

CAsT 2020: The Conversational Assistance Track Overview

Jeffrey Dalton¹, Chenyan Xiong², and Jamie Callan³

University of Glasgow¹, Microsoft Research², Carnegie Mellon University³
jeff.dalton@glasgow.ac.uk¹, chenyan.xiong@microsoft.com², callan@cs.cmu.edu³

1 INTRODUCTION

CAsT 2020 is the second year of the Conversational Assistance Track and builds on the lessons from the first year. Teams tried a wide range of techniques to address conversational search challenges. Some methods used proven techniques such as query difficulty prediction and query expansion. Given the text understanding challenges in the task, teams also used traditional NLP models that incorporate coreference resolution. One important development was the application of generative query models and ranking models using pre-trained neural language models. The results showed that both traditional and neural techniques provided complementary effectiveness.

Based on participant feedback, CAsT 2020 task is similar to 2019. The task is identify relevant passages for conversational queries that evolve through a trajectory of a discussion on a topic. For consistency, the corpus is unchanged with it including both MS MARCO passage and Wikipedia (Complex Answer Retrieval) passages. This stability enables the community to explore the challenges and best practices in a consistent setup and to accumulate high quality annotation labels on more topics for long-term use.

The biggest change for CAsT 2020 is that utterances refer to previous *responses* given by a system (e.g., see turn 8 in Table 1). In contrast, in 2019 turns could only refer to information mentioned in previous user utterances. This make the setup more realistic by eliminating the need to restate information presented in a previous result. It also expands the contextual information that systems can, and sometimes *must*, use to understand a turn. To allow for repeatability for new systems the dependence on previous results is done in a controlled manner. A single canonical result is selected as the response for each turn, as detailed in Section 2. Section 5 describes how participating systems handle this additional dependency. Another minor change to make the conversations realistic compared with 2019, the topics in 2020 no longer include a title or description – the information need unfolds only through the turns.

To add additional realism the topics in 2020 are based on real user needs mined from information seeking sessions in Bing sessions [6]. Sessions are manually reviewed and filtered to ensure they have meaningful trajectories and are then manually rewritten by an organizer to allow them to be used for reference. The organizer manually construct 2020 topics from these derived sessions. The topics reflect the organizers’ vision of user behavior for conversational search systems of the future, while also being grounded in real information needs and current search behavior. We provide details on topic construction in Section 2.

Year two had strong participation from more than a dozen teams worldwide. The results show that the second year of CAsT is more

Table 1: CAsT 2020 Topic 91.

Turn	Conversation Utterances
1	What is the purpose of GDPR?
2	What is different compared to previous legislation?
3	What are the privacy implications of those technologies?
4	Oh, IP addresses are considered PII? What is the full range of personal data?
5	How do big companies adapt to GDPR?
6	OK. Tell me about the privacy issues in social networks.
7	What do they get in return for their privacy?
8	What are the symptoms of that addiction?

challenging than the first year. While similar in structure, the 2020 topics have greater complexity. We find that turns requiring previous result context to be particularly challenging.

The long-term vision of CAsT remains unchanged: to support natural conversations between a person and a search engine to satisfy information needs and support complex information tasks. The first two years of CAsT focused on retrieving relevant short content from passages. Future iterations will evolve this setup and continue to challenge and advance research in CIS.

2 TASK, DATA, AND RESOURCES

The core of the CAsT 2020 task is unchanged from CAsT 2019. The goal of the task is to satisfy a user’s complex information need expressed through multi-turn conversational queries/utterances (u) for each turn $T = \{u_1, \dots, u_i, \dots, u_n\}$, by retrieving and ranking passages from MS MARCO [1] and Wikipedia – the Complex Answer Retrieval (CAR) corpora [4]. The CAsT 2019 overview and dataset paper provide detail about the this setup [2, 3].

CAsT 2020 has twenty five information needs (topics) with an average length of 8.6 utterances, for a total of 216 turns. In comparison, the CAsT 2020 topics are slightly shorter than the 2019, which averaged 9.5 turns. An example topic is shown in Table 1.

The rest of this section focuses on the two major changes for the second year: the more realistic information needs derived from commercial search logs, and the possible dependence of a query on a previous system response.

Information Needs. A majority of the topics are based on multi-turn information-seeking sessions from a commercial search engine. The reference sessions are constructed via the following steps.

- (1) Filter the raw web search sessions to conversational-alike search sessions, using the procedure to obtain about 20,000 “QA-Gen” sessions as described in Rosset et al. [6].

- (2) One organizer manually examines the QA-Gen sessions and selects approximately two thousand diverse topics and non-trivial topics with non-personal information for privacy reasons.
- (3) The sessions provide inspiration for the organizer to manually write several hundred derived conversational-like ad-hoc search sessions, with realistic and challenging information needs.

The result of this step is synthetic web sessions inspired by one or more real complex web sessions.

The organizers use the manually written web sessions to build natural language conversational topics. The same guidelines from year one are used to ensure that topics are complex (requiring multiple rounds of elaboration), diverse (across different information categories), open-domain (not requiring expert domain knowledge to access), and answerable (sufficient coverage in the corpus).

Result Dependence. A major difference in year two is the added possible dependence on the system responses of previous turns. When interacting with a conversational assistant, a user often not only refers to their past queries, but also the assistant's responses earlier in the conversation.

To simulate this, the organizers built a baseline search system (discussed below) that uses standard BM25 retrieval followed by a BERT-based reranker. The organizers used the baseline system when curating queries to check the number and nature of answer passages. One challenge is that the baseline did not include an automatic query rewriter, particularly one that used previous result context and so organizers used their best judgment to construct manual queries.

When curating the next turn in the conversational sequence (u_j), the organizers examine the results of an earlier conversational turn ($u_i, i < j$), *often manually rewriting the earlier query u_i until a reasonable response is provided by the system.* Then the next query (u_j) may be written with contextual dependence on *queries and/or responses from previous turns.* There is no explicit guidelines on the number of queries that should depend on previous questions or answers, however the majority of turns have some dependence on previous utterances or responses. The resulting dataset has approximately a quarter of the turns depend on a previous system response.

Canonical Response for Reusability. The interactive nature of the response dependence makes it necessary to include a fixed system's response as part of benchmark. Otherwise it would be nearly impossible to resolve this dependence because of differences in system responses.

For the system response the organizers provide a single *canonical passage response*, the response provided by a hypothetical agent. All turns have a response. In most cases the *canonical response* is sampled from the top three results of the baseline ranking. Note that the organizers provide full rankings for the baseline systems.

To identify a passage that continues the conversation in a natural direction the organizers sometimes identified passages outside the baseline results; either deeper or from alternative query formulations. Participants are not informed about the source of the canonical response or whether the canonical response is referenced in later turns.

There are two versions of canonical responses that correspond to two categories of submission:

The *automatic canonical* response is sampled from the top results by the baseline system for queries rewritten by an automatic conversation query rewriter built for CAsT 2019 [7].

The *manual canonical* response is sampled from the baseline system for the manually rewritten query, those released as the oracle context-free queries. Systems that use the manual canonical responses are considered "manual" runs in the participating guideline, as the responses are for oracle manual queries.

Although it organizers preferred responses returned by the baseline system, it was not required. When the response passage is not returned at an appropriate rank by the canonical query, the organizers insert it into the ranking for both the *manual* and the *automatic* canonical responses. We discuss these elements in more depth later.

Baseline System. The initial ranking is produced by Solr using BM25 ($k_1=1.2, b=0.75$), distributed (multi-partition) idfs, stopword removal, and KStem stemming. Based on preliminary experiments, it reranks the top 500 documents using a BERT-base reranker with a publicly available Hugging Face software¹ model. It is fine-tuned using 12.8M query-passage pairs from the MS-MARCO passage ranking corpus, following Nogueira, et al. [5] (batch size=32, gradients accumulated over 4 time steps, passages truncated to 256 tokens). The search engine is available for interactive search by track participants. Although not formally benchmarked, most queries returned results in under five seconds.

Other Data Resources. The organizers release similar resources released as year one [2, 3] that include the canonical results and baseline runs. In addition, the organizers release the contextual dependence labels for queries and results developed during topic curation. In each turn the corresponding curator manually notes which query or response from a previous turn is referred to if a contextual dependency exists. These labels are verified by at least one other organizer. In some cases the references may be ambiguous - to a query, result, or combination. The earliest reference to a concept in the conversation is used. We use this fine-grained annotation to analyze the influence of contextual dependencies for participating runs in Section 5 and believe they can also be useful for future research.

Evaluation. The evaluation of CAsT 2020 follows CAsT 2019. Refer to the previous year for details [2, 3].

The overlap among submissions is significantly higher in year two than in year one. This allowed assessors to judge deeper for some topics. All document pools are formed from all four runs across all participants. The pools for thirteen topics (81, 83, 84, 85, 87, 88, 90, 92, 95, 98, 99, 101, 102) use documents from ranks 1-10, as last year. The pools for eleven topics (82, 86, 89, 91, 93, 94, 96, 97, 103, 104, 105) include documents from ranks 1-15. Topic 100 use depth 15 for all queries except 100_8, which use depth 10. This variation enables us to analyze the effect of the pool size on recall.

Table 2 shows that the deeper pool increases the average number of judged documents per query by 47%, from 145 to 213. The number of relevant documents per query increases by about 33% for each level of relevance. This difference includes relevance levels 2-4 that

¹<https://huggingface.co/>

have few relevant documents. It suggests that judging four runs per participant up to depth 10 documents misses some relevant documents.

We advise researchers to monitor whether returned results are assessed because judgments may be incomplete. The queries that have deeper pools may more complete.

2.1 Result Dependence Considerations

The introduction of response dependence more closely models the challenges faced in conversational search systems, but the offline nature of this benchmark and our goal to ensure re-usability introduces challenges and limitations, which we highlight in this section.

Canonical response limitations. Adding result dependence presented challenges developing topics, particularly when providing canonical responses for all turns. A canonical response is always provided - regardless of its relevance and whether it is referred to in subsequent turns. However, due to the nature of how they are constructed the results that are referred to later are more likely to be relevant because they are building block for discussion.

Further, artificially adding results not present in the baseline response list is problematic. For both manual and automatic versions the ground truth response dependence is hidden, embedded in the provided responses. But where there is dependence the results much be shared to keep the automatic and manual conversations consistent. As a result, there is the the potential to implicitly indicate result dependence in an artificial way if different setups are compared.

Systems using the canonical responses have access to data that systems without access do not. Systems using canonical results may have a limited implicit advantage. As a result, we separate comparison of systems that use canonical results from those that only use previous utterances and recommend this be clearly stated when reporting future results. Due to the setup, systems that use a response are likely to require access to canonical responses.

Possible bias towards responses from the baseline. By design, all the contextual dependencies on responses came from the baseline system used to curate the topic. The dependent responses are manually selected by the organizers and are frequently relevant to the previous turn. The dependency on these baseline system's output may make it challenging for other systems that have different behavior.

Separate evaluation for automatic canonical. To ensure a fair comparison of systems this year we separate runs that use the automatic canonical responses *in their inference and testing*. As previously described, they have additional access to data that may partially indicate the relevance of some results. Although, we note that systems are still automatic and have far less information than systems using the manually resolved utterances (and results).

There are three categorizes in this year's runs, based on the data they used in the *testing phrase*:

- (1) *Manual*: Runs that use the manually rewritten (resolved) context-free queries, and/or manual canonical responses.
- (2) *Automatic-Canonical*: Automatic runs that use the provided automatic canonical system responses.

Table 2: The effect of pool depth on the average number of judged documents per query, averaged over 106 queries. Both pools were formed from up to 4 runs per team.

Label	Depth 10	Depth 15	Increase
0	125	186	49%
1	9	12	33%
2	6	8	33%
3	2	3	50%
4	3	4	33%
Total	145	213	47%

- (3) *Automatic*: Runs that only use the provided raw conversational queries, automatic baseline results, automatic query rewrites, and/or other data source that do only contain manual or automatic-canonical information.

We also suggest that future research make this clear distinction to avoid confusion and for fair comparison. We envision that importance of dependence on results will increase and new methods of incorporating it in the benchmark in a reusable way will evolve in the future.

3 PARTICIPANTS

CAST received 51 run submissions from 15 teams shown in Table 3. When submitting, we asked the participants to provide metadata describing certain properties of their runs.

3.1 Submitted team descriptions

Below are brief summaries of approaches from each participant. Teams are listed in alphabetical order.

- **ASCFDA** - We first use a T5-based conversational query rewriting (CQR) model to solve the co-reference problem of queries. Then rewritten queries are exploited to do further expansion via three query expansion methods: RM3 from Anserini, keyword distillation method, and sentence-based query extraction. With the expanded queries, we retrieve candidate passages with BM25. A T5-based reranker is used to rerank the retrieved passages.
- **CMU-LTI** - The pipeline consisted of two stages. The first stage used a BERT-based classifier (trained with weak supervision) to de-contextualize the input by selecting relevant terms from the dialog history. The selected terms were appended to the input question to create a query that Indri for initial retrieval. The second stage used BERT for reranking. It began by creating two types of queries for each question within a conversation using the BERT-based classifier used in the first stage. The first query consisted of relevant terms from the dialog history whereas the second query consisted of relevant terms from the top 5 passages (retrieved in the first round of retrieval). The queries were individually used to rerank passages. Finally, a fusion over the reranked list was performed to produce the final rankings.
- **DUTH** - Our approach consists of two main steps: i) linguistic analysis and ii) query reformulation. We apply the AllenNLP co-reference resolution model to every query of

Table 3: Participants and their runs.

Group	Run ID	Run Type	Group	Run ID	Run Type
ASCFDA	ASCFDA_baseline	automatic	ielab	ielab-bertAQ	automatic
ASCFDA	ASCFDA_d2q_emb	automatic	ielab	ielab-bertPRFAQ	automatic
ASCFDA	ASCFDA_qa	manual	ielab	ielab-bm25AQ	automatic
ASCFDA	ASCFDA_rm3	automatic	ielab	ielab-bm25T5QLM	manual
CMU-LTI	PAS_BQUERY_MER	automatic	nova-search	AUTO_BERT100	automatic
CMU-LTI	PAS_RQUERY_MER	automatic	nova-search	AUTO_T5_RRF	automatic
CMU-LTI	PASO_BQUERO_MER	automatic	nova-search	T5_BERT100	automatic
DUTH	duth	automatic	POLYU_SOME	man_polyu1	manual
DUTH	duth_arq	automatic	POLYU_SOME	raw_polyu1	automatic
DUTH	duth_manual	manual	UMass_CIIIR	umass_4prev_rm3	automatic
grill	grill_bmDuo	manual	UMass_CIIIR	umass_curr_rm3	manual
grill	grill_ctxDuo	manual	USI	castur_albert	automatic
grill	grill_duoBART	automatic	USI	hist_attention	canonical
grill	grill_fuseDuo	automatic	USI	hist_concat	canonical
h2oloo	h2oloo_RUN1	canonical	USI	rewrite_albert	automatic
h2oloo	h2oloo_RUN2	canonical	UvA.ILPS	baselineQR	automatic
h2oloo	h2oloo_RUN3	automatic	UvA.ILPS	humanQR	manual
h2oloo	h2oloo_RUN4	automatic	UvA.ILPS	qretrecNoRerank	canonical
HBKU	HBKU_t2_1v1	automatic	UvA.ILPS	qretrecQR	canonical
HBKU	HBKU_t2_1v2	automatic	WaterlooClarke	WatACBase	automatic
HBKU	HBKU_t5_1v1	automatic	WaterlooClarke	WatACBaseRe	automatic
HBKU	HBKU_t5_1v2	automatic	WaterlooClarke	WatACGPT2Re	automatic
HPCLab-CNR	HPCLab-CNR-run1	canonical	WaterlooClarke	WatACReAll	automatic
HPCLab-CNR	HPCLab-CNR-run2	automatic	WLU	WLU_ManUttOnly	manual
HPCLab-CNR	HPCLab-CNR-run3	canonical	WLU	WLU_RawUttOnly	automatic
HPCLab-CNR	HPCLab-CNR-run4	automatic			

each conversational session. Then, we use the SpaCy model for part-of-speech tagging, and keyword extraction from the current and the previous turns to be used as context. In the second step, we reformulate the initial query into a weighted query by keeping the keywords from the current turn and adding conversational context from previous turns before retrieval using Indri. The goal is for the conversational context of previous turns to have less impact than the keywords from the current turn while adding informational value.

- **GRILL** - We leverage leverage query re-writing by fine-tuning a BART model on a full context generation objective for automatic runs. This is integrated into a multi-stage re-ranking pipeline with mono & duo BERT for point-wise and pair-wise scoring, both for auto and manual runs.
- **h2oloo** - We use a multi-stage pipeline for conversational search comprised of a query reformulation model followed by a two-stage retrieve-and-rerank text ranking model. Our query reformulation model is based on T5, fine-tuned on the conversational question rewriting dataset: CANARD. As for our two-stage text ranking model, we had a dense-sparse hybrid retrieval model followed by a T5 reranker, and the ranking model is fine-tuned on MS MARCO passage ranking dataset. During inference, we have different schemes to extract sentences from system responses, and the extracted sentences are treated as part of historical context for query reformulation.
- **HBKU** - Passages were retrieved by implementing a multi-stage retrieval pipeline inspired by last year's winning solution with the addition of a pseudo-relevance feedback step where the query is expanded using top-k retrieved passages. Passages were re-ranked using a pre-trained BERT re-ranker.
- **HPCLab-CNR** - Our utterance rewriting techniques enrich the raw utterances with the context extracted from the previous utterances. Two of our approaches are completely unsupervised, while the other two rely on utterances manually classified by human assessors. These approaches also employ the canonical responses for the automatically rewritten utterances provided by the organizers.
- **ielab** - We investigated two methods to improve both the retrieval and re-ranking stages of a conversational IR system. The first method focused on query adaptation, which extracted context from the first query only to expand all subsequent queries for a conversational session. The second method utilized the text-to-text transfer transformer (T5) as a scoring function within query likelihood model (QLM) for re-ranking passages. All submissions employed Anserini (with BM25 and RM3) for passage retrieval.
- **nova-search** - Our submission is based on a three-stage pipeline composed of: 1) query rewriting, 2) passage retrieval, and 3) passage re-ranking. Query rewriting is performed using a T5 model fine-tuned on the conversational query rewriting task using the CANARD dataset. Retrieval is carried out by a Language Model with Dirichlet smoothing

(LMD). Passage re-ranking uses a BERT model fine-tuned on MS MARCO on a relevance classification task.

- **POLYU_SOME** - We use a 2-stage conversational search architecture, including the initial rule-based retrieval and BERT-based re-ranking.
- **UMass_CIIR** - Our submission investigates the use of unsupervised query expansion. We submit two runs, in the first run we study the RM3 formulation for each turn in the conversation. In our second, we explore if the top retrieved passages from previous turns can be improve recall for the current turn.
- **USI** - The system first performs query expansion by concatenating raw, relevant previous utterances, as predicted by an independent model trained on CASTUR, with the current one. Initial ranking is performed by BM25, followed by ALBERT re-ranker trained on MS MARCO passage ranking task. Modifications of the approach include methods for using context: 1) feeding the previous utterance and its top-1 response to the model alongside the current one; 2) feeding up to 3 relevant utterances to the model and performing an attentive-sum to aggregate context information. Our last run uses automatically rewritten queries without context.
- **Uva.ILPS** - Our main run (quretecQR) uses the QuReTeC term classification model applied on raw utterance and automatic canonical responses. For passage retrieval, we use Anserini with BERT-based reranker. The retrieval score is combined with the score of a reader model trained on the MRQA dataset reformulated as a binary classification task.
- **WaterlooClarke** - Our runs this year were based solely on the raw utterances. For each utterance, we used two different query generation methods, executed these queries against the collection, and merged the top-500 passages produced by each query into a single pool. We then re-ranked this pool using BERT.
- **WLU** - Our approach first uses Indri search engine to retrieve top 1000 sample answer paragraphs as the candidate set. Then a hierarchical session-based learning model with BERT encoding further matches how a candidate document is related to the sequence of utterances and their sample answer paragraphs.

Most teams used a multi-step pipeline consisting of: 1) conversational rewriting (optionally using the responses), 2) passage retrieval using traditional IR model, and 3) reranking with a fine-tuned neural language model. Almost all teams leverage pre-trained Transformer-based language models for rewriting (GPT-2, BART, T5) and ranking (BERT, ALBERT, T5).

4 OVERALL RESULTS

In this section we present results of the submitted runs. We also report three organizer baselines (prefixed *org*) that are included in the pooling and are available in the CAST Github repository.

The main results are turn-level macro-averaged response effectiveness. We use four standard evaluation measures: Recall, Mean Average Precision (MAP@1000), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG@1000). Continuing with Year one, the primary measure is NDCG@3 to focus on

Table 4: Automatic response retrieval results. The first four metrics use cutoff@1000. Binary relevance metrics, Recall, MAP, and MRR, use relevance scale ≥ 2 as positive.

Group	Run	Recall	MAP	MRR	NDCG	NDCG@3
h2oloo	h2oloo_RUN4	0.633	0.302	0.593	0.526	0.458
ASCFDA	ASCFDA_baseline	0.587	0.279	0.597	0.494	0.458
ASCFDA	ASCFDA_d2q_emb	0.588	0.278	0.595	0.493	0.458
h2oloo	h2oloo_RUN3	0.639	0.296	0.582	0.522	0.452
ASCFDA	ASCFDA_rm3	0.612	0.274	0.584	0.500	0.451
grill	grill_duoBART	0.506	0.190	0.520	0.403	0.398
USI	rewrite_albert	0.507	0.189	0.495	0.389	0.339
WaterlooClarke	WatACReAll	0.617	0.204	0.458	0.435	0.337
WaterlooClarke	WatACGPT2Re	0.523	0.191	0.435	0.393	0.325
CMU-LTI	PAS_BQUERY_MER	0.511	0.174	0.461	0.381	0.325
CMU-LTI	PASO_BQUERO_MER	0.492	0.174	0.451	0.378	0.323
UvA.ILPS	baselineQR	0.238	0.129	0.427	0.261	0.319
CMU-LTI	PAS_RQUERY_MER	0.523	0.170	0.450	0.382	0.318
HBKU	HBKU_t5_1v2	0.494	0.176	0.445	0.379	0.313
HBKU	HBKU_t2_1v2	0.495	0.177	0.440	0.377	0.309
HBKU	HBKU_t5_1v1	0.494	0.171	0.428	0.374	0.307
nova-search	AUTO_BERT100	0.521	0.151	0.422	0.364	0.304
nova-search	AUTO_T5_RRF	0.547	0.151	0.420	0.377	0.302
nova-search	T5_BERT100	0.502	0.148	0.416	0.353	0.301
-	org_auto_bertbase	0.308	0.134	0.408	0.284	0.300
WaterlooClarke	WatACBaseRe	0.524	0.175	0.425	0.384	0.298
HBKU	HBKU_t2_1v1	0.495	0.169	0.418	0.369	0.296
HPCLab-CNR	HPCLab-CNR-run4	0.489	0.164	0.421	0.359	0.292
USI	castur_albert	0.475	0.160	0.423	0.356	0.281
HPCLab-CNR	HPCLab-CNR-run2	0.508	0.154	0.401	0.360	0.275
POLYU_SOME	raw_polyu1	0.503	0.128	0.408	0.346	0.265
UMass_CIIR	umass_4prev_rm3	0.548	0.114	0.332	0.342	0.211
DUTH	duth_arq	0.524	0.096	0.278	0.257	0.167
WaterlooClarke	WatACBase	0.527	0.086	0.236	0.295	0.151
DUTH	duth	0.422	0.087	0.235	0.311	0.144
ielab	ielab-bm25AQ	0.445	0.078	0.195	0.246	0.133
ielab	ielab-bertAQ	0.445	0.052	0.147	0.217	0.079
ielab	ielab-bertPRFAQ	0.350	0.043	0.134	0.179	0.078
WLU	WLU_RawUttOnly	0.279	0.010	0.059	0.111	0.022

Table 5: Automatic systems using automatic canonical result context. The first four metrics use cutoff@1000. Binary relevance metrics use relevance scale ≥ 2 as positive.

Group	Run	Recall	MAP	MRR	NDCG	NDCG@3
h2oloo	h2oloo_RUN2	0.705	0.326	0.621	0.575	0.493
h2oloo	h2oloo_RUN1	0.705	0.284	0.576	0.549	0.444
grill	grill_fuseDuo	0.770	0.243	0.584	0.536	0.444
UvA.ILPS	quretecQR	0.264	0.147	0.476	0.283	0.340
HPCLab-CNR	HPCLab-CNR-run3	0.561	0.193	0.449	0.422	0.331
HPCLab-CNR	HPCLab-CNR-run1	0.545	0.181	0.434	0.403	0.313
USI	hist_concat	0.475	0.160	0.424	0.354	0.281
USI	hist_attention	0.475	0.125	0.340	0.321	0.214
UvA.ILPS	quretecNoRerank	0.264	0.081	0.262	0.216	0.171

high-precision and quality responses in the top ranks. The other evaluation measures are computed at the standard 1000 cutoff. For the official results we threshold using a relevance cutoff of **two** as positive, because the value of one is marginal in the guidelines.

We distinguish between three types of runs: *automatic*, *automatic-canonical*, and *manual*. Automatic runs use the provided test topics raw or with automatic rewriting. Automatic-canonical runs use the test topics as well as the provided canonical system responses. Manual runs use the manually rewritten (resolved) queries, where coreference and result dependence are removed to create clear and unambiguous utterances, and also optionally the canonical responses provided (the result dependence is resolved in the manual query).

Table 6: Manual retrieval results. These runs used the manually resolved queries and/or manual canonical results. The first four metrics use cutoff@1000. Binary relevance metrics use relevance scale ≥ 2 as positive.

Group	Run	Recall	MAP	MRR	NDCG	NDCG@3
grill	grill_bmDuo	0.747	0.302	0.684	0.571	0.530
grill	grill_ctxDuo	0.770	0.338	0.655	0.607	0.507
UvA.ILPS	humanQR	0.395	0.241	0.640	0.414	0.498
—	org_manual_bertbase	0.498	0.252	0.652	0.451	0.479
ASCFDA	ASCFDA_qa	0.632	0.282	0.593	0.513	0.466
ielab	ielab-bm25T5QLM	0.526	0.231	0.581	0.447	0.410
POLYU_SOME	man_polyu1	0.693	0.215	0.547	0.488	0.398
—	org_manual_sdm	0.770	0.195	0.479	0.493	0.321
UMass_CIIR	umass_curr_rm3	0.560	0.130	0.364	0.358	0.247
DUTH	duth_manual	0.691	0.153	0.380	0.419	0.243
WLU	WLU_ManUttOnly	0.723	0.037	0.148	0.287	0.067

Automatic run results. Table 4 shows the results for the 33 automatic runs. The median NDCG@3 score of all automatic runs is 0.304. The organizer provided GPT2 + BERT baseline is slightly below the median. For the best runs, there is a three-way tie between the top automatic runs. All of the top runs leverage neural language models for ranking (T5, Albert, and others). Further, all of the top five systems use T5-based re-rankers trained on MS MARCO. As we analyze in more detail below, generative sequence-to-sequence query rewriting with pre-trained language models (T5, BART) is widely used by the top performing runs. There also appears to be a strong relationship between system effectiveness in NDCG@3 and Recall of the overall run. This indicates that rewriting and first-pass retrieval phases are important for overall end-to-end effectiveness.

Automatic-canonical results. There are 9 automatic runs that utilize the provided canonical results. Table 5 shows the results on all measures. The median NDCG@3 is 0.331, higher than the automatic runs. The best performing run for canonical and automatic overall is *h2oloo_RUN2*. It is the only run to outperform the best non-canonical runs. It is also notably the only automatic run to outperform the organizer’s manual BERT baseline. It uses rules to extract sentences from the canonical and retrieved results and incorporate them into the query reformulation process.

Manual run results. Table 6 shows the results for the 9 manual runs. The median run has an NDCG@3 value of 0.410. This is higher than the organizer provided SDM run, but below the organizer BERT baseline. The best performing run is *grill_bmDuo* with an NDCG@3 value of 0.530. It leverages a pairwise DuoBERT reranker on top mono-BERT results with a BM25 baseline. Similar to automatic runs, all of the top performing runs use a pre-trained language model for reranking. It’s noteworthy that the gap between the best manual and automatic runs shrunk significantly in 2020 compared with 2019. The gap between manual and automatic systems is: 24% in the median; 16% for the best automatic, and 8% for the best automatic-canonical.

4.1 Results by turn depth

We now study how the systems performed as the conversation progresses and turn depth increases. Turns beyond ten (up to 13 for some topics) are truncated due to small sample size. The sample size for depth 9 is 10 queries and depth 10 is 6 queries. The results in Figure 1 show the average NDCG@3 at each turn depth. To

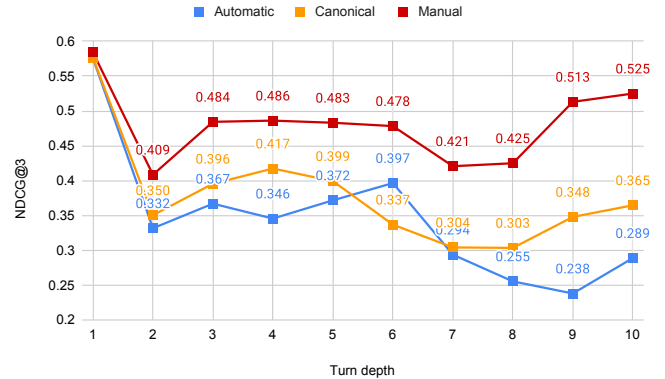


Figure 1: NDCG@3 at varying conversation turn depth.

focus on strong systems the figure only shows data for runs that outperform the organizer baselines - *auto_bertbase* for automatic and canonical - automatic (20), canonical runs (4) and *manual_sdm* for manual (6); organizer runs are not included.

The results show that all systems perform well at the start of a topic. For automatic and canonical runs there is a sharp approximately 30-40% drop for turns two through four compared with the first turn. There is also a drop for manual systems, but it is smaller - 17-30%. This indicates that these queries are harder as the conversation progresses. For automatic runs, the decreasing effectiveness trend continues, with it approximately halved at the end compared with the first turn. The automatic-canonical results also decrease, but not as much as the automatic runs.

In contrast, the results for manual runs show a different pattern. Effectiveness dips at turn 2 (by 30%), but then recovers and remains (mostly) stable. For all classes we observe a small decrease in effectiveness around turns 7 and eight, possibly due to topic shifts. The gap between manual and automatic systems appears to widen as the turn depth increases.

4.2 Results by Topic

Figure 2 provides a per-topic analysis comparing the three classes of systems across topics. It uses data from all submitted runs. The results show that the the topic difficulty varies widely across topics.

5 RESULTS ANALYSES

We encouraged the groups to submit metadata along with their runs. We designed a questionnaire with Yes/No questions in three categories: Query Understanding Methods/Data, Training/Retrieval Models Methods/Data, and the Utilization of Context information. Each question asks the teams to provide whether their run uses a specific type of resource or technique. This section discusses the impact of the varying approaches and resource usage on system effectiveness. We use all automatic runs - both automatic and automatic-canonical. This year we focus on the use of context. A complete breakdown of methods and their influence is included in Appendix A.

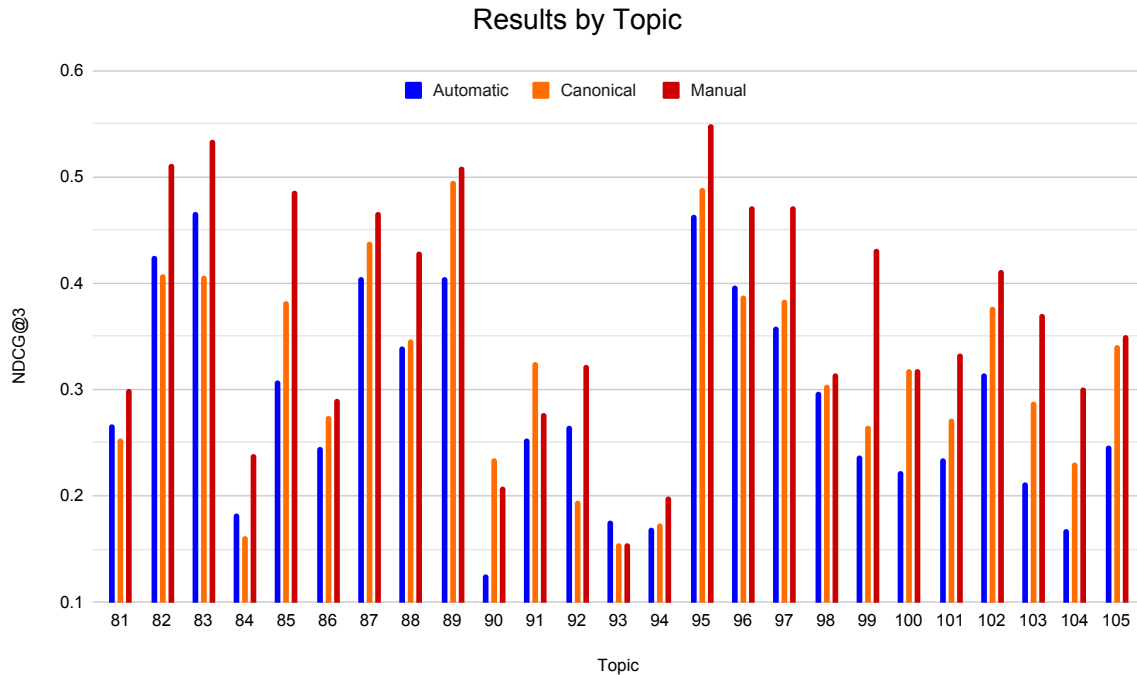


Figure 2: NDCG@3 aggregated for each topic.

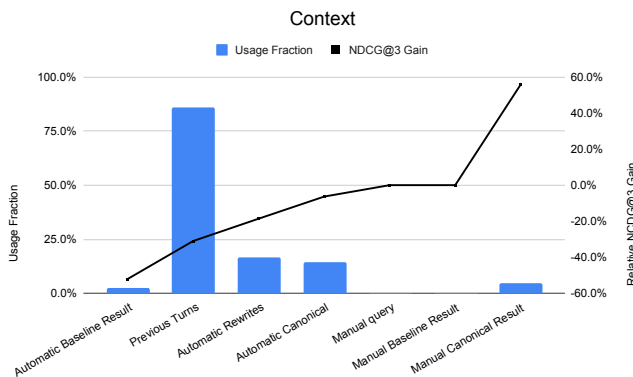


Figure 3: Retrieval context influences on system effectiveness for automatic runs.

5.1 Conversational Context

A key challenge in conversational search is to infer information needs through the use of the conversation history. Year two represents a shift from the first year in that the title and description are not provided. In addition, some turns explicitly rely on result context provided in the single canonical retrieved result (provided for ‘canonical’ runs). Below are the questions on context usage.

- (1) *Previous Raw*. The method uses the previous raw utterances before the current turn.

- (2) *Previous automatic*. The method uses the previously automatically rewritten queries in the topic.
- (3) *Previous manual*. The method uses the previous manual utterances for the topic.
- (4) *Canonical result*. The method uses the previous canonical results provided for the topic.
- (5) *Automatic result*. The method uses the provided automatic baseline results for the topic.
- (6) *Manual results*. The method uses the previous manual baseline results for the topic.

The results in Figure 3 show the impact. It shows that almost all runs used the previous turn history (86%). Despite this, on average there is a 31% decrease over teams that did not use them. The runs that did not use the previous turns instead used the provided automatically rewritten queries and performed competitively. The biggest gain is for runs that use the manual canonical results. We note that these results rely on the (noisy) team self-assigned metadata.

5.2 Explicit Query and Turn Dependence

In this section we study the effect of context. In the organizers’ dependence annotation, there are 118 queries that depend on a query from a previous turn in the conversation. There are 56 queries that depend on results from previous turns. Note that a turn may depend on multiple previous queries, results, or a combination of both.

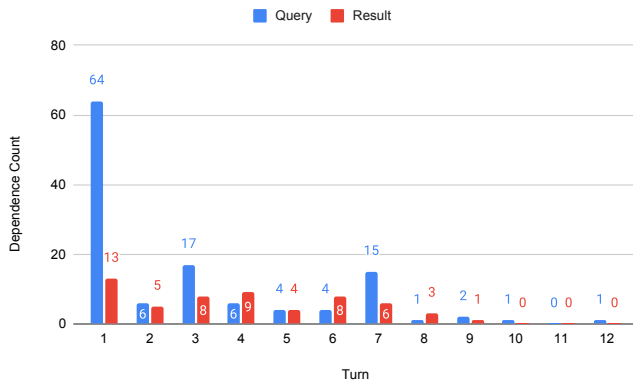


Figure 4: Statistics on the source of contextual information in query and response (used by later turns) by turn depth.

Table 7: Dependence results on automatic and canonical runs. We report the average across runs outperforming the organizer’s baseline org_auto_bertbase.

Dependence	Turns	Auto. NDCG@3	Canon. NDCG@3
All turns	208	0.360	0.384
None	45	0.463	0.473
Query	118	0.379	0.387
Query (hard)	22	0.375	0.329
Result	56	0.226	0.285
Result (hard)	5	0.217	0.235

Figure 4 shows the breakdown of these dependencies by turn depth is shown. It shows source turn that is referenced. Unsurprisingly, the figure shows that there is a strong dependence on the first turn. But, it also highlights a significant number of dependencies to turns deeper in the conversation. Further analysis shows a strong dependence on the previous turn, with over two thirds of the query dependencies being to the immediate previous turn. Less than a fifth of the query dependencies are *hard*, defined to be references not to the first or immediate preceding turn.

Table 7 shows the results broken down by different types of conversational context. Some turns (for example all first turns) do not rely on a previous turn at all. We label these *None* and they represent approximately 20% of turns. Others only depend on previous utterances *Query*. Over half of all turns depend on a previous utterance with the majority of these being the first turn. We also further split the dependence into a *Query hard* subset (approximately 10% of turns) where there is a dependence on a previous query that is not the first turn and not the immediate preceding turn. The bottom two rows focus on result dependence. The *Result* type has 27% of turns and *Result Hard* has 2% of turns. Similar to queries, the hard variant for results is result dependence beyond the first or immediate preceding turn.

The results show that turns without dependence are the easiest and systems perform the best on this subset (the best run is h2oloo_RUN2 with a mean NDCG@3 value of 0.571 on this subset). Turns with a query dependence perform in the middle - between

turns without dependence and all turns. Notably, the hard variant performs only slightly worse than all query dependencies.

The most significant observation is that turns with result dependence perform much worse than those with query dependence (a 40% relative reduction compared with query dependence). We also observe that systems that use the results (canonical) perform only slightly better than those that don’t. On the result dependence queries the h2oloo_RUN2 performs the best with a NDCG@3 value of 0.403 (vs 0.571 on queries without context). The only automatic run that outperforms the canonical mean (and median) on this subset of queries is h2oloo_RUN3 (NDCG@3 is 0.291). Interestingly, the best performing automatic run h2oloo_RUN4 performs worse on this subset (0.267). This indicates that methods to result dependence is an area that needs improvement for the future.

6 CONCLUSION

The second year of TREC CAsT continued developing resources for studying conversational information seeking and added to the community’s understanding of the topic. Conversations became a little more natural, and contextual information became more complex. The number of assessed conversations and queries increased, providing further training and evaluation data.

- **Conversational Language Understanding.** Instead of traditional NLP pipelines this year we observe the dominance and effectiveness of sequence-to-sequence neural rewriting methods introduced in Y1. In particular, the training of query rewrite models with T5 and BART is now feasible with available datasets (CAsT Y1 or CANARD).
- **Conversational Context.** The results on the manual runs show that clean context has the potential to maintain or even improve effectiveness over the course of the conversation as an information need unfolds. This year we find the most effective automatic run is able to use result context, but methods are still primitive and most simply ignoring context. We find that result dependence is more challenging for current systems than query dependence.
- **Ranking.** The use of pre-trained neural language models for ranking is the de facto standard for the most effective systems. We observe that recent advances to larger models and ones that go beyond pointwise ranking.
- **Automatic vs Manual.** The gap between automatic and manual methods shrunk for the median systems. This indicates that automatic rewriting methods made significant gains. However, automatic methods show a significant degradation in the effectiveness as turn depth increases. There remains a large gap between automatic and manual systems at deeper turn depths.

After the success of year two we look forward to year three.

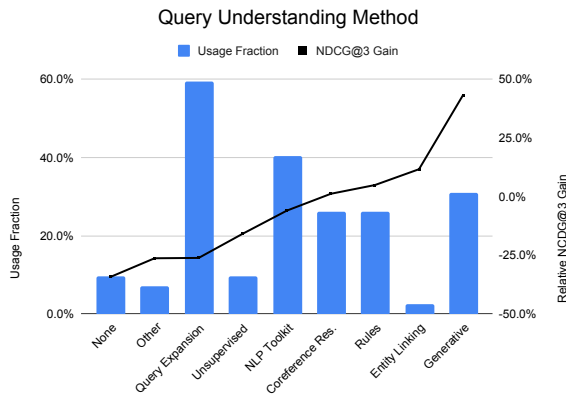


Figure 5: Query understanding method influences on system effectiveness for automatic and automatic canonical runs.

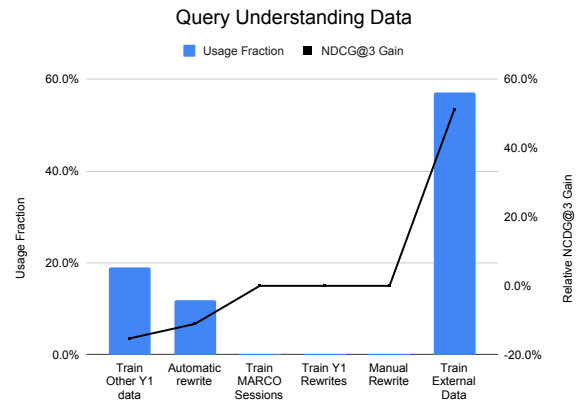


Figure 6: Query understanding data influences on system effectiveness for automatic and automatic canonical runs.

7 ACKNOWLEDGMENTS

We thank Vaibhav Kumar and Carlos Gemmel for their work in developing topics. Vaibhav Kumar created the BERT reranker that generated canonical rankings. We thank Cameron VandenBerg for setting up the Solr instances that generated the initial rankings. We thank Shi Yu for running the automatic query rewriter on Year 2 topics. We thank the Deep Learning track organizers for helping align the two tracks. We also are deeply thankful for Ellen Voorhees’ experience, patience, and persistence in running the assessment process. Finally we thank all our participants.

REFERENCES

- [1] Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., et al.: Ms marco: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268 (2016)
- [2] Dalton, J., Xiong, C., Callan, J.: Cast 2019: The conversational assistance track overview. In: The Twenty-Eighth Text REtrieval Conference Proceedings (TREC 2019). National Institute of Standards and Technology, special publication (2020)
- [3] Dalton, J., Xiong, C., Kumar, V., Callan, J.: Cast-19: A dataset for conversational information seeking. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1985–1988. ACM (2020)
- [4] Dietz, L., Gamari, B., Dalton, J., Craswell, N.: Trec complex answer retrieval overview. In: The Twenty-Seventh Text REtrieval Conference Proceedings (TREC 2018). NIST Special Publication 500-331 (2019)
- [5] Nogueira, R., Yang, W., Cho, K., Lin, J.: Multi-stage document ranking with BERT. CoRR abs/1910.14424 (2019)
- [6] Rosset, C., Xiong, C., Song, X., Campos, D., Craswell, N., Tiwary, S., Bennett, P.: Leading conversational search by suggesting useful questions. In: Proceedings of The Web Conference 2020. pp. 1160–1170 (2020)
- [7] Yu, S., Liu, J., Yang, J., Xiong, C., Bennett, P., Gao, J., Liu, Z.: Few-shot generative conversational query rewriting. arXiv preprint arXiv:2006.05009 (2020)

A META-ANALYSES

We provide additional analysis of the results across metadata dimensions for reference purposes and comparison with previous analyses. These include all automatic runs – both automatic and automatic-canonical.

A.1 Query Understanding

The query understanding methods show the techniques used in understanding the conversational queries, including but not limited

to query rewriting, term re-weighting, query expansion, coreference resolution, and others.

The first set of questions is on query understanding methods.

- (1) *None*. Whether no query understanding method is used.
- (2) *Supervision*. Whether the method uses any training data.
- (3) *Rules*. Whether the method uses heuristic rules.
- (4) *NLP Toolkit*. Whether the method uses a standard NLP toolkit.
- (5) *Entity Linking*. Whether the method uses entity linking.
- (6) *Coreference*. Whether the method uses coreference resolution.
- (7) *Expansion*. Whether the method performs query expansion.
- (8) *Rewriting*. Whether the method performs generative query rewriting.
- (9) *Other*. Whether the method uses other understanding methods.

Figure 5 shows the use and effect of varying query understanding methods. Runs that did not use query understanding had the biggest loss, approximately 34% compared with those that did. The most commonly used approach is a query expansion with approximately 60% of runs, but the effect was mixed with runs using it performing approximately 26% worse than teams that did not. The approach that appears to have the greatest positive effect is the use of generative rewriting models, 30% of runs with a gain of 43% relative to those that did not. Entity linking also shows a positive 12% gain, but was used by only one run.

The second set of query understanding focuses on the datasets used to train query understanding and rewriting methods, including CAsT Y1, and others.

- (1) *CAsT Y1*. Whether the method uses data provided in CAsT Y1
- (2) *MARCO Sessions*. Whether the method uses data provided in the MARCO Conversational Sessions dataset.
- (3) *Y1 rewrites*. Whether the method is trained on CAsT Y1 manual query rewrites.
- (4) *Y2 automatic*. Whether the method uses the CAsT Y2 automatically rewritten queries.

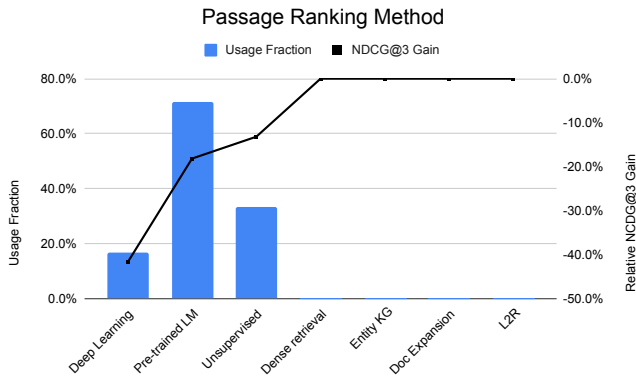


Figure 7: Retrieval method influences on system effectiveness for automatic and canonical runs.

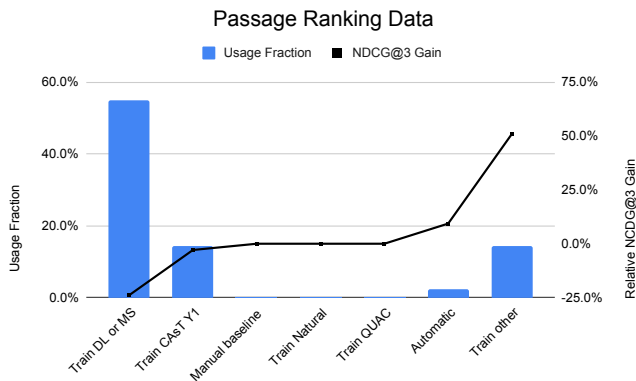


Figure 8: Training data influences on system effectiveness for automatic and canonical runs.

- (5) *Y2 Manual*. Whether the method uses CAsT Y2 manually rewritten queries.
- (6) *External*. Whether the method uses an external dataset.

The results in Figure 6 show the datasets used. It shows that using other forms of data resulted in a relative 15% decrease. In contrast, external datasets (such as CANARD) are commonly used and show a relative 51% gain. Note that external datasets were interpreted by some groups to include language models pre-trained on external collections (e.g. T5).

A.2 Retrieval and Ranking

We also looked at the effect of different ranking methods and datasets used for training.

We first look at ranking methods. The first set of questions focused on the retrieval method.

- (1) *Unsupervised*. Whether any training data has been used.
- (2) *Dense retrieval*. Whether a vector-based model is used for retrieval.
- (3) *Entities*. Whether an entity-knowledge graph is used.
- (4) *Document Expansion*. Whether document expansion is used.
- (5) *L2R*. Whether the method uses a learning to rank method.
- (6) *Neural*. Whether the method uses a neural ranking model
- (7) *Pre-trained LM*. Whether the method uses a pre-trained language model (e.g. BERT).

The fraction of each technique used and their influence for automatic runs is shown in Figure 7. The result is clear – teams use pre-trained neural language models heavily with 71% of runs using them for ranking. However, we observe that their effect is varies widely and is quite mixed, with an average of 18% decrease – despite them being used in all of the top performing systems. There was no method that had a positive effect overall, which is surprising.

We now examine the training data used to train / fine-tune the models used.

- (1) *Auto baseline*. Whether the method uses the provided automatic baseline run.
- (2) *Manual baseline*. Whether the method uses the manual baseline run.
- (3) *CAsT Y1*. Whether the model is trained on CAsT Y1 data.
- (4) *DL / MARCO Training*. Whether the method is trained with Deep Learning or MS MARCO dataset.
- (5) *Natural Questions*. Whether the model is trained using the Google Natural Questions dataset.
- (6) *QUAC*. Whether the model is trained on the Question Answering in Context (QUAC) dataset.
- (7) *Other Training*. Whether the method is trained with other datasets.

The results are shown in Figure 8. It shows that the majority (55%) of runs used data from MS MARCO and/or the Deep Learning track, followed by CAsT Y1 (14%). Overall, runs trained on DL / MS MARCO had a decrease in effectiveness of approximately 24%. Inspecting the run descriptions, it appears that most of the Train Other runs actually used MS MARCO and the metadata is incorrectly reported. The remaining runs are trained on other QA datasets (notably MRQA).