A Test Collection for Coreferent Mention Retrieval

Rashmi Sankepally
Human Language Technology Center of Excellence
and University of Maryland, College Park
rashmi@umd.edu

Tongfei Chen
Johns Hopkins University
tongfei@cs.jhu.edu

Benjamin Van Durme
Human Language Technology Center of Excellence
and Johns Hopkins University
vandurme@cs.jhu.edu

Douglas W. Oard
Human Language Technology Center of Excellence
and University of Maryland, College Park
oard@umd.edu

ABSTRACT

This paper introduces the coreferent mention retrieval task, in which the goal is to retrieve sentences that mention a specific entity based on a query by example in which one sentence mentioning that entity is provided. The development of a coreferent mention retrieval test collection is then described. Results are presented for five coreferent mention retrieval systems, both to illustrate the use of the collection and to specify the results that were pooled on which human coreference judgments were performed. The new test collection is built from content that is available from the Linguistic Data Consortium; the partitioning and human annotations used to create the test collection atop that content are being made freely available.

CCS CONCEPTS

• **Information systems** → **Information extraction**; Structured text search;

KEYWORDS

coreference, mention retrieval, entity linking

ACM Reference Format:
Rashmi Sankepally, Tongfei Chen, Benjamin Van Durme, and Douglas W. Oard. 2018. A Test Collection for Coreferent Mention Retrieval. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3209978.3210139

1 INTRODUCTION

We define *Coreferent Mention Retrieval* (CMR) as an information retrieval task in which one passage mentioning a specific entity is presented as a query, and the system's task is to find all other sentences in the test collection in which that same entity is mentioned. A CMR system might be used directly by an end user to find other mentions of the same entity when performing a task that involves learning about new entities, or it might be used as a component of other downstream systems (e.g., question answering, entity linking, or knowledge base population). The CMR task shares much in common with the well studied *cross-document coreference resolution* task in which the goal is to cluster all mentions of the same entity [2, 8]. However, cross-document coreference resolution suffers from several challenges, including the need to assemble a collection of substantial size before representative results can be obtained, the time complexity of clustering methods that find all possible coreferent pairs over the entire collection, and the well known challenges of evaluating clustering techniques in ways that support reliable comparisons among systems. It is those challenges that give rise to our interest in the more focused query-by-example setting of CMR in which we can benefit from the efficiency of modern ranked retrieval methods, well developed methods for comparing retrieval results from different systems, and the potential for operating on collections that grow over time, doing so from their earliest stages (although for our purposes in this paper we focus only on a single static test collection). It can be regarded as a special case of cross-document coreference resolution in which rather than operating on the entire mention graph, we use retrieval techniques to limit our focus to an implicit subgraph anchored by the given query mention.

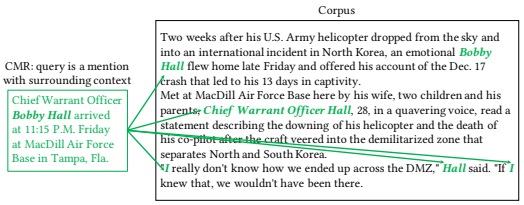

Figure 1: Example of a CMR query and its candidates.

The CMR task also has close links with several other tasks. The Web People Search task in SemEval 2007 [1] is one such task, in which systems were evaluated based on how well they clustered web pages that result from a search for a person's name, in as many sets as entities that share the name. In addition, the underlying theme of the Knowledge Base Population (KBP) track hosted by Text Analysis Conference (TAC) [7] is to link mentions in documents to entities in a knowledge base and to fill in information about their attributes. On similar lines, the TREC entity track, which ran between 2009-11 [3] aims at finding target entities that satisfy a specified relationship with an input entity.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210139

Table 1: Statistics of the collections.

Collection	# queries	# docs	# relevant
2014	84	117,132	4,172
2015	5,344	166	194,512

Our task differs from these in several ways. First, the online and dynamic nature of the CMR task potentially caters to user needs quickly, while overcoming the shortcomings of clustering based approaches. Second, our task does not necessarily result in a knowledge base. This allows agents to look up and find information about an entity in other documents. That said, CMR results could also be used for several downstream knowledge acquisition tasks. Another particularly relevant variant in this connection is *Pocket KBP* [10], which strives to construct pocket Knowledge Bases (KBs), i.e. dense entity-centric KBs dynamically constructed for a specific query. But pocket KBs focus on relationships between entities, while our task aims to retrieve mentions of entities.

2 TASK DEFINITION AND TEST COLLECTION

Coreferent Mention Retrieval (CMR) is a structured search task defined as follows: given a mention of an entity within a background document context, find other mentions of the entity in other documents. CMR is an instance of query by example. In our case, a document is a news article or a discussion forum thread. A mention is a set of words that specify an entity. We focus on specific entity types, namely person (`PER`), organization (`ORG`) and location (or, more formally, Geo-Political Entities, `GPE`).

Our test collection is built atop the documents in the Text Analysis Conference (TAC) 2014 [6] and TAC 2015 Entity Detection and Linking (EDL) [7] training data, which are available from the Linguistic Data Consortium.[1] The TAC 2014 collection has about one million newswire articles and some entity linking annotations that specify the entity IDs of mentions in a knowledge base. A subset of these mention annotations, along with the documents in which they appear, are used as queries for our collection. The collection was filtered to obtain a subset of documents with dates that fall between the dates of the chronologically earliest and latest documents from which the query mentions were selected. The TAC 2015 collection has 166 newswire articles and discussion forum threads. Although this is a small collection, the TAC 2015 EDL collection has exhaustive mention annotations, making it useful as a development test set. All mention annotations (not just unique entities) were used as queries for TAC 2015.

A subset of mention annotations were used as queries and the rest formed gold standard relevant mentions for the corresponding queries. Since the TAC 2014 EDL collection has quite sparse mention annotations, we used Amazon Mechanical Turk to obtain additional relevance judgments for a set of pooled candidates from five systems. This process is described in detail in Section 3. For the TAC 2015 EDL collection, all other mentions were considered non-relevant.

All documents were processed by the Stanford CoreNLP[2] package for tokenization, sentence boundary detection, named entity recognition (NER), entity type detection, and within-document coreference annotation. We use these annotations as a basis for constructing features for our ranking systems.

Each mention is identified by a unique mention ID which includes a document ID for the document in which it appears, a sentence ID for the sentence in which it appears and a pair of begin and end token indices that point to the position where it appears in the sentence. Query mentions were similarly processed to be identified in this format. For example - the mention ID `doc5:sent8:10:11` implies 10th token in the 8th sentence of the 5th document.

We evaluate retrieval performance of ranking systems using each sentence as the unit of evaluation, as evaluating on mentions down to their token indices might be too strict for some applications. Other window based approaches could be used, but we leave that as future work. Since our collection has incomplete annotations, we use inferred Average Precision (infAP) [11] as our evaluation measure.

3 ANNOTATIONS AND RELEVANCE JUDGMENTS

In order to make the TAC 2014 EDL collection usable for our task, we obtained mention annotations for query-candidate mention pairs using Amazon Mechanical Turk. We worked with a randomly selected subset of 196 entities which had 688 corresponding mention annotations. Among all the mentions that referred to a unique entity, we chose the one with the largest number of characters as the canonical query, breaking ties randomly, and we used the rest as known relevant coreferent mentions.

Query/candidate mention pairs for annotation were identified by pooling top-30 results from the 5 systems described in Section 4. Each query had about 90 candidates in its pool on average, thus indicating substantial diversity among the systems pooled.

Assessors were shown the first sentence from the candidate document, the mention sentence with the mention highlighted, and the sentence following the mention sentence. The reason for choosing this restricted set of sentences for display was to decrease the cognitive load on the assessor and control annotation costs, while hopefully giving sufficient context for a reasonably accurate assessment. The query mention was highlighted in yellow, and system generated candidate mentions were highlighted in blue. An example is shown in figure 2, which is best viewed in color. Each Human Intelligence Task (HIT) had 10 such query/candidate mention pairs. Assessors were provided with instructions and example annotations. Users were allowed to express the degree to which they believe that the mentions are coreferent on a 5 point Likert scale which ranged from "very unlikely" to "very likely" as shown in figure 2. Assessors took an average of about 1 minute to annotate a HIT. After initial experiments with different rates, we decided on $0.13 as the payment for each HIT in phase 2 for adequate accuracy and efficiency.

After running preliminary tests with 6 queries (which we did not use for evaluation), we ran the annotation task in 2 phases. In phase 1, a single human annotation was obtained for each HIT for a set of 50 query mentions. Quality control was performed by including one test pair with known annotation from the TAC 2014 EDL collection for each HIT of 10 pairs. All HITs which got the

[1] Linguistic Data Consortium catalog numbers `LDC2014E54`, `LDC2014E13`, `LDC2015E75`.
[2] http://nlp.stanford.edu:8080/corenlp/

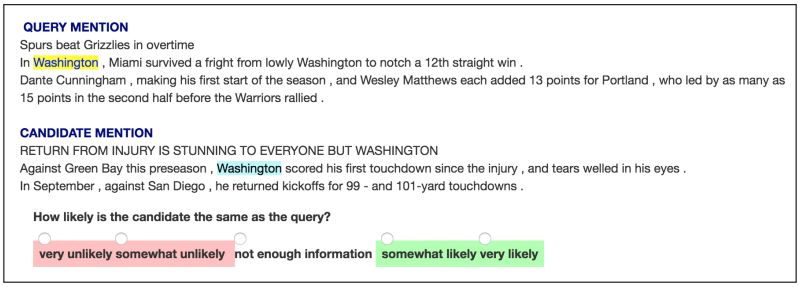

Figure 2: An example candidate in a HIT (best viewed in color)

test pair wrong were re-uploaded for 3-way redundant annotations from a new set of assessors.

The 5-point Likert scale assessments thus obtained were binarized by considering candidates marked with the first three options (red and white options in figure 2) as non-relevant (0) and the last two options (green options) as relevant (1). Candidates with equal numbers of annotators disagreeing were marked as "unknown" (-1). Some queries which had large number of disagreements were marked as hard queries and removed from the collection and stored separately for later use.[3] Examples of hard queries are: Utah Jazz (the basket ball team confused with Utah state), New York (city or state), Congress.

Among the reuploaded HITs in phase 1, 58% of the time, all three assessors agreed on a binary assessment. On uploading the rest for a third time for two-way redundant assessment, we retained only those cases on which at least 4 assessors agreed on a binary assessment and left the rest as unknown. This process resulted in 26 queries that had 5 or more relevant candidates and an average incompleteness of at least 20% (due to the marking of low agreement cases as unknown) in phase 1. The other 30 queries were considered unusable due to having 4 or fewer relevant candidates, which risks introducing quantization noise in the evaluation measure. Some consistently bad assessors were also identified in this phase and were prevented from working on future HITs.

In phase 2, annotations were collected for 100 more queries, each with an average of 90 candidates, in a similar fashion, after making corrections from what we learned from phase 1. In this phase, we obtained 3-way redundant assessments at the outset, and for that reason we did not include any test pairs for which the truth was known (from the sparse annotations on the EDL collection). 61% of the time all three agreed on a binary judgment. Disagreement cases were re-uploaded for further 2-way redundant assessments, and the binarized assessment which was agreed by at least 4 assessors was used as the final assessment, while marking other cases as unknown. This yielded 84 queries with 5 or more relevant candidates, after removing hard queries and queries with 4 or fewer relevance sentences. This has an average incompleteness of at least 5% (due to unknown cases). Thus we obtained TREC-style relevance judgments for 84 queries consisting of 4,172 relevant candidates in this phase. There are 20 GPE, 22 ORG and 42 PER queries in our "2014" collection.

[3]Three queries were removed from the 2015 collection; 11 were removed from the 2014 collection.

4 EXPERIMENTS

Our approach to the CMR task is to model the scoring function between a pair of mentions as a featurized linear model [9]. We adopt the following sequential steps, which are common in ranked retrieval systems:

(1) ***Featurization***: This step involves obtaining features for query and candidate mentions. We call each category of feature type a *field*. Each field had several binary or real-valued features. Binary fields included mention string, mention type (PER/ORG/GPE/none), trigrams of mention string and acronym of the mention - which is the concatenation of first alphabetic character of each mention word (with stopwords such as "the" or "of" removed). Real-valued fields were weighted using either BM25 or IDF, as described below. These features consist of mention words, words from the surrounding sentence, top-scoring words from surrounding document and words in the coreference chain of the mention (as is produced by the Stanford CoreNLP within-document coreference resolution tool).
(2) ***Field weighting***: There are different ways to assign weights to different fields for scoring candidates for retrieval and ranking. One way is to manually craft weights by deciding the prominence of the fields intuitively, for example by giving mention words 10 times more weight than surrounding document words. Another way is to learn the weights from training data, based on which features are more discriminative for the task. In our case, the training set consists of pairs of mentions with a binary judgment to tell whether they corefer or not. We call the approach where we *learn* the field weights DiscK, described in more detail in section 4.1.
(3) ***Indexing***: Lucene[4] was used for indexing the features.
(4) ***Retrieval***: In this step we score the candidates by obtaining the inner product of feature vectors, and rank them in descending order of scores using Lucene's retrieval scoring function.

Within this general framework, we developed and experimented with models that differed in query and candidate features and field weighting schemes.

[4]http://lucene.apache.org.

4.1 Models

Mention baseline (mention). This is a basic system that represents query and candidate mentions using the single field of mention words, which are weighted using IDF. The IDF of words was computed from an external Wikipedia corpus, as it is larger and may give a more accurate representation.

Document baseline (doc). This system uses two fields - (1) mention words and (2) background document text for representing candidate mentions. Query mentions are represented using mention words only. Lucene's multi-field Query Parser is used to project the query features to score the candidates. The document text field is weighted at 10% of the mention words field. All these features are weighted using BM25.

Query expansion (QE). This builds on the doc system and has the same set of fields for queries and candidates. Retrieval is performed twice. The first time, features from the same two fields as the doc system for top ranking candidates are stored. These terms are added to the query for performing retrieval the second time in a Rocchio query expansion framework [4]. The query terms used for expansion are further down-weighted using manually picked weights.[5]

Coreference chains system (coref). This is another single-field system that represents query and candidate mentions using words from the corresponding coreference chain in the document. BM25 scoring was used for feature weighting.

DiscK. DiscK is a machine learning approach based on previous work by Chen and Van Durme [5]. It was originally a method to retrieve sentences from a large corpus that may answer a specific natural language question (as queries). DiscK learns field weights from training data consisting of mention pairs. Both queries and candidates are represented by the following features: mention strings, mention words, mention types (PER/GPE/ORG/etc.), trigrams, acronyms, document context words and coreference chain features. These features are sometimes correlated, e.g., the acronym of organizations (e.g. World Health Organization, WHO), are useful but the acronym of locations are probably not so useful. To take this type-dependent information into account, the type of an entity mention is paired with every other feature. Additionally, this can help to eliminate most candidates whose types are different from our query.

Table 2: System Results: mean infAP

	QE	doc	coref	DiscK	mention
GPE	0.38	0.38	0.26	0.21	0.20
ORG	0.42	0.40	0.36	0.30	0.29
PER	0.40	0.39	0.34	0.37	0.37
all	0.40	0.39	0.32	0.31	0.31

[5]Original query terms are weighted 0.9, expanded terms are weighted 0.1.

5 QUALITY OF TEST COLLECTION AND RESULTS

We evaluate retrieval performance of our systems using relevance judgments of TAC 2014 from phase 2 annotations as described in section 3. We use the mean of inferred Average Precision (infAP) over all 84 queries as our evaluation measure.

For evaluating DiscK, we used 10-fold cross-validation in order to leverage the size of our training set. Each training fold has 76 queries; each test fold has about 8. We average the infAP results over the 10 test folds. Training mention pairs were obtained by collecting all possible mention pairs from the relevance judgments for the training queries. Each training fold has an average of 6,500 mention pairs.

As table 2 shows, the baseline IR systems perform fairly well with about 0.4 infAP. These scores indicate that the retrieval performance of our systems used for creating the test collection was quite credible. The results are consistent across different entity types.

6 CONCLUSION

We presented Coreferent Mention Retrieval (CMR), a search task for finding coreferent mentions. We also built an evaluation test collection for the task and presented results for different baseline approaches. CMR will be useful as an upstream task for various other tasks including cross-document coreference resolution itself. We leave evaluating on those tasks as future work. The topics and relevance judgments are available at https://github.com/hltcoe/CoreferentMentionRetrieval. The TAC 2014 and 2015 EDL collections can be obtained from the Linguistic Data Consortium.

REFERENCES

[1] Javier Artiles, Julio Gonzalo, and Satoshi Sekine. 2007. The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*. ACL, 64–69.

[2] Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proc. ACL*, Vol. 1. 79–85.

[3] Krisztian Balog, Pavel Serdyukov, and Arjen P de Vries. 2010. *Overview of the TREC 2010 entity track.* Technical Report. Norwegian University of Science and Technology, Trondheim.

[4] Chris Buckley, Gerard Salton, and James Allan. 1994. The effect of adding relevance information in a relevance feedback environment. In *Proc. SIGIR*. 292–300.

[5] Tongfei Chen and Benjamin Van Durme. 2017. Discriminative information retrieval for question answering sentence selection. In *Proc. EACL*, Vol. 2. 719–725.

[6] Heng Ji, Joel Nothman, Ben Hachey, et al. 2014. Overview of tac-kbp2014 entity discovery and linking tasks. In *Proc. Text Analysis Conference (TAC2014)*. 1333–1339.

[7] Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. 2015. Overview of TAC-KBP2015 tri-lingual entity discovery and linking. In *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.

[8] James Mayfield, David Alexander, Bonnie J Dorr, Jason Eisner, Tamer Elsayed, Tim Finin, Clayton Fink, Marjorie Freedman, Nikesh Garera, Paul McNamee, et al. 2009. Cross-Document Coreference Resolution: A Key Technology for Learning by Reading.. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, Vol. 9. 65–70.

[9] Donald Metzler and W Bruce Croft. 2007. Linear feature-based models for information retrieval. *Information Retrieval* 10, 3 (2007), 257–274.

[10] Travis Wolfe, Mark Dredze, and Benjamin Van Durme. 2017. Pocket Knowledge Base Population. In *Proc. ACL*, Vol. 2. 305–310.

[11] Emine Yilmaz and Javed A Aslam. 2006. Estimating average precision with incomplete and imperfect judgments. In *Proc. CIKM*. 102–111.