start with an entity of their choice (instead of spoonfeeding them with one, which could be counterproductive if the user has no interest or knowledge about the provided entity). Real conversations between humans, or several search sessions, often start when users have queries about such seed or topical entities. With the first question initiating the conversation, we collected four follow-up questions (total of five turns) that build upon the ongoing conversational context.

Quality control. The study was conducted on the popular crowdsourcing platform Amazon Mechanical Turk (AMT), where we allowed only Master Workers to participate, for quality assurance. We also blocked single and sets of annotators who demonstrated evidence of excessive repetition or collusion in their annotations. Since the task is non-trivial for an average Turker (requires understanding of factoid questions, and familiarity of knowledge sources like Wikidata and Wikipedia, along with entities and literal answers), we also included quite a few internal checks and warnings that could prompt users for unintentional mistakes before task submission. Workers with notably diverse and interesting conversations were awarded with a 5 USD bonus. Interestingly, several

Turkers providing high-quality conversations seemingly found the task engaging (we provided a free-text feedback box), and submitted more than 20 HITs. The authors conducted semi-automatic post-processing, validation and cleaning of the benchmark. Several issues were also resolved by meticulous manual inspection. For example, we ran CLOCQ [7] on the initial questions, and manually inspected cases for which no answering evidences were found, to identify and rectify cases in which the initial questions themselves were unanswerable. Such cases are specifically problematic, because the whole conversation might become unanswerable.

Ensuring heterogeneity. Last but not the least, ensuring answer coverage over heterogeneous sources was a key concern. Here, we again kept it natural, and encouraged users not to forcibly stick to any particular source during their conversation. Interestingly, out of 3000 conversations, only 374 used exactly one source. A majority (1280) touched three sources, 572 touched four, while 774 used two inputs. Finally, note that this is only the source that the annotator used during her search process: it is quite possible that the answer can be located in other information sources (see the field [·] below answers in Table 4), thereby enabling future benchmark users to exploit answer redundancy.

Collecting longer conversations. We initially collected 2800 conversations with five turns (referred to as CONVMIX-5T). However, there can also be cases in which users wish to dive deep into a specific topic, or other curiosities arise as the conversation continues. In such situations, conversations can easily go beyond five turns, making the understanding of the conversational flow even more challenging for the ConvQA system. Therefore, we collected 200 additional conversations with ten turns (denoted CONVMIX-10T), to test the generalizability of ConvQA systems over longer conversations. On manual investigation, we found that there are naturally more topic drifts within these conversations. These 2k (200×10) questions are only used as an additional test set to serve as a robustness check for pre-trained models. Thus, our complete benchmark CONVMIX (3000 conversations in total) is made up of subsets CONVMIX-5T (2800 conversations, 5 turns each) and CONVMIX-10T (200 conversations, 10 turns each).

Collected fields. We collected the following annotations from crowdworkers: i) conversational questions, ii) intent-explicit versions of follow-up questions, iii) gold answers as plain texts and Wikidata URLs, iv) question entities, v) question paraphrases, and vi) sources used for answer retrieval. We believe that this additional metadata will make our resource useful beyond QA (in question rewriting and paraphrasing, for example). Most questions had exactly one correct answer, with the maximum being six. Table 3 summarizes notable properties of our study and benchmark, while Table 4 reports interesting representative examples. Note that HIT specific entries, like the payment per HIT, are given for the collection of conversations with five turns. The respective numbers were doubled for the collection of conversations with ten turns.

5 EXPERIMENTAL SETUP

We conduct all experiments on the CONVMIX benchmark. We split the part of CONVMIX with five turns, CONVMIX-5T, into train, development and test sets with the ratio 60:20:20. CONVMIX-10T is used only as a separate test set.

5.1 Heterogeneous sources

CONVINSE and all baselines run on the same data collections. As our knowledge base, we take the 31 January 2022 complete NTriples dump[1] of Wikidata, one of the largest and best curated KBs today. It consists of about 17B triples, consuming about 2 TB disk space. We access the KB via the recently proposed CLOCQ [7] interface, that reduces the memory overhead and efficiently returns KB-facts for queried entities. The text collection is chosen to be the English Wikipedia (April 2022). The benchmark-relevant subset of Wikipedia is comprised of the pages of entities detected via CLOCQ. Documents are split into sentences using spaCy[2]. All tables and infoboxes originating from the retrieved Wikipedia pages together constitute the respective answering sources. We parse Wikipedia tables using WikiTables[3], and concatenate entries in the obtained JSON-dictionary for verbalization. This procedure also includes conversions for tables with nested structure. Infoboxes are detected using Beautiful Soup[4].

5.2 Baselines

There are no prior works for ConvQA over heterogeneous sources. Thus, to compare the proposed CONVINSE pipeline with alternative choices, we adapt state-of-the-art question understanding (in this case, rewriting and resolution) methods from the IR and NLP literature. These serve as competitors for our SR generation phase. We then provide these baselines with exactly the same ERS and HA phases that CONVINSE has, to complete end-to-end QA pipelines.

Prepending history turns. Adding turns from the history to the beginning of the current question is still considered a simple yet tough-to-beat baseline in almost all ConvQA tasks [8, 20, 33, 51], and so we investigate the same here as well. Specifically, we consider four variants: i) add only the initial turn $\langle q^0, a^0 \rangle$, as it often establishes the topic of the conversation (**Prepend init**); ii) add only the previous turn $\langle q^{i-1}, a^{i-1} \rangle$, as it sets immediate context for the current information need (**Prepend prev**); iii) add both initial and previous turns (**Prepend init+prev**); and iv) add all turns $\{\langle q^t, a^t \rangle\}_{t=0}^{i-1}$ (**Prepend all**).

Question rewriting. We choose a very recent T5-based rewriting model [36]. The method is trained on the CANARD question rewriting benchmark [9]. The model is fine-tuned on CONVMIX, using the ⟨full history, current question⟩-pairs as input, and the respective completed questions (available in the benchmark) as the gold label.

Question resolution. We use QURETEC [52] as a question resolution baseline, treating context disambiguation as a term classification problem. A BERT-encoder is augmented with a term classification head, and predicts for each history term whether the word should be added to the current question. The same distant supervision strategy (Sec. 3.4) as used by CONVINSE is employed for generating annotated data for QURETEC (trained on CONVMIX).

5.3 Metrics

Measuring retrieval effectiveness. To evaluate retrieval quality, we use **answer presence** as our metric. It is a binary measure of

[1] https://dumps.wikimedia.org/wikidatawiki/entities/
[2] https://spacy.io/api/sentencizer
[3] https://github.com/bcicen/wikitables
[4] https://beautiful-soup-4.readthedocs.io/en/latest/

Table 4: Representative conversations in ConvMix. The types of sources which can be used for answering are given in brackets.

Turn	Books	Movies	Music	TV series	Soccer
q^0	*Who wrote Slaughterhouse-Five?*	*Who played Ron in the Harry Potter movies?*	*What was the last album recorded by the Beatles?*	*Who is the actor of Rick Grimes in The Walking Dead?*	*Which national team does Kylian Mbappé play soccer for?*
a^0	`Kurt Vonnegut` [KB, Text, Info]	`Rupert Grint` [KB, Text]	`Let It Be` [KB, Text, Table]	`Andrew Lincoln` [KB, Text, Table]	`France football team` [KB, Text, Info, Table]
q^1	*Which war is discussed in the book?*	*Who played Dumbledore?*	*Where was their last paying concert held?*	*What about Daryl Dixon?*	*How many goals did he score for his home country in 2018?*
a^1	`World War II` [KB, Text]	`R. Harris, M. Gambon` [Text, Table]	`Candlestick Park` [Text]	`Norman Reedus` [KB, Text, Table]	`9` [Table]
q^2	*What year was it's first film adaptation released?*	*What's the run time for all the movies combined?*	*What year did they break up?*	*did he also play in Saturday night live?*	*place of his birth?*
a^2	`1972` [KB, Text, Table, Info]	`1179 minutes` [KB, Info]	`1970` [KB, Text, Info]	`Yes` [Text]	`Paris` [KB, Text, Info]
q^3	*Who directed it?*	*Who was the production designer for the films?*	*Who was their manager?*	*whom did he play?*	*award he got in 2017?"*
a^3	`George Roy Hill` [KB, Text, Table, Info]	`Stuart Craig` [KB, Text, Table]	`Brian Epstein` [KB, Text]	`Daryl Dixon` [Text]	`Golden Boy` [KB, Table]
q^4	*What was the final film that he made?*	*Which movie did he win an award for working on in 1980?*	*What was their nickname?*	*production company of the series?*	*Who is the award conferred by?*
a^4	`Funny Farm` [KB, Text, Table]	`The Elephant Man` [Text]	`Fab Four` [KB, Text]	`NBC Studios` [KB, Text, Info]	`Tuttosport` [KB, Text, Info]

whether one of the gold answers is present in the top-e evidences ($e = 100$ in all experiments).

Measuring answering effectiveness. We use one of the standard metrics for factoid QA [40], **precision@1 (P@1)**, since FiD generates a unique answer. FiD generates plain strings as answers: evaluation for such strings with exact match for computing P@1 can often be problematic [45], since the correct answer could be expressed in different ways (e.g. {*"Eddard Stark"*, *"Ned Stark"*}, or {*"11 June 1969"*, *"11-06-1969"*, *"June 11, 1969"*}). Therefore, we try to normalize the answer to the KB, whenever possible, to allow for a fair comparison across systems. We search through ⟨entity mention, KB-item⟩ pairs coming from the evidence retrieval phase (Sec. 3.2). If there is a perfect match between the entity mention and the predicted answer string, we return the corresponding KB-item as the answer. If there is no such perfect match, we compute the Levenshtein distance [23] between the predicted answer and entity mentions from ⟨entity mention, KB-item⟩ pairs. The KB-item for the entity mention with the smallest edit distance is used as the answer in such cases. Note that such KB-items may also be normalized strings, dates, years, or numbers. These normalized KB-items are compared to the gold answers in the benchmark for the computation of P@1.

5.4 Configurations

Convinse uses a fine-tuned BART-base model for generating structured representations. The default hyperparameters from the Hugging Face library were used[5]. The maximum sequence length was set to 20, and early stopping was enabled. Three epochs were used during training, with a batch size of ten. 500 warmup steps with a weight decay of 0.01 turned out to be the most effective. Clocq, that was used for evidence retrieval inside Convinse and all baselines, has two parameters: k (number of disambiguations to consider for each question word), and p (a pruning threshold). In this paper, we set k = `Auto` (Clocq dynamically sets the number of disambiguations), and $p = 1000$, as these performed the best on our dev set. We used a Python implementation of BM25, with default parameters[6]. Code for FiD is publicly available[7]. FiD was trained on ConvMix for its use in this work. The number of input passages (e in this paper) was retained at 100 as in the original work [15]. The maximum length of an answer was set to 10 words. A learning rate of 5×10^{-5} really proved effective, with a weight decay of 0.01 and AdamW as the optimizer. All systems were trained on the ConvMix train set, and all hyperparameters were tuned on the dev set. All code was scripted using `Python`, making use of the popular PyTorch library[8]. Whenever a neural model was used, code was run on a GPU (single GPU, NVIDIA Quadro RTX 8000, 48 GB GDDR6).

6 RESULTS AND INSIGHTS

We run Convinse and the baselines on the test set of ConvMix (combined test sets of -5T and -10T), and report results in Tables 5 and 6. All metrics are micro-averaged over each conversational question that the systems handle, i.e. we measure the performance at a question-level. Throughout this section, best performing variants in columns are marked in **bold**. An asterisk (*) denotes statistical significance of Convinse over the nearest baseline. The McNemar's

[5] https://huggingface.co/facebook/bart-base

[6] https://pypi.org/project/rank-bm25/

[7] https://github.com/facebookresearch/FiD

[8] https://pytorch.org

Table 5: Comparison of answer presence within top-100 retrieved evidences after QU + ER on the ConvMix test set.

QU + ERS Method	KB	Text	Table	Info	KB+Text	KB+Table	KB+Info	Text+Table	Text+Info	Table+Info	All
Prepend init + BM25	0.380	0.298	0.120	0.331	0.415	0.386	0.406	0.297	0.329	0.331	0.419
Prepend prev + BM25	0.342	0.284	0.095	0.295	0.382	0.347	0.372	0.284	0.317	0.306	0.392
Prepend init+prev + BM25	0.440	0.366	0.137	0.420	0.486	0.443	0.479	0.359	0.407	0.409	0.495
Prepend all + BM25	0.431	**0.367**	**0.148**	**0.430**	0.476	0.437	0.468	**0.361**	**0.411**	**0.419**	0.482
Q. Resolution [52] + BM25 + FiD	0.414	0.311	0.115	0.329	0.445	0.419	0.437	0.312	0.356	0.341	0.453
Q. Rewriting [36] + BM25 + FiD	0.434	0.315	0.114	0.347	0.460	0.435	0.461	0.319	0.362	0.336	0.465
Convinse (Proposed)	**0.475***	0.352	0.117	0.369	**0.528***	**0.486***	**0.507***	0.353	0.408	0.381	**0.542***

Table 6: Comparison of end-to-end (QU + ERS + HA) answering performance (P@1) on the ConvMix test set.

QU + ERS + HA Method	KB	Text	Table	Info	KB+Text	KB+Table	KB+Info	Text+Table	Text+Info	Table+Info	All
Prepend init + BM25 + FiD	0.211	0.174	0.065	0.200	0.246	0.211	0.240	0.174	0.203	0.195	0.254
Prepend prev + BM25 + FiD	0.179	0.190	0.052	0.212	0.238	0.184	0.233	0.185	0.224	0.211	0.257
Prepend init+prev + BM25 + FiD	0.234	0.233	**0.074**	0.276	0.290	0.238	0.292	**0.229**	0.274	**0.272**	0.312
Prepend all + BM25 + FiD	0.230	**0.234**	**0.074**	**0.282**	0.290	0.238	0.282	0.224	0.267	0.265	0.300
Q. Resolution [52] + BM25 + FiD	0.222	0.190	0.063	0.219	0.261	0.227	0.257	0.185	0.241	0.221	0.282
Q. Rewriting [36] + BM25 + FiD	0.216	0.183	0.062	0.219	0.252	0.221	0.261	0.187	0.227	0.223	0.271
Convinse (Proposed)	**0.251***	0.220	0.062	0.258	**0.317***	**0.257***	**0.310***	0.220	**0.276**	0.253	**0.342***

test was performed for binary variables like P@1, and the paired t-test otherwise, with $p < 0.05$. All results are reported on the test set, except the ablation study, that was naturally conducted on the dev set. If not stated otherwise, we make use of *gold* answers for the previous turns in the conversation. For example, for answering q^4 we assume gold answers a^0-a^3 to be known.

6.1 Key findings

Convinse is viable for heterogeneous QA. The first and foremost takeaway is that our proposed pipeline is a viable approach for handling incomplete questions in conversations, given heterogeneous input sources. Convinse and most of the baselines consistently reach 40-50% on answer presence in their evidence pools after QU and ERS, with Convinse leading with 54% (see Table 5). These numbers set the upper bounds for end-to-end QA after the answering phase, which are in the ballpark of 25-34% (see Table 6).

It is noteworthy that even the basic prepending baselines, despite generating fairly verbose question formulations (15 − 26 words, Table 9) have very good answer presence in the top-100 evidences. This is largely due to the Clocq entity-based retrieval module, which turned to be quite robust even for long queries. Subsequently, BM25 scoring serves as a necessary filter, to prune the evidence sets. Sizes of these evidence sets varied from 2.3k evidences (Convinse) to 7.5k (Prepend all), which would have posed efficiency challenges to the final answering stage had the BM25 filter not been applied.

Convinse outperforms baselines. We observe that the SRs in Convinse are significantly more effective than question rewriting/resolution and the prepend baselines. SRs provide the right balance between conciseness and coverage. Conciseness (SRs are just 6-7 words on average, Table 9) helps effective retrieval with the IR model at the ERS stage, while expressive representation of the conversational context is crucial, too. The expressivity in the SRs helps achieve the highest answer presence after QU and ERS (0.542 in Table 5), outperforming all baselines by substantial margins. This advantage is carried forward to the answering stage and results in the best end-to-end QA performance (0.342 in Table 6).

The simple prepend baselines sometimes come surprisingly close, though, and perform better than the sophisticated rewriting or resolution methods. Nevertheless, it is noteworthy that SRs and question rewriting/resolution have clear advantages in terms of interpretability. When a user wonders what went wrong upon receiving an incorrect (if not outright embarrassing) answer, the system could show the CFG, SRs or the inferred complete questions as informative explanations – helping the user to better understand and cope with the system's limitations and strengths.

Altogether, the absolute P@1 numbers still leave huge room for improvement. This shows that ConvMix is indeed a challenging benchmark, with difficult user inputs where inferring the intent is hard, for example: *Which war is discussed in the book?* or *What was the final film he made?* (see Table 4 for more examples).

Combining heterogeneous sources helps. Another across-the-board observation is that combining knowledge sources is a strong asset for ConvQA. Consider the values in the *"All"* columns of Tables 5 and 6. These numbers are systematically and substantially higher than those in the columns for individual source types and even for pair-wise combinations.

To inspect whether these gains are not only from enhanced coverage, but also leverage *redundancy of information* across source types, we measured the average P@1 for all cases where the questions had 1, 2, 3, or 4 evidences (among the top-e) containing the gold answer in the output of the ERS stage. The P@1 improves steadily as the answer can be found several times, i.e. as information becomes redundant, being 0.428 for one, 0.658 for two, 0.713 for three, and 0.763 across instances with four answering evidences.

Convinse excels in realistic setup with *predicted answers*. The experiments so far are conducted assuming the gold answers for the previous turns in the conversation to be given. However, in a realistic setup, the ConvQA system would not know these answers. Therefore, we conducted an additional experiment, in which we used the *predicted answers* by the system for the previous turns, when generating the outputs of the QU phase. The results of this experiment are shown in Table 7. Convinse outperforms all methods significantly on conversations of length both five and

Table 7: P@1 of ConvQA systems when using the *predicted answers* for the previous turns in the ongoing conversation.

Method (+ BM25 + FiD)	ConvMix-5T (2800 questions)	ConvMix-10T (2000 questions)	ConvMix (4800 questions)
Prepend init	0.276	0.178	0.235
Prepend prev	0.190	0.123	0.162
Prepend init+prev	0.277	0.195	0.243
Prepend all	0.284	0.168	0.236
Q. Resolution [52]	0.283	0.188	0.243
Q. Rewriting [36]	0.258	0.168	0.221
Convinse	**0.321***	**0.217***	**0.278***

Table 8: P@1 of ConvQA systems over turns.

Method (+ BM25 + FiD)	ConvMix-5T 1	ConvMix-5T 2–5	ConvMix-10T 1	ConvMix-10T 2–4	ConvMix-10T 5–7	ConvMix-10T 8–10
Prepend init	0.388	0.271	0.320	0.235	0.185	0.128
Prepend prev	0.371	0.243	0.325	0.243	0.232	0.218
Prepend init+prev	0.370	0.315	**0.335**	**0.310**	0.308	0.245
Prepend all	0.375	0.323	0.305	0.278	0.270	0.195
Q. Resolution [52]	0.391	0.300	0.315	0.232	0.213	0.220
Q. Rewriting [36]	0.366	0.280	0.295	0.220	0.245	0.215
Convinse	**0.395**	**0.368***	0.320	0.298	**0.338***	**0.252**

Table 9: Average QU output length in words on test sets.

QU Method	ConvMix-5T	ConvMix-10T	ConvMix
Original	5.73	5.34	5.57
Prepend init	14.39	14.84	14.58
Prepend prev	12.23	12.17	12.21
Prepend init+prev	18.73	20.61	19.52
Prepend all	23.03	40.70	30.39
Q. Resolution [52]	8.24	8.54	8.36
Q. Rewriting [36]	9.07	9.04	9.06
Convinse	**6.43***	**6.53***	**6.48***

ten, even though it has never seen such 10-length conversations during training. Another observation is that the performance for the *"Prepend all"* baseline drops for ConvMix-10T, which might be due to the much longer QU outputs generated as the conversation continues longer. In general, the performance of all methods drops a bit (about 0.057 P@1 on average, c.f. Tables 6 and 7) when changing from gold answers to predicted answers.

6.2 In-depth analysis

Convinse is stable over turns. One striking finding when drilling down into the results, is that the performance of Convinse stays fairly stable as the conversation continues – see Table 8. In contrast, as one would naturally expect, the baselines exhibit systematic degradation from turn to turn, as it becomes harder to capture implicit cues about the user intents with progressing conversation (this was the main reason for collecting ConvMix-10T). This is most pronounced for the *"Prepend all"* model (diff. FiD models make T1 diff.). For most methods, there is a significant performance drop for the last three turns, indicating that further investigation of generalization to longer conversations might be worthwhile.

SRs are compact. SRs are indeed succinct representations of user intents, as indicated by Table 9 on question lengths. Also, for interpretability, SRs are an easy-to-inspect gist that are comprehensible

Table 10: Domain- and source-wise results (Convinse P@1).

Source	Books	Movies	Music	TV Series	Soccer
KB	0.255	0.273	0.219	0.245	0.264
Text	0.226	0.229	0.244	0.234	0.165
Table	0.021	0.106	0.052	0.058	0.073
Info	0.282	0.272	0.226	0.265	0.243
All	0.329	0.357	0.353	0.338	0.333

Table 11: Ablation study of the Convinse SR.

Method	Answer presence (ERS)	P@1 (HA)
Convinse	0.559	**0.371**
w/o Context entity slot	0.546	0.362
w/o Question entity slot	0.078	0.054
w/o Predicate slot	0.421	0.176
w/o Answer type slot	**0.572**	0.350
w/o Ordering	0.560	0.361

to a human user and at the same time being amenable for use with most standard IR models. Notably, for the more sophisticated models, the output length of the QU phase is almost stable on the two test sets with five turns and ten turns.

Convinse is stable over domains. Zooming into the results over the five thematic domains in ConvMix, we find that performance is relatively stable (working best for the movies domain) – see Table 10 columns. The same holds when contrasting this with distributions over source types (Table 10 rows). Infoboxes consistently provide knowledge easily harnessed, while tables turn out to be the trickiest to handle, with proper verbalizations being a likely issue [28, 50].

Slots in SR are vital. A systematic ablation study shows that each of the SR slots plays an important role – see Table 11. We blanked out the contents of the respective slots during retrieval, and proceeded with these weaker SRs to the answering phase. Question entities clearly are the most pivotal; answer types do not help much at retrieval time, but justify their importance during answering (a shared insight w.r.t. expected answer types in many QA models [40]). We also examined the effect of our proposed ordering of the SR slots (e.g. predicates first). As expected, there is hardly an effect during ERS (both Clocq and BM25 are word-order-agnostic), but ordering proves beneficial when generating answers from evidences using sequence-aware models like FiD.

Error analysis. Convinse cannot answer the question correctly 65.8% of the time, arising from three error cases: i) the evidence retriever cannot retrieve any answering evidence (42.4%), which can be due to the QU phase missing important cues in the conversation, failures within the evidence retriever, or the information sources not containing necessary information for answering, ii) the evidence retriever retrieves at least one answering evidence but none of these is among the top-e after ERS (27.2%), calling for more informed evidence-scoring mechanisms; or iii) FiD fails to detect the correct answer (30.4%), which indicates that more sophisticated reasoning over the set of evidences might be required.

Anecdotal results. For a better understanding of how our SRs look like when contrasted with rewritten and resolved questions, we provide a few representative examples in Table 12.

Table 12: Cases where only CONVINSE answered correctly.

Domain	Books
Original	country of origin?
Q. Res.	country of origin? expanse leviathan
Q. Rew.	What country was it taken from?
SR	⟨ Leviathan Wakes | Expanse | country origin | sovereign state ⟩
Domain	**Movies**
Original	What actor portrayed Magneto?
Q. Res.	What actor portrayed Magneto? x-men movie
Q. Rew.	Which actor played the character Magneto in the X-Men movie?
SR	⟨ X-Men | Magneto | actor portrayed | human ⟩
Domain	**Music**
Original	Her date of birth?
Q. Res.	Her date of birth? shakira
Q. Rew.	When was Shakira born?
SR	⟨ Waka Waka (This Time for Africa) | Shakira | date birth | date ⟩
Domain	**TV Series**
Original	What TV series did he first appear on?
Q. Res.	What TV series did he first appear on? tv show appeared the shows george
Q. Rew.	What TV series did George Grizzard first appear on?
SR	⟨ _ | George Grizzard | tv series first appear | television series ⟩
Domain	**Soccer**
Original	Who won?
Q. Res.	Who won? shakira
Q. Rew.	Who won the FIFA FIFA World Cup in France?
SR	⟨ _ | 2010 FIFA World Cup | won | national association football team ⟩

7 RELATED WORK

Conversational question answering. Some methods for ConvQA over text use a *hard history selection* to obtain a question-relevant subset of the conversation history [19, 31, 33]. However, the more prevalent approach is to construct a question-aware encoding with attention on the conversational history (*soft history selection*) [5, 13, 34]. This is then used by a neural reader for predicting the answer in the given passage. While this works well in ConvQA over text, the whole pipeline does not easily generalize to the heterogeneous setting, since this would require carefully designed mechanisms for representing inputs from different sources in a shared latent space. It may seem that hard history selection can be generalized to the heterogeneous setting more easily, but we found that such an approach hurts end-to-end answering performance in preliminary experiments. We employ attention over conversation history in the QU phase, which can be seen as a soft history selection.

In ConvQA over KBs, the conversational flow is typically captured in a context graph, either explicity [8] using iterative expansion, or implicitly [10, 20, 26] by maintaining a set of context entities and predicates. Conversational KB history can be incorporated using encodings [10, 26, 44] or heuristic subsets of the history [20], similar to ConvQA over text. All existing methods consider only one information source for answering conversations, which limits their answer coverage. Moreover, such works often adopt source-specific modeling of the conversational flow, like KB subgraphs, and cannot easily be extended to capture the conversational flow in a heterogeneous setting.

Question completion. One line of work that suggests a more general approach to the ConvQA problem aims to create a self-contained question from the incomplete utterance that can be answered by standalone QA systems [9, 36, 51, 52, 56]. Such approaches can either take the form of question rewriting, which generates a complete question from scratch [36, 51], or question resolution, which adds relevant terms from the conversational history to the question [52]. Question rewriting can entail unnecessary complexity since recent QA systems may not require the question to have perfect syntax or grammar [15, 16].

Question resolution [52] operates on the surface level, without aiming for grammatically correct questions, creating a completed question similar to a keyword query. However, a potential downside is that there is no structure in the completed form.

We show in experiments (Sec. 6.1) that both, completed forms via question rewriting, and question resolution, perform worse than our generated structured representations, which can be perceived as logical forms for matching with heterogeneous sources.

Heterogeneous answering. There has been work on combining text and KBs [30, 42, 47, 48, 54, 55] for conventional QA with single self-contained questions. Oğuz et al. [28] and Ma et al. [25] integrate tables as an additional type of information source. There has also been work on combining text and tables [3, 4, 57], or text, tables and images [12, 49] for answering full-fledged questions. However, these approaches cannot easily be extended to a conversational setting with incomplete follow-up questions.

8 CONCLUSIONS AND FUTURE WORK

We presented CONVINSE, the first end-to-end framework for answering conversational questions over heterogeneous sources, covering knowledge bases (KBs), text corpora, Web tables, and Wikipedia infoboxes in a seamless manner. Our approach marries ideas from different communities and their methodological paradigms. The learning of intent-explicit structured representations (SRs) is inspired by semantic frames in symbolic AI, but leaves its constituents in surface form without normalization. For retrieving relevant evidences from heterogeneous sources, we follow the line of IR thinking. The design of SRs is vital here: the SRs provide enough structure to enhance effectiveness, but are lightweight and can be directly applied to KBs, text, tables, and infoboxes alike. Finally, the last stage of extracting answers adopts the NLP paradigm of QA over text. We believe that extending conversational QA into a heterogeneous retrieval problem is a useful contribution to the community. The proposed CONVMIX benchmark, with thousands of real-user conversations, will help to further advance the state-of-the-art. One of the most promising avenues for future work involves developing more informed retrieval and answering phases, making direct use of the semantic roles in structured representations.

Acknowledgments. We thank Magdalena Kaiser from the MPI for Informatics for her useful inputs on this work. Part of this work was supported by the ERC Synergy Grant "imPACT" (No. 610150).

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a Web of open data. In *Sem. Web.*

[2] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *ACL*.

[3] Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Wang, and William W Cohen. 2021. Open question answering over tables and text. In *ICLR*.

[4] Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A Dataset of Multi-Hop Question Answering over Tabular and Textual Data. In *EMNLP*.

[5] Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2020. GraphFlow: Exploiting Conversation Flow with Graph Neural Networks for Conversational Machine Comprehension. In *IJCAI*.

[6] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *EMNLP*.

[7] Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2022. Beyond NED: Fast and Effective Search Space Reduction for Complex Question Answering over Knowledge Bases. In *WSDM*.

[8] Philipp Christmann, Rishiraj Saha Roy, Abdalghani Abujabal, Jyotsna Singh, and Gerhard Weikum. 2019. Look before you hop: Conversational question answering over knowledge graphs using judicious context expansion. In *CIKM*.

[9] Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can You Unpack That? Learning to Rewrite Questions-in-Context. In *EMNLP-IJCNLP*.

[10] Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2018. Dialog-to-action: conversational question answering over a large-scale knowledge base. In *NeurIPS*.

[11] Manish Gupta and Michael Bendersky. 2015. Information retrieval with verbose queries. In *SIGIR*.

[12] Darryl Hannan, Akshay Jain, and Mohit Bansal. 2020. ManyModalQA: Modality Disambiguation and QA over Diverse Inputs. In *AAAI*, Vol. 34.

[13] Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. 2018. FlowQA: Grasping Flow in History for Conversational Machine Comprehension. In *ICLR*.

[14] Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *ACL*.

[15] Gautier Izacard and Édouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *EACL*.

[16] Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Complex Temporal Question Answering on Knowledge Graphs. In *CIKM*.

[17] Hideaki Joko, Faegheh Hasibi, Krisztian Balog, and Arjen P de Vries. 2021. Conversational Entity Linking: Problem Definition and Datasets. In *SIGIR*.

[18] Endri Kacupaj, Joan Plepi, Kuldeep Singh, Harsh Thakkar, Jens Lehmann, and Maria Maleshkova. 2021. Conversational Question Answering over Knowledge Graphs with Transformer and Graph Attention Networks. In *EACL*.

[19] Magdalena Kaiser, Rishiraj Saha Roy, and Gerhard Weikum. 2020. Conversational Question Answering over Passages by Leveraging Word Proximity Networks. In *SIGIR*.

[20] Magdalena Kaiser, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Reinforcement Learning from Reformulations in Conversational Question Answering over Knowledge Graphs. In *SIGIR*.

[21] Vineet Kumar and Sachindra Joshi. 2017. Incomplete follow-up question resolution using retrieval based sequence to sequence learning. In *SIGIR*.

[22] Yunshi Lan and Jing Jiang. 2021. Modeling Transitions of Focal Entities for Conversational Knowledge Base Question Answering. In *ACL*.

[23] Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*.

[24] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL*.

[25] Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. 2022. Open Domain Question Answering with A Unified Knowledge Interface. In *ACL*.

[26] Pierre Marion, Paweł Krzysztof Nowak, and Francesco Piccinno. 2021. Structured Context and High-Coverage Grammar for Conversational Question Answering over Knowledge Graphs. In *EMNLP*.

[27] Thomas Mueller, Francesco Piccinno, Peter Shaw, Massimo Nicosia, and Yasemin Altun. 2019. Answering Conversational Questions on Structured Data without Logical Forms. In *EMNLP-IJCNLP*.

[28] Barlas Oğuz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2021. UniK-QA: Unified Representations of Structured and Unstructured Knowledge for Open-Domain Question Answering. *arXiv* (2021).

[29] Joan Plepi, Endri Kacupaj, Kuldeep Singh, Harsh Thakkar, and Jens Lehmann. 2021. Context Transformer with Stacked Pointer Networks for Conversational Question Answering over Knowledge Graphs. In *ESWC*.

[30] Soumajit Pramanik, Jesujoba Alabi, Rishiraj Saha Roy, and Gerhard Weikum. 2021. UNIQORN: Unified Question Answering over RDF Knowledge Graphs and Natural Language Text. *arXiv* (2021).

[31] Minghui Qiu, Xinjing Huang, Cen Chen, Feng Ji, Chen Qu, Wei Wei, Jun Huang, and Yin Zhang. 2021. Reinforced History Backtracking for Conversational Question Answering. In *AAAI*, Vol. 35.

[32] Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *SIGIR*.

[33] Chen Qu, Liu Yang, Minghui Qiu, W Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. BERT with history answer embedding for conversational question answering. In *SIGIR*.

[34] Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W Bruce Croft, and Mohit Iyyer. 2019. Attentive history selection for conversational question answering. In *CIKM*.

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR* (2020).

[36] Gonçalo Raposo, Rui Ribeiro, Bruno Martins, and Luísa Coheur. 2022. Question rewriting? Assessing its importance for conversational question answering. In *ECIR*.

[37] Siva Reddy, Danqi Chen, and Christopher Manning. 2019. CoQA: A conversational question answering challenge. *TACL* (2019).

[38] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* (2009).

[39] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, et al. 2021. Recipes for Building an Open-Domain Chatbot. In *EACL*.

[40] Rishiraj Saha Roy and Avishek Anand. 2021. *Question Answering for the Curated Web: Tasks and Methods in QA over Knowledge Bases and Text Collections.* Morgan & Claypool Publishers.

[41] Amrita Saha, Vardaan Pahuja, Mitesh Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In *AAAI*.

[42] Denis Savenkov and Eugene Agichtein. 2016. When a knowledge base is not enough: Question answering over knowledge bases with external text data. In *SIGIR*.

[43] Mingyue Shang, Tong Wang, Mihail Eric, Jiangning Chen, Jiyang Wang, Matthew Welch, Tiantong Deng, Akshay Grewal, Han Wang, Yue Liu, et al. 2021. Entity Resolution in Open-domain Conversations. In *NAACL*.

[44] Tao Shen, Xiubo Geng, QIN Tao, Daya Guo, Duyu Tang, Nan Duan, Guodong Long, and Daxin Jiang. 2019. Multi-Task Learning for Conversational Question Answering over a Large-Scale Knowledge Base. In *EMNLP-IJCNLP*.

[45] Chenglei Si, Chen Zhao, and Jordan Boyd-Graber. 2021. What's in a Name? Answer Equivalence For Open-Domain Question Answering. In *EMNLP*.

[46] Fabian Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A core of semantic knowledge. In *WWW*.

[47] Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and Text. In *EMNLP-IJCNLP*.

[48] Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text. In *EMNLP*.

[49] Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. MultiModalQA: Complex Question Answering over Text, Tables and Images. In *ICLR*.

[50] James Thorne, Majid Yazdani, Marzieh Saeidi, Fabrizio Silvestri, Sebastian Riedel, and Alon Halevy. 2020. Neural databases. *arXiv* (2020).

[51] Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021. Question rewriting for conversational question answering. In *WSDM*.

[52] Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. In *SIGIR*.

[53] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *CACM* (2014).

[54] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving Question Answering over Incomplete KBs with Knowledge-Aware Reader. In *ACL*.

[55] Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question Answering on Freebase via Relation Extraction and Textual Evidence. In *ACL*.

[56] Zihan Xu, Jiangang Zhu, Ling Geng, Yang Yang, Bojia Lin, and Daxin Jiang. 2020. Learning to Generate Reformulation Actions for Scalable Conversational Query Understanding. In *CIKM*.

[57] Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance. In *ACL*.

[58] David Ziegler, Abdalghani Abujabal, Rishiraj Saha Roy, and Gerhard Weikum. 2017. Efficiency-aware Answering of Compositional Questions using Answer Type Prediction. In *IJCNLP*.