A Context Mechanism for an Inference-based Question Answering System

A Context Mechanism for an Inference-based Question Answering System

Kwabena Nuamah[1], **Alan Bundy**[1], **Yantao Jia**[2],
[1] School of Informatics, University of Edinburgh, UK
[2] Poisson Lab, Distributed and Parallel Software Lab, Huawei Technologies Co., Ltd, China
k.nuamah@ed.ac.uk, a.bundy@ed.ac.uk, jiayantao@huawei.com

Abstract

Question answering (QA) techniques have predominantly focused on improving semantic parsing and information retrieval steps. Recent work has seen significant advances using deep neural networks to tackle these problems. However, not much emphasis has been put on incorporating contextual information into the QA process. More so in inference-based QA methods where, in addition to information retrieval (IR), there is the need for a non-deterministic composition of different operations on data from diverse sources.

In this paper, we formalise the idea of context and describe how it can be injected into a question answering process which, in addition to the retrieval of facts, requires the use of deductive, statistical and mathematical operations. We refer to this as an inference-based QA process. We show how this can improve the answers returned by constraining the key operations in the QA pipeline to contextual information. Context includes a user's environment and preferences such as how they might want to trade off accuracy over speed in the inference process. The latter informs the choice of inference methods that are used to answer the question. We explore these ideas using an inference-based QA framework that draws on structured data from diverse knowledge graphs, including commonsense knowledge found in sources such as Wikidata, decomposes questions recursively and combines retrieved facts using arithmetic and statistical operations, including making predictions. Experiments on questions based on Wikidata and the World Bank Open Data set validates the effectiveness of the proposed approach.

Our primary contribution is our approach to incorporating context information in the QA process, especially when inferring answers that cannot be found by traditional IR methods.

Introduction

Question answering using web data is a challenging problem given the huge amount of knowledge that is stored across multiple sources. Several of the techniques in this domain, such as (Fader, Zettlemoyer, and Etzioni 2014; Savenkov and Agichtein 2016), are evaluated only on questions whose answers are not influenced by a user's context. By context, we mean information about a user or the environment within which a user makes a query such as the device type and the current date/time.

This paper tackles the problem of incorporating context into inference-based question answering using web data, such as commonsense knowledge bases. A simple question such as "What is the population of London?" should result in different answers depending on which 'London' a user is referring to (it could be London in England, Ontario, etc.), as well as the date the question is posed (see figure 1). Additionally, when dealing with questions that go beyond the retrieval of pre-stored answers, such that it requires the combination of data from different sources and a dynamic composition of inference operations, there is the need to factor in details specific to the user: their preferences (e.g. prefer accuracy of answer over speed of response), as well as and their situational context (such as location and time). We define these details as the context within which the question is answered.

Furthermore, in order to answer a user's question, an automated question system often requires information that is not explicitly requested in the question. While it is reasonable to expect users to be explicit in their queries, logs from search engines show that users ask queries with a minimal set of information, and only refine it with more keywords when the answers returned fall outside their context.

In this paper, we describe how context can be injected into a question-answering process that leverages both automated reasoning techniques and statistical methods. This tackles the space of problems where simply looking up data from a knowledge graph or performing multi-hop graph traversal is not enough to answer a query. In particular, we use the FRANK (Functional Reasoning for Acquiring Novel Knowledge) (Bundy, Nuamah, and Lucas 2018) inference-based QA framework. Because FRANK does not just retrieve already stored knowledge, but draws inferences by combining knowledge from diverse sources, its context mechanism must represent more than the usual user preferences and current environment. For instance, as FRANK can make predictions, it needs to know the current date. Since it has a choice of inference mechanisms that trade-off speed against accuracy in different ways, it needs to know how quickly the answer is required and how accurate it needs to be. Our hypotheses are that (1) incorporating context in the reasoning process improves the quality of answers through personalization, and (2) hybrid reasoning methods ensure that assumptions made by the system based on user context are transparent to the user. Our contribution does not include the process of gathering context information since we assume that context information, obtained from sources such as user preferences and query history, is already available at the time of answering a user's question.

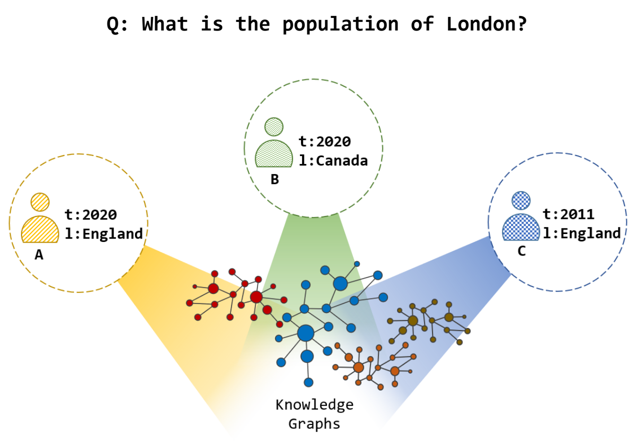


Figure 1: For three users with different context (in this case, time and location), the same question posed will focus on different aspects of the knowledge graphs and, hence, result in different answers. These contexts are not explicit in the question, and so the QA system needs to have a way of incorporating them in its inference process.

Background

We provide background to two concepts that are vital to our approach: the idea of 'context', and the inference-based QA within which we develop our ideas.

What is Context?

The concept of 'context' in question answering is used in different ways. For instance in (Datla et al. 2017) which tackles open-domain QA based on context-driven retrieval, context refers to the domain or category (from a pre-defined set or categories like (Dumais, Cutrell, and Chen 2001)) into which the question falls. Hence the processing of an open-domain question involves both parsing of the question and parsing of the context. The authors also add context to the answer by padding the sentence in the corpus containing the answer with the text before it and the text after it. Similar work includes (Jimmy et al. 2003) where the text surrounding the answer serves as a natural source of context. In (Park and Kim 2000), the context information refers to two types, i.e., the structural and temporal context. The structural context gives a preview extended from the current position to all other positions where the user asks a question such as "Where can I go from here?" to facilitate forward navigation. While the temporal context is related to the historically visited locations when the user asks "How did I get there" to help backward navigation.

Other systems try to figure out the context within which to answer a question by eliciting preferences from the user. For instance, in preference-based search with adaptive recommendations (Viappiani, Pu, and Faltings 2008), the authors developed a method that adapts suggestions to a user according to observations in the user's behaviour. This is done through example-critiquing with adaptive model-based suggestion, a kind of QA system presented as a conversational recommender system. That is, the system provides additional answers (recommendations) beyond what the search query would ordinarily return given its knowledge base. This is aimed at stimulating the user's preference expression in order to obtain an underlying model of the user's actual preferences. Suggestions are also adapted to the user's reaction to previously shown examples. This idea comes from traditional recommendation systems, e.g., (Wu et al. 2020) where context information includes the history of a user's check-in, temporal and spatial information.

Overview of the FRANK QA System

FRANK applies inference to structured knowledge sources on the Internet to derive estimates of novel knowledge and reliably assigns an uncertainty to it. It applies deductive, arithmetic and statistical reasoning to the results of information retrieval. FRANK's main focus is on estimating the values of numeric attributes, but it can also infer qualitative answers, e.g., the question "Which country will have the largest population in Africa in 2025?", it returns the name of the African country with the maximum estimated population.

FRANK recursively constructs an acyclic inference graph by decomposing a user query into sub-queries using decomposition rules (see Figure 2). At the leaves of this graph, sub-queries are answered by matching them against one of the many knowledge sources it consults, e.g. Wikidata (Vrandečić and Krötzsch 2014), Geonames(Wick and Vatant 2012), ConceptNet (Speer, Chin, and Havasi 2016), Google Knowledge Graph (Singhal 2012) and the World Bank Open Data (World Bank 2012) on country development indicators. For instance, to answer the question above, geospatial decomposition is used to create sub-queries about the population of each country in Africa. Then temporal decomposition is used to create sub-queries for census data for each of those countries. Regression is applied to this census data to create a function, that is then extrapolated to the year 2025. The country with the highest predicted population is then identified and returned as the answer.

FRANK adopts a Gaussian view of uncertainty (Nuamah and Bundy 2018). The mean of a Gaussian distribution is returned as the answer and the coefficient of variation (CoV: the standard deviation normalised by this mean) provides an error bar around it. A CoV for each knowledge sources is dynamically estimated based on its agreement with the other sources. These CoVs are inherited up the inference tree to be returned with the answer. The uncertainties associated with inference methods, e.g., regression, are incorporated into this process.

Additionally, in (Nuamah and Bundy 2020), the authors show how the combination of deductive and inductive reasoning used by the FRANK QA system makes possible the generation of explanations for users.

Terminology and Notation

We use the terminology and formal notation in (Bundy, Nuamah, and Lucas 2018).

- Let $\mathcal{A} = [\langle a_1, v_1\rangle, \ldots, \langle a_n, v_n\ \rangle]$ be an alist with attributes a_1 to a_n, each with their respective values v_1 to v_n. Attributes include, but are not limited to, $h =$ aggregation operation, $v =$ operation variable, $s =$ subject, $p =$ predicate, $o =$ object, $t =$ time, $u =$ uncertainty.
- $\mathcal{A}(t)$ is an association list (alist) of attribute/value pairs, where t is a distinguished value within it.
- A decomposition rule in FRANK for an alist **A**, decomposition type τ and aggregation operation h is defined as:

$$Decompose(\mathcal{A}, \tau) = [\mathcal{A}_j | 1 \leq j \leq m] \wedge \bigwedge_{j=1}^{m} \mathcal{A}_j(?\vec{x_j})$$
$$\implies \quad \mathcal{A}(\vec{h}(\epsilon ?\vec{x}_1.\ \mathcal{A}_1(?\vec{x}_1), \ldots, \epsilon ?\vec{x}_m.\ \mathcal{A}_m(?\vec{x}_m))/\vec{z})$$

- The Hilbert epsilon operation $\epsilon ?\vec{x}.\mathcal{A}(?\vec{x})$ returns the values of the vector of projection variables $?\vec{x}$ that make $\mathcal{A}(?\vec{x})$ true.
- $\mathcal{A}[a]$ represents the value of attribute a in alist $\mathcal{A}$.

Injecting Context into the Inference Process

The very large size of the web means that users are often only aware of portions of the knowledge available based on the individual context such as their location and the time. However, these contexts are often implied in their queries and, without making them explicit before answering these questions, answers that are either retrieved or computed by an automated QA system, could easily be wrong. To avoid this, we inject context information into the query and the inference mechanism. We formalise the idea of context, and show three ways in which context is injected into the QA process for FRANK.

Formalizing Context

Let $\mathcal{A}$ be an alist as defined in the section on terminology. We define a context attribute, c, as the tuple $\langle c_u, c_s\rangle$, with elements representing *user context* and *situational context respectively*. Each element of the context tuple is a set of attribute value-pairs.

Example:

$c = \langle$\{ gender: male, nationality: British \},
\{ place: United Kingdom,
device_type: phone,
datetime: 2020-04-30T13:20:00 \}$\rangle$

The *user* context contains user-specific information and preferences. The *situational* context contains information about (or from) the user's device (e.g. mobile phone) and other environmental information (e.g. location and time).

Unlike attributes such as 'subject' and 'object' in an alist, the context attribute cannot be a variable (or an unknown value). If the context attribute is used in an alist, a valid value must be provided.

Context versus Query Filters

While context and filters are similar, we treat them differently in FRANK.

- **Contexts** specify constraints over an entire alist.
- **Filters** specify constraints over an attribute of the alist.

Example:

country in Africa with the largest population

Suppose the alist of this query is $\mathcal{A}$ and the answer is $\epsilon ?x.\mathcal{A}(?x)$, where $?x$ is instantiated with the value of the country. Then $?x$ in the alist is defined with the following filter

{type: country, predicate: locatedIn, object: Africa}

However, a temporal context of this question by default is the current date. That is,

$$A[c] = \langle\{\ldots\}, \{\text{datetime : 2020-04-30T13:20:00}\}\rangle$$

Hence all population values retrieved will be constrained by this date since no explicit time is provided in the query.

Using Context

Context plays different roles in FRANK. Figure 3 illustrates how the three main components of the FRANK system are affected by context information.

As Constraints on Questions

An inference session is a set of all the operations that are performed to answer a question in FRANK. Context is used as a constraint on the question, restricting answers to those that satisfy the context for a given session. That is, for a question q (an alist), context c and an inference session I, FRANK finds an answer a such that

$$a = I(q|c)$$

Formally, if a query alist, $\mathcal{A}$, does not contain values for the temporal and geospatial attributes, then we inject those values in as context if similarly typed attributes exist in the context tuple.

Suppose $\mathcal{A}$ is an alist where c is the context attribute, t is the time attribute, $\mathcal{A}[t] = \emptyset$ and $\mathcal{A}[c] \neq \emptyset$. Then we perform the following substitution on the query alist:

$$\mathcal{A}(\xi(\mathcal{A}[c])/t)$$

where ξ is a function that extracts the appropriate context value.

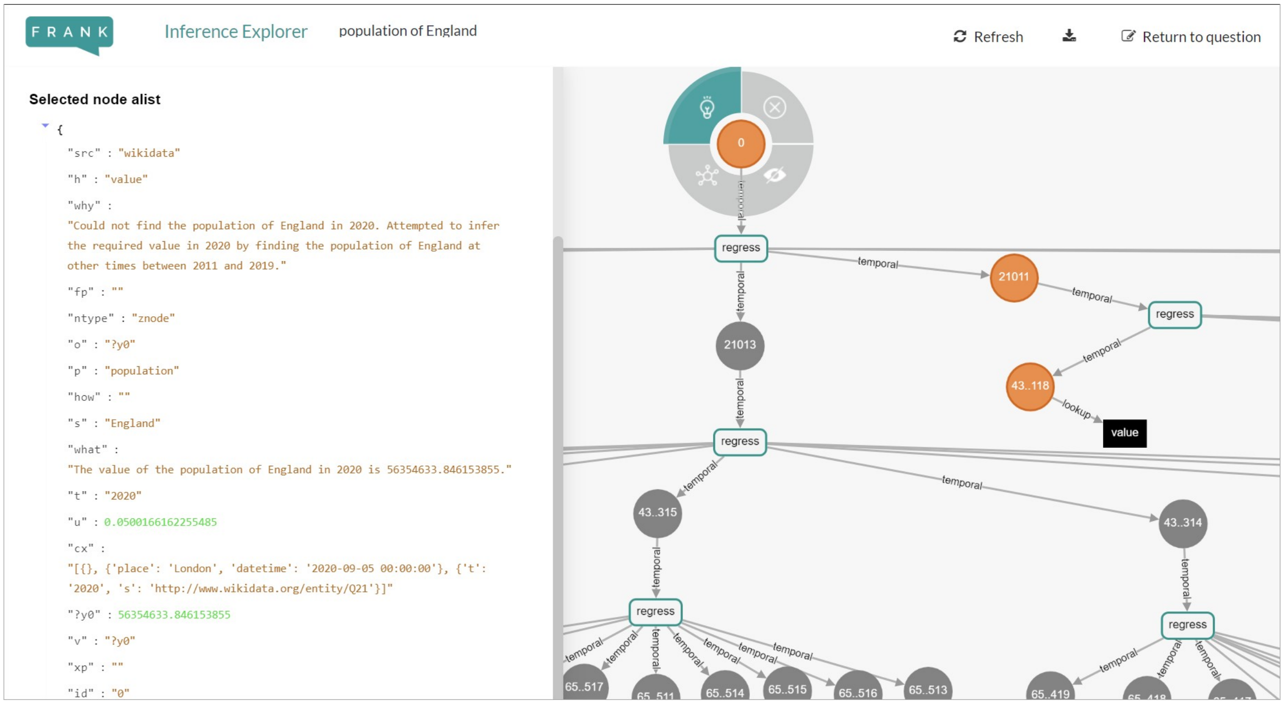


Figure 2: A screenshot of the user interface of FRANK (Inference Explorer) showing an inference graph used to answer the question "*What is the population of England*" with the context values {*"place": "London", "datetime": "2020-09-05 00:00:00"*}. The context attribute (*cx*) in the selected alist (shown in the side panel) indicates the context values injected into the alist for the time (*t*) and subject (*s*) attributes.

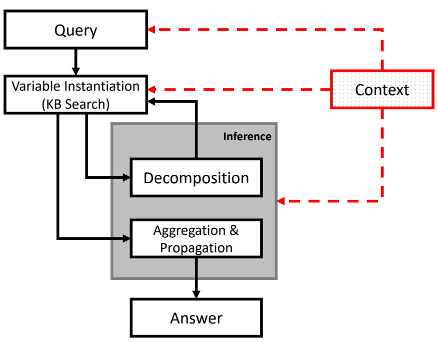


Figure 3: Context affects the query, inference and information retrieval components.

As Conditions for Decomposition and Aggregation

Context influences the decomposition of alists during inference. The presence of context attributes such as time and location can trigger temporal and geospatial decompositions (respectively) during inference. Thus, a function Δ that selects decomposition rules τ to apply to an alist can be conditioned on the context of the alist.

$$\tau = \Delta(\mathcal{A}|\mathcal{A}[c])$$

We can inject context into this function such that it influences the selection of the aggregation function, h, where

$$h = f(\tau, \mathcal{A}[c])$$

where f is a function that selects an aggregation function given the decomposition rule τ and the context of the alist $\mathcal{A}[c]$. This allows FRANK to apply user preferences among statistical operations, for instance, to select how aggregations are performed for a given decomposition. An example of this is seen in the selection of a prediction function given a user's preference for speed over accuracy. Given two regression function such as linear regression and Gaussian Process regression (both of which are available as aggregation operations in FRANK), it will opt for the former given that it is easier and faster to compute than the latter. However, the former has lower accuracy in some cases.

As Criteria for Information Retrieval

Context serves as a criteria for selecting knowledge sources and relevant facts within them. This is important when dealing with diverse knowledge sources and the ambiguities in search terms. For instance, in a question about restaurants in a town, if a user's context attribute specifies that s/he is a vegetarian, then FRANK will prioritise knowledge sources and facts containing information about vegetarian restaurants.

Concretely, for an alist $\mathcal{A}$, a query generator g and a knowledge source s, FRANK generates the native query a_s.

That is,

$$a_s = g(\mathcal{A})$$

Context adds additional filters, a_f, to the generated native query. We use the modified generator, g', that factors in the context of the alist as follows:

$$a_s \cup a_f = g'(\mathcal{A}, \mathcal{A}[c])$$

A typical example is the use of the date/time from the context value as a filter on query to retrieve data for that specific date.

Context Information is Defeasible

As described above, context allows one to add latent information (situational information and preferences) to the query in order to constrain answers to those that will be relevant to the user. However, a user may explicitly provide information in the query that should override the context. For instant, in the example, "What is the population of London?" context information on the current year and the user's current location will be injected into the query. However, if the user is explicit about the date, "What was the population of London in 2011", then the user provided date is used instead of injecting temporal context into the query.

Propagating Context

The context attribute in an alist is propagated to its child nodes during decomposition. This ensures that FRANK keeps track of all contextual information injected into the query alist and ensures that each alist in the inference graph is context-aware. Once an attribute in an alist is substituted with a context value, all child nodes inherit that assignment. For example, if no time is provided in the query and time is injected from context, then all child alists will use the 'assumed' time from context. Subsequent decompositions (temporal) of the alist using that time attribute proceeds in the same way as the case where time is explicit in the query, following the idea of defeasible contexts described above.

Evaluation

For convenience, we repeat our hypotheses here:

1. *Incorporating context in the reasoning process improves the quality of answers through personalization.*
2. *Hybrid reasoning methods ensure that assumptions made by the system based on user context are transparent to the user.*

At the time of writing this paper, we could not find any benchmark datasets for evaluating the above hypotheses. Hence, we created a synthetic test set of questions and contexts within which we answer them. Although this test set is small, it is representative of the diverse kinds of contextual information required for inference-based QA. However, the data to infer the answers to these questions are retrieved in real time from Wikidata and the World Bank knowledge sources on the web. We classified the World Bank as a more trusted knowledge source than Wikidata, which we consider as noisy given that it is based on crowd-sourced information from sources such as Wikipedia (Wikipedia 2001).

For context, we provided synthetic values for the user's location, the current data/time, and the user's preferences for speed and accuracy trade-offs. Context attributes and their valid values are listed in table 1. We considered 3 classes of questions, C1 (questions requiring quantitative answers), C2 (questions requiring qualitative answers) and C3 (questions requiring prediction) each with variations based on explicit information in the question, as well as their respective contexts. The various classes of questions are executed on FRANK without context and with context (FRANK+). We manually verified the correctness of answers by querying the data sets by hand and aggregating the retrieved data as required by the queries. The results are shown in table 2. Code and data are publicly available on GitHub and are attached as supplementary[1] files to this paper.

Discussion and Related Work

The results in table 2 are based on a variety of question types whose answers vary depending on contextual information. Due to the lack of an existing benchmark questions and datasets, we use a representative examples of questions and context, shown in table 2, and data from the Wikidata knowledge graph and the World Bank dataset. They validate our hypothesis that incorporating context during inference improves the quality of answers. In cases with vague questions, such as C2-0 and C2-1, answers can only be found by the injection of the location. We also observe that in the absence of context, the popularity of entities in a knowledge graph, (based, for instance, on the number of edges connected to a node) does not guarantee that the correct facts are retrieved. C1-0 highlights this where, without context, '*London*' is assumed to be the one in England based on its popularity in most knowledge bases.

Related techniques for question answering using structured data such as Question Answering over Linked Data (QALD) (Unger et al. 2014), (Fader, Zettlemoyer, and Etzioni 2014) and (Savenkov and Agichtein 2016) focus on translating questions in natural language to SPARQL queries (World Wide Web Consortium, W3C 2013) that are executed on a Linked Data (or other structured form) knowledge sources such as Wikidata. A limitation of such methods is that the absence of explicit filters in the question means that the generated SPARQL are no different from the manual Wikidata queries composed in table 2, and hence, result in incorrect answers.

In C3-0, for the query with a high accuracy user preference, a Gaussian process regression function, which provides a better fit to the data is used for prediction. This instance also retrieves and uses more data points for prediction than the instance without context. Figure 4 highlights the portions in the inference graphs

[1] https://qa-eval.s3-eu-west-1.amazonaws.com/aaai21ws.zip

Context Attribute	Values	Description
User Context		
nationality	plain text	the nationality (country name) of the user
accuracy	high, low	User's preference of accuracy.
speed	high, low	User's preference of speed.
trust	high, low, none	User's preference for the trusted knowledge sources.
Situational Context		
place	plain text	name of a town, country
datetime	date/time	the date/time within which to situate the question
device	computer, phone	type of computing device used to send the question

Table 1: A list of context attributes used in this evaluation and the values that can be provided for each. Default values are underlined. Context attributes are, however, not limited to these.

Question	Context	FRANK	FRANK+
C1-0. *What is the population of London?*	⟨{}, {"place": "Ohio", "datetime": "2025-09-01 12:00:00"}⟩	9138891	10060
C1-1. *What is the population of the capital?*	⟨{}, {"place": "Ghana", "device": "phone", "datetime": "2020-09-01 12:00:00"}⟩	–	1665000
C2-0. *Who is the prime minister?*	⟨{"nationality": "United Kingdom"}, {"place": "United Kingdom", "device": "phone", "datetime": "2020-09-01 12:00:00"}⟩	–	Boris Johnson
C2-1. *Who is the mayor?*	⟨{"nationality": "United Kingdom"}, {"place": "London", "device": "phone", "datetime": "2020-09-01 12:00:00"}⟩	–	Sadiq Khan
C2-2. *Who is the prime minister?*	⟨{"nationality": "United Kingdom"}, {"place": "London", "device": "phone", "datetime": "2020-09-01 12:00:00"}⟩	–	Sadiq Khan
C3-0. *What will be the GDP of Brazil in 2025?*	⟨{"accuracy":"high"}, {"device":"computer"}⟩	1.704e+12	2.051e+12
C3-1. *What will be the GDP of Brazil in 2025?*	⟨{"accuracy":"high","trust":"high"}, {"device":"computer"}⟩	1.704e+12	2.052e+12
C3-2. *What is the GDP?*	⟨{"accuracy":"high","trust":"high"}, {"place": "Brazil", "device":"computer", "datetime": "2025-09-01 12:00:00"}⟩	–	2.052e+12

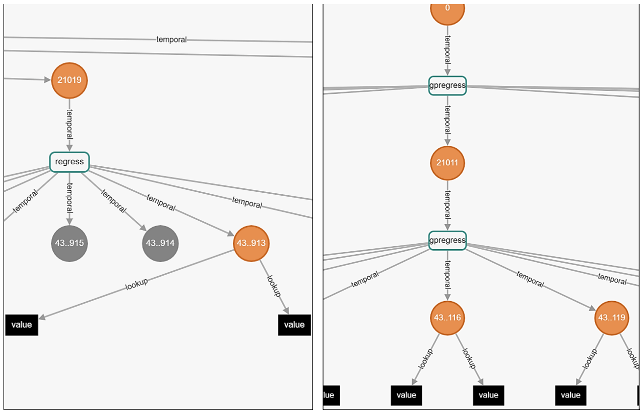
Table 2: Questions, context provided and the answers returned by the FRANK system with context (FRANK+) and without context (FRANK). Facts were retrieved from the Wikidata and the World Bank knowledge bases. Note that the situational context (e.g. place, device and date/time) are not provided by the user and so are not considered as part of the user's input when they pose the query. For questions C2-0, C2-1, C2-2, C1-1 and C3-2, FRANK is unable to answer the questions without context because they are missing information with which to reason and constrain information retrieval. Note that the answer to C2-2 for FRANK+ is wrong as explained in the discussion section.

where the regression operations differ. In C3-1, the context specifies a user preference for more trustworthy sources. Given our prior classification of the two knowledge sources, the query with the *trust* preference uses data from only the World Bank at its leaf nodes (see figure 5(a)). However, in the absence of any context information, data from both knowledge sources are used (figure 5(b)).

In C3-2, the FRANK system first injects Brazil as the subject and 2025 as the time attributes of the query. When the attempt to retrieve the GDP in 2025 fails, the temporal decomposition operation is applied in order to predict the 2025 GDP from past GDP values. The presence of *trust* preference forces FRANK to use data from the World Bank only. The selection of the prediction function is also similar to that of C3-0 and C3-1. However, in the instance without context, there is insufficient information to successfully answer the query.

A primary concern that is often raised about systems that personalize user experience is one of trust and transparency and the lack thereof. For instance, in online shopping systems, shoppers are recommended items primarily for the commercial interests of the vendor. This arises when the basis of the recommendation is not clear, and users cannot trace recommendations back to their specific preferences or to assumptions that the automated system has made with respects to a user's context. The benefit of using FRANK is that the entire inference trace (see figure 2) is available to the user: from context injection to decompositions of the query, variable instantiations and aggregations of data.

There are challenges with context and the granularity of the context provided. For instance, in the questions *"Who is the mayor?"* and "*Who is the*



(a) Without context

(b) With context ({“accuracy”:“high”}, {“place”: “Brazil”, “device”:“computer”})

Figure 4: FRANK inference graphs that answer the question “*What will be the GDP of Brazil in 2025?*” with and without context, respectively. The choice of inference operation for regression is influenced by the user preference for ‘accuracy’. Graph (a) without context uses the ‘regress’ operation (linear regression), while (b), with high accuracy preference, uses the ‘gpregress‘ (Gaussian Process regression) operation. Black leaf nodes represent data retrieved from a knowledge source and which are used to instantiate variables in the circular nodes.

(a) Without context

(b) With context ({“trust”:“high”,“accuracy”:“high”, {“device”:“computer”})

Figure 5: Inference graphs that question “*What will be the GDP of Brazil in 2025?*”. Graph (a) without context instantiates variable in leaf nodes with data from both Wikidata and the World Bank. Graph (b) having a preference for *trust* and *accuracy* restricts its leaf nodes to data from the World Bank given our prior classification of the World Bank as a more trustworthy data source.

prime minister?” the context provided could result in the correct or wrong answer. If the ‘place’ and ‘date’ contexts are set to *(*United Kingdom) and ‘2020’ respectively, the answer returned is *‘Boris Johnson’*, which is correct as at 2020. However, when the ‘place’ context is changed to ‘London’, the answer changes to ‘Sadiq Khan’, which is incorrect. This is due to how data is coarsely represented and stored in the knowledge graphs. In Wikidata, the properties ‘mayor’ and ‘prime minister’ are both represented by the same term labelled ‘head of government’ (https://www.wikidata.org/wiki/Property:P6). FRANK, therefore, has no way of distinguishing between the two properties in the two questions, and the fact retrieved is dependent on the location provided. The inclusion of more knowledge sources could remedy this.

A common limitation of neural networks and deep learning approaches to QA is that they are unable to deal with questions that require non-trivial statistical or arithmetic operations. Also, their inferences are opaque. (Li et al. 2017) also create a context-aware attention network mechanism in an encoder-decoder neural network model for interactive question answering domain. Although the types of questions targeted by the authors differ from those that we consider in this paper, we observe that our approach makes it a lot easier for users to see any assumptions (contextual information) used by the QA system to infer answers.

Finally, while the goal of incorporating context into QA involves the task of named entity disambiguation (Cucerzan 2007), our use of context in this work goes beyond that task.

Conclusion

We provide a mechanism for making an inference-based question answering system, such as FRANK, context-aware. Context is implemented as a data structure containing a tuple of attribute/value pairs. These attribute/value pairs can include user preferences, such as that accuracy is preferred over speed, to influence the choice decomposition rules that sacrifice speed for accuracy. They can also include information about the user and situation obtaining when the user asked a question, such as the user’s current location and the current time. We validated our method using a variety of questions with different contextual information and answering them with data from Wikidata and the World Bank data set on country development indicators.

We see, as future work, the potential to personalize the inference process even further by allowing users to interact with the inference graph and to change preferences or default contexts applied at different decision points (e.g. decomposition of alists). We will also explore adding the context and preference elicitation process into the reasoning mechanism.

Acknowledgments

The research reported in this paper was funded by Huawei Technologies Co., Ltd under project HO2017050001B8. We also thank the anonymous reviewers of this paper for their constructive feedback.

References

Bundy, A.; Nuamah, K.; and Lucas, C. 2018. Automated Reasoning in the Age of the Internet. In *13th International Conference on Artificial Intelligence and Symbolic Computation*, 3–18. Springer, Cham. Invited Talk.

Cucerzan, S. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 708–716.

Datla, V. V.; Arora, T. R.; Liu, J.; Adduru, V.; Hasan, S. A.; Lee, K.; Qadir, A.; Ling, Y.; Prakash, A.; and Farri, O. 2017. Open domain real-time question answering based on asynchronous multiperspective context-driven retrieval and neural paraphrasing. In *TREC*.

Dumais, S.; Cutrell, E.; and Chen, H. 2001. Optimizing search by showing results in context. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 277–284.

Fader, A.; Zettlemoyer, L.; and Etzioni, O. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1156–1165.

Jimmy, L.; Quan, D.; Sinha, V.; Bakshi, K.; Huynh, D.; Katz, B.; and Karger, D. R. 2003. What makes a good answer? The role of context in question answering. In *Proceedings of the Ninth IFIP TC13 International Conference on Human-Computer Interaction (INTERACT 2003)*, 25–32.

Li, H.; Min, M. R.; Ge, Y.; and Kadav, A. 2017. A context-aware attention network for interactive question answering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 927–935.

Nuamah, K.; and Bundy, A. 2018. Calculating error bars on inferences from web data. In *Proceedings of SAI Intelligent Systems Conference*, 618–640. Springer.

Nuamah, K.; and Bundy, A. 2020. Explainable Inference in the FRANK Query Answering System. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020)*.

Park, J.; and Kim, J. 2000. Effects of contextual navigation aids on browsing diverse Web system. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 257–264.

Savenkov, D.; and Agichtein, E. 2016. When a knowledge base is not enough: Question answering over knowledge bases with external text data. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 235–244.

Singhal, A. 2012. Introducing the knowledge graph: thing, not strings. *https://blog.google/products/search/introducing-knowledge-graph-things-not/, accessed on 17/12/2020* .

Speer, R.; Chin, J.; and Havasi, C. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. *arXiv preprint arXiv:1612.03975* .

Unger, C.; Forascu, C.; Lopez, V.; Ngomo, A.-C. N.; Cabrio, E.; Cimiano, P.; and Walter, S. 2014. Question answering over linked data (QALD-4).

Viappiani, P.; Pu, P.; and Faltings, B. 2008. Preference-based search with adaptive recommendations. *AI Communications* (2-3): 155–175.

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

Wick, M.; and Vatant, B. 2012. The geonames geographical database. *https://www.geonames.org, accessed on 17/12/2020* .

Wikipedia. 2001. Wikipedia, the free encyclopedia. *https://www.wikipedia.org/, accessed on 17/12/2020* .

World Bank. 2012. World Bank Open Data. *https://data.worldbank.org/, accessed on 17/12/2020* .

World Wide Web Consortium, W3C. 2013. SPARQL 1.1 overview URL https://www.w3.org/TR/sparql11-overview/.

Wu, Y.; Li, K.; Zhao, G.; and Xueming, Q. 2020. Personlized Long-and Short-term Preference Learning for Next POI Recommendation. *IEEE Transactions on Knowledge and Data Engineering* .