Follow-on Question Suggestion via Voice Hints for Voice Assistants

Besnik Fetahu[1], **Pedro Faustini**[2]*, **Giuseppe Castellucci**[1], **Anjie Fang**[1]
Oleg Rokhlenko[1], **Shervin Malmasi**[1]
Amazon, Seattle, WA, USA
Macquarie University, Sydney, NSW, Australia
pedro.arrudafaustini@hdr.mq.edu.au
{besnikf,giusecas,njfn,olegro,malmasi}@amazon.com

Abstract

The adoption of voice assistants like Alexa or Siri has grown rapidly, allowing users to instantly access information via voice search. Query suggestion is a standard feature of screen-based search experiences, allowing users to explore additional topics. However, this is not trivial to implement in voice-based settings. To enable this, we tackle the novel task of suggesting questions with compact and natural *voice hints* to allow users to ask follow-up questions. We define the task, ground it in syntactic theory and outline linguistic desiderata for spoken hints. We propose baselines and an approach using sequence-to-sequence Transformers to generate spoken hints from a list of questions. Using a new dataset of 6,681 input questions and human written hints, we evaluated the models with automatic metrics and human evaluation. Results show that a naive approach of concatenating suggested questions creates poor voice hints. Our approach, which applies a linguistically-motivated pretraining task was strongly preferred by humans for producing the most natural hints.

1 Introduction

Voice assistants, like Alexa or Google Assistant provide ubiquitous services through a variety of devices (e.g. smart speakers, phones, TVs, etc.). Users interact with voice assistants for different purposes (Rzepka, 2019; Lopatovska et al., 2019) such as question answering, e-commerce, or entertainment. With increasing adoption, user expectations also grow and related content recommendation is a valued feature (Tabassum et al., 2019).

The question of *how* to present proactive suggestions is an open one, and recent work has examined how content such as news articles can be recommended over voice (Sahijwani et al., 2020). Query and question recommendation (see Fig. 1 (a)) have become well-established research topics,

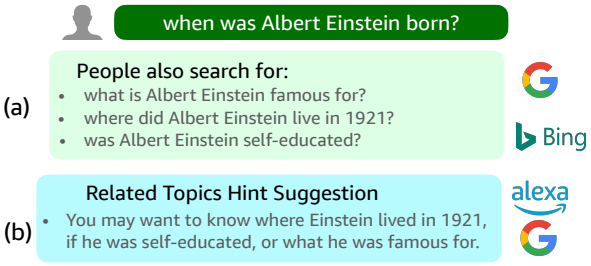

Figure 1: (a) Question suggestion in web search (available in Google/Bing) for a user question. (b) Proposed voice-based hint for the same questions users can ask as follow-on questions to a voice assistant such as Alexa.

and are well integrated in screen-based Web search experiences (i.e., those from Google/Bing). However, such functionality does not exist for voice-based systems. Suggestions enable highly useful exploratory search capabilities, and we aim to provide a similar experience over voice (see Fig. 1 (b)), where through a *follow-on hint* we suggest related topics they can ask about.

Contrary to suggestions on Web search, integrating recommendations in voice assistants poses unique challenges (Ma and Liu, 2020), such as (i) *modality*: voice lacks the advantages of visual interfaces used on the Web (e.g., showing a list), (ii) *transmitted information*: to ensure comprehension, the amount of transmitted information in an utterance is limited in terms of time and number of words, and (iii) *shape*: simply reading out a list of questions is not natural over voice.

We propose a new approach on ***how*** to deliver voice-based question suggestions using hints. We do not consider ***what*** to suggest as this is widely explored in existing work. Figure 1 provides an overview. For an input question, we assume the voice assistants can retrieve related questions[1] from which a suggestion hint is generated. Differently from questions recommendation in Web search

*Work done during an internship at Amazon.

[1]Related questions can be log based, or retrieved from a question bank using a similarity metric.

Findings of the Association for Computational Linguistics: EMNLP 2023, pages 310–325
December 6-10, 2023

(Fig. 1 (a)) where new related questions are listed, we aim to synthesize a natural utterance (Fig. 1 (b)) suggesting the same questions.

The hint does not contain questions, rather, it contains several subordinate clauses describing facts or knowledge that the user can ask about.

Our overarching contribution is a framework for generating voice-friendly hints. We begin with a grounded linguistic description of the task, outlining the characteristics of a good hint (e.g., cohesion, length), and the syntactic transformations needed to construct such utterances. Next, we frame the task as a *seq2seq* approach (Lewis et al., 2020a), where for an input question and its top–3 related questions, covering a diverse set of topics (unrelated topics to the initial question's topic), a voice hint is synthesized to meet the desiderata in Table 1. While newer large language models like ChatGPT are very capable in tasks like ours (Ouyang et al., 2022), generating real-time voice hints requires low latency (<150ms), which cannot be met by such models, hence the need for our task-specific model.

We create a dataset of voice-friendly hints, consisting of the triple: *initial question*, *related questions*, *follow-on hint*, in 9 different domains. We evaluate hint generation on our dataset by means of automated metrics and human evaluation studies. To summarize, our contributions are:

1. To our knowledge, we are the first to define the task of question suggestion via voice hints;
2. A large real-world hint generation dataset of $6,681$ instances, covering 9 domains, that will become publicly available;[2]
3. A seq2seq approach with task-specific training strategies for voice hint generation;
4. A detailed human evaluation protocol for evaluating different aspects of voice hints.

2 Linguistic Task and Background

To generate a spoken hint, our objective is to take a set of standalone questions (interrogative sentences), and convert them into a single sentence that informs the listener about the different pieces of information available. Figure 2 shows the overview of the linguistic tasks that are needed to be performed in order for a set of input questions to generate a voice-friendly hint.

Direct questions ("*can a dog eat peanuts?*") can be presented as an *indirect question* ("*Alice asked if dogs can eat peanuts.*") (Suñer, 1993). All direct questions can have an indirect equivalent, and the embedded clause of the indirect version is said to refer to the direct question (Puigdollers, 1999).

While both direct and indirect questions can be used to *ask*, when an indirect question's main clause reports information (e.g. "*I know . . .*"), their pragmatic purpose is to *provide* information (Puigdollers, 1999). Our task requires transforming independent questions into *subordinate clauses*, and then embedding them into a new sentence whose main verb is one of cognition or reporting, and takes the clauses as direct objects (Appendix A).

The most interesting syntactic transformation is that of converting a question to a dependent clause. In English, this can be done using content clauses (also known as noun clauses), which describe the inquired information in a main clause. The contents of a question can be framed as an *interrogative content clause* which represents the knowledge or entity that is being interrogated in the question.

The syntactic transformations needed to construct the content clause vary depending on the question type and its complexity. In general, these are the same changes used to generate reported or indirect speech, and can include subject-auxiliary inversion, changes in tense, and other lexical substitutions. This resulting subordinate clause is a syntactic unit which can be used as a direct object in a declarative sentence. Multiple subordinates can be combined to compose a single sentence.

Since these transformations between direct and reported speech are commonly used in English, representing our questions this way sounds very natural, and allows listeners to effortlessly convert any of the clauses into a fully formed question.

2.1 Characteristics of Natural Voice Hints

For a hint to be considered voice-friendly, i.e., sound like a natural spoken utterance, several aspects detailed in Table 1 must be fulfilled.

These desiderata are based on the principles of cohesion and coherence (Halliday and Hasan, 1976) and Gricean maxims of conversation (Grice, 1975). They ensure that constructed hints sound natural and are easy to comprehend. The characteristics were derived from our preliminary experiments on how English speakers create hints.

[2] https://github.com/bfetahu/spoken_hints/

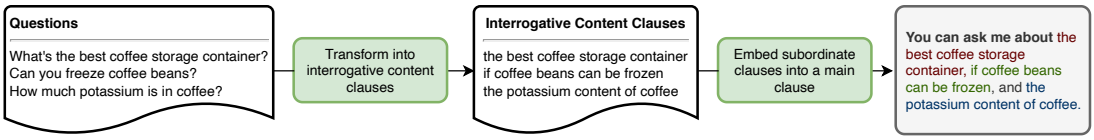

Figure 2: An overview of the linguistic processes for transforming a set of questions into a declarative statement.

Aspect	*Description*
Naturalness	The hint should reference facts or knowledge that can be asked.
Actionability	The main hint clause should be action oriented, e.g., *You can/may/might/could ask/also ask/be interested/also be interested*.
Information content	Questions must be converted to an interrogative content clause, just as they would be embedded in an indirect version of the same question.
Length	The hint utterance should not be exceedingly long in terms of words and listening time.
Coherence, Cohesion	• The hint is syntactically correct and semantically coherent. • Subordinate clauses Q_{rel} are connected through *coordinating conjunctions* (Webber et al., 2003). • Lexical repetitions, e.g., entity mentions, should be replaced by anaphora where appropriate.

Table 1: Linguistic properties of a natural spoken hint.

2.2 Voice-Friendly Hint Generation Task

The task is orchestrated as follows: (a) for an input question q, defined as a sequence of tokens $q = \{x_1, \ldots, x_n\}$ with a subject entity e; (b) a *follow-on hint* is generated from a set of top–k questions Q_{rel} about e (cf. §B.1), which cover *related topics* not covered in q. The generated hint does not contain explicit questions, but related topics that the user can ask about e.

The task is to learn the mapping function $\mathcal{F}(q, Q_{rel}) \rightarrow h$, which learns the transformation described in §2, i.e., mapping q and Q_{rel} into h, and meets the criteria in Table 1, with the most challenging tasks being:

Content Clause Generation: $\mathcal{F}$ must map Q_{rel} into subordinate content clauses in reported speech format (Lucy and Lucy, 1993), e.g. "*how many children does Cristiano Ronaldo have?*" $\rightarrow$ "*Alice asked about how many children Cristiano Ronaldo has*".[3]

Anaphora: $q_{rel} \in Q_{rel}$ typically contain variable surface forms of e, hence its repetitions in h are unnatural. $\mathcal{F}$ needs to learn how and when to replace e in q_{rel} with anaphoric expressions.

3 Method

3.1 Hints Generation Architecture

The function $\mathcal{F}(q, Q_{rel})$ corresponds to a generative Transformer model (Vaswani et al., 2017), which for an input question q and its top–k related questions Q_{rel} produces the hint h. We experiment with BART (Lewis et al., 2020a) and T5 models (Raffel et al., 2020).

We encode the input question q and its related questions Q_{rel} as follows:

$$\mathbf{s} := \{q, \texttt{[SEP]}, q^1_{rel}, \texttt{[SEP]}, q^2_{rel}, \texttt{[SEP]}, q^3_{rel}\}$$

Representation $\mathbf{s}$ is used by the decoder to generate the hint h. During the training of $\mathcal{F}$, the model learns to map the input $\mathbf{s}$ to h through operations, such as: (i) using *start patterns*, serving as the main clause of h, (ii) converting Q_{rel} into subordinate clauses, (iii) avoid entity repetitions through *anaphora*, and, (iv) ensuring hint *coherence* by connecting the subordinate clauses.

While seq2seq models show remarkable natural language generation capabilities, fine-tuning them for all the criteria above is challenging, resulting in hints that are *incoherent* and *unnatural* (cf. §6). Hence, we propose a pretraining strategy to overcome such challenges.

3.2 Reported Speech Pretraining

A key aspect of ensuring that h is correct is creating the subordinate clauses from Q_{rel}, as they would be in reported speech (RS) format. Generating RS requires the model $\mathcal{F}$ to perform the most significant rewrite operations, including performing the subordinate clause syntax change, such as verb tense, pronoun and word order alterations.

We propose a two stage training strategy, where: (1) we pretrain $\mathcal{F}$ in converting individual questions into their RS format, and finally (2) fine-tune $\mathcal{F}$ for

[3] Pronoun and verb tense changes are required.

the full hint generation task, ensuring that the hint is coherent and there are no repetitions.

RS Pre-training: For pretraining, we change the input of the model to be a single question and output its reported speech equivalent. This is the same as generating a hint from a single question, with the only difference that there is no initial input question q to the model. Constraining the pretraining phase to a single question it allows the model to learn how to perform all the necessary rewrite operations for converting a question to RS format.

Fine-Tuning: Next, we fine-tune the pretrained model to learn to convert the input s (containing the q and its related questions Q_{rel}) into a hint. By this stage, the model already has pretrained knowledge for converting questions into RS, and can focus on learning to use anaphora, conjunctions, etc.

4 VoFH – Voice-Friendly Question Suggestion via Hints Dataset

We now describe the process of generating a new voice-friendly hints dataset.[4] We first construct tuples of input questions and related questions $\langle q, Q_{rel} \rangle$. We then annotate spoken hints for each tuple, creating a dataset of $6,681$ samples composed of the triples $Q = \{\langle q, Q_{rel}, h \rangle_i \ldots\}$. The input question Q and related questions Q_{rel} datasets are described in Appendix B.

4.1 Hint Annotation

Using the question bank Q and the related questions Q_{rel}, we collect spoken hints for suggesting related questions. From a random sample of $6,681$ input questions and their related questions Q_{rel}, we create two *disjoint* hint sets, namely:

1. **SINGLE-HINTS**: follow-on hints generated from only a single related question, and
2. **MULTI-HINTS**: follow-on hints generated from multiple distinct related questions.

4.1.1 Hint Generation Guidelines

Based on the intuitions from §2, we provide guidelines to annotators to create voice-friendly hints. For the tuple $\langle q, Q_{rel} \rangle$, annotators follow the steps below to write a hint. Details about the crowdsourcing setup, worker payment and hint generation quality are provided in §C.

[4] https://github.com/bfetahu/spoken_hints/

Step 1. Annotators are asked to start the hint with one of the provided *start patterns* (cf. Table 1).

Step 2.a. The questions in Q_{rel} are converted into RS format. RS conversion templates are provided to annotators:

- "***Q**: Did Samuel Adams plan the Boston Tea Party?*"
- "*Bob wants to know if Samuel Adams planned the Boston Tea Party?*"

Step 2.b. For MULTI-HINTS, annotators need to avoid *repetitions* and replace them with *anaphora* where necessary. Next, subordinate clauses from Q_{rel} are connected with the correct *conjunctive discourse markers*, e.g. given two questions: "*Did Samuel Adams plan the Boston Tea Party?*" and "*What was the role of Samuel Adams in the American Revolution?*", the example below shows the correct use of anaphora and conjunctions.

- "*You may also want to know if Sam Adams planned the Boston Tea party, or/and about his role in the American Revolution.*"

4.1.2 Data Collection

We create two disjoint subsets: SINGLE-HINTS (hints from a single related question) and MULTI-HINTS (hints for multiple questions). Table 2 shows a detailed overview of our collected dataset.

	SINGLE-HINTS		MULTI-HINTS		
domain	#	ratio	#	ratio ($\lvert Q_{rel}\rvert=2$)	ratio ($\lvert Q_{rel}\rvert=3$)
Animal	2,806	-	2,780	24.1%	75.9%
Place	2,105	-	1,369	3.2%	96.8%
Technology	928	-	897	5.4%	96.4%
Politician	956	-	766	8.2%	91.8%
Food	537	-	329	59.3%	40.7%
Athlete	352	-	209	16.3%	83.7%
Wearables	180	-	177	-	100%
Holiday	60	-	54	5.6%	94.4%
total	7,932	-	6,581	1,132	5,449

Table 2: Follow-on voice friendly hints data statistics for SINGLE-HINTS and MULTI-HINTS, respectively.

Our main focus is in generating hints from top–3 related questions Q_{rel}, however, to ensure data diversity, we also collect hints constructed from the top–1 and top–2 related questions. This increases the utility of our dataset, as hint generation approaches must ensure hint coherence with a variable number of related questions.

As shown in Table 2, we collect a larger sample of SINGLE-HINTS. Most of it is used for pretraining of our hint generation approaches.

5 Experimental Setup

We evaluate different models and assess hint quality using automatic and human evaluation metrics. Table 3 shows the statistics about the dataset used in our experiments.

5.1 Datasets

Pre-training RS Dataset. SINGLE-HINTS, which we refer to as reported speech data, is used for pretraining the hint generation approaches (cf. §3.2).
Hint Generation Dataset. For the main task of hint generation, we randomly sample questions from Table 2, and split with 60%/10%/30% for training, development, and testing. Majority of the hints are MULTI-HINTS, with 81% generated from three questions, 17% with two questions, and the remaining 2% are SINGLE-HINTS.

	train	dev	test
RS pretraining	4,262	1,831	-
Hint Generation	4,008	668	2,005

Table 3: Pretraining and training hint generation datasets, sampled randomly from Table 2.

5.2 Baselines and Approaches

For all Transformer-based approaches, we experimented with BART-BASE (Lewis et al., 2020b) and T5-BASE (Raffel et al., 2020) models. Details about model training, along with the hyperparameter setup, are provided in Appendix D.

Template Baseline – TB. Hints are constructed based on manually defined templates, by first choosing a start pattern (cf. Table 1) and then concatenating question from Q_{rel} with an *"or"*.

Reported Speech Baseline – RSB. We train a seq2seq model on SINGLE-HINTS only, where questions are first converted into their RS format, then using TB different questions are concatenated into a hint. RSB represents an ablation of PTG (only the *pretraining stage*).
Direct Hint Generation – DHG. This represents our approach without pretraining. The limitation of DHG are that it has to jointly learn all aspects of constructing voice-friendly hints, which may lead to cases where subordinate clauses are not in the desired syntax, or the hint lacks coherence.

Hint Generation with RS Pretraining – PTG. This represents our final approach with pretraining on the RS task. Breaking down the training into two stages, PTG first learns RS rewriting, then it learns to avoid *repetitions* and ensure hint coherence and right order of subordinate clauses.

5.3 Evaluation Metrics

Evaluating hint quality is not trivial. Given the task novelty and the lack of metrics that capture voice-friendliness, we opt for a combination of automatic metrics and human evaluations.

5.3.1 Automated Metrics

To assess the closeness of the generated hints with respect to their ground-truth counterparts generated by human annotators, we use BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and F1-BertScore (Zhang et al., 2020a). BLEU captures the accuracy in terms of the *n-grams*, whereas ROUGE quantifies coverage of the ground-truth *n-grams* in the generated hint. Finally, BERTScore computes the semantic similarity between two hints, thus accounting for the use of equivalent phrases or synonyms in the hints.

5.3.2 Human Evaluation

Automated metrics are good quality indicators, but they do not capture hint voice-friendliness. We devise a set of human evaluations which judge the correctness and naturalness of a hint. For a realistic evaluation, all human studies[5] are performed in a voice modality.[6] We consider the following studies: (i) syntactic correctness, (ii) input question coverage, (iii) hint pairwise comparison from different approaches, and (iv) question retention.
Syntactic Correctness. Annotators judge whether a hint is *syntactically correct*, and if the hint uses *idiomatic* expressions in English.
Question Coverage. Given a hint h and Q_{rel}, annotators assess if h covers all questions in Q_{rel}.
Pairwise Hint Comparison. For two generated hints h_a and h_b from the same set of questions Q_{rel} and two different approaches, annotators choose their preferred hint. To reduce any positional bias, hints are ordered randomly. Finally, for each comparison we collect three judgements, achieving an inter-annotator absolute agreement rate of 0.77.
Question Retention. We consider retention of a hint's information in memory as a proxy for its

[5] *Question coverage* and *syntactic correctness* are done in text, as the annotators need to map questions to hints.

[6] We use Amazon's AWS Polly text-to-speech service to convert the generated texts into spoken utterances.

simplicity and comprehensibility. Hints cannot be considered actionable if listeners cannot remember them. We assess how well annotators can recall the conveyed information in a hint and ask questions about *one* of the conveyed topics in the hint. To emulate interaction with a voice assistant, annotators first listen to the hint, after which a mandatory 5 seconds pause is enforced. Then they need to choose the correct question covered in h from a set of four questions shown to them. Only one of the questions is present in h. We select the three distractor questions, one chosen at random, and the other two are either relevant to the entity and topic covered by h, or the entity only.

6 Evaluation on Automated Metrics

Table 4 shows the performance measured on the automated metrics for the different approaches.

Baseline Performance: TB achieves the lowest scores across all metrics (except for BERTScore). This is expected, since concatenated questions are compared w.r.t the ground-truth hints, written by annotators. RSB obtains a consistent improvement across all metrics. It rewrites individual questions into content clauses, which then are concatenated using the conjunction "*or*". However, RSB does not reduce lexical repetition via anaphora, and simple concatenation results in lower coherence. Overall TB and RSB, achieve low scores as expected. More insights are provided by the human evaluation studies, which capture hint voice friendliness.

Approach Performance: Our approaches, DHG and PTG, show a consistent improvement over TB and RSB across all automated metrics. This is intuitive since they are optimized to generate hints.

Comparing PTG and DHG in Table 4, we note a *significant* improvement in terms of BLEU scores due to the pretraining phase. This follows our intuition that pretraining helps PTG to convert questions into subordinate clauses, a key aspect of natural hints. In the fine-tuning stage, PTG can already reasonably convert questions into RS syntax, and thus can focus on reducing lexical redundancy, resulting in more coherent hints. While PTG employs multi-stage training, in DHG all operations are learned end-to-end. This represents a complex training regime, requiring optimization of several rewrite tasks, listed in Table 1.

The difference in performance between PTG and DHG, demonstrates that for complex rewriting tasks, end-to-end training may be sub-optimal. Decomposing the problem into specific pretraining subtasks before fine-tuning in an end-to-end manner yields significant improvements. Similar finding are reported in (Arora et al., 2021).

For ROUGE metrics, only PTG-T5 obtains significantly better results than DHG-T5 for ROUGE1. For the rest, although PTG has higher ROUGE scores, the differences are not significant. Finally, for BERTScore the differences are significant between PTG-BART over DHG-BART.

Robustness: Table 5 shows an out-of-domain evaluation, for PTG-BART and DHG-BART. This assesses model robustness on unseen domains during training. Comparing the performance of PTG-BART and DHG-BART, we note that across all domains, pretraining in PTG allows the model to achieve significantly better results than DHG. Only for `Wearables` do we not observe any significant difference. This can be attributed to the smaller test set size, with only 45 instances. Additional evaluation results are shown in Appendix E.1.

7 Human Evaluation Studies

7.1 Syntactic Correctness and Coverage

Table 6 shows the performance of the different models in terms of input questions coverage and the syntactic correctness. For a random sample of 500 hints and the corresponding Q_{rel}, we assess if all input questions are present in a generated hint, and if the hint is syntactically correct.

Syntactic Correctness. Table 6 shows a consistent pattern in terms of syntactic correctness: the baseline RSB and PTG-BART have the highest portion of syntactically correct hints as judged by the annotators, with 92% and 91%, respectively. Generating hints from multiple questions is not trivial, as it involves syntactic and stylistic changes in h, allowing room for errors for generative models, especially in terms of syntactic errors.

The high RSB and PTG-BART scores can be interpreted as follows. RSB is trained on SINGLE-HINTS, which does a syntactic conversion of the input question into their RS format, and through simple rules concatenates content clauses. This allows the model to generate hints that are syntactically correct in 92% of the cases. Similarly, PTG-BART, that is pretrained on SINGLE-HINTS, has the same capabilities as RSB, and generates in 91% of the cases syntactically correct hints. How-

	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE1	ROUGE2	ROUGE3	ROUGE4	BERTScore
TB	0.509	0.401	0.323	0.254	0.713	0.488	0.358	0.278	0.536
RSB	0.519	0.415	0.341	0.274	0.717	0.501	0.375	0.292	0.494
DHG-T5	0.616	0.510	0.428	0.358	0.728	0.525	0.400	0.320	0.632
DHG-BART	0.616	0.509	0.427	0.359	0.734	0.529	0.402	0.322	0.628
PTG-T5	0.629	0.524	0.442	0.373	0.739	0.534	0.410	0.329	**0.643**
PTG-BART	**0.630**	**0.527**	**0.446**	**0.378**	**0.742**	**0.539**	**0.413**	**0.333**	0.642

Table 4: PTG-BART achieves the highest performance across nearly all evaluation metrics, obtaining statistically highly significant results ($p < 0.01$) against all its counterparts. Appendix F shows a comparison between PTG-BART and ChatGPT.

Domain	BLEU1		BLEU2		BLEU3		BLEU4		ROUGE1		ROUGE2		ROUGE3		ROUGE4		BertScore	
	DHG	PTG	DHG	PTG	DHG	PTG	DHG	PTG	DHG	PTG	DHG	PTG	DHG	PTG	DHG	PTG	DHG	PTG
Animal	0.604	**0.618**‡	0.498	**0.511**‡	0.413	**0.426**‡	0.344	**0.356**‡	0.717	**0.732**‡	0.509	**0.521**‡	0.377	**0.388**‡	0.298	**0.308**‡	0.634	**0.642**‡
Athlete	0.563	**0.573**†	0.440	**0.450**	0.345	**0.356**	0.261	**0.275**	0.696	**0.705**	0.478	**0.488**	0.347	**0.357**	0.266	**0.276**	**0.587**	0.586
Food	0.636	**0.643**‡	0.540	**0.550**‡	0.466	**0.478**‡	0.401	**0.413**‡	0.749	**0.752**	0.561	**0.568**†	0.443	**0.453**‡	0.364	**0.374**†	0.686	**0.695**‡
Holiday	0.572	**0.586**	0.459	**0.472**	0.369	**0.387**	0.289	**0.320**†	0.717	**0.727**	**0.511**	0.504	**0.399**	0.395	**0.327**	0.325	0.592	**0.608**†
Places	0.602	**0.619**‡	0.493	**0.512**‡	0.411	**0.431**‡	0.341	**0.360**‡	0.754	**0.760**†	0.550	**0.554**	0.427	**0.429**	0.342	**0.343**	0.599	**0.617**‡
Politician	0.561	**0.579**‡	0.440	**0.459**‡	0.348	**0.370**‡	0.277	**0.298**‡	0.715	**0.719**†	0.491	**0.494**	0.360	**0.362**	0.283	**0.284**	0.552	**0.573**‡
Technology	0.598	**0.608**‡	0.484	**0.497**‡	0.398	**0.411**‡	0.329	**0.340**‡	0.738	**0.750**‡	0.537	**0.550**‡	0.418	**0.427**‡	0.342	**0.348**†	0.576	**0.589**‡
Wearables	0.658	**0.659**	0.572	**0.573**	0.505	**0.506**	0.447	**0.448**	0.794	**0.795**	**0.620**	0.619	**0.502**	0.499	**0.425**	0.424	0.626	**0.628**

Table 5: Comparison on out-of-domain hint generation performance for PTG-BART and DHG-BART. With † are denoted statistically significant ($p < 0.05$) and with ‡ highly significant results ($p < 0.01$).

Approach	**Syntactic Correctness**	**Question Coverage**
TB	449 (89.8%)	500 (100%)
RSB	461 (92.2%)	484 (96.8%)
DHG-T5	434 (86.8%)	464 (92.8%)
DHG-BART	428 (85.6%)	466 (93.2%)
PTG-T5	431 (86.2%)	485 (97.0%)‡
PTG-BART	455 (91.0%)‡	485 (97.0%)‡

Table 6: Syntactic correctness and input question coverage results. Significant differences between PTG and DHG are marked with ‡. No significant difference exists between DHG and RSB (as per binomial test of proportions).

ever, contrary to RSB, PTG-BART additionally fine-tunes for voice-friendliness, which ensure hint coherence and redundancy. While RSB generates syntactic hints, its hints are far less natural than those of PTG-BART (cf. §7.2).

Coverage. For question coverage, we note that the PTG approaches achieve the highest coverage among the learning based approaches, with 97% of the hints covering all the questions. TB has perfect coverage, given that its hints are generated by simply concatenating the input questions.

Finally, the DHG approaches have the lowest coverage, with 92.8% of hints having full coverage. This indicates that end-to-end learning of all hint generation tasks is challenging.

7.2 Pairwise Hint Comparison

Here we measure which approaches generate hints that are considered more natural by humans. As DHG has consistently lower performance than PTG, we only compare PTG-BART, RSB, and TB. To understand the naturalness of the hints in a spoken format, they are converted to audio. After listening to the hints, annotators judge which hint they find more *natural* and *easier to understand*. To avoid positional bias, the order in which the hints are played is randomized.

Table 7 shows the pairwise comparisons the different models. We run the comparison on the 441 hints that were judged to be syntactically correct in Table 6. This is done to avoid any bias stemming from syntactically incorrect hints.

Comparison	**PTG-BART chosen**	**Baseline chosen**
PTG-BART vs. TB	300 (68%)	141 (32%)
PTG-BART vs. RSB	267 (61%)	174 (39%)

Table 7: Pairwise hint comparison. PTG-BART hints are significantly ($p < 0.01$, as per binomial test of proportions) considered to be more voice-friendly than the baselines hints.

In both comparisons, PTG-BART produces more natural hints than baselines. Against TB, it is preferred in 68% of the cases, whereas against RSB, this is in 60% of the cases. Both results represent statistically highly significant differences (as per Wilcoxon's signed-rank test). Table 8 shows the pairwise comparison at the domain level, for all domains PTG-BART is preferred by human annotators as having more voice friendly hints.

Domain	PTG-BART vs. TB	PTG-BART vs. RSB
Animal	135/69	122/82
Places	59/20	43/36
Tech	39/20	37/22
Politician	30/15	30/15
Food	14/10	15/9
Athlete	10/6	11/5
Wearables	10/1	7/4
Holiday	3/0	2/1

Table 8: Per-domain pairwise hint comparison results.

7.3 Question Retention Evaluation

In the final human evaluation from §5.3.2, we measure how actionable the generated hints are. Beyond being natural or correct, the main aim of generating follow-on hints is for them to be actionable such that listeners (i.e., users of voice assistants) can ask follow-up questions.

Using the same set of 441 syntactically correct hints (cf. Table 6), annotators listen to the hints, after which a set of four questions is shown, where only one was actually part of the hint. The ability to correctly *recognize* this question is a proxy for whether the listeners could comprehend and remember the hint's information content.[7] In a conversational scenario with a voice assistant, they could follow-up by asking this question.

Table 9 shows the retention for different approaches. PTG-BART and DHG-BART achieve significantly better retention than the baselines TB and RSB. This finding demonstrates that retention is negatively impacted by incoherent (TB due to simple concatenation) and repetitive (RSB due to it not using anaphora) hints.

Model	# Recognized Questions	Hint Length (# characters)
Templates (TB)	356 (80.7%)	152.72 ± 34.6
RSB	348 (78.9%)	158.02 ± 34.6
DHG-BART	383 (86.8%)	139.85 ± 33.8
PTG-BART	384 (87.1%)	140.78 ± 34.5

Table 9: Number of hints correctly recognized as being part of the hint by annotators, who selected between four questions, where only one is correct.

8 Related Work

Our task is novel and thus has no directly comparable works, closest being on question generation.

Question Generation. Rus et al. (2010) for a given input paragraph generate questions. The works in (Chaudhri et al., 2014; Raynaud et al., 2018) make use of knowledge graphs (KG) and predefined templates, such as "*what is X*", where X is some entity from the KG.

Rosset et al. (2020) propose an approach for *conversational* question generation based on the GPT-2 (Radford et al., 2019). Given a user question, a follow-on question is suggested to the user, that can be seen as a continuation of their search trajectory. Rao et al. (2020) generate follow-up questions for interviews, where after a question, an answer, a follow-up question is generated.

Our approach can be seen related to these works, especially to (Rosset et al., 2020) given that we both aim at increasing user engagement. Yet, we differ in two fundamental ways: 1) we do generate questions but hints about questions that can be asked, and 2) through hints we allow users to explore additional topics. Finally, we do not focus on *what* but rather *how* to generate hints.

Conversational Text Generation. Su et al. (2020) propose a pretraining approach for diversifying seq2seq models in generating non-conversational text for dialogues, by additionally training on non-conversational text extracted from books. Similarly, in (Zhang et al., 2020b) a GPT-2 model is pretrained over Reddit conversation chains. Targeted conversational question generation approaches (Pan et al., 2019; Gu et al., 2021) take into account the conversation history and the topic of interest, and generate possible next questions that can be answered. These methods deal with how to generate conversational text, and thus are very different to our use case. Past works on follow-up conversation turn generation, either considers a question (Pan et al., 2019; Gu et al., 2021) or other non-conversational snippet (Zhang et al., 2020b), and focus on generating snippets that are extracted from a single sentence or passage, thus not directly dealing with text coherence. Additionally, no voice-friendly aspects are considered, diminishing their utility on voice assistants.

Text Summarization. Generating compact summaries from lengthy documents has been the focus of various approaches (Kryscinski et al., 2019). Abstract text summarization (Jiang and Bansal, 2018; Paulus et al., 2018; Durrett et al., 2016) are typically deployed in scenarios where the input text needs to be *summarized* and at the same time

[7] More details about hint length/retention are in §E.1

paraphrased. On the contrary, our task, instead of paraphrasing, requires *stylistic* changes such as rewriting questions in their *indirect speech* form. Moreover, instead of summarizing, our task entails *syntactic* changes, such as use of pronouns to avoid redundancy, and coordinating the different subordinate clauses using conjunctive phrases. The two tasks have inherently different aims and as such require optimizing for different objectives. We experimented with several pre-trained summarization models, however, expectedly their performance was poor, thus, do not include those results as baselines in the paper.

Paraphrasing. Related works on paraphrasing (Witteveen and Andrews, 2019; Niu et al., 2021; Bannard and Callison-Burch, 2005) make use of pre-trained language model to paraphrase input sentences into semantically equivalent sentences, which make use of different phrases and wording. The main difference of our task to paraphrasing lies in combining different interrogative clauses from related questions into a coherent hint, while paraphrasing does not enforce strict syntactic patterns as required in voice friendly hints (cf. Table 1).

Evaluation Metrics. Guy (2018) in his analysis of spoken and Web search queries identifies that voice questions have phonetic properties such as *speed* and *intonation* that are not present in text queries. This poses challenges when using automated metrics such as BLEU, ROGUE, where the output of a model is *voice*, but it is trained on text data. Similar to the work in (Mehri and Eskenazi, 2020), which introduces several task specific evaluation metrics to measure dialog quality, e.g. *fluency*, *engagement*, *correctness*, we follow a similar strategy and propose several human evaluations to measure voice friendliness of a hint.

9 Conclusions

We presented a novel approach for question suggestion using spoken hints. Our work enables the creation of new voice-based experiences where users can receive compact and natural hints about additional questions they can ask. Question suggestion is a standard feature in screen-based search experiences, and our work takes an important first step in bringing this capability to voice interfaces.

Our contributions are manifold: (i) a novel task of suggesting questions with voice hints; (ii) outlined the linguistic desiderata and processes to decompose questions into interrogative content clauses, and recompose them into declarative hints; and (iii) a new dataset of over 14k input questions and hints, using carefully constructed annotation guidelines and quality checks.

We defined seq2seq models to generate hints. Using both automatic metrics and human evaluations, we conclusively showed that our most sophisticated approach PTG, which utilizes a linguistically motivated pretraining task was strongly preferred by humans with most natural hints.

Limitations

Languages. We limited our work to the English language for obtaining training and testing data for generating voice-friendly hints. As a next step, we foresee adding other languages, such as German, Korean, and Chinese, and understanding the implications in terms of the required syntactic and semantic operations to generate voice-friendly hints.

Scenarios. Our work focused only on a single turn conversations, where after a user asks a question to a voice assistant, a hint suggesting related questions are uttered back to the user. Future steps include multi-turn conversations, where user interests and actions after each hint will impact the generated hints for follow-up turns. There are several strategies that can be considered, and we aim at investigating the following: dive deeper in a topic of user's interest (suggest more targeted questions on a specific topic about the entity of interest), or broaden user's knowledge on a given topic (i.e., suggest questions about *related entities*).

Large Language Models. While in this work we do not focus on recent multi-billion parameter LLMs, in Appendix F we present an evaluation of the performance of ChatGPT on our test set for the task of generating voice friendly hints. We do not go in depth in our analysis for ChatGPT and similarly large models for two key reasons. First, ChatGPT can be considered as a black box, where there is no scientific reporting on the models parameters and its training. Second, due to the strict latency requirements in voice assistants, such large models are not feasible to be used for applications like ours where the hint must be generated in 150 milliseconds or less.

References

Siddhant Arora, Alissa Ostapenko, Vijay Viswanathan, Siddharth Dalmia, Florian Metze, Shinji Watanabe, and Alan W. Black. 2021. Rethinking end-to-end evaluation of decomposable tasks: A case study on spoken language understanding. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 1264–1268. ISCA.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg. Springer-Verlag.

Pooja Rao S B, Manish Agnihotri, and Dinesh Babu Jayagopi. 2020. Automatic follow-up question generation for asynchronous interviews. In *Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation*, pages 10–20, Santiago de Compostela, Spain. Association for Computational Lingustics.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604, Ann Arbor, Michigan. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Vinay K. Chaudhri, Peter E. Clark, Adam Overholtzer, and Aaron Spaulding. 2014. Question generation from a knowledge base. In *Knowledge Engineering and Knowledge Management*, pages 54–65, Cham. Springer International Publishing.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1998–2008, Berlin, Germany. Association for Computational Linguistics.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Jing Gu, Mostafa Mirshekari, Zhou Yu, and Aaron Sisto. 2021. Chaincqg: Flow-aware conversational question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2061–2070. Association for Computational Linguistics.

Ido Guy. 2018. The characteristics of voice search: Comparing spoken with typed-in mobile web search queries. *ACM Trans. Inf. Syst.*, 36(3).

M. A. K. Halliday and R. Hasan. 1976. *Cohesion in English*. Longman, London.

Yichen Jiang and Mohit Bansal. 2018. Closed-book training to improve summarization encoder memory. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4067–4077. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020b. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. 2020. Efficient one-pass end-to-end entity linking for questions. In *EMNLP*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Irene Lopatovska, Katrina Rink, Ian Knight, Kieran Raines, Kevin Cosenza, Harriet Williams, Perachya Sorsche, David Hirsch, Qi Li, and Adrianna Martinez. 2019. Talk to me: Exploring user interactions with the amazon alexa. *Journal of Librarianship and Information Science*, 51(4):984–997.

John A Lucy and Lucy A Lucy. 1993. *Reflexive language: Reported speech and metapragmatics*. Cambridge University Press.

Xiao Ma and Ariel Liu. 2020. Challenges in supporting exploratory search through voice assistants. In *Proceedings of the 2nd Conference on Conversational User Interfaces, CUI 2020, Bilbao, Spain, July 22-24, 2020*, pages 47:1–47:3. ACM.

Shikib Mehri and Maxine Eskenazi. 2020. Unsupervised evaluation of interactive dialog with DialoGPT. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 225–235, 1st virtual meeting. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Tong Niu, Semih Yavuz, Yingbo Zhou, Nitish Shirish Keskar, Huan Wang, and Caiming Xiong. 2021. Unsupervised paraphrasing with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5136–5150, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Boyuan Pan, Hao Li, Ziyu Yao, Deng Cai, and Huan Sun. 2019. Reinforced dynamic reasoning for conversational question generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2114–2124. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Antonio Revuelta Puigdollers. 1999. Indirect questions in ancient greek: meaning and internal classification. In *Les complétives en grec ancien: actes du Colloque international de Saint-Etienne, 3-5 septembre 1998*, volume 18, page 129. Université de Saint-Etienne.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Tanguy Raynaud, Julien Subercaze, and Frédérique Laforest. 2018. Thematic question generation over knowledge bases. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 1–8.

Corbin Rosset, Chenyan Xiong, Xia Song, Daniel Campos, Nick Craswell, Saurabh Tiwary, and Paul Bennett. 2020. *Leading Conversational Search by Suggesting Useful Questions*. Association for Computing Machinery, New York, NY, USA.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Christian Moldovan. 2010. The first question generation shared task evaluation challenge. In *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics.

Christine Rzepka. 2019. Examining the use of voice assistants: A value-focused thinking approach.

Harshita Sahijwani, Jason Ingyu Choi, and Eugene Agichtein. 2020. *Would You Like to Hear the News? Investigating Voice-Based Suggestions for Conversational News Recommendation*. Association for Computing Machinery, New York, NY, USA.

Hui Su, Xiaoyu Shen, Sanqiang Zhao, Xiao Zhou, Pengwei Hu, Randy Zhong, Cheng Niu, and Jie Zhou. 2020. Diversifying dialogue generation with non-conversational text. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7087–7097. Association for Computational Linguistics.

Margarita Suñer. 1993. About indirect questions and semi-questions. *Linguistics and Philosophy*, pages 45–77.

Madiha Tabassum, Tomasz Kosinski, Alisa Frik, Nathan Malkin, Primal Wijesekera, Serge Egelman, and Heather Richter Lipford. 2019. Investigating users' preferences and expectations for always-listening voice assistants. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(4):153:1–153:23.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Bonnie L. Webber, Matthew Stone, Aravind K. Joshi, and Alistair Knott. 2003. Anaphora and discourse structure. *Comput. Linguistics*, 29(4):545–587.

Sam Witteveen and Martin Andrews. 2019. Paraphrasing with large language models. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 215–220, Hong Kong. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 270–278. Association for Computational Linguistics.

Appendix

The appendix contains details about the question data collection, crowdsourcing job setup for the hint data annotation task, and more detailed model evaluation on different aspects, such as examples and approach robustness.

Finally, it contains the workflows of our voice-friendly hint generation approaches.

A Question Suggestion via Voice Friendly Hints

In Figure 3 are shown the different steps that are invoked when a user interacts with a voice assistant in order to obtain related question suggestions via voice friendly hints.

B Question & Related Question Retrieval

Input Questions: We first create a pool of input questions by extracting 521k questions from the MS-MARCO dataset (Nguyen et al., 2016), extracted from sentences that start with *wh–** phrases from community QA websites (e.g. Quora, Yahoo Answers). By limiting to QA community pages, we can get a diverse and high quality questions.

B.1 Related Question Retrieval

For q to retrieve its related questions Q_{rel}, we first determine the entity and topic of a question.

Entity Linking: Entities are extracted from questions using the Blink approach (Li et al., 2020), which are then mapped to their *types* from DBpedia (Auer et al., 2007), restricted to only the following domains: $D = \{$`Animal`, `Athlete`, `Food`, `Holiday`, `Places`, `Politician`, `Technology`, `Wearables`, `Video Game`$\}$.

Topic Extraction: As topics we consider the *predicates* associated to entities in DBpedia. For instance, $T_d = \{$*"birth place"* ... *"death place"*$\}$ are extracted from entities of domain `Politician`. A question q is associated to a predicate from T_d based on the highest semantic similarity (Cer et al., 2018) between the topic and question keywords (and enforce a minimum threshold cosine similarity of 0.1). Finally, the question bank becomes the set of quadruples $Q = \{\langle q, e, d, t\rangle_1, \ldots \langle q, e, d, t\rangle_n\}$.

Related Question Retrieval: For an input question $q_i \in Q$, by filtering the quadruples in Q we obtain the top–k related questions Q_{rel} as the questions that have the same subject entity as q_i and which cover a different topic from q_i, namely, $Q_{rel} = \{\langle q, e, d, t\rangle_j | e_j = e_i \wedge t_j \neq t_i\}, \forall j \leq |Q|$. The top–$k$ ranges with $k = \{1, 2, 3\}$, chosen from most frequent topics in Q.

C Hint Annotation Guidelines

Annotators from the Appen crowdsourcing platform[8] are given the related questions Q_{rel}, and asked to compose a corresponding hint h. Figure 4 shows the annotation interface, while the guidelines and steps are explained in the following.

We rely only on annotators with highest level of competence[9] that were also English native speakers, and paid according to the time spent in a task at a rate of $15 (USD) per hour.[10]

Finally, we enforce a set of validation mechanisms to avoid malicious behavior from annotators. Table 10 shows the set of validators used to ensure quality of obtained annotations. Any generated hint that does not meet any of the validators in the table below is discarded. Furthermore, hints are run through Gramformer[11] to correct any potential grammar mistakes by the human annotators.

Minimum/maximum *characters* per hint[12]	Minimum/maximum *words* per hint[13]
Hint Coherence[14]	*Language* constraint (`EN`)
Presence of *start pattern*	Presence of *entity*
Presence of *anaphora*	Hint/Question(s) similarity[15]

Table 10: Validation mechanisms to ensure data quality.

D Model Setup & Hyperparameters

For all transformer based models, namely, TB, DHG, and PTG, we use the following hyperparameters for model training. We consider a learning rate of $lr = 3e^{-5}$ with a weight decay of $d = 0.01$. We train the model with a maximum 50 epochs and batch size of 8. The training stops after 10 epoch of non-decreasing validation loss.

[8] https://appen.com

[9] Level 3 workers in Appen

[10] This is the minimum hourly wage in WA, USA

[11] https://github.com/PrithivirajDamodaran/Gramformer

[12] Minimum of 70 characters, and a maximum that does not exceed the number of characters from the input question.

[13] We set the minimum and maximum number of words to be related to the length of input questions.

[14] Use Appen's natural language coherence functionality and use a minimum threshold, which the annotators need to pass in order for them to proceed further in the task.

[15] Cosine similarity computed between the generated hint and the input questions.

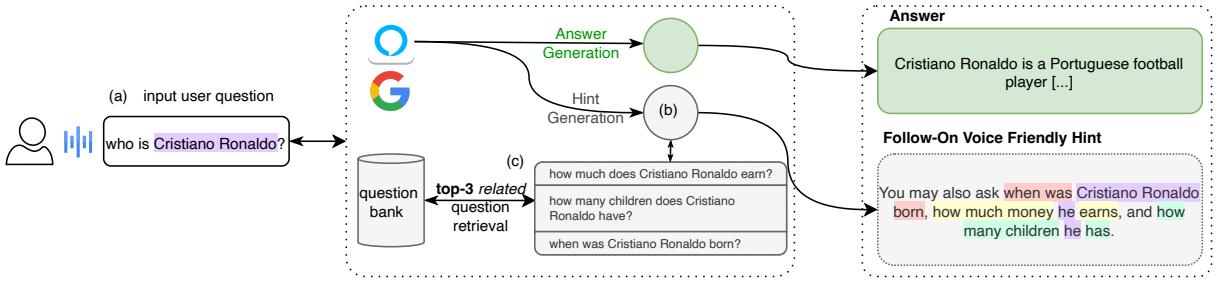

Figure 3: Voice-friendly Hint Generation Task: (a) for an input user question, the voice-assistant generates a voice-friendly hint (c) from top-3 related questions about the entity in (a) retrieved from its question bank.

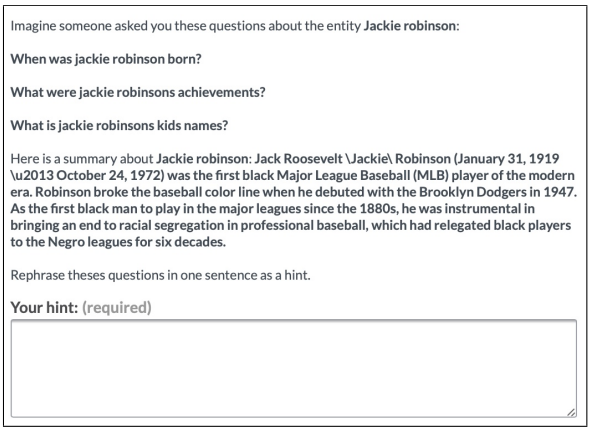

Figure 4: Annotation interface for obtaining voice-friendly hints showing three related questions about the entity *"Jackie Robinson"* along with a short summary of the entity itself, extracted from Wikipedia.

E Hint Generation Performance

The examples below show hints for the same set of questions, generated from all competing approaches. Depending on the questions in Q_{rel}, TB may or may not produce voice-friendly hints, as the questions are simply concatenated using templates. For RSB on the other hand we see that it does a series of rewrites. The differences between DHG and PTG are subtle, such as, rewriting "*does earn*" $\rightarrow$ "*earns*", which is attributed to PTG's pretrained RS knowledge. This allows the model to express the same information with fewer words and in a more voice-friendly manner.

Q_{rel}	*How much money does Cristiano Ronaldo earn? How many children does Cristiano Ronaldo have? Who is the mother of Cristiano Ronaldos child?*
TB	*You may want to know how much money does Cristiano Ronaldo earn, or how many children does Cristiano Ronaldo have, or who is the mother of Cristiano Ronaldos child.*
RSB	*You may want to know how much money Cristiano Ronaldo earns, or how many children Cristiano Ronaldo has, or who is the mother of Cristiano Ronaldo child.*
DHG	*You may want to know how much money does Cristiano Ronaldo earn, or how many children he has, or who is the mother of his child.*
PTG	*You may want to know how much money Cristiano Ronaldo earns, or how many children he has, or who is the mother of his child.*

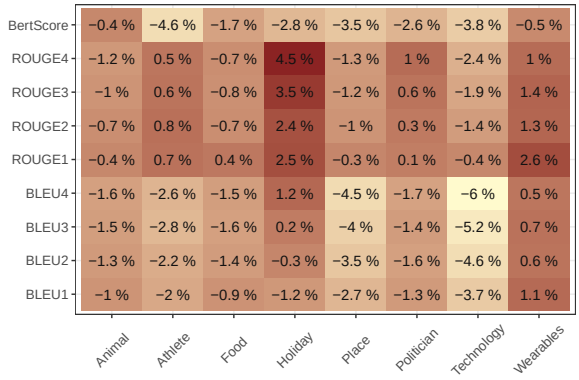

Figure 5: Performance gap of PTG-BART when evaluated in a zero-shot setting on a target domain (not seen during training) when compared to its performance when the model has been trained on questions from the target domain.

E.1 Approach Robustness

Figure 5 shows the gap in terms of performance across the different evaluation metrics for the PTG-BART when applied in a zero-shot setting on a *target* domain, compared to when the model is trained with questions from that domain. We note that overall, the gap is quite small, with many domains having a gap of 1-2%, with th exception of `Holiday`, `Place` and `Technology`, which have higher gaps. Such results show a promising generalization of PTG-BART across domains, an indicator that the models effectively learn how to perform the various syntactic operations (cf. Table 1) to produce voice-friendly hints.

Hint Length vs. Question-Retention: We measure the Pearson correlation between hint length (in characters) and the question retention from the generated hints. We note a negative moderate correlation of $\rho = -0.47$ between length and retention rate. Longer hints impact annotators' comprehension performance, resulting in their inability to correctly identify the suggested question in the hint. This confirms our hypothesis, that a key aspect to voice-friendliness such as length, has a negative

impact in a conversational setting between user and a voice-assistant in consuming such hints.

E.2 Hint Examples

Table 12 shows hints generated from the different competing approaches on the same set of input questions.

F Large Language Models for Voice Friendly Hint Generation

Large language models (LLMs) like GPT3.5 or ChatGPT,[16] which leverage billions of parameters, are shown to have great zero-shot capabilities for various tasks in NLP. While, LLMs are impractical in our setting, where the latency requirements make it nearly impossible to use such models, nonetheless we compared our models PTG and DHG against ChatGPT.

We prompted ChatGPT with the input related questions Q_{rel}, and asked to generate the hint using the prompt show in Figure 6.

Example prompts to ChatGPT

Summarize the following questions into a single question, start it with *"You may also ask"* and keep each question as a clause:
{{Q_{rel}}}

Related Questions

- What state is toronto in?
- Is toronto the largest city in canada?
- What time is it in toronto right now?

ChatGPT Output: You may also ask in which state is Toronto located, is Toronto the largest city in Canada, and what is the current time in Toronto?

Target: You could ask if Toronto is a city or a state, if it is the largest city in Canada and what time it is right now.

Figure 6: The input prompt for ChatGPT, along with an example output and human ground truth (target).

We find that ChatGPT in a zero-shot setting has significantly worse performance in terms of BLEU and ROUGE metrics, achieving the following performance on automated metrics shown in Table 11.

Finally, while ChatGPT has reasonable performance in zero-shot settings, there are limitations in terms of fine-tuning such LLMs. First, models like ChatGPT are not scientifically reported and the model is not publicly available. Second, the sheer size of the model makes it impractical and impossible to use in voice assistants, where such hints are generated in real-time based on the user's questions, requiring the models to meet very strict latency requirements where the hint must be generated in less than 150 milliseconds.

BLEU1	BLEU2	ROUGE1	ROUGE2
0.508 (▼ 12.2%)	0.297 (▼ 23%)	0.652 (▼ 9%)	0.403 (▼ 13.6%)

Table 11: ChatGPT zero-shot performance on the task of hint generation. The relative difference w.r.t PTG-BART is shown in parentheses.

[16] https://chat.openai.com

input	model	hint
how many times can i enter wrong passcode on iphone? [SEP] can i unlock my iphone even if i am still paying for it? [SEP] why does messages on iphone 8 show half moon?	DHG-Bart	You may be interested to know how many times you can enter wrong passcode on iPhone, or if you still paying for it and why messages on iPhone 8 show half moon.
	PTG-BART	You may want to know how many times you can enter wrong passcode on iPhone, or if you can unlock it even if you are still paying for it, or why messages on iPhone 8 show half moon.
	RSB	You could ask how many times can i enter wrong passcode on iphone, or if i can unlock my iphone even if i still paying for it, or why messages on iphone 8 show half moon
	TB	You could ask how many times can i enter wrong passcode on iphone, or can i unlock my iphone even if i still paying for it, or why does messages on iphone 8 show half moon?
What is the largest horse that is alive? [SEP] Where does the word horse come from? [SEP] What is the collective name for a group of horses?	TB	You can ask what is the largest horse that is alive, or where does the word horse come from, or what is the collective name for a group of horses?
	RSB	You might be interested to know what is the largest horse that is alive, or where the word horse comes from, or what is the collective name for a group of horses
	DHG-BART	You may want to know what is the largest horse that is alive, where it comes from and what is the collective name for a group of horses.
	PTG-BART	You may want to know what is the largest horse that is alive, where the word horse comes from and what is its collective name for a group of horses.
Who is the mother of cristiano ronaldo's twin's child? [SEP] Who is cristiano ronaldo's real wife? [SEP] How much money does earn cristiano ronaldo?	DHG-BART	You may want to know who is the mother of Cristiano Ronaldo's twin's child, or who is his real wife.
	PTG-BART	You may want to know who is the mother of Cristiano Ronaldo's twin's child, or who is his real wife, or how much money he earns.
	RSB	you might also be interested to know who is the mother of cristiano ronaldo's twin's child, or who is cristiano ronaldo's real wife, or how much money cristiano ronaldo earns
	TB	You can ask who is the mother of cristiano ronaldo's twin's child, or who is cristiano ronaldo's real wife, or how much money does earn cristiano ronaldo?

Table 12: Example hints generated by each model.