INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Composing Spoken Hints for Follow-on Question Suggestion in Voice Assistants

Pedro Faustini[1*], *Besnik Fetahu*[2], *Giuseppe Castellucci*[2], *Anjie Fang*[2],
Oleg Rokhlenko[2], *Shervin Malmasi*[2]

[1]Macquarie University, Sydney, NSW, Australia
[2]Amazon.com, Inc., Seattle, WA, USA

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 instant access to 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 first define the task of composing speech-based hints, ground it in syntactic theory, and outline linguistic desiderata for spoken hints. We propose a sequence-to-sequence approach to generate spoken hints from a list of questions. Using a new dataset of $6,681$ input questions and human written hints, we evaluate models with automatic metrics and human evaluation[1]. Results show that a naive approach of concatenating suggested questions creates poor voice hints. Our most sophisticated approach applies a linguistically-motivated pretraining task and 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, etc.). Users interact with voice assistants for different purposes [1, 2] such as question answering, e-commerce, or entertainment. With increasing adoption, user expectations also grow and related content recommendation is a valued feature [3].

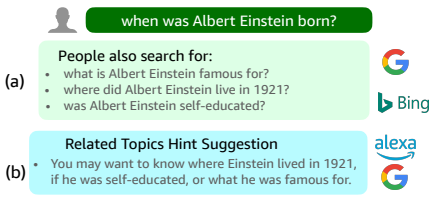

Figure 1: *(a) Question suggestion in web search (available in Google/Bing) for a user question. (b) Proposed voice-based hint for the same questions users can ask as follow-on questions to a voice assistant such as Alexa.*

The issue of *how* to present proactive suggestions is an open one, and recent work has examined how content like news articles can be recommended over voice [4]. Query and question recommendation (see Fig. 1 (a)) have become well-established research topics, and are common in screen-based Web search experiences (i.e., those from Google/Bing). However, such functionality does not exist for voice-based systems. Suggestions enable useful exploratory search capabilities, and we aim to provide a similar experience over voice (see Fig. 1 (b)), where a *follow-on hint* suggests related topics users can ask about.

Contrary to Web search suggestions, this poses several challenges in voice assistants [5], 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) *content structure*: simply reading out a list of questions is not natural over voice.

We propose an approach on *how* to deliver voice-based question suggestions using hints. We do not consider *what* to suggest as this is widely explored [6, 7]. Figure 1 provides an overview. For an input question, we assume the voice assistants can retrieve related questions from which a suggestion hint is generated. Differently from Web search suggestions (Fig. 1 (a)) where new related questions are listed, we aim to synthesize a spoken hint (Fig. 1 (b)) suggesting the same questions. The hint does not contain questions, rather, it contains several subordinate clauses describing knowledge that the user can ask.

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 based on Transformers [8]. For an input question and a set of 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.

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 speech-based hints, allowing users to explore related topics about their original question;
2. A large real-world hint generation dataset of $6,681$ instances, covering 9 domains, that will become publicly available;
3. A seq2seq approach with task-specific training strategies for voice hint generation.

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 (or independent clause) that informs the listener about the different pieces of information available.

Direct questions ("*can a dog eat peanuts?*") can be presented as an *indirect question* ("*Alice asked if dogs can eat*

*Work done during an internship at Amazon.
[1]https://github.com/bfetahu/spoken_hints/

10.21437/Interspeech.2023-1240

peanuts.") [9]. All direct questions can have an indirect equivalent, and the embedded clause of the indirect version is said to refer to the direct question [10].

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 [10]. 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.

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

Table 1: *Linguistic properties of a natural spoken hint.*

Aspect	*Description*
Start Patterns	You may also *want to know/be interested . . .*.
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 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* [11]. • Lexical repetitions, e.g. entities to be replaced by anaphora where appropriate.

For a hint to be considered as 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 [12] and Gricean maxims of conversation [13]. They ensure that hints sound natural and is easy to comprehend. The characteristics were derived from our preliminary experiments on how English speakers create hints.

3. Spoken Hint Generation Approach

3.1. Hints Generation Architecture

To learn $\mathcal{F}(q, Q_{rel})$, the hint generation model, we propose using sequence-to-sequence (seq2seq) models. More specifically we experiment with BART [8] and T5 models [14]. We encode the input question q and related questions Q_{rel} by concatenating them using the `[SEP]` token[2]:

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

Encoder representation of $\mathbf{s}$ is used by the decoder to generate the hint h. During training, $\mathcal{F}$ 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 language generation performance, fine-tuning them for all the criteria above is challenging, resulting in *incoherent* and *unnatural* hints (cf. §6). 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 $\mathcal{F}$ to perform the most significant rewrite operations, including performing the subordinate clause syntax change, such as verb tense, pronoun and word order alterations. Hence, we propose a two stage training strategy, where: (1) pretrain $\mathcal{F}$ in converting individual questions into their RS format, and (2) fine-tune $\mathcal{F}$ for the full hint generation task, ensuring that the hint has no repetitions and is coherent.

RS Pre-training: To pretrain $\mathcal{F}$, we change the input to consist of a single question $\mathbf{s} := \{q^1_{rel}\}$, and output the rewritten q^1_{rel} in its RS format. This is equivalent to a hint generated from the top–1 related question Q_{rel}, with the only difference that there is no initial input question q as input to the model. Our intuition is that by constraining the pretraining phase to a single question it allows $\mathcal{F}$ to learn how to perform the necessary rewrite operations for converting a question to RS format.

Fine-Tuning. The pretrained $\mathcal{F}$ is fine-tuned to learn how to convert an input $\mathbf{s}$ with multiple related questions Q_{rel}. By this stage, $\mathcal{F}$ already possesses pretrained knowledge for converting a single $q_{rel} \in Q_{rel}$ into RS format, and can focus on learning to use anaphora, conjunctions, etc.

4. Dataset Collection

Here we describe the process of generating a new voice-friendly hints dataset. We first construct tuples of input and related questions $\langle q, Q_{rel} \rangle$, then annotate spoken hints, creating a dataset of $6,681$ samples composed of the triples $Q = \{\langle q, Q_{rel}, h \rangle_i \ldots\}$.

Hint Annotation Using the question bank Q and the related questions Q_{rel}, we collect hints for suggesting the related questions. From a sample of $6,681$ input questions and their related questions, we create two *disjoint* hint sets: 1) **SINGLE-HINTS**: generated from only one related question, and 2) **MULTI-HINTS**: generated from multiple distinct related questions.

Hint Generation Guidelines Based on the intuitions from §2, we provide guidelines to annotators to create voice-friendly hints. For the tuple $\langle q, Q_{rel} \rangle$, following the steps below annotators write the hint h: **Step 1)** Annotators are asked to start the hint with one of the provided *start patterns* (cf. Table 1); **Step 2.a)** Next, questions in Q_{rel} are converted into their RS; and **Step 2.b)** For MULTI-HINTS, annotators need to avoid *repetitions* and replace them with their *anaphora* where necessary. Next, subordinate clauses from Q_{rel} are connected with the correct *conjunctive discourse markers*.

Data Collection Table 2 shows an overview of our dataset. Our main focus is in generating hints from top–3 related questions Q_{rel}, but to ensure diversity, we also collect hints constructed from the top–1 and top–2 related questions. This increases the utility of our data, as hint generation approaches must ensure 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 pre-training of our hint generation approaches.

[2] A special token in pre-trained models for marking text boundaries.

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

	SINGLE-HINTS		MULTI-HINTS		
domain	#	ratio	#	ratio ($\|Q_{rel}\|$=2)	ratio ($\|Q_{rel}\|$=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

5. Experimental Setup

5.1. Datasets

Pre-training RS. SINGLE-HINTS also referred to as RS data are used for pretraining the hint generation approaches.
Hint Generation. For the main task, we randomly sample questions from Table 2, where 81% are hints generated from 3 questions, 17% with 2 questions, and the remaining 2% are SINGLE-HINTS.

Table 3: *Pretraining and training hint generation datasets.*

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

5.2. Baselines and Approaches

For all Transformer-based approaches, we experimented with both BART [15] and T5 [14].
Template Baseline – TB. Hints are constructed according to manually defined templates, by first choosing a start pattern (cf. Table 1) and then concatenating question from Q_{rel} using *"or"*.
Reported Speech Baseline – RSB. We train a seq2seq model on SINGLE-HINTS only, where questions in Q_{rel} are converted into RS format, then using TB, they are concatenated into a hint.
Direct Hint Generation – DHG. This represents our approach without pretraining. The limitation of DHG is that it has to jointly learn all aspects of constructing voice-friendly hints.
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.

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 similarity of generated hints against the ground truth, we use BLEU [16], ROUGE [17] and F1-BertScore [18]. BLEU captures accuracy in terms of *n-grams*, while ROUGE quantifies coverage. BERTScore computes hint semantic similarity, accounting for the use of different paraphrases in hints.

5.3.2. Human Evaluation

We devise a set of human evaluations which judge the correctness and naturalness of a hint. For a realistic evaluation, the studies are performed in voice modality, apart from *Question coverage* and *syntactic correctness*. We consider the following studies:

Syntactic Correctness. Assess 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.
Question Retention. We consider retention of a hint's information in annotator's memory as a proxy for its simplicity and comprehensibility. Hints cannot be considered actionable if listeners cannot remember them. 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 wtih Automated Metrics

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

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. B1–B4 represent BLEU scores, and R1–R4 represent ROUGE scores.*

	B1	B2	B3	B4	R1	R2	R3	R4	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

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 TB. However, RSB does not reduce lexical repetition via anaphora, and simple concatenation results in lower coherence. Overall, as expected, TB and RSB, achieve low scores, however better insight are provided in Section 7, which capture hint voice friendliness.

Approach Performance: Our approaches, DHG and PTG, show a consistent improvement over TB and RSB across all metrics. 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 focus only 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.

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

Finally, the difference in performance between PTG and DHG, demonstrates that for complex rewriting tasks, end-to-end training may be sub-optimal.

7. Human Evaluation Studies

7.1. Syntactic Correctness and Coverage

Table 5 shows the results on question coverage and hint 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.

Table 5: *Syntactic correctness and question coverage results (significant results between* PTG *and* DHG *are marked with ‡).*

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%)‡

Syntactic Correctness. Table 5 shows a consistent pattern in terms of syntactic correctness: the baseline RSB and PTG-BART have the most 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. However, 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. Section 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 6: *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.*

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

Table 6 shows the pairwise comparisons the different models. We run the comparison on the 441 hints that were judged to be syntactically correct in Table 5. This is done to avoid any bias stemming from syntactically incorrect hints. In both comparisons, PTG-BART produces more natural hints than baselines. Against TB and RSB, it is preferred in 68% and 60% of the cases, respectively. Both results represent statistically highly significant differences (as per Wilcoxon's signed-rank test).

7.3. Question Retention Evaluation

Here 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 5), 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. In a conversational scenario with a voice assistant, they could follow-up by asking this question.

Table 7 shows the retention for different approaches. PTG-BART and DHG-BART achieve significantly better retention than 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.

Table 7: *Number of hints correctly recognized by annotators.*

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

7.4. Examples of Model Generated Spoken Hints

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.*

8. Conclusions

We presented a novel approach for spoken question suggestion. 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 a key 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 $6,681$ input questions and hints (14k when considering both SINGLE-HINTS and MULTI-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.

9. References

[1] C. Rzepka, "Examining the use of voice assistants: A value-focused thinking approach," 2019.

[2] I. Lopatovska, K. Rink, I. Knight, K. Raines, K. Cosenza, H. Williams, P. Sorsche, D. Hirsch, Q. Li, and A. Martinez, "Talk to me: Exploring user interactions with the amazon alexa," *Journal of Librarianship and Information Science*, vol. 51, no. 4, pp. 984–997, 2019.

[3] M. Tabassum, T. Kosinski, A. Frik, N. Malkin, P. Wijesekera, S. Egelman, and H. R. Lipford, "Investigating users' preferences and expectations for always-listening voice assistants," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 4, pp. 153:1–153:23, 2019. [Online]. Available: https://doi.org/10.1145/3369807

[4] H. Sahijwani, J. I. Choi, and E. Agichtein, *Would You Like to Hear the News? Investigating Voice-Based Suggestions for Conversational News Recommendation*. New York, NY, USA: Association for Computing Machinery, 2020, p. 437–441. [Online]. Available: https://doi.org/10.1145/3343413.3378013

[5] X. Ma and A. Liu, "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*, M. I. Torres, S. Schlögl, L. Clark, and M. Porcheron, Eds. ACM, 2020, pp. 47:1–47:3. [Online]. Available: https://doi.org/10.1145/3405755.3406152

[6] P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna, "Query suggestions using query-flow graphs," in *Proceedings of the 2009 workshop on Web Search Click Data, WSCD@WSDM 2009, Barcelona, Spain, February 9, 2009*, N. Craswell, R. Jones, G. Dupret, and E. Viegas, Eds. ACM, 2009, pp. 56–63. [Online]. Available: https://doi.org/10.1145/1507509.1507518

[7] C. Rosset, C. Xiong, X. Song, D. Campos, N. Craswell, S. Tiwary, and P. N. Bennett, "Leading conversational search by suggesting useful questions," in *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Y. Huang, I. King, T. Liu, and M. van Steen, Eds. ACM / IW3C2, 2020, pp. 1160–1170. [Online]. Available: https://doi.org/10.1145/3366423.3380193

[8] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "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*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 7871–7880. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.703

[9] M. Suñer, "About indirect questions and semi-questions," *Linguistics and Philosophy*, pp. 45–77, 1993.

[10] A. R. Puigdollers, "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*, vol. 18. Université de Saint-Etienne, 1999, p. 129.

[11] B. L. Webber, M. Stone, A. K. Joshi, and A. Knott, "Anaphora and discourse structure," *Comput. Linguistics*, vol. 29, no. 4, pp. 545–587, 2003. [Online]. Available: https://doi.org/10.1162/089120103322753347

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

[13] H. P. Grice, "Logic and conversation," in *Speech acts*. Brill, 1975, pp. 41–58.

[14] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[15] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "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*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. [Online]. Available: https://aclanthology.org/2020.acl-main.703

[16] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://aclanthology.org/P02-1040

[17] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[18] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr