TREC Complex Answer Retrieval Overview

Laura Dietz,[*] Ben Gamari, Jeff Dalton, Nick Craswell

Homepage: `http://trec-car.cs.unh.edu`
Google group mailinglist: `TREC-CAR`

1 Introduction

The SWIRL 2012 and 2018 workshops on frontiers, challenges, and opportunities for information retrieval [1, 3] noted many important challenges, such as generating information objects, sub-document retrieval, answer aggregation, retrieving material for conversational agents. These challenges share commonalities: We desire answers to complex information needs, and wish to find them in a single and well-organized page. Such a page may not yet exist, but it rather needs to be synthesized from multiple information sources. Advancing the state of the art in this area is the goal of this TREC track.

Such a tool would benefit users that investigate new and unfamiliar topics. It would support information access in mobile environments with restricted interaction capabilities. In contrast to extensive work on finding the best short answer, this track reaches towards the retrieval of longer answers that cover a range of different subtopics. We envision answers to be composed of multiple text fragments from multiple sources, recycling information about related topics, but selected to highlight insightful connections. CAR tools would support automatic authoring of pages on new topics.

Retrieving high-quality comprehensive answers is challenging as it is not sufficient to choose a lower rank-cutoff with the same techniques as for short answers. Instead, we need new approaches for finding and organizing relevant information units of a complex answer space.

Many examples of manually created complex answers exist on the Web: how-stuff-works.com, travel guides, fanzines, or educational text books. These are collections of articles where each constitutes a long answer to an information need expressed by the title of the article. Ideally, by reading an article, users will gain new information about the topic. We can measure this information gain by testing how well the text enables users to answer questions about the topic.

The fundamental task of collecting references, facts, and opinions into a single point-of-entry has traditionally been a manual process. We envision that automated information retrieval systems can relieve users from a large amount of manual work though sub-document retrieval, consolidation and organization. Ultimately, the goal is to retrieve synthesized *information* rather than documents.

2 A Worked Example

To motivate a brief example, consider a user wondering about how coffee preparation techniques lead to different tastes. With this intention in mind, she enters the query "Coffee preparation". A possible answer could look as follows:

Coffee Preparation

Grinding
Arabic coffee and Turkish coffee require that the grounds be almost powdery in fineness, finer than can be achieved by most burr grinders. Pounding the beans with a mortar and pestle can pulverize the coffee finely enough.

[*] dietz@cs.unh.edu

> *The fineness of the grind strongly affects brewing. Brewing methods that expose coffee grounds to heated water for longer require a coarser grind than faster brewing methods.*
>
> **Steeping**
>
> *The softer flavors come out of the coffee first and the more bitter flavors only after some time, so a large brew will tend to be both stronger and more bitter.*
>
> *Another variation is cold brew coffee, sometimes known as "cold press." Cold water is poured over coffee grounds and allowed to steep for eight to twenty-four hours. This process produces a very strong concentrate which can be stored in a refrigerated, airtight container for up to eight weeks.*

Here "grinding" and "steeping" are two query facets. After reading this article, the user is able to tell that the taste is affected by the fineness of the ground, the time of exposure, and the temperature.

Of course, one might envision other responses that would satisfy the information need equally well. While this example was taken from Wikipedia[1] it should be possible to identify such information from a Web collection with passage retrieval, consolidation, and organization.

3 Task Description

While the long-term goal of this track is to retrieve complex answers without any more information than the given complex topic, in the first year we focus on a simpler task, where both the topic and an appropriate outline is provided as a query. An example outline is given in Figure 1. We run two tasks: passage and entity.

Passage Task: Given an outline for complex topic outline Q, retrieve for each of its sections H_i, a ranking of relevant passages S.

Entity Task: Given outline for complex topic Q, retrieve for each of its sections H_i, a ranking of relevant entities E and with support passages S. These support passage should motivate the why the entity is relevant for the query.

The passage S is taken from the provided passage corpus. The entity E refers to an entry in the provided knowledge base. We define a passage or entity as relevant if the passage content or entity is appropriate for the knowledge article.

4 TREC CAR Data Set (v2.1)

The 2018 Complex Answer Retrieval track uses topics, outlines and that are extracted from English Wikipedia (XML dump from June 2018) and the Text Book Question Answering (TQA) dataset[2]. Paragraphs, entities and training data is extracted from previous years's English Wikipedia (XML dump from Dec 20th, 2016). Wikipedia articles are split into the outline of sections and the contained paragraphs.

All paragraphs from all articles are gathered and deduplicated to form the paragraph corpus. Due to a bug fix in the Wikipedia parser the paragraph collection for Y2 is much larger and cleaner than in Y1.

```
Title: Protecting the Water Supply
1. Rationing Water
2. Reducing Water Pollution
3. Saving Water in Irrigation
4. Conserving Water
5. Water Treatment
6. What You Can Do
7. Saving Water at Home
8. Controlling Water Pollution
```

Figure 1: Example outline for TQA topic "Protecting the Water Supply".

[1] https://en.wikipedia.org/wiki/Coffee_preparation

[2] Available at http://data.allenai.org/tqa/

MUST be mentioned:
Many water-saving devices (such as low-flush toilets) that are useful in homes can also be useful for business water saving. Other water-saving technology for businesses includes:

CAN be mentioned:
Recycling one gallon of paint could save 13 gallons of water, 1 quart of oil, and 250,000 gallons of water pollution, 13.74 pounds of , save enough energy to power the average home for 3 hours, or cook 6 meals in a microwave oven, or blow dry someone's hair 27 times.

Roughly on TOPIC but non-relevant
Dual piping is a system of plumbing installations used to supply both potable and reclaimed water to a home or business. Under this system, two completely separate water piping systems are used to deliver water to the user. This system prevents mixing of the two water supplies, which is undesirable, since reclaimed water is usually not intended for human consumption.

Paragraph IDs:
left: dbcef592762b4711012041f6bdf1bdff7cb5a521
center: f26730da3b7860c727411480b08ae6466dcc9a54
right: 21e6e381383e392cb7d1432200c51c095cdf3fbe

Figure 2: Example passages and relevance for "Protecting the Water Supply / Saving Water at Home" (Query ID "tqa:protecting%20the%20water%20supply/Saving%20Water%20at%20Home").

Each section outline is a description of a complex topic. By keeping the information which paragraph originates from which article and section, we have a means of providing training data for the passage retrieval task. By preserving hyperlinks inside paragraphs that point to Wikipedia pages (also known as entities in the DBpedia knowledge graph), we have a means of providing training data for the entity retrieval task.

A difficulty in this year's TREC CAR evaluation is that the paragraphs and outlines were taken from a different collections. After filtering and processing procedures described in Section 4.1, several datasets for training and evaluation are derived. As depicted in Figure 3, only a small subset of Wiki-18 pages in the benchY2test dataset contained paragraphs available in the provided collection. This allows us to study how well Wikipedia content can be recycled to populate articles on new and unseen topics. A downside is that the "automatic evaluation" option from Y1 cannot be applied as the provided corpus (based on Wiki-16) does not include the paragraphs of Wiki-18 and TQA. The size of the datasets is given in Table 1. The paragraph collection contains 29,678,367 unique paragraphs.

4.1 Data Set Creation Pipeline

The TREC Complex Answer Retrieval benchmark (v1.5) is derived from Wikipedia so that complex topics are chosen from articles on open information needs, i.e., not people, not organizations, not events, etc. However, any paragraph or entity on Wikipedia is a legal paragraph/entity for the retrieval task even if a person entity or a paragraph from an article on an event. The data set creation process is as follows:

1. Mediawiki format of each article in the Wikipedia dump is parsed, preserving paragraph boundaries, intra-Wikipedia hyper links, and section hierarchy. The TQA collection is provided preprocessed in

	benchY1train	benchY1test	train	benchY2test **
number of articles (complex topics)	117	133	285,924	27
hierarchical sections (queries)	1,816	2,125	2,180,868	269 / 271
total positive paragraphs assessments	4,530	5,820	5,276,624	9,633
total positive entity assessments	13,031	15,085	12,310,616	1,356

Table 1: Data set sizes in terms of articles, section, and automatic positive assessments. (**) used in this years' evaluation.

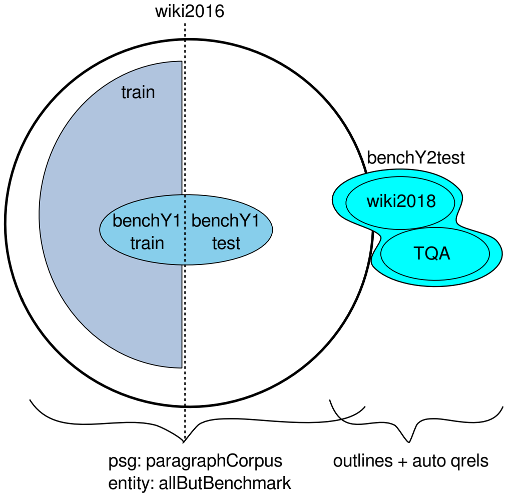


Figure 3: Y1 and Y2 datasets. Manual Qrels for benchY1test were collected in the previous year's evaluation.

JSON format.

2. Name information from redirect and disambiguation pages and Category information added to the article page. Redirects in hyperlinks are resolved. Templates, talk pages, portals, disambiguation, redirect, and category pages are discarded. (Disambiguation, redirects, and category information only provided in training set).
3. Articles in benchY1train, benchY1test, benchY2test, and test200 are witheld, all remaining articles in Wikipedia 2016 are released as a basis for entity retrieval **(allButBenchmark).**
4. Articles tagged with categories that indicate people, organizations, music, books, films, events, and lists are discarded.
5. Sections with headings that do not contain prose are discarded, for example external links, references, bibliography, notes, gallery etc.
6. Each article is separated into (1) the outline of section headings and (2) paragraphs.
7. The set of paragraphs across all of Wiki-16 are collected, and unique paragraph IDs are derived through SHA256 hashes on the text content (ignoring links).
8. The paragraphs are further deduplicated with min hashing using word embedding vectors provided by GloVe. For each set of duplicates, one representative paragraph is chosen.
9. The collection representative paragraphs is released as the **paragraphCorpus**.
10. Articles are rewritten, replacing paragraphs with duplicates with the representative paragraph.
11. The set of articles is further filtered to remove images, lead sections, sections with very long (>100 character), and very short headings (<3 letters). Articles with less than three remaining sections are discarded.
12. The set of Wiki-16 articles is split into training and holdout data, training data is further split into five folds. To ensure uniform distribution and reproducibility, these decisions are made based on the SipHash of the article title.
13. The five folds of the training data, with separated outlines and paragraphs and extracted automatic qrels are made available as **train**.
14. The set of pages used for benchmarks used in Y1 (**benchY1train, benchY1test, test200**) were re-processed with the new Wikipedia parser, manual judgments were translated to new paragraph ids, the re-released.
15. A manual selection of articles in Wikipedia 2018 and the TQA corpus were only released as outlines as **benchY2test.public**. Official contributed runs were submitted on these topics. A complete **benchmarkY2test** will be released after the TREC workshop. It will contain:

    - Manual ground truth for paragraphs (qrels)

	Paragraph		Entity	
	Automatic	Manual	Automatic	Manual
train	yes	-	yes (original entity)	-
benchmarkY1train	yes	-	yes (TagMe)	-
benchmarkY1test	yes	yes	yes (TagMe)	no
benchmarkY2test Wiki-18	no, too small	yes	yes (TagMe)	yes
benchmarkY2test TQA	no	yes	yes (TagMe)	yes

Table 2: Availability of manual and automatic assessments for different benchmarks.

- Automatic and manual ground truth for entities (qrels)
- Original articles

4.2 Automatic Ground Truth

Two kinds of ground truth signals are collected: automatic and manual. For each, we release true paragraphs and true entities. While the manual ground truth is assessed after participants submit runs, the automatic ground truth is derived along with the dataset from the Wikipedia/TQA dump.

For the benchmarkY1 datasets, the automatic ground truth is derived as follows

- If a paragraph is contained in the page/section it is defined as relevant, and non-relevant otherwise.
- If the page/section contains an entity link, then the (link target) entity it is defined as relevant, and non-relevant otherwise. For benchmarkY1train, benchmarkY1test, and benchmarkY2test, the TagMe[5] entity linker was used. For the much larger “train” collection, entity links that were manually added by Wikipedia editors were used.

Qrels are derived with several different levels:

- Hierarchical: Only content of leaf sections in considered.
- Article: All content of the page is considered (independent of the section). However, due to a mistake in processing, lead text is missing.
- Top-level: Only relevance for top-levels sections is provided. All content in this section or a child section is considered relevant.
- Tree: For all page titles and headings, all content in the subtree is considered relevant. Tree qrels also contain Article-level relevance assessments, although these are not part of the TREC CAR evaluation.

The benchmarkY2 dataset is constructed from pages outside the Wiki-16 dump (depicted in Figure 3). Only a small fraction of paragraphs on Wiki-18 pages already existed in Wiki-16 (yet on a different page). The paragraph sets from TQA and Wiki-16 are disjoint. Thus, the automatic evaluation procedure for paragraphs, used in Y1, is not applicable to the Y2 dataset.

The automatic entity ground truth is available on benchmarkY2test as it is constructed from entity links and does not rely on an overlap in paragraphs. An overview for which benchmarks on which kinds of ground truth is available is given in Table 2

5 Submission

For the passage ranking task, participants were asked to submit a ranking of paragraph IDs per heading in the outlines of benchY2test. For the entity ranking task, participants were asked to submit a ranking of entity IDs per heading in the outlines of benchY2test. To support assessing entity relevance, participants were asked to provide a paragraph clarifying why the entity is relevant for this section. This helps to arrive at consistent annotations for each entityId.

Table 3: Assessment scale for manual assessments. Horizontal line: Cutoff for positive/negative assessments.

	binary scale	manual scale	manual lenient scale
MUST be mentioned	1	3	5
SHOULD be mentioned	1	2	4
CAN be mentioned	1	1	3
Non-relevant, but roughly on TOPIC	0	0	2
NO, non-relevant	0	-1	0
Trash	0	-2	-2

Participants were allowed to consider all headings in the outline at once, use external resources such as knowledge graphs, entity linking tools, pre-trained word embeddings, and any of the provided TREC CAR data sets. **The participants were not allowed to directly use a dump of Wikipedia**, as this would allow them to look up the paragraphs on the page—the information used in the automatic ground truth.

Each participating team was allowed to submit up to three runs to the passage task and three runs to the entity task. Nine teams participated in this second year of the track.

6 Assessment of the Manual Ground Truth

The five elements of participant contributed runs were merged to build the assessment pools for each heading in the query outlines. Additionally, paragraphs and entities that are relevant according to the ground truth were added to the pool for verification. In the previous year, complex topics were only partially judged (to obtain a larger topical variety). In contrast, this year, all sections of complex topics were assessed. This yielded manual assessments for 65 complex topics (31 TQA and 34 Wiki-18). One section was annotated by all assessors in order to measure inter-annotator agreement across the six NIST assessors (cf. Section 6).

For the passage task, the assessor is presented with the complex topic (page title) and the topic sections as a heading hierarchy, followed by a randomized list of paragraphs from the assessment pool. In the case of the entity task, the list displayed the canonical entity names together with the provenance paragraph if given. As not all participants submitted provenance, the list also displays an entry of the canonical entity name together with first paragraph from the entity's Wikipedia pages as provenance. This information was intended to support the assessment process. However, the first paragraph of the entity's Wikipedia page turnedout to be not useful. As only one team submitted passage provenance in entity rankings, the assessors had no choice by to resort to world-knowledge.

Assessors were asked to envision writing a Wikipedia article on the given complex topic. A graded assessment scale was used based on how much the paragraph/entity should be mentioned in this section of the article.

- MUST be mentioned
- SHOULD be mentioned
- CAN be mentioned
- Non-relevant, but roughly on TOPIC of the page
- NO, non-relevant
- Trash

The "Trash" grade is assigned to paragraphs/entities have a low quality or no topical dependence and therefore would not be relevant for any topic imaginable.

Label distribution

Six assessors created 13,616 passage annotations on a total of 269 topic sections for passages. The grade histogram per annotator and the overall grade distribution is given in Table 4a. We notice that only a third of all assessments are graded as relevant, while an additional third were annotated as being on topic.

Table 4: Grade histogram and distribution.

(a) Manual judgments overall.

	annotator1	annotator2	annotator3	annotator4	annotator5	annotator6	Total %
Trash	40	118	0	14	5	2	1
No	1213	980	810	1124	801	1066	42
Topic	439	674	613	361	770	946	27
Can	241	489	469	261	212	115	13
Should	210	304	131	305	140	472	11
Must	145	214	65	288	191	51	7

(b) Manual judgments on relevant entities.

	annotator1	annotator2	annotator3	annotator4	annotator5	annotator6	Total %	TQA	Wiki2018
Trash	0	7	0	0	0	0	1	2	5
No	1	11	0	2	0	4	4	2	16
Topic	7	13	2	0	8	7	8	3	34
Can	3	29	7	3	3	0	9	8	37
Should	3	19	20	20	15	44	25	28	93
Must	62	25	37	25	91	17	53	86	171

For 271 topics sections, entity assessments were created with 8415 assessments in total. Histogram and distribution is given in Table 4b.

70% of passages (62% of entities) in the assessment pool were marked as non-relevant. This demonstrates that the task is feasible, but challenging.

Inter-annotator agreement

One section (both passage and entity) was selected for annotation by all assessors to measure inter-annotator agreement in the middle of the assessment period.

We measure inter-annotator agreement using Cohen's κ for pairwise comparison and Fleiss' κ across all annotators. We analyze agreement on the derived graded assessments counting "off by one" as agreement. (Table 5a), e.g., grades "SHOULD" and "MUST", are also counted as agreements for both for p_0 and p_e.

Inspecting Cohen's κ, we find that on the whole the pair-wise agreement is relatively similar across all pairs of assessors. In other words, there is no "odd one out" which speaks to the quality of NIST's assessment procedures. As expected, the agreement for binarized paragraph judgments (Fleiss $\kappa = 0.500$) is higher than for graded judgments (Fleiss $\kappa = 0.304$). For entity judgments, we find that Annotator 2 was an outlier, after removing assessment from Annotator 2, binarized agreement is Fleiss $\kappa = 0.416$, and graded Fleiss $\kappa = 0.374$. This may sound small, yet it is comparable to previous work [2]. However, once neighboring grades are counted as agreement ("off by one"), the inter-annotator agreement is even a bit higher agreement on binarized assessments.

We conclude that, aside from subtle nuances in the grading scale, assessors agree on the whether the passage or entity should be included in the article on the complex topic.

Annotation Time

Over the course of two weeks, six assessors were hired for 40 hours each, yielding a total of 240 hours. Including breaks and training, the average annotation time per passage or entity judgments depends on the annotator and ranges between 17 and 64 seconds (median: 30 seconds).

7 Participant Submitted Runs

In total nine teams contributed runs.Below, detailed descriptions of submitted runs:

Table 5: Inter annotator agreement according to Cohen's κ on graded scale, counting grades that are "off by one" as agreement.

(a) Passage assessment.

	annotator1	**annotator2**	**annotator3**	**annotator4**	**annotator5**	**annotator6**
annotator1		0.687	0.430	0.781	0.628	0.449
annotator2	0.687		0.568	0.632	0.644	0.871
annotator3	0.430	0.568		0.409	0.422	0.859
annotator4	0.781	0.632	0.409		0.407	0.740
annotator5	0.628	0.644	0.422	0.407		0.512
annotator6	0.449	0.871	0.859	0.740	0.512	

(b) Entity assessment.

	annotator1	**annotator2**	**annotator3**	**annotator4**	**annotator5**	**annotator6**
annotator1		0.342	0.542	0.708	0.705	0.134
annotator2	0.342		0.526	0.096	0.500	0.514
annotator3	0.542	0.526		0.422	0.716	0.457
annotator4	0.708	0.096	0.422		0.643	0.467
annotator5	0.705	0.500	0.716	0.643		0.408
annotator6	0.134	0.514	0.457	0.467	0.408	

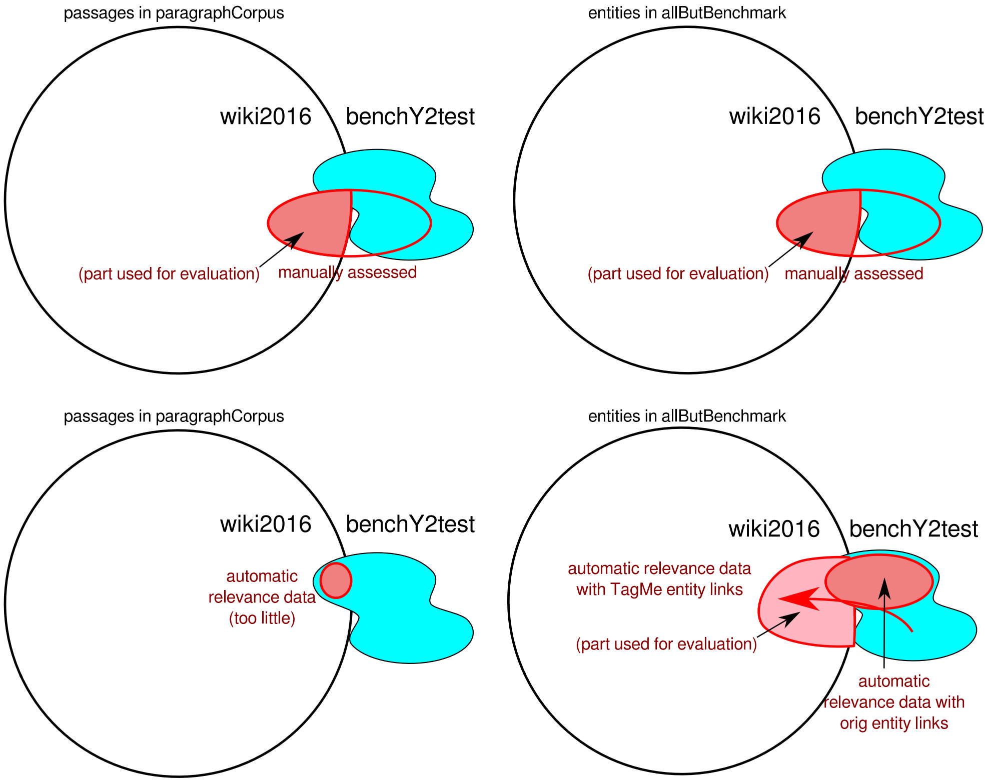

passages in paragraphCorpus
wiki2016
benchY2test
(part used for evaluation)
manually assessed
entities in allButBenchmark
wiki2016
benchY2test
(part used for evaluation)
manually assessed
passages in paragraphCorpus
wiki2016
benchY2test
automatic relevance data (too little)
entities in allButBenchmark
wiki2016
benchY2test
automatic relevance data with TagMe entity links
(part used for evaluation)
automatic relevance data with orig entity links

Figure 4: Manual (top) versus automatic (bottom) benchmarks for passages (left) and entities (right). The black circle marks the paragraph/entity collection, blue area the gold articles, and the red circle marks the set of assessed paragraphs. the light pink area makes entities determined as relevant via TagMe entity links. The automatic passage collection has too few entries and is not included in this study.

- guir: The guir team submitted two passage retrieval runs. The first run (`guir`) is an approach based off [6], in which an ad-hoc neural ranking architecture is modified for the CAR task by incorporating heading frequency statistics from the training data and by incorporating separate matching phases for each heading in the query. The second run (`guir-exp`) uses the same approach, and adds the top-scoring query expansion terms for each heading as determined by a learned match-gating mechanism. The models were trained on the automatic relevance judgments from the train dataset, and validated using the manual relevance judgments on the benchmarkY1-test dataset.

- NYU: Lucene is the underlying retrieval engine. We train 20 query reformulators on random disjoint partitions of the training set as in Nogueira et. al., 2018 [7]. For each query, each reformulator produces a list of ranked documents. We re-rank the union of these 20 lists using a simple ranking model that scores each query-document pair using a feed-forward neural network whose input is the concatenation of the average word embeddings of the query and document. To further improve the performance of the system, we train an ensemble of 9 ranking models whose network architectures are randomly chosen. For each query, we re-rank the union of the 9 lists produced by the 9 ranking models using the best ranking model in the ensemble. All the models are trained on the first 4 folds of TREC-CAR v2.1 queries and the last fold is used for dev/hyperparameter tuning. Y1 test queries are used as test queries. We use paragraphCorpus v2.0 to retrieve documents for train, dev, and test queries.

- CG: We developed a deep learning model for information retrieval TREC Complex Answer Retrieval (CAR) task. We used an attention based sequence-to-sequence model to first translate content passages to outlines. Then across all extracted outlines, use sentence embedding to rank outlines based on the available query. We used attention based bidirectional LSTM model for encoder and decoder layers. In order to capture rare words while limit our dictionary size, we used Byte Pair Encoding subword units to tokenize sentences. The main advantage of using seq2seq model is to preform main inference computation offline and only use the model to re-rank all outlines based on the query in inference time which is much faster compare to recent neural IR models. Using attention based model also provides position-dependent information required to assess the relevance of a snippet of a document to a given query. Attention signals illustrate term dependencies between query and given passage. The model is trained on TREC V2 data set which has 50% of Wikipedia Articles and test on TREC benchmarkY1 which is official evaluation topics for TREC CAR task.

- UTD: We extended our approach from last year to create the TRANSformer Complex Answer PAragraph Retrieval (TRANS-CAPAR) system to perform complex answer retrieval consisting of the following five modules: (1) The Paragraph Indexing Module creates a searchable index of paragraphs from Wikipedia articles; (2) The Query Processing Module processes a Wikipedia article outline into a set of queries - one for each section of the outline; (3) The Paragraph Search Module searches each query against the paragraph index, resulting in a list of relevant paragraphs for each section in the article outline; (4) The Feature Extraction Module is used to extract features from each paragraph; (5) The Paragraph Ranking Module produces a separate ranking of the retrieved paragraphs for each section. To calculate relevance scores for each paragraph we use a neural relevance model that combines dynamic IR features with two semantic matching networks that capture complementary relevance signals. One is based on the Transformer sequence-to-sequence model and the other uses the cosine similarity matrix. The model was trained on train.v2.0 dataset and validated using the test200.v2.0 dataset.

- uog: Our 2018 runs study entity-aware expansion models tailored to the TREC CAR task. In particular, we use fine-grained expansion features based on heading components of the TREC CAR query topics. We employ query entity linking, entity retrieval, and performed expansion over diverse matching vocabularies (words, entity ids, aliases). For entity retrieval we experimented with using feedback on the paragraph collection. All methods parameters (hyper-parameters and LTR models) used Y1Train (hierarchical qrels) and runs were selected by performance on Y1Test (tree qrels).

- CUIS: Team CUIS provided two passage runs and three entity runs. The system consists of two stages. The first stage chooses top 1000 candidate passages based on Lucene's BM25 method. The second stage reranks these candidate passages using a Markov random field based model where unigrams, bigrams and concepts induced by query terms from differnt query sections are considered. Besides, the system incorporates the Wikipedia article information or query entity mentions based on a Dirichlet prior smoothed language model. The run "CUIS-F150" incorporates the Wikiepdia article information. The run "CUIS-MX5" incorporates both Wikipedia article and query entity mentions. The system is trained on benchmarkY1-train v2.0 dataset. The entity runs are derived by replacing the paragraph id with the containing Wikiepdia article id.

- UMass (entityEmbedLambdaMart): For the `umass_entityEmbedLambdaMart` run we use the `benchmarkY1-train.v2.0` as the training set, `benchmarkY1-test-public.v2.0` as the validation set and the `benchmarkY2test-public.v2.1.1` evaluation topics for the submitted run. In this run, we learned a joint entity-word embedding representation based on Wikipedia corpus. In TREC CAR topics, each query topic is consist of three subtopic: Root subtopic(R), Intermediate subtopic (I) and leaf subtopic(H). We retrieve a set of documents with three baseline methods: SDM (Sequential Dependence Model), RM3 and query likelihood with the subtopic combinations of R-H, R-I-H and R-I-H, respectively. Furthermore, We represent each document and query based on their entity embeddings. Each query has fine-grained subtopic word vector representations as well as the complete topic representation. To be more specific, we represent an entity by the average embedding vectors of entities only in Root, entities only in Leaf (H), and all of the entities in the topic. We use cosine similarity between the document vector representation and each query representation as a feature in LambdaMart learning-to-rank model as well as the retrieval scores from the base retrieval model.

- TREMA-UNH (UNH): We provided three passage runs and three entity runs that were all based on combination of low-level input runs such as bm25, Query likelihood, SDM, RM3 and Entity Context model [4]—with combinations trained with coordinate ascent Learning-to-Rank optimized for MAP. We also experimented with a new "Learning-to-Walk" methods for supervised graph walks. Our best performing passage run is a combination of BM25, Query Likelihood with and without RM3. All combinations were trained on benchmarkY1train, and the best three methods were selected on benchmarkY1test. In our notebook also includes evaluation results on Y1 benchmarks.

- DWS-UMA: Our Trec-CAR submission is a simple unsupervised method. At query time we perform semantic query expansion in combination with term specifity boosting on a Lucene Index. Our model first represents the query by its lemmatized query terms. In the next step, for each term, the query is expanded by including the top k nearest neighbors from a semantic word embedding space. The expanded query is executed against a Lucene Index with BM25. Query terms at lower levels in the outline, i.e. more specific query terms, are boosted and receive a higher weight. For our submission we used a pre-trained embedding space and the value for k is tuned on benchmarkY1-train.

7.1 Assessment Interface and Fixed Mistakes in Submitted Runs

The assessment interface was populated by a) pooling the top 5 of all submitted runs and b) paragraphs and entities on the original article (i.e., the automatic ground truth).

Several submitted runs contained mistakes. We fixed those mistakes post-hoc, but we were unable to do so before the assessment. The following issues were fixed post-hoc

- Team NYU assigned all ranked items the score of 1.0, all information was contained in the rank information. We derived corrected runs turning rank information into scores. As a result random 5 elements of their ranking were assessed.

- Several teams submitted illegal entity ids. Despite recommended otherwise, participants created entity IDs from page names by replaceing spaces with %20. This will not address non-ascii characters such as accents or umlauts. We derived corrected runs post-hoc. As a result, for affected runs entities containing accents or umlauts were not assessed.

- Team CG submitted rankings that only contained three elements. As a result, fewer elements were assessed for CG.

8 Results

8.1 Passage Task

We evaluate participant-contributed passage runs on manual assessments with respect to four standard TREC evaluation measures, R-Precision (RPrec), Mean-average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG). Of these measures only NDCG includes a graded scale, for all other methods will use the positive/negative cutoff indicated in Table 3.

While the manual assessments included paragraphs within and outside the paragraph collection, we evaluate participating runs only on passages that are included in paragraph corpus (red shaded area in Figure 4). The automatic passage benchmark creation paradigm used in TREC CAR Y1 does not apply, as most passages in the ground truth pages are not included in the paragraphCorpus of Wikipedia from 2016.

Results for the benchY2test passage retrieval task are presented in Figure 5 on the manual graded scale, and a lenient variant of the manual graded scale. Standard error bars and paired-t-test with respect to the best performing method are given for reference. All analyzes across all measures are painting the same picture. Acknowledging consistent patterns in the results for different metrics, here the ranking of passage methods by across-the-board performance (tied methods on the same rank):

1. uog-heading–rh-sdm, UNH-p-l2r

2. uog-linear-lrt-hier, UNH-p-sdm, UNH-p-mixed

3. entityEmbedLambdaMart (UMass), guir-exp, UTDHLTRI2

4. guir, uog-linear-raw-expansion, NYU-XL-f, CUIS-MX5

5. NYU-L-f, NYU-M-f, CUIS-F150

6. DWS-UMASemQueryExp, DWS-UMASemQueryExp20, DWS-UMASemQueryExp30

7. CG-Seq2Seq

To study whether the differences are due to better performance on easy queries, difficult queries, or overall, we include divide the set of all annotated topic sections into percentiles ranging from easy to difficult according to the best performing method. The results are presented in Figure 9 and show a consistent ranking of methods for difficult and easy queries (interquartile ranges 25%-50%, 50%-75%, and 75%-95%).

Separating results for queries originating from Wiki18 and TQA (Figures 5d and 5e, demonstrates that methods performing well on Wiki-18, also perform well on TQA. However, the difference between methods is less pronounces for the Wiki18 subset. TQA queries seem to be slightly easier, which is probably because the TQA outlines containe fewer sections.

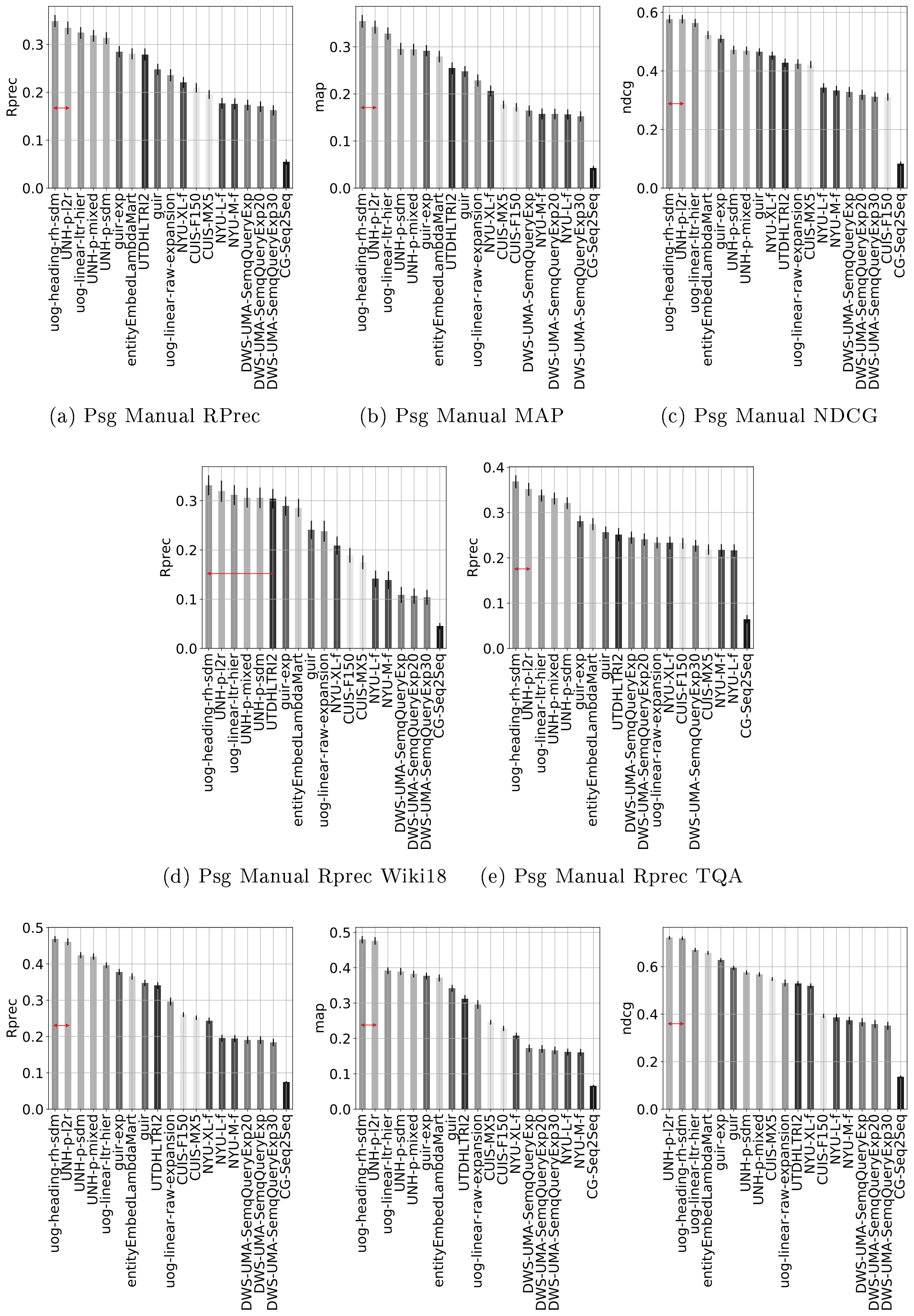
8.2 Entity Task

We evaluate participant-contributed entity runs on automatic and manual assessments with respect to tree standard TREC evaluation measures, R-Precision (RPrec), Mean-average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG).

While the manual assessments included entities within and outside the legal entity set (allButBenchmark), we evaluate participating runs only on entities that are included in legal set (red shaded area in Figure 4). We further evaluate on the automatic benchmark derived with TagMe entity links on ground truth pages. (The previous method of using only entity links included manually by Wikipedia editors does not apply to TQA articles.)

Results for the benchY2test entity retrieval task are presented in Figure 6 on the manual graded scale, lenient variant of the manual graded scale, and automatic assessment based on TagMe entity links, including error bars and paired-t-tests. All analyzes are providing a coherent picture, resulting in this ranking of entity methods across different measures (tied methods on the same rank):

1. UNH-e-L2R, UNH-e-mixed, UNH-e-graph, uog-rf-ent



(a) Psg Manual RPrec

(b) Psg Manual MAP

(c) Psg Manual NDCG

(d) Psg Manual Rprec Wiki18

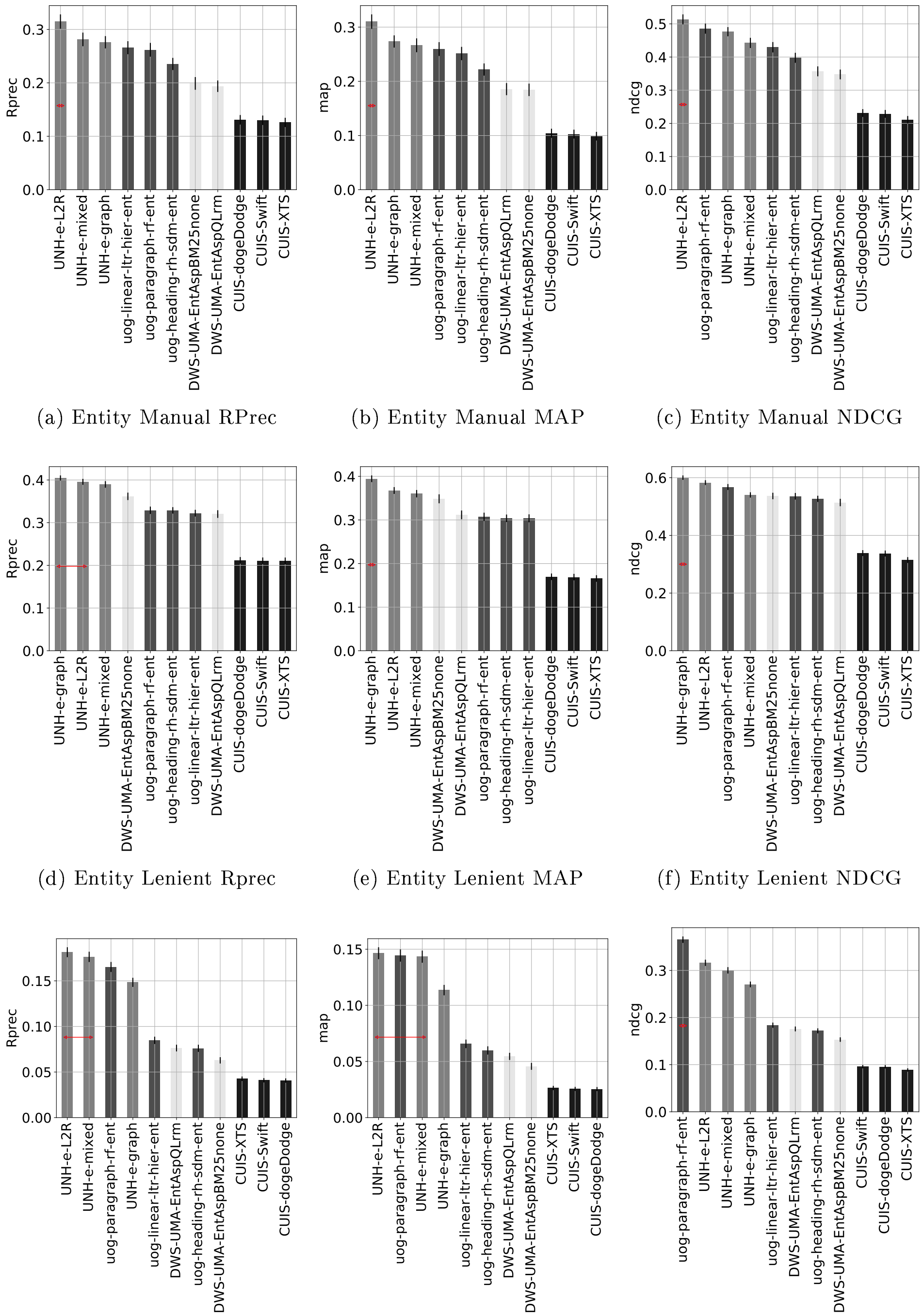
(e) Psg Manual Rprec TQA

(f) Psg Lenient Rprec

(g) Psg Lenient MAP

(h) Psg Lenient NDCG

Figure 5: Results of contributed passage runs under the manual ground truth. Lenient is based on the manual graded scale, but counting TOPIC as relevant. The red arrow markes systems for which no significant difference to the best system could be detected.



(a) Entity Manual RPrec

(b) Entity Manual MAP

(c) Entity Manual NDCG

(d) Entity Lenient Rprec

(e) Entity Lenient MAP

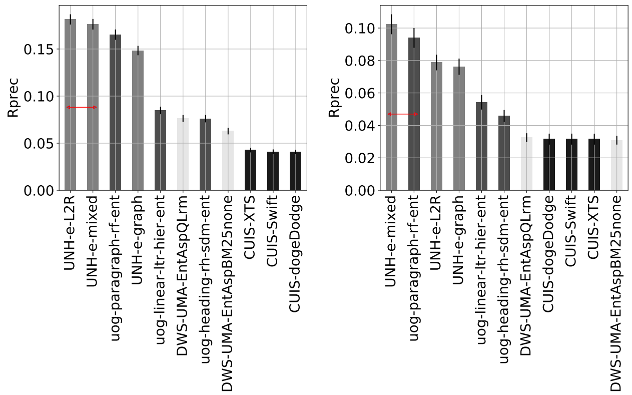
(f) Entity Lenient NDCG

(g) Entity Automatic Rprec

(h) Entity Automatic MAP

(i) Entity Automatic NDCG

Figure 6: Results of contributed entity runs under automatic and manual ground truth. Lenient is based on the manual graded scale, but counting TOPIC as relevant. The red arrow markes systems for which no significant difference to the best system could be detected.



(a) Entity Automatic Rprec (b) Entity Auto-Orig Rprec

Figure 7: Automatic evaluation using entity links from TagMe versus evaluation using links in Wikipedia source (Auto-Orig, only available for Wiki18). The red arrow markes systems for which no significant difference to the best system could be detected.

2. uog-linear-ltr-hier-ent, uog-heading-rh-sdm-ent, DWS-UMA-AspQLrm, DWS-UMA-EntAspBM25none
3. CUIS-dodgeDodge, CUIS-Swift, CUIS-XTS

Comparing results under the automatic entity benchmark (using TagMe) with the automatic entity benchmark derived from entity links included by Wikipedia editors (Auto-Orig, provided as training data for the "train" benchmark), we see a similar pattern emerging, but some systems swap ranks.

When analyzing queries originating from Wiki18 and TQA presented in Figure 8, we observe less sharp distinctions on Wiki18 (with two or three systems tied for rank 1), where on TQA, the method UNH-e-L2R is consistently and with significant difference leading.

9 Conclusion

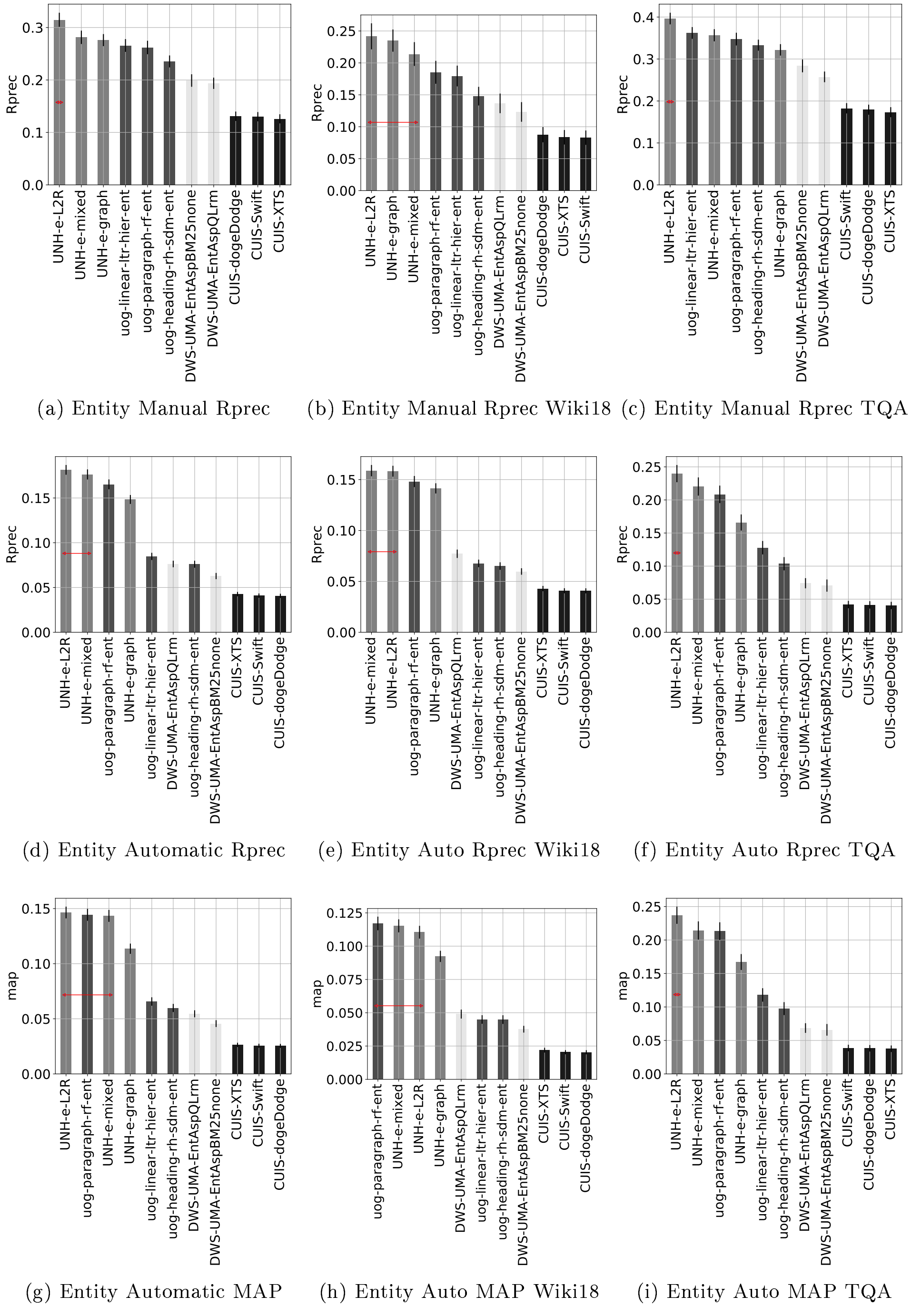
In contrast to the previous year of TREC CAR, where neural network methods were dominating the leaderboard, in this second year we see that learning-to-rank with unsupervised retrieval models such as BM25, SDM and query expansion significantly outperformed state-of-the-art neural methods. We see that systems that perform well in passage retrieval tasks, are also performing well in the entity retrieval task.

Regarding benchmark construction for this project. In previous year, we confirmed that automatically derived relevance data for passages agrees with human-created benchmarks on the ranking of systems. In this year we further show that automatically derived relevance data for entities agrees with human-created benchmarks. This means, that we have an effective test bed for method development in TREC CAR, when the outline is provided.

In the next year of TREC CAR, we will move beyond population of existing outlines and leave it to participants to also identify a suitable outline to automatically populate a complete article given just a short information need.

Acknowledgement

We express our gratitude for many suggestions of several experts in the field, who helped to make this track successful. We thank the University of New Hampshire for providing computational resources and web servers. We are deeply thankful for Ellen Voorhees' experience, patience, and persistence in running the assessment process. Finally we thank all our participants.



(a) Entity Manual Rprec

(b) Entity Manual Rprec Wiki18

(c) Entity Manual Rprec TQA

(d) Entity Automatic Rprec

(e) Entity Auto Rprec Wiki18

(f) Entity Auto Rprec TQA

(g) Entity Automatic MAP

(h) Entity Auto MAP Wiki18

(i) Entity Auto MAP TQA

Figure 8: Entity performance on Wiki18 versus TQA. The red arrow markes systems for which no significant difference to the best system could be detected.

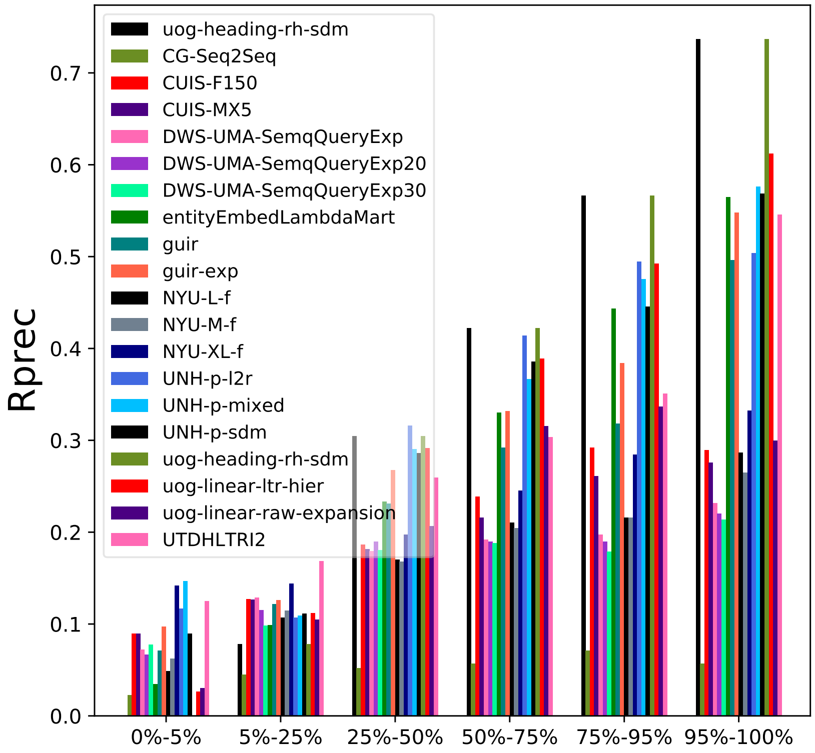


Figure 9: Performance according to manual truth on difficulty percentiles according to best performing method.

References

[1] James Allan, W. Bruce Croft, Alistair Moffat, and Mark Sanderson. Frontiers, challenges, and opportunities for information retrieval: Report from SWIRL 2012 the second strategic workshop on information retrieval in Lorne. *SIGIR Forum*, 46(1):2–32, 2012. doi: 10.1145/2215676.2215678. URL `http://doi.acm.org/10.1145/2215676.2215678`.

[2] Omar Alonso and Stefano Mizzaro. Using crowdsourcing for trec relevance assessment. *Information Processing & Management*, 48(6):1053–1066, 2012.

[3] J Shane Culpepper, Fernando Diaz, and Mark D Smucker. Research frontiers in information retrieval: Report from the third strategic workshop on information retrieval in lorne (swirl 2018). In *ACM SIGIR Forum*, volume 52, pages 34–90. ACM, 2018.

[4] Jeffrey Dalton, Laura Dietz, and James Allan. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 365–374. ACM, 2014.

[5] Paolo Ferragina and Ugo Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628. ACM, 2010.

[6] Sean MacAvaney, Andrew Yates, Arman Cohan, Luca Soldaini, Kai Hui, Nazli Goharian, and Ophir Frieder. Characterizing question facets for complex answer retrieval. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2018.

[7] Rodrigo Nogueira, Jannis Bulian, and Massimiliano Ciaramita. Learning to coordinate multiple reinforcement learning agents for diverse query reformulation. 2018.