Computing (2017) 99:313–349
DOI 10.1007/s00607-016-0490-0

A systematic review and comparative analysis of cross-document coreference resolution methods and tools

Seyed-Mehdi-Reza Beheshti[1] · Boualem Benatallah[1] · Srikumar Venugopal[1] · Seung Hwan Ryu[1] · Hamid Reza Motahari-Nezhad[1,2] · Wei Wang[1]

Received: 15 November 2013 / Accepted: 22 March 2016 / Published online: 7 April 2016

Abstract Information extraction (IE) is the task of automatically extracting structured information from unstructured/semi-structured machine-readable documents. Among various IE tasks, extracting actionable intelligence from an ever-increasing amount of data depends critically upon cross-document coreference resolution (CDCR) - the task of identifying entity mentions across information sources that refer to the same underlying entity. CDCR is the basis of knowledge acquisition and is at the heart of Web search, recommendations, and analytics. Real time processing of CDCR processes is very important and have various applications in discovering must-know information in real-time for clients in finance, public sector, news, and crisis management. Being an emerging area of research and practice, the reported literature on CDCR challenges and solutions is growing fast but is scattered due to the large space, various applications, and large datasets of the order of peta-/tera-bytes. In order to fill this gap, we provide a

✉ Seyed-Mehdi-Reza Beheshti
sbeheshti@cse.unsw.edu.au

Boualem Benatallah
boualem@cse.unsw.edu.au

Srikumar Venugopal
srikumarv@cse.unsw.edu.au

Seung Hwan Ryu
seungr@cse.unsw.edu.au

Hamid Reza Motahari-Nezhad
hamidm@cse.unsw.edu.au; motahari@us.ibm.com

Wei Wang
weiw@cse.unsw.edu.au

[1] School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

[2] IBM Almaden Research Center, San Jose, CA, USA

systematic review of the state of the art of challenges and solutions for a CDCR process. We identify a set of quality attributes, that have been frequently reported in the context of CDCR processes, to be used as a guide to identify important and outstanding issues for further investigations. Finally, we assess existing tools and techniques for CDCR subtasks and provide guidance on selection of tools and algorithms.

Keywords Information extraction · Cross-document coreference Resolution · Large datasets

Mathematics Subject Classification 68 Computer Science · 68-02 Research exposition (monographs, survey articles) · 68U15 Text processing; mathematical typography

1 Introduction

The majority of the digital information produced globally is present in the form of Web pages, text documents, news articles, emails, and presentations expressed in natural language text. Collectively, such data is termed *unstructured* as opposed to *structured* data that is normalized and stored in a database. The domain of information extraction (IE) is concerned with identifying information in unstructured documents and using it to populate fields and records in a database [1]. In most cases, this activity concerns processing human language texts by means of natural language processing (NLP) [2]. In this context, identifying and linking named entities across various information sources can be considered as the basis of knowledge acquisition and at the heart of Web search, recommendations, and analytics. An important problem in this context is cross-document coreference resolution (CDCR): computing equivalence classes of textual mentions denoting the same entity, within and across documents [3–6].

A CDCR process involves multiple stages. A simple, but typical, process might include entity identification and classification. There are many possible choices for each stage, and only some combinations are valid. For example, traditional approaches to CDCR [5,7] employ ranking, clustering, or probabilistic graphical models using syntactic features and distant features from knowledge bases. These methods exhibit limitations regarding run-time, grow exponentially in time with the increase in the number of documents, and robustness. Recent approaches to CDCR [2,4], are considering the transition from documents and keywords to data, knowledge, and entities. Google Knowledge Graph [8] and the IBM Watson technology for deep question answering[1] are examples of this. Such semantic resources can be used for the recognition and disambiguation of named entities in Web and user contents.

Real time processing of CDCR processes are very important and have various applications in discovering must-know information in real-time for clients in finance, public sector, news, and crisis management. To address this challenge, recent approaches [9–13] to CDCR focuses on scaling CDCR techniques and provides solutions for difficulties in clustering and grouping large numbers of entities and mentions across large

[1] http://www.research.ibm.com/deepqa/.

datasets, i.e. document collections sized of the order of tera-bytes and above. Parallel and distributed architectures such as Apache Hadoop [14], Spark [15] and distributed graph processing techniques [16] have been used to solve the scalability issues. Being an emerging area of research and practice, the reported literature on CDCR challenges and solutions is growing fast but is scattered. It is difficult for researchers and practitioners to have an easy access to systematically identified and peer reviewed studies reporting challenges, solutions, and valid stages in the CDCR process. In order to fill this gap, we decided to conduct a Systematic Literature Review (SLR) [17] of related challenges and solutions to CDCR approaches. The primary contributions of this work include:

- A systematic review of the state of the art challenges of and solutions to CDCR approaches. The systematically discovered and synthesized knowledge can be leveraged by the practitioners to understand the central concepts, subtasks, and the current state-of-the-art in CDCR.
- A taxonomy of CDCR research for studying and categorizing the identified state of the art challenges and solutions. The main categories in the taxonomy includes intra-document co-reference resolution, knowledge enrichment, similarity computation, and clustering.
- Identification of a set of quality attributes, such as scalability and complexity, that have been frequently reported in the context of CDCR; these quality attributes can be used as a guide for designing and evaluating CDCR systems over large text datasets.
- Assessing existing tools/techniques for CDCR subtasks and providing guidance on the selection of tools and algorithms for different stages of CDCR process.

The remainder of this document is organized as follows. Section 2 provides background information on the CDCR process and the motivations behind its scalable processing. Section 3 provides an overview of the methodology and presents a taxonomy for the CDCR process. In Sect. 4 we analyze the state-of-the-art approaches through the lens of the CDCR taxonomy and explain a taxonomy to analyze CDCR processes. In Sect. 5 we discuss challenges and provide solutions in applying CDCR process to large datasets. In Sect. 6 we evaluate the state-of-the-art tools and techniques for CDCR processes before concluding the paper in Sect. 7.

2 Background and motivation

Information extraction (IE) is the task of automatically extracting structured information from unstructured/semi-structured machine-readable documents. Figure 1 illustrates a taxonomy of the information that can be extracted from unstructured/semi-structured documents. In this paper we focus on entity extraction from text documents, where an *entity* is a real-world person, place, organization, or object, such as the person who serves as the 44th president of the United States and an *entity mention* is a string which refers to such an entity, such as “Barack Hussein Obama”, “Senator Obama” or “President Obama”. Among various IE tasks, extracting actionable intelligence from an ever-increasing amount of data depends critically upon coreference

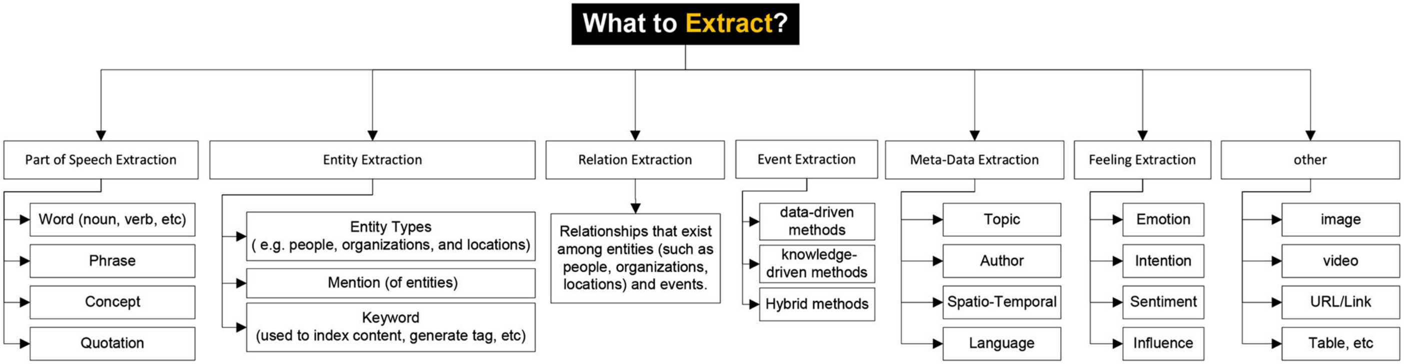


Fig. 1 A taxonomy of the information that can be extracted from unstructured/semi-structured documents

resolution (CR), i.e. the task of finding all expressions that refer to the same entity in a text.

Intra-document CR approaches provide techniques for the identification of entity mentions in *one document* that refer to the same underlying entity, while cross-document CR (CDCR) approaches provide techniques for the identification of entity mentions in *different documents*. Given a collection of mentions of entities extracted from large number of documents, CDCR involves various subtasks, from extracting entities and mentions to clustering the mentions. The overall objective is to cluster mentions such that mentions referring to the same entity (termed "co-referent") are in the same cluster and no other entities are included [13].

Perhaps the most important value-adding component in this setting is the recognition and disambiguation of named entities in Web and user contents. Named nntity disambiguation (NED) [18] maps an extracted mention string, e.g., a person name like 'Obama', onto its proper entity if present in a knowledge bases (KB). A KB typically consists of a set of concepts (e.g. a person) organized into a taxonomy, instances for each concept (e.g. Barack Obama), and the relationships among them. Freebase [19] and YAGO [20] are examples of a KB. Notice that, CDCR does not involve mapping mentions to the entities of a KB. In this context, unlike named entity disambiguation, CDCR can deal with long-tail or emerging entities that are not captured in the KB or are merely in very sparse form.

Real time processing of CDCR processes are very important and have various applications in e-Health (processing the electronic health records), legal databases, opinions, sentiment analysis, and also understanding what is happening around us. Consider open source intelligence as a motivating example, where millions of people broadcast events and opinions every second. In this context, cross document coreference occurs when the same person, place, event or concept is discussed in more than one text source, e.g. tweets in Twitter (an online social networking service that enables users to send and read short messages, 'tweets'). Consequently, Real time processing of CDCR can help in analyzing very large number of tweets generating in seconds, linking related tweets, and discovering more insight from them to understand what is happening now and predict what may happen later.

Designing and evaluating a suitable CDCR process are not only extremely important but also hugely challenging. Analyzing the state of the art, shows that a CDCR process involves multiple stages, where there are many possible choices for each stage, and only some combinations are valid. In this context, a systematic review of the state-of-the-art challenges of and solutions to CDCR approaches can help practitioners to understand the central concepts, subtasks, and the current state of the art in CDCR. In the following we explain our research methodology.

3 Research methodology

In this paper, we follow the systematic literature review (SLR) [17] method, which is a systematic and repeatable research process to identify, extract, assess, synthesize and report all available evidence (or information) on a particular research topic. Our research began by systematically designing and rigorously reviewing and imple-

Table 1 Research questions and their respective rationale

Research questions	Rationale
What are the different dimensions of a CDCR process that are addressed by researchers?	This research question is aimed at identifying different areas of research focused by the CDCR research community and identifying different stages/subtasks in CDCR
What are the publication venues and trends of studies on CDCR?	This research question aims at highlighting the important publication venues of CDCR research and provide information on the research publication trends
What quality attributes are primarily focused for evaluation of the proposed solution?	This question aims at identifying the important quality attributes that have attracted the research efforts for the accuracy of the result and the efficiency of CDCR approaches
What is the maturity level of published studies and the reliability level of proposed solutions?	This question determines the maturity and reliability of the published research studies
What are the major challenges of and solutions for designing and implementing a CDCR process?	This question is aimed at identifying the main challenges and solutions reported in the literature on CDCR approaches

menting research protocols. Following are the activities and artifacts of this research study. The research study protocol included research study background and motivation, research objectives, research questions, criteria for inclusion and exclusion of target studies, search strategies, and selection of target data sources. The protocol also specified a set of measures to assess the quality of the selected studies. Our research questions were derived from the objectives of our study. The research questions of this study and their respective rationale have been reported in Table 1.

Search query We performed searches on electronic databases that had been accessible online, including IEEE (http://ieee.org), ACM (http://acm.org), Springer (http://springerlink.com), and ScienceDirect (http://sciencedirect.com). We did not look for information in printed sources. We performed searches on the chosen digital libraries to retrieve the relevant studies. We used following criteria to obtain the keywords for the search queries.

- Derived the major terms from the research objectives and the research questions.
- Identified alternatives and related terms. Literature related to CDCR is often referred with different terms such as information extraction, intra-/within-document processing, cross-/multi-document processing, coreference resolution, unsupervised/supervised resolution, entity resolution, entity linking, entity pairs filtering, featurization, classification, and clustering. Therefore, we included these alternative names while preparing our search string. We also included different names used for scalable processing including large scale, big data, scalable, and massive in our search string.
- Used Boolean ‘or’ and ‘and’ operators to link the major terms of the strings for target databases when the search engines allowed the use of Boolean operators.

Table 2 Technology maturity model of the selected publications

Maturity stage	Description
Basic research [21–29]	The studies that are classified in this maturity stage provide theoretical solutions for the problems
Prototype implementation [2,4,5,12,18,30–41]	The studies that are classified in this maturity stage propose solutions to the stated problems and provide prototype implementation
Evaluated in real environments [1,3,3,7,9,11,13,42–61]	The studies that are classified in this maturity stage provide solutions to the described problems, and provide implementation and evaluation details along with results
Popularization [10,19,20,62–76]	The studies that are classified in this maturity stage demonstrate the applicability of the proposed solutions in real world applications

- Performed pilot searches to validate the effectiveness of the constructed search queries.

Considering the inclusion and exclusion criteria, we have selected papers published in peer-reviewed journals and ERA[2] ‘A’ ranked conferences till February 2015. Also we considered publication having more than 30 citations. Following search string represents our generic search query based upon terms related to CDCR and combining AND and OR operators:

> (“cross-document” OR “multi-document” OR “intra-document” OR “within-document”) AND (“coreference resolution” OR “co-reference resolution” OR “resolution”) AND (“featurization” OR “classification” OR “clustering” OR “entity linking” OR “entity pairs”) AND (“unsupervised resolution” OR “supervised resolution” OR “entity resolution”) AND (“large scale” OR “big data” OR “bigdata” OR “scalable” OR “massive”)

We included “*coreference resolution” and “*co-reference resolution” in the search query to make sure that no potential study is missed in search results. We customized the generic search query according to standard of each of the target electronic database to get more accurate search results. We performed searches using customized search strings on documents’ metadata including both title and abstract.

Literature synthesis and classification We determine the maturity of the solutions based on the implementation and evaluation reported in the selected study using four maturity stages of the technology maturity model [82]. Table 2 illustrates the technology maturity model for this study. We synthesized the collected papers with respect to CDCR subtasks challenges and corresponding solutions. We used a multi-stage approach of thematic synthesis that has been proposed by Cruzes et al. [83]. This approach begins by identifying codes corresponding to the concepts of interest. Then the codes are translated into themes, where the relations between themes are investigated to create higher order themes. We developed a catalogue of codes, and assigned it to selected

[2] http://www.arc.gov.au/era/.

Table 3 Classification of the selected studies

Category	Description
Intra-document processing [2,3,7,22,25,26,29,30,36,38,40, 41,44,46,53–56,59,64,69,77–79]	Studies in this category, start with casting the input documents into plain text (pre-processing), identifying the location of names of things in the text (entity identification), and the identification of entity mentions in one document that refer to the same underlying entity (intra-document CR)
Knowledge enrichment [11,23,31, 35,39,41,43,58,60,73,80]	Studies in this category, start with extracting all the entity pair candidates (entity pair filtering), featurizing the entity pair itself and the context around it (featurization), and computing the similarity between the entity pairs and the context patterns (similarity computation)
Clustering [4,5,21,22,27,33,35,41, 43,45,46,48,52,56,57,62,77,78]	Studies in this category, group mentions into clusters such that the similarities among members within the same cluster are maximal while similarities among data members from different clusters are minimal
Scalable approaches to CDCR [12,13,32,37,42,47,48,61,72,74, 75,81]	These studies provide solutions for scalable computing of CDCR over large text datasets. Four main directions have been identified in this category including extracting entities from a huge number of documents (e.g. Web pages and news articles), entity pairs filtering and featurization of the extracted entities, classification of all possible pairs, and co-referent entity clustering

studies according to: the main challenges they were addressing, the venues that published the selected studies, different environments that were utilized for evaluation of the solutions, different maturity stages and delivery model of the solutions.

As the result, we classified the selected studies into four categories based on the main focus of the studies including Intra-Document Processing, Knowledge Enrichment, Clustering, and Scalable Approaches to CDCR. Table 3 illustrates the classification of the selected studies. Studies in the 'Intra-Document Processing' category, includes (i) *pre-processing* cast the input documents into plain text; (ii) *entity identification* the work in this category, uses 'Entity Identification' techniques using standard tools such as Stanford CoreNLP [69] tool suite to detect mentions and anaphora (i.e. the interpretation of which depends upon another expression in context and used to bind different syntactical elements together at the level of the sentence); and (iii) *intra-document CR* these approaches provide techniques for the identification of entity mentions in *one document* that refer to the same underlying entity.

Studies in the 'Knowledge Enrichment' category, provide solutions for entity pair filtering, featurization, and similarity computation of the local mention groups obtained in the previous step. Prior to entity pairing [9], various *features* [53] may be extracted to annotate entities and their mentions. Then, the similarity scores for a pair of entities are computed using appropriate similarity functions for each type of feature. Studies in the 'Clustering' category, provide solutions for computing and classifying the cross-document coreference equivalence classes of mentions. The goal of clustering is to group identified mentions from previous stages into clusters such that the similarities

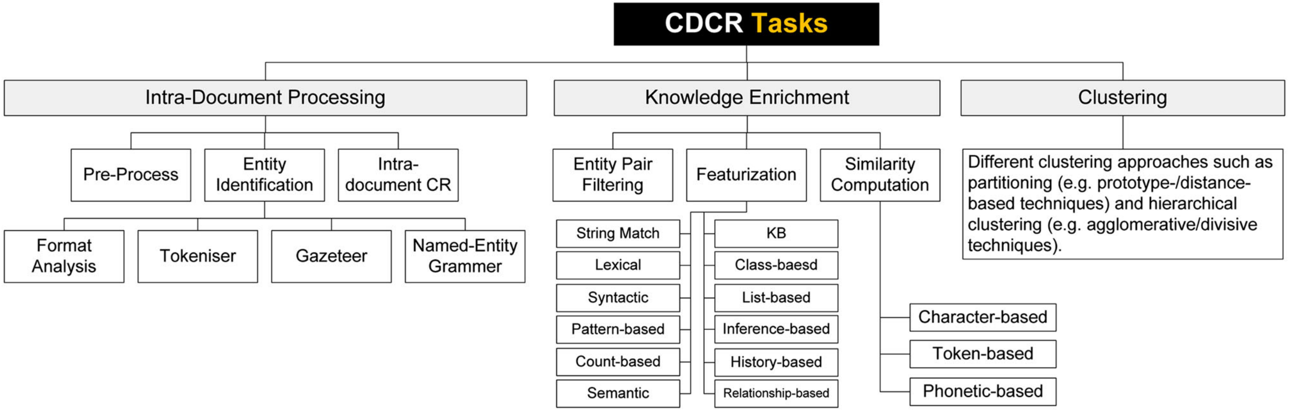


Fig. 2 The CDCR Taxonomy

among members within the same cluster are maximal, while similarities among data members from different clusters are minimal. Studies in the ‘Scalable Approaches to CDCR’ category, provide solutions for scalable computing of CDCR over large text datasets. Figure 2 shows a structural view of the CDCR taxonomy, which, at the highest level, groups the CDCR subtasks into: intra-document processing, knowledge enrichment, and clustering. Each group is further subdivided. At the leaves of this tree are the actual algorithms and techniques for CDCR sub-tasks. We constructed this prototype taxonomy by first considering the types of operations applied to information sources. We chose the three stages of the CDCR process that have been used in the state of the art approaches to CDCR. We then divided these into subcategories, focusing on entity extraction, coreference resolution, featurization, similarity computation, and clustering. Finally, we chose and assessed tools, techniques, and datasets that have been implemented and widely used in the literature, for evaluating each sub task.

4 CDCR methods analysis

In this section, we analyze the selected publications through the lens of the CDCR taxonomy and explain a taxonomy to analyze the main stages in CDCR processes. Such a taxonomy can help the analysts in deciding which mechanism is best suited to carry out the CDCR sub tasks. We apply the taxonomy to the technique of extracting coreference entities from different documents and exemplify it by referring to three main CDCR subtasks: intra-document processing (in Sect. 4.1), knowledge enrichment (in Sect. 4.2), and clustering (in Sect. 4.3).

4.1 Intra-document processing

The first step in intra-document processing includes pre-processing input documents to cast them into plain text. Tools such as *Boilerpipe*[3] and *Jsoup* (http://www.jsoup.org) can be used for pre-processing the documents. The next step in intra-document processing is to use ‘Entity Identification’ techniques which require extracting entities from the text. In this context, Named Entity Recognition (NER), also known as Entity Extraction (EE), techniques can be used to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, and percentages. In particular, NER is a key part of an information extraction system that supports robust handling of proper names essential for many applications, enables pre-processing for different classification levels, and facilitates information filtering and linking. However, performing coreference, or entity linking, as well as creating templates is not part of NER task. A basic entity identification task can be defined as follows: *Let* $\{t_1, t_2, t_3, \ldots, t_n\}$ *be a sequence of entity types denoted by* T *and let* $\{w_1, w_2, w_3, \ldots, w_n\}$ *be a sequence of words denoted by* W*, then the identification task can be defined as ‘given some* W*, find the best* T*’*.

[3] https://code.google.com/p/boilerpipe/.

In particular, entity identification consists of three subtasks: identifying entity names, temporal expressions, and number expressions, where the expressions to be annotated are ‘unique identifiers’ of entities (organizations, persons, locations), times (dates, times), and quantities (monetary values, percentages). Most research on entity extraction systems has been structured as taking an un-annotated block of text (e.g., “Obama was born on August 4, 1961, at Gynecological Hospital in Honolulu”) and producing an annotated block of text, such as the following:[4]

```
<ENAMEX TYPE=‘‘PERSON’’>Obama</ENAMEX> was born on <TIMEX TYPE=‘‘DATE’’>August 4, 1961,</TIMEX> at
<ENAMEX TYPE=‘‘ORGANIZATION’’>Gynecological Hospital</ENAMEX> in <ENAMEX TYPE=‘‘CITY’’>Honolulu</ENAMEX>.
```

where, entity types such as person, organization, and city are recognized.

However, NER is not just matching text strings with pre-defined lists of names. It should recognize entities not only in contexts where category definitions are intuitively quite clear, but also in contexts where there are many grey areas caused by metonymy. Metonymy is a figure of speech used in rhetoric in which a thing or concept is not called by its own name, but by the name of something intimately associated with that thing or concept. Metonyms can be either real or fictional concepts representing other concepts real or fictional, but they must serve as an effective and widely understood second name for what they represent. For example, (i) *person vs. artefact* “The Ham Sandwich (a person) wants his bill. vs “Bring me a ham sandwich.”; (ii) *organization vs. location* “England won the World Cup” vs. “The World Cup took place in England”; (iii) *company vs. artefact* “shares in MTV” vs. “watching MTV”; and (iv) *location vs. organization* “she met him at Heathrow” vs. “the Heathrow authorities”.

To address these challenges, the Message Understanding Conferences (MUC) were initiated and financed by Defense Advanced Research Projects Agency (DARPA) to encourage the development of new and better methods of information extraction. The tasks grew from producing a database of events found in newswire articles from one source to production of multiple databases of increasingly complex information extracted from multiple sources of news in multiple languages. The databases now include named entities, multilingual named entities, attributes of those entities, facts about relationships between entities, and events in which the entities participated. MUC essentially adopted the simplistic approach of disregarding metonymous uses of words, e.g. ‘England’ was always identified as a location. However, this is not always useful for practical applications of NER, such as in the domain of sports.

MUC defined basic problems in NER as follows: (i) variation of named entities: for example John Smith, Mr Smith, and John may refer to the same entity; (ii) ambiguity of named entities types: for example John Smith (company vs. person), May (person vs. month), Washington (person vs. location), and 1945 (date vs. time); (iii) ambiguity with common words: for example ‘may’; and (iv) issues of style, structure, domain, genre etc. as well as punctuation, spelling, spacing, and formatting. To address these

[4] In this example, the annotations have been done using so-called ENAMEX (a user defined element in the XML schema) tags that were developed for the Message Understanding Conference in the 1990s.

challenges, existing approaches to entity extraction proposed four primary steps [26, 64,77,84], described as follows:

Format analysis In this step, the goal is the identification and handling of the formatting content embedded within documents that controls the way the document is rendered on a computer screen or interpreted by a software program. For example, HTML documents contain HTML tags specifying formatting information such as new line starts, bold emphasis, and font size or style. Format analysis is also referred to as structure analysis, format parsing, tag stripping, format stripping, text normalization, text cleaning, and text preparation.

Tokeniser Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens.This module is responsible for segmenting text into tokens, e.g., words, numbers, and punctuation. The list of tokens becomes input for further processing such as parsing or text mining.

Gazetteer The role of the gazetteer is to identify entity names in the text based on lists. These lists are used to find occurrences of these names in text, e.g. for the task of named entity recognition. Gazetteers usually do not depend on Tokens or on any other annotation and instead find matches based on the textual content of the document. As an output, this module will generate a set of named entities (e.g., towns, names, and countries) and key words (e.g., company designators and titles).

Grammar This module is responsible for hand-coded rules for named entity recognition. NER systems are able to use linguistic grammar-based techniques as well as statistical models. Hand-crafted grammar-based systems typically obtain better precision, but at the cost of lower recall and months of work by experienced computational linguists. Statistical NER systems typically require a large amount of manually annotated training data.

4.2 Knowledge enrichment

The goal of knowledge enrichment is to facilitate the process of identifying co-referring entity sets. To achieve this, extracted mentions in the Intra-Document Processing phase should be compared by applying various features to pairs of entities. Figure 3 illustrates a simple example for calculating various featurization classes for the pair of mentions {'Barack Obama', 'Barack Hussein Obama'}. As illustrated in the figure, these classes can be defined for entities, words around the entities (document level), and meta-data about the documents such as their type. Such features can be divided into various classes such as:

- *String match* [13,25,40,69] This feature is used to find strings that match a pattern approximately, rather than exactly. Techniques such as substring match, string overlap, pronoun match, and normalized edit distance can be used to extract this type of feature.
- *Lexical* [23,26] This feature contains computable features of single words including the n-gram (i.e. a contiguous sequence of n items from a given sequence of text

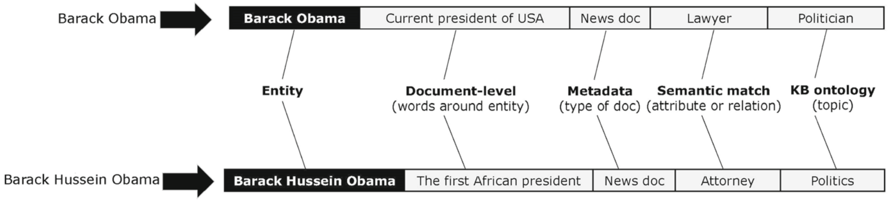


Fig. 3 Calculating featurization classes for the pair of entities (‘Barack Obama’, ‘Barack Hussein Obama’)

or speech) and the word stem (i.e. a part of a word). Techniques such as n-gram (unigram, bigram, trigram, etc.) can be used to extract this type of feature.

- *Syntactic* [59,79] This feature is based on running an in-house state of the art part of speech tagger and syntactic chunker on the data. Techniques such as phrase chunking and part-of-speech tagging can be used to extract this type of feature.
- *Pattern-based* [32] This feature is used for surrounding the words using lexical and part of speech patterns. Techniques such as pattern mining (e.g. mining item-set and temporal pattern) and Binary/categorical/numeric extraction can be used to extract this type of feature.
- *Count-based* [29,32] This feature can be applied to the coreference task and attempt to capture regularities in the size and distribution of coreference chains. Techniques to extract the total number of entities and mentions, the size of the hypothesized entity chain, the entity to mention ratio, etc can be used to extract this type of feature.
- *Semantic* [32,51,85] This feature can be used to express the existence or non-existence of pre-established semantic properties between extracted entities. Techniques such as extracting semantic properties from WordNet considering the synset (synonym ring) and hypernym (a word with a broad meaning that more specific words fall under) can be used in this caregory.
- *Knowledge-based* [24,32,41,72,86] This feature can be used to provide information about extracted entities from existing knowledge bases. Techniques for extracting information from KBs such as Google Knowledge Graph [8], Wikidata (www.wikidata.org), YAGO [20], Freebase [19], etc can be used to extract this type of feature.
- *Class-based* [9,11,12,57,61] This feature can be used to get around the sparsity of data problem while simultaneously providing new information about word usage. Techniques such as Web-scale distributional similarity, clustering and entity set expansion can be used to extract this type of feature.
- *List-/inference-based* [32] List-based features can be used to generate a list of related entities, e.g. common places, organization, names, etc. from census data and standard gazetteer information listing countries, cities, islands, ports, provinces and states. Inference-based features can be used to derive logical conclusions from premises known or assumed to be true, e.g. mentions that corefers with ‘she’ is known to be singular and female, etc.
- *History-based* [32] This feature can be used in the detection phase of entity extraction, e.g., by adding features having to do with long-range dependencies between

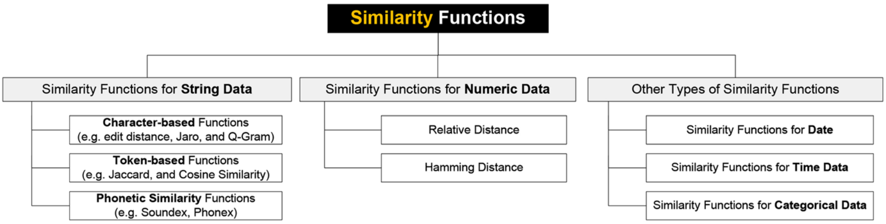

Similarity Functions
Similarity Functions for String Data
Character-based Functions (e.g. edit distance, Jaro, and Q-Gram)
Token-based Functions (e.g. Jaccard, and Cosine Similarity)
Phonetic Similarity Functions (e.g. Soundex, Phonex)
Similarity Functions for Numeric Data
Relative Distance
Hamming Distance
Other Types of Similarity Functions
Similarity Functions for Date
Similarity Functions for Time Data
Similarity Functions for Categorical Data

Fig. 4 Different categories of similarity functions

words within document processing. Extracting provenance [32,87] can play an important role to extract this feature.

- *Relationship-based* [47,51] A relationship extraction task requires the detection and classification of semantic relationship mentions within a set of entities. The output from Relationship extraction can be used as a feature in subsequent CDCR processing.

Recently, linked data [31] and Knowledge Graphs (e.g. Google Knowledge Graph [8]) have become a prominent source of information about entities. Linked data describes a method of publishing structured data so that it can be interlinked and become more useful, and provides a publishing paradigm in which not only documents, but also data, can be a first class citizen of the Web. Projects such as Wikidata, Freebase, YAGO, WikiTaxonomy [74], and DBpedia [63] have constructed huge knowledge bases of entities, their semantic classes, and relationships among entities [81]. These systems can be used to enrich the entities with additional features and consequently to improve the effectiveness of the results. Next step is to compute the similarity scores between mention groups based on the features extracted above.

4.2.1 Similarity functions and their characteristics

Approximate data matching usually relies on the use of a similarity function, where a similarity function $f(v_1, v_2) \mapsto s$ can be used to assign a score s to a pair of data values v_1 and v_2. These values are considered to be representing the same real world object if s is greater then a given threshold t. Similarity functions play a critical role in dealing with data differences caused by various reasons, such as misspellings, typographical errors, incomplete information, lack of standard formats, and so on. For example, personal name mentions may refer to the same person, but can have multiple conventions (e.g., *Barack Obama* versus *B. Obama*). In the last four decades, a large number of similarity functions have been proposed in different research communities, such as statistics, artificial intelligence, databases, and information retrieval. They have been developed for specific data types (e.g., string, numeric, or image) or usage purposes (e.g., typographical error checking or phonetic similarity detection). Figure 4 illustrates different categories of similarity functions. The functions can be categorized as follows:

Similarity functions for string data For *string* data types, in addition to exact string comparison, approximate string comparison functions [49] can be used for computing the similarity between two strings. They can be roughly categorized into three groups: *character-based*, *token-based* and *phonetic* functions. *Character-based functions* consider characters and their positions within strings to estimate the similarity [39]. Examples are: (i) *edit distance* is measured based on the smallest number of edit operations (insertions, deletions, and substitutions) required to transform one string to the other. This function is expensive or less accurate for measuring the similarity between long strings; (ii) *Jaro and Jaro-Winkler* is measured by computing the string similarity by considering the number of common characters and transposed characters. These functions are likely to work well for comparing short strings (e.g., personal names); and (iii) *Q-grams* Given a string S, its positional q-grams are obtained by 'sliding' a window of length q over the characters of S. Monograms, bigrams, and trigrams are examples of a q-gram.

Token-based functions might be appropriate in situations where the string mismatches come from rearrangement of tokens (e.g., "James Smith" versus "Smith James") or the length of strings is long, such as the content of a document or a message [34]. Examples are: (i) *Jaccard* This function tokenizes two strings `s` and `t` into tokensets `S` and `T`, and quantifies the similarity based on the fraction of common tokens in the sets: $\frac{(S \cap T)}{(S \cup T)}$. For example, the jaccard similarity between "school of computer science" and "computer science school" is $\frac{3}{4}$. This function works well for the cases where word order of strings is unimportant; and (ii) *TF/IDF Cosine Similarity* This function computes the closeness by converting two strings into unit vectors and measuring the angle between the vectors. In situations where the word frequency is important, e.g. in search engines, this function is likely to work better than Jaccard similarity which is insensitive to the word frequency.

Phonetic similarity functions describe how two strings are phonetically similar to each other in order to compute the string similarity. Examples are: (i) *Soundex* [50] this function is one of the best known phonetic functions, converts a string into a code according to an encoding table; (ii) *Phonex/Phonix* [55] this function is an alternative function to Soundex, which was designed to improve the encoding quality by preprocessing names based on their pronunciations. (iii) *Double Metaphone* [73] this function performs better for string matching in non-English languages, like European and Asian, rather than the soundex function that is suitable for English. Thus it uses more complex rules that consider letter positions as well as previous and following letters in a string, compared with the soundex function.

Similarity functions for numeric data For *numeric* attributes, one can treat numbers as strings and then compare them using the similarity functions for string data described above or choose different functions for comparing numeric values as follows: (i) *relative distance* The relative distance is used for comparing numeric attributes x and y (e.g., price, weight, size): $R(x, y) = \frac{|x-y|}{max\{x,y\}}$; and (ii) *Hamming distance* The Hamming distance is the number of substitutions required to transform one number to the other. Unlike other functions (e.g., relative distance), it can be used only for comparing two numbers of equal length. For example, the Hamming distance between "2121" and “2021” is 1 as there is one substitution (1 → 2). The Hamming distance is used mainly for numerical fixed values, such as postcode and SSN [66].

Similarity functions for date or time data Date and time values must be converted to a common format in order to be compared with each other. For example, possible formats for date type (considering day as ‘dd’, month as ‘mm’, and year as ‘yyyy’/‘yy’) include: ‘ddmmyyyy’, ‘mmddyyyy’, ‘ddmmyy’, ‘mmddyy’, and so on. For time type, times could be given as strings of the form ‘hhmm’ or ‘mmhh’ in 24 hours format.
Similarity functions for categorical data Other examples of similarity functions include the functions for categorical data [62]. For *categorical* features (whose values come from a finite domain), the similarity can be computed in a similar way to binary data types. For example, the score ‘1’ is assigned for a match and the score ‘0’ for a non-match.

4.3 Clustering

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar (in some sense or another) to each other than to those outside the cluster. In information extraction, identifying the equivalence classes of entity mentions is the main focus: it is important that an entity and all its mentions are placed in the same equivalence class. In this context, the goal of coreference resolution will be to identify and connect all textual entity mentions that refer to the same entity. There are different clustering approaches such as hierarchical and partitional clustering. For example, hierarchical approaches groups data objects with a sequence of partitions, either from singleton clusters to a cluster including all individuals or vice versa. Hierarchical procedures can be either agglomerative[5] or divisive.[6] Partitional clustering approaches attempt to divide the data set into a set of disjoint clusters without the hierarchical structure. The most popular partitional clustering algorithms are the prototype-based [88] and distance-based [89] clustering algorithms.

In the context of CDCR process, an intra-document coreference system can be used to identify each reference and to decide if these references refer to a single individual or multiple entities. The classic works on clustering [43,45,48] adapted the Vector Space Model (VSM, an algebraic model for representing text documents as vectors of identifiers) or deployed different information retrieval techniques for entity disambiguation and clustering. For example, the clustering approach presented in [45], proposed some extensions to VSM for cross-document coreference clustering by constructing a vector space representation derived from local/global contexts of entity mentions in documents and then performed some form of clustering on these vectors. Such works showed that clustering documents by their domain specific attributes such as domain genre will affect the effectiveness of cross-document coreferencing.

[5] Agglomerative algorithms begin with each element and merge them in successively larger clusters.

[6] Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

The state-of-the-art approaches in CDCR clustering, rely on mention (string) matching, syntactic features, and linguistic resources like English WordNet [70]. For example, some techniques [33,56] create a coreference chain for each unique entity within a document and then group related coreference chains in similar clusters. Then, they use a streaming clustering approach with common coreference similarity computations to achieve high performance on large datasets. Dynamic clustering approach [41] consists of two stages: update and merge. Update adds points to existing clusters or creates new clusters, while merge combines clusters to prevent the clusters from exceeding a fixed limit. Comparing to the agglomerative clustering approach (which has the quadratic cost), the streaming clustering provides a potentially linear performance in the number of observations since each document need only be examined a single time. Furthermore, supervised [44], semi-supervised [42], and unsupervised [27,90] approaches have used clustering to group together different nominal characteristics referring to the same entity.

Another line of related work, e.g. [5,46], added a discriminative pairwise mention classifier to a VSM-like model. For example, Mayfield et al. [5], clustered the resulting entity pairs by eliminating any pair with an SVM output weight of less than 0.95, then they treated each of the connected components in the resulting graph as a single entity. Ah-Pine et al. [21] proposed a clique-based clustering method based upon a distributional approach, which allows one to extract, analyze and discover highly relevant information for corpus specific NEs annotation. Another line of related work [33] proposed techniques for clustering text mentions across documents and languages simultaneously. Such techniques may produce cross-lingual entity clusters. Some later work [22,35] relies on the use of extremely large corpora which allow very precise, but sparse features. For example, Ni et al. [35] enhanced the open-domain classification and clustering of named entity using linked data approaches.

5 CDCR and large datasets

We analyzed the studies that classified into ‘scalable approaches to CDCR’ [12,13,32,37,42,47,48,61,72,74,75,81] and identified the key challenges in CDCR over large datasets. Figure 5 illustrates the main challenges including pre-processing the documents, identification of entities, partitioning the entities, entity pairs filtering and featurization, similarity computation, coreference classification, and clustering. These challenges have been identified by considering two main characteristics: (i) large amounts of detailed information; and (ii) advanced analytics including artificial intelligence, natural language processing, data mining, statistics and so on. Also the metadata for imbuing the entities with additional semantics will generate another line of challenges in CDCR, namely the metadata collected to model the evolution of entities over time. For example, ‘Barack Obama’ can be a student in the Harvard Law School in a period of time and can be the president of the United States in another time. It is important to consider the evolution of entities over time.

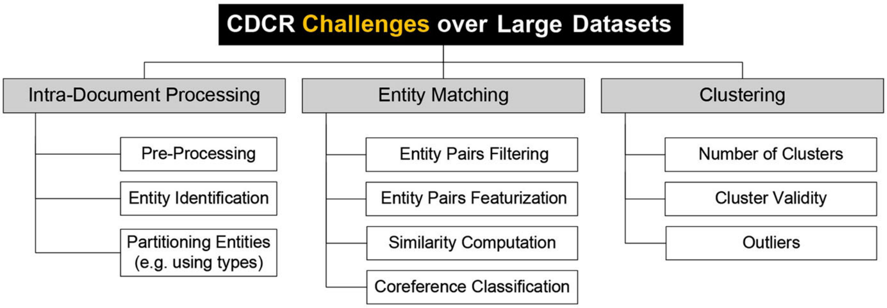

CDCR Challenges over Large Datasets
Intra-Document Processing
Pre-Processing
Entity Identification
Partitioning Entities (e.g. using types)
Entity Matching
Entity Pairs Filtering
Entity Pairs Featurization
Similarity Computation
Coreference Classification
Clustering
Number of Clusters
Cluster Validity
Outliers

Fig. 5 Identification of key challenges in CDCR over large datasets

5.1 Challenges

Intra-document processing Challenges in this category starts with pre-processing large number of documents. Also, in the entity identification phase, entity extraction subtasks such as format analysis, tokeniser, gazetteer, and grammar would have to be applied to very large number of documents. This is challenging as, in terms of scalability, entity extraction outputs more data than it takes. For example, the English Gigaword dataset contains more than nine million documents and will produce orders of magnitude more information. Before matching entity pairs and for reducing the number of pairs, some works propose to partition entities based on their types/subtypes: the larger the dataset, the more the variety of different entities.

Entity matching Currently, the dominant methods for co-reference measure the compatibility between pairs of mentions. These methods suffer from a number of drawbacks including difficulties scaling to large numbers of mentions and limited representational power [61]. In particular, in terms of scalability, pairwise entity comparison will become exponential across documents. In the coreference classification step, various similarity metrics should be calculated for all generated paired entities, and then the huge number of coreferent entities should be clustered and placed in the same equivalence class. Recent research [7,40,54,61] have studied methods that measure the compatibility between mention pairs (i.e., the dominant approach to coreference) and showed that these approaches suffer from a number of drawbacks including difficulties scaling to large numbers of mentions.

For example, Wick et al. [61] proposed to replace the pairwise approaches with a more expressive and highly scalable alternatives, e.g., discriminative hierarchical models that recursively partitions entities into trees of latent sub-entities. Wellner et al. [7] proposed an approach to integrated inference for entity extraction and coreference based on conditionally-trained undirected graphical models. Luo et al. [53] proposed an approach for coreference resolution which uses the Bell tree to represent the search space and casts the coreference resolution problem as finding the best path from the root of the Bell tree to the leaf nodes. Wick et al. [40] proposed a discriminatively-trained model that jointly performs coreference resolution and canonicalization, enabling features over hypothesized entities.

Clustering Many of the solutions to CDCR involve data-driven techniques, such as clustering. In particular, clustering techniques facilitate the processing of larger amounts of data. In this context, clustering should help in obtaining better CDCR results as the size of the datasets increases. But in reality, as the size of the dataset becomes larger, the variety of different entities and contexts that have to be dealt with also increases. Consequently, as the contexts in which mentions occur become more diverse, clustering approaches potentially become harder [12], where challenges include the huge number of clusters, clustering results validation, and also outliers.[7]

5.2 Solutions

A recent line of work [9–13] uses Apache Hadoop solution to tackle CDCR challenges over large datasets. Hadoop is an open source framework that uses a simple programming model to enable distributed processing of large data sets on clusters of computers. Four key elements of an Apache Hadoop solution include: (i) Apache Hadoop Distributed File System (Apache HDFS): is a distributed file system designed to run on commodity hardware. (ii) Apache MapReduce [14], is the heart of Hadoop. It is the programming paradigm that allows for large scalability across hundreds or thousands of servers in a Hadoop cluster; (iii) Apache Pig [91], is a high-level platform for creating MapReduce programs; and (iv) Apache HCatalog, is a table and storage management layer for Hadoop that enables users to easily read and write data using Apache Pig and MapReduce.

MapReduce and intra-document processing Singh et al. [13] proposed a distributed inference that uses parallelism to enable large scale processing. The approach uses a hierarchical model of coreference that represents uncertainty over multiple granularities of entities. The approach facilitates more effective approximate inference for large collections of documents. They divided the mentions and entities among multiple machines, and propose moves of mentions between entities assigned to the same machine. This ensures all mentions of an entity are assigned to the same machine. Kolb et al. [10] proposed a tool called Dedoop (Deduplication with Hadoop) for MapReduce-based entity resolution of large datasets. Dedoop automatically transforms the entity resolution workflow definition into an executable MapReduce workflow.

MapReduce and entity matching Elsayed et al. [9] proposed a MapReduce algorithm for computing pairwise document similarity in large document collections. The authors focused on a large class of document similarity metrics that can be expressed as an inner product of term weights. They proposed a two step solution to the pairwise document similarity problem: (i) indexing, where a standard inverted index algorithm

[7] An outlier is an observation point that is distant from other observations.

[92] in which each term is associated with a list of document identifiers for each document; and (ii) pairwise similarity, where the MapReduce mapper generates key tuples corresponding to pairs of document IDs. Later on, the key tuples will be associated with the product of the corresponding term weights. Pantel et al. [11] proposed a scalable MapReduce-based implementation based on distributional similarity, where the approach followed a generalized sparse-matrix multiplication algorithm [58].

MapReduce and clustering Sarmento et al. [12] proposed a multi-pass semantic graph-based clustering approach to large scale named-entity disambiguation. In particular, semantic graphs are semi-structured data with very different contextual clues and need different approaches to identify potentially coreferent entities. The proposed MapReduce-based algorithm is capable of dealing with an arbitrarily high number of entities types and is able to handle unbalanced data distributions while producing correct clusters both from dominant and non-dominant entities. Complementary to this approach, Sleeman et al. [38] proposed an approach to reduce the computational cost of identifying coreferent instances in heterogeneous semantic graphs. The approach use techniques to map the attributes to a common dictionary and to perform entity typing for heterogeneous graphs. According to these related works, MapReduce algorithm design could lead to data skew and the curse of the last reducer and consequently careful investigation is needed while mapping an algorithm into the MapReduce plan.

6 Evaluation of CDCR tools and techniques

So far we have classified the selected studies in CDCR (Sect. 4) into different categories including intra-document processing (Sect. 4.1), knowledge enrichment (Sect. 4.2), clustering (Sect. 4.3), and scalable approaches to CDCR (Sect. 5). Notice that, during the synthesis and classification of selected papers, we have also analyzed the information about tools/algorithms proposed in selected publications to guarantee the completeness of the results. In Sect. 6.5 we present the tools/algorithms that can be used for each of the specific tasks of CDCR process.

In this section we present the evaluation dimensions that can be used to evaluate the performance of CDCR techniques. To illustrate the applicability of these evaluation dimensions, we assess state-of-the-art tools/algorithms for two CDCR subtasks (entity identification, similarity computation, and coreference classification) and highlight challenges in each of them to help readers identify important and outstanding issues for further investigation. We have also identified a set of *benchmarking datasets* that have been used in the extracted publications. We classified these datasets to be useful for evaluating the effectiveness (concerns with achieving a high quality coreference result) and efficiency (concerns performing the coreference resolution as fast as possible for large datasets) of CDCR approaches.

The remainder of this section is organized as follows. In Sect. 6.1 we present the benchmarking datasets. In Sect. 6.2 we describe the various dimensions for evaluating a CDCR system's performance. Finally, we assess state-of-the-art tools/algorithms for entity identification (Sect. 6.3), similarity computation and coreference classification (Sect. 6.4) as an important subtasks of CDCR process.

Table 4 Examples of benchmarking datasets

Dataset	Description
John Smith corpus	This dataset is one of the first efforts for creating corpora to train and evaluate cross-document co-reference resolution algorithms. The corpus is a highly ambiguous dataset which consisted of 197 articles from 1996 and 1997 editions of the NY Times
ACE (2008) corpus	This dataset includes approximately 10,000 documents from several genres (predominantly newswire). The dataset includes 400 annotated documents
reACE	This dataset consists of English broadcast news and newswire data originally annotated for the ACE dataset. Comparing to ACE dataset, this dataset provides more annotation to offer a sufficient level of ambiguity and reasonable ground-truth
English Gigaword	This dataset is a comprehensive archive of newswire text data and the fifth edition of this dataset includes 7 distinct international sources of English newswire and contains more than 9 million documents. This large dataset is not annotated
TAC-KBP corpus	This dataset links entity mentions to corresponding Wikipedia entities, focusing on ambiguous person, organization, and geo-political entities mentioned in newswire
	The dataset contains over 1.2 million documents, primarily newswire
Google's Wikilinks	This dataset comprises of 40 million mentions over 3 million entities gathered using an automatic method based on finding hyperlinks to Wikipedia from a web crawl and using anchor text as mentions [75]

6.1 Benchmarking datasets

Measuring the effectiveness of CDCR task on a large corpus is challenging and needs large datasets providing sufficient level of ambiguity (the ability to express more than one interpretation) and sound ground-truth (the accuracy of the training set's classification for supervised learning techniques). Several manually/automatically labeled datasets have been constructed for training and evaluation of coreference resolution methods', however, CDCR supervision will be challenging as it has a exponential hypothesis space in the number of mentions. Consequently, the manual annotation task will be time-consuming, expensive and will result in small number of ground-truths. A few projects [3,30,65] have introduced manually-labeled, small datasets containing high ambiguity. Using such small datasets, it would be hard to evaluate the *effectiveness* of the CDCR techniques. Several automatic methods for creating CDCR datasets have been proposed to address this shortcoming. For example, recently, Google released the Wikilinks Corpus [75] which includes more than 40 million total disambiguated mentions over 3 million entities within around 10 million documents. Table 4 provides more details about John Smith [3], ACE [93], reACE [68], English Gigaword,[8] TAC-KBP [94] and Google's Wikilinks [75] datasets.

[8] http://catalog.ldc.upenn.edu/LDC2012T21.

6.2 Evaluation dimensions

An important problem in CDCR task is how to evaluate a system's performance. There are two requirements that lie at the heart of CDCR task: effectiveness and efficiency. Effectiveness is concerned with achieving a high quality coreference result. For the evaluation of accuracy, well-known measures such as *precision* (the fraction of retrieved instances that are relevant) and *recall* (the fraction of relevant instances that are retrieved) [95] can be used. Efficiency is concerned with performing the coreference resolution as fast as possible for large datasets. In this context, a good performance metric should have the following two properties [28]: (i) discriminativity: the ability to differentiate a good system from a bad one. For example, precision and recall have been proposed to measure the effectiveness of information retrieval and extraction tasks, where high recall means that an algorithm returned most of the relevant results and high precision means that an algorithm returned substantially more relevant results than irrelevant; and (ii) interpretability: emphasizes that a good metric should be easy to interpret. In particular, there should be an intuitive sense of how good a system is when a metric suggests that a certain percentage of coreference results are correct. For example, when a metric reports 95 % or above correctness for a system, we would expect that the vast majority of mentions are in right coreference chains.

As the complementary to precision/recall, some approaches such as link-based F-measure [60], count the number of common links between the truth (or reference) and the response. In these approaches, the link precision is the number of common links divided by the number of links in the system output, and the link recall is the number of common links divided by the number of links in the reference. The main shortcoming of these approaches is that they fail to distinguish system outputs with different qualities: they may result in higher F-measures for worse systems. Some other value-based metric such as ACE-value [96] count the number of false-alarm (the number of misses) and the number of mistaken entities. In this context, they associate each error with a cost factor that depends on things such as entity type (e.g., location and person) as well as mention level (e.g., name, nominal, and pronoun). The main shortcoming of these approaches is that they are hard to interpret. For example a system with 90 % ACE-value does not mean that 90 % of system entities or mentions are correct: the cost of the system, relative to the one outputting zero entities is 10 %. To address this shortcoming, approaches such as Constrained Entity-Aligned F-Measure (CEAF) [28] have been proposed to measure the quality of a coreference system where an intuitively better system would get a higher score than a worse system, and is easy to interpret. B-cubed metric [43], a widely used approach, proposed to address the aforementioned shortcomings by first computing a precision and recall for each individual mention and then taking the weighted sum of these individual precisions and recalls as the final metric. The key contributions of this approach include: promotion of a set-theoretic evaluation measure, B-CUBED, and the use of TF/IDF weighted vectors and cosine similarity in single-link greedy agglomerative clustering. In particular, B-Cubed looks at the presence/absence of entities relative to each of the other entities in the equivalence classes produced: the algorithm computes the precision and recall numbers for each entity in the document, which are then combined to produce final precision and recall numbers for the entire output.

6.3 Tools for entity identification and their evaluation

In this section, we assess a set of named entity extraction systems including: Stanford-NLP,[9] OpenNLP,[10] LingPipe,[11] Supersense Tagger,[12] AFNER,[13] and AlchemyAPI.[14] We have selected these tools as they are the result of studies classified in the Popularization category (Sect. 3): studies that demonstrate the applicability of the proposed solutions in real world applications. The assessment only consider the names of persons, locations and organizations. The motivation behind this assessment is to provide a complementary vision for the results of domain independent systems that permit the processing of texts. For this assessment we use reACE and English Gigaword datasets. These datasets have been introduced in Sect. 6.1. The reACE dataset was developed at the University of Edinburgh, which consists of English broadcast news and newswire data originally annotated for the ACE (Automatic Content Extraction) program. This dataset contains few annotated documents and cannot be used to evaluate the efficiency of state-of-the-art approaches in CDCR. We use reACE to evaluate the effectiveness of CDCR tools. English Gigaword dataset is a comprehensive archive of newswire text data that has been acquired over several years at the University of Pennsylvania. The fifth edition of this dataset includes seven distinct international sources of English newswire and contains more than 9 million documents. This large dataset is not annotated. We use this dataset to assess the efficiency of the CDCR approaches. Table 5 provides a brief description of the selected tools.

Analysis The data analysis has been realized having focused on a comparison of results obtained by the tools, for entities found in the test corpus: the English Gigaword corpus has been used in order to evaluate the behavior of the tools. The assessment only consider the names of persons, locations and organizations. For the evaluation of accuracy, we use the well-known measures of precision and recall [95]. As discussed earlier, precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved. In particular, precision measures the quality of the matching results and is defined by the ratio of the correct entities to the total number of entities found:

$$Precision = \frac{Number\text{-}of\text{-}correct\text{-}entities\text{-}found}{Total\text{-}number\text{-}of\text{-}entities\text{-}extracted}$$

Recall measures coverage of the matching results and is defined by the ratio of the correct entities matched to the total number of all correct entities that should be found.

$$Recall = \frac{Number\text{-}of\text{-}correct\text{-}entities\text{-}found}{Total\text{-}number\text{-}of\text{-}correct\text{-}entities\text{-}that\text{-}should\text{-}be\text{-}found}$$

[9] http://nlp.stanford.edu/.

[10] http://opennlp.apache.org/.

[11] http://alias-i.com/lingpipe/.

[12] http://sites.google.com/site/massiciara/.

[13] http://afner.sourceforge.net/afner.html.

[14] http://www.alchemyapi.com/.

Table 5 Selected entity identification tools for the evaluation purpose

Tool	Description
Stanford-NLP	This tool is an integrated suite of NLP tools for English in Java, including tokenization part-of-speech tagging, named entity recognition, parsing, and coreference. Stanford NER provides a general implementation of linear chain conditional random field (CRF) sequence models coupled with well-engineered feature extractors for named entity recognition
OpenNLP	This tool is a machine learning based toolkit for the processing of natural language text. It supports the most common NLP tasks, such as tokenization, sentence segmentation part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution
LingPipe	This tool is used to detect named entities in news, classify Twitter search results into categories, and suggest correct spellings of queries. It includes multi-lingual, multi-domain, and multi-genre models as well as training with new data for new tasks
Supersense tagger	This tool is an open-source tagger designed for the semantic tagging of nouns and verbs based on WordNet categories which include persons, organizations, locations, temporal expressions and quantities [97]. It is based on automatic learning, offering three different models for application: CONLL (used in our evaluation), WSJ and WNSS
AFNER	AFNER is an open-source NERC tool which is capable of recognizing persons names, organizations locations, miscellanea, monetary quantities, and dates in English texts [98]. This tool uses regular-expressions to find simple case named entities. It supports finding the parts of text matching listed named entities
AlchemyAPI	This tool provides APIs to utilize NLP technology and machine learning algorithms. It supports multiple languages and offers comprehensive disambiguation capabilities solutions

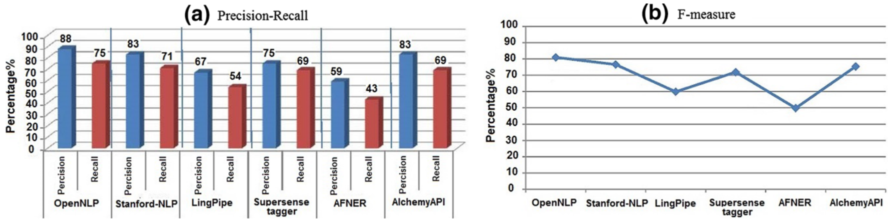
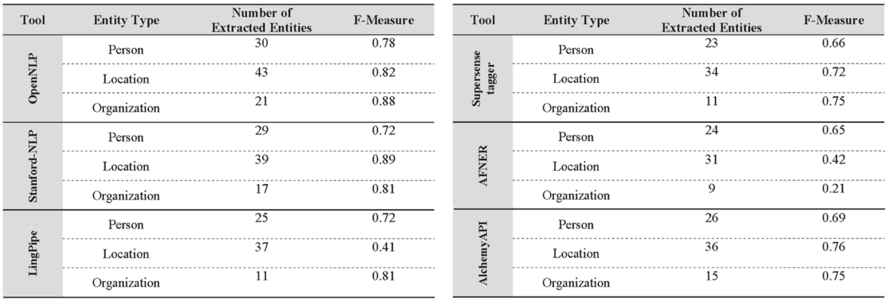


Fig. 6 Precision-recall (**a**) and F-measure (**b**) in entity identification (English Gigaword Dataset)

For an approach to be effective, it should achieve a high precision and high recall. However, in reality these two metrics tend to be inversely related [95]. The evaluation has been realized through distinct measures of precision and recall based on identification of the entities and false-positives (i.e. the incorrect rejection of a true null hypothesis, which may lead one to conclude that a thing exists when really it does not) in the identification.

Comparison discussion Figure 6 illustrates the precision-recall and F-measure (harmonic mean of precision and recall) for Stanford-NLP, OpenNLP, LingPipe, Supersense Tagger, AFNER, and AlchemyAPI. In this evaluation, OpenNLP, StanfordNLP, and AlchemyAPI stand out with their behavior for obtaining F-measure rate

Table 6 Results by entity type (English Gigaword Dataset)



Tool	Entity Type	Number of Extracted Entities	F-Measure	Tool	Entity Type	Number of Extracted Entities	F-Measure
OpenNLP	Person	30	0.78	Supersense tagger	Person	23	0.66
	Location	43	0.82		Location	34	0.72
	Organization	21	0.88		Organization	11	0.75
Stanford-NLP	Person	29	0.72	AFNER	Person	24	0.65
	Location	39	0.89		Location	31	0.42
	Organization	17	0.81		Organization	9	0.21
LingPipe	Person	25	0.72	AlchemyAPI	Person	26	0.69
	Location	37	0.41		Location	36	0.76
	Organization	11	0.81		Organization	15	0.75

identification (81 % OpenNLP). One interesting point here would be to know if combination of these approaches could result in a higher F-measure than OpenNLP. To achieve this goal, there is a need to provide the precision-recall for each entity type (e.g. persons, locations, and organizations) separately. It would be important to consider the false positive errors too. Consequently, we took a further analysis to account for the false positive errors by calculating the number of persons, locations, and organizations and the F-measure for them. Table 6 illustrates the results by entity type. In this experiment, we did not analyze the number of categories that each tool can recognize, as the utility and difficulty of recognition of some types against some others is different and demonstrates the need for a study based on the entity's type. In this context, the study was carried out for person, location, and organization types that the tools were able to recognize in the corpus. The analysis illustrated in Table 6 allows us to observe what combination of the approaches could result in a higher F-measure to identify a specific type of entity. For example, it is remarkable how OpenNLP has an F-measure on the entity type Person of 0.78, whilst AFNER achieves 0.65. As another example, Stanford-NLP has an F-measure on the entity type location of 0.89, whilst LingPipe achieves 0.41.

6.4 Similarity computation and coreference classification

In order to compare pairs of entities, each entity needs to be augmented with several features extracted from documents in the featurization step. Next step is to determine whether these pairs of entities are coreferent or not. This step consists of two consecutive tasks: similarity computation and coreference decision. Figure 7 illustrates the coreference classification process. The similarity computation task takes as input a pair of entities and computes the similarity scores between their features (e.g., character-, document-, or metadata-level features) using different appropriate similarity functions for the features. The coreference decision task classifies entity pairs as either "coreferent" or "not coreferent" based on the computed similarity scores between their features.

There are two alternative methods for the final coreference decision as follows: (i) threshold-based: the feature similarity scores of an entity pair might be combined by taking a weighted sum or a weight average of the scores. The entity pairs whose com-

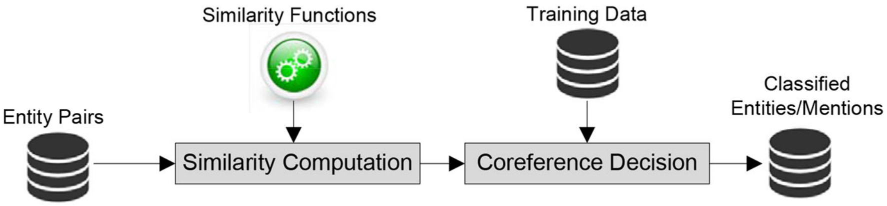
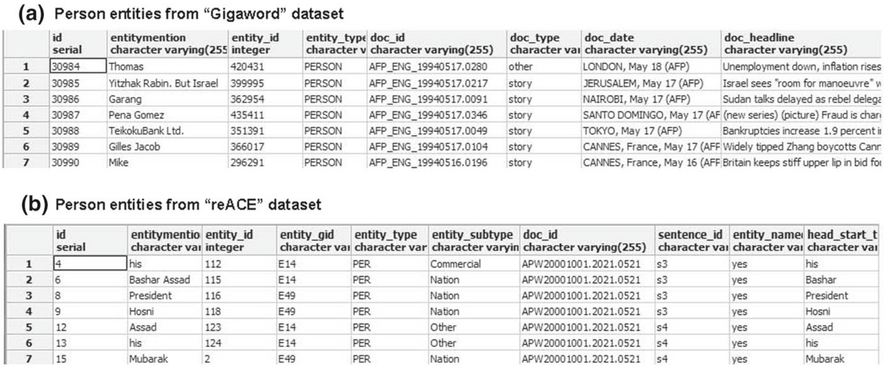


Fig. 7 Coreference classification process



(a) Person entities from "Gigaword" dataset

	id	entitymention	entity_id	entity_type	doc_id	doc_type	doc_date	doc_headline
	serial	character varying(255	integer	character v	character varying(255)	character vaı	character varying(255)	character varying(255)
1	30984	Thomas	420431	PERSON	AFP_ENG_19940517.0280	other	LONDON, May 18 (AFP)	Unemployment down, inflation rises
2	30985	Yitzhak Rabin. But Israel	399995	PERSON	AFP_ENG_19940517.0217	story	JERUSALEM, May 17 (AFP)	Israel sees "room for manoeuvre" w
3	30986	Garang	362954	PERSON	AFP_ENG_19940517.0091	story	NAIROBI, May 17 (AFP)	Sudan talks delayed as rebel delega
4	30987	Pena Gomez	435411	PERSON	AFP_ENG_19940517.0346	story	SANTO DOMINGO, May 17 (AF	(new series) (picture) Fraud is chari
5	30988	TeikokuBank Ltd.	351391	PERSON	AFP_ENG_19940517.0049	story	TOKYO, May 17 (AFP)	Bankruptcies increase 1.9 percent ii
6	30989	Gilles Jacob	366017	PERSON	AFP_ENG_19940517.0104	story	CANNES, France, May 17 (AFF	Widely tipped Zhang boycotts Canr
7	30990	Mike	296291	PERSON	AFP_ENG_19940516.0196	story	CANNES, France, May 16 (AFF	Britain keeps stiff upper lip in bid foı

(b) Person entities from "reACE" dataset

	id	entitymentio	entity_id	entity_gid	entity_type	entity_subtype	doc_id	sentence_id	entity_namei	head_start_t
	serial	character vaı	integer	character vaı	character var	character varyin	character varying(255)	character vaı	character vaı	character vaı
1	4	his	112	E14	PER	Commercial	APW20001001.2021.0521	s3	yes	his
2	6	Bashar Assad	115	E14	PER	Nation	APW20001001.2021.0521	s3	yes	Bashar
3	8	President	116	E49	PER	Nation	APW20001001.2021.0521	s3	yes	President
4	9	Hosni	118	E49	PER	Nation	APW20001001.2021.0521	s3	yes	Hosni
5	12	Assad	123	E14	PER	Other	APW20001001.2021.0521	s4	yes	Assad
6	13	his	124	E14	PER	Other	APW20001001.2021.0521	s4	yes	his
7	15	Mubarak	2	E49	PER	Nation	APW20001001.2021.0521	s4	yes	Mubarak

Fig. 8 Example person entities from two datasets

bined score is above a given threshold are considered as "coreferent"; and (ii) machine learning-based: a classifier is trained by one of machine learning techniques (e.g., SVM or decision tree) using a training data and entity pairs are classified based on the trained classifier. The similarity scores between entity pairs are used as features for classification.

For the similarity computation and the threshold-based coreference decision, we evaluate the following open-source packages: (i) SecondString[15] and SimMetrics,[16] are open-source packages that provide a variety of similarity functions used for comparing two feature attribute values. They provide different sets of similarity functions, e.g., `SecondString` does not provide `cosine` function supported by `SimMetrics`. Thus, we use both of the packages as we want to test different similarity functions for different cases; and (ii) Weka [80], is a free software package under the GNU public license, which is a collection of various machine learning algorithms developed in Java. It also provides functionalities for supporting some standard data mining tasks, such as data preprocessing, classification, clustering, regression and feature selection. The package can be applied in this project if a sufficient, suitable and balanced training data is available.

Analysis In this assessment, we use both reACE and English Gigaword datasets. Figure 8 shows some example person entities (including metadata such as document

[15] http://secondstring.sourceforge.net.

[16] http://sourceforge.net/projects/simmetrics/.

Table 7 Execution times (in seconds) for the two datasets

	Edit distance	Q-grams	Jaccard	Cosine
Gigaword (# of person entities= 20,308, # of entity pairs= 412 million)	3794	12364	1151	813
reACE (# of person entities= 3,243, # of entity pairs= 10.5 million)	37	145	17	16

identifier, type, and title) from the two datasets. We measured the overall performance using both *efficiency* and *effectiveness*. First, the efficiency is commonly determined in terms of the execution time, which is taken in comparing feature attributes using similarity functions and then making coreference decisions based on their computed similarity scores. Second, the effectiveness is determined with the standard measures precision, recall, and F-measure with respect to "perfect coreference results", which are manually determined. Let us assume that TP is the number of true positives, FP the number of false positives (wrong results), TN the number of true negatives, and FN the number of false negatives (missing results).

$$
\begin{aligned}
Precision &= \frac{TP}{TP+FP}; \quad recall = \frac{TP}{TP+FN};\\
F\text{-}measure &= \frac{2 * precision * recall}{precision + recall};
\end{aligned}
$$

For the initial evaluations, we focus on making coreference decisions for entity pairs of `Person` entity type. It should be noted that the same techniques described below would be applied to the other entity types, such as `organization`, `location`, and `date/time`. We have used two character-based (edit distance and Q-grams) and two token-based (jaccard and cosine) similarity functions. For the *Gigaword* dataset, we only measured the *execution time* as the perfect coreference results are not available. We applied the four similarity functions on the entity mention feature. For the *reACE* dataset, we measured the *execution time* as well as the *accuracy*. As in the "Gigaword" dataset, we used the four similarity functions in comparing the entity mention feature. Table 7 lists the execution times taken for making coreference decisions by comparing `person` entities of the two datasets. The table shows significant differences among the applied similarity functions. The token-based functions achieved fast execution time, when compared to the character-based functions. This may be influenced by the algorithms of those functions, e.g., the character-based functions consider characters and their positions within strings to estimate the similarity, rather than considering tokens within strings as in the token-based functions. For the both datasets, among all the functions, the `Q-grams` function is the slowest one while the `cosine` function is the fastest. When comparing 20,308 entities (the number of entity pairs is 412 millions) from "Gigaword" dataset, an execution time of 12,364 s is needed with the `Q-grams` function while an execution time of 813 s is needed with the `cosine` function.

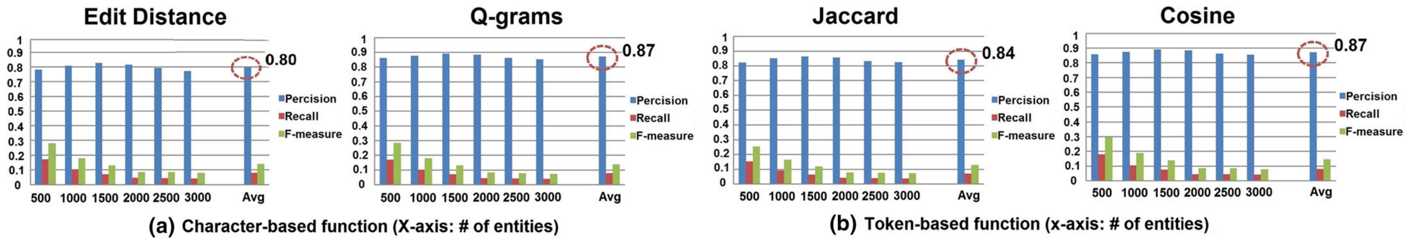
Figure 9 shows the coreference quality (precision, recall, and F-measure) results for the "reACE" dataset with different similarity functions. For each similarity function, we measured the quality results with respect to different number of entities (x axis): groups of 500, 1000, 1500, 2000, 2500, and 3000 entities; where the first 500 entities is the subset of next 1000 entities and so on. Figure 9a shows the results obtained by applying the character-based functions on just one single feature attribute of "reACE" dataset, namely `person name` entity mention. To see the overall performance (e.g., precision) of each similarity function with respect to the different entity number, we calculated the average score of returned results. For instance, considering Fig. 9a, for edit distance function, we got the average precision '0.80': (0.78 + 0.81 + 0.83 + 0.79 + 0.77) = 0.80. Among the character-based functions, the `Q-grams` function (average precision 0.87) worked better than the `edit distance` function (average precision 0.80). Figure Fig. 9(b) illustrates the results achieved by applying the token-based functions on the same feature attribute. Among the token-based functions, the `cosine` function (average precision 0.87) achieved slightly better results, compared with the `jaccard` function (average precision 0.84). Among the character-based functions, the coreference decision using the `Q-grams` function performed best while the one using the `edit distance` performed worst.

6.5 Guidance on CDCR methods and tools

Cross document coreference resolution is central to knowledge base construction and also useful for joint inference with other NLP components. The essential key to apply CDCR approaches to large datasets is to divide the CDCR process into understandable stages and scale up to the exceptionally large volume of data for each stage. The next step is to use an effective computing platform as typical extraction algorithms require all data to be loaded into the main memory. For example, intra-document processing requires computational intensive computing units for data analysis and comparisons. Currently, scalable processing mainly depends on parallel programming models like MapReduce, as well as providing a cloud computing platform. MapReduce is a batchoriented parallel computing model. An interesting topic of research here would be to improve the performance of MapReduce and enhance the real-time nature of large-scale data processing.

To address this challenge, an approach such as CDCR can be divided into several stages and each stage can be assigned to multiple MapReduce jobs. Figure 10 illustrates how we can divide a CDCR process into several stages and assign each stage into a specific MapReduce job, where each MapReduce (MR) job allows the user to configure the job, submit it, control its execution, and query the state. In each MR job, a set of existing tools/algorithms can be used. We have identified the list of tools/algorithms that can be used in each of the MR jobs. Figure 11 illustrates these tools/algorithms. In order to select a proper tool for this study, we considered the following characteristics: (i) they are not domain dependent; (ii) they do not require the user to provide resources necessary for its operation; and (iii) they process texts in the English language. We have implemented a software prototype based on the MapReduce process presented in Fig. 10. Software prototype evaluation results along with technical details of the above mentioned tools can be found in [99]. In the following, we highlight some of the challenges and lessons learnt.

Entity extraction and large datasets The entity extraction task outputs more data than it takes. Large number of documents can be used as an input to this task, and very large number of entities can be extracted. In this context, the performance of entity extraction as well as the accuracy of extracted named entities should be optimized. In



(a) Character-based function (X-axis: # of entities)

(b) Token-based function (x-axis: # of entities)

Fig. 9 Evaluation results with the four different similarity functions (threshold = 0.5)

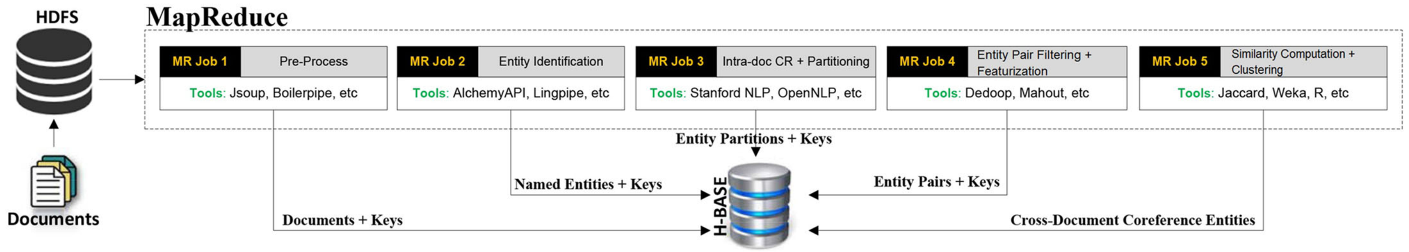

HDFS
Documents
MapReduce
MR Job 1
Pre-Process
Tools: Jsoup, Boilerpipe, etc
MR Job 2
Entity Identification
Tools: AlchemyAPI, Lingpipe, etc
MR Job 3
Intra-doc CR + Partitioning
Tools: Stanford NLP, OpenNLP, etc
MR Job 4
Entity Pair Filtering + Featurization
Tools: Dedoop, Mahout, etc
MR Job 5
Similarity Computation + Clustering
Tools: Jaccard, Weka, R, etc
Entity Partitions + Keys
Named Entities + Keys
Entity Pairs + Keys
H-BASE
Documents + Keys
Cross-Document Coreference Entities

Fig. 10 Dividing a CDCR process into several stages and assign each stage into a specific MapReduce job

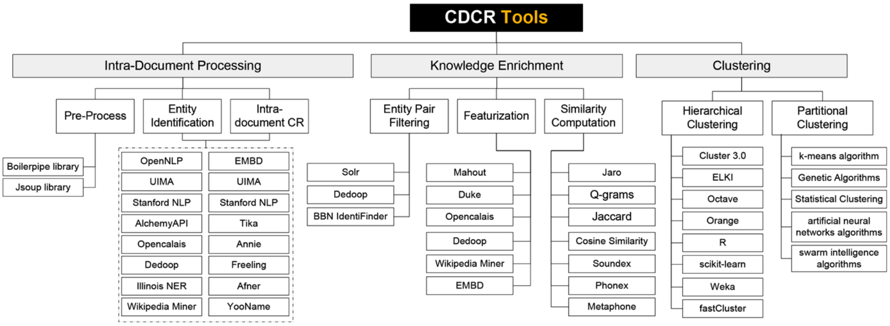


Fig. 11 Tools/algorithms that can be used for each CDCR sub-task

the second MR job (entity identification) we enable users to choose between a set of NLP tools, including OpenNLP, StanfordNLP, and AlchemyAPI. In this context, part of speech taggers performs well on the edited texts, such as news articles and Wikipedia pages. For example, the entity identification result presented in Sect. 6.3, shows a good performance which is generally over 50 %, except the recall value of the Afner tool. A detailed analysis should additionally take in account the false positive errors, i.e. the elements erroneously identified as entities, as this could result more damaging in a project than partial identification or erroneous classification. To analyze the false positive errors, we tested these NLP tools with non-edited texts, such as Twitter text. The unique style and also the noise in Twitter text reduces the performance of NLP tools and increases false positive errors. For example, capitalization is a key feature for named entity extraction within edited texts, but this feature is highly unreliable in tweets. In this context, there is a need to train the named entity tagger for the domain specific text, e.g. Twitter data. Various dictionaries and knowledge bases such as YAGO, freebase, DBpedia, and reACE can be used for training, which may help to optimize the accuracy of extracted entities.

Entity pairs filtering and featurization of extracted entities For a huge number of extracted entities, it is generally not feasible to exhaustively evaluate the Cartesian product of all input entities, and generate all possible entity pairs. To address this challenge, various blocking techniques (e.g., blocking strategy for all non-learning and learning-based match approaches) can be used to reduce the search space to the most likely matching entity pairs. Moreover, featurization of the corpus as well as extracted entities, will facilitate the filtering step and also will quickly eliminate those pairs that have little chance of being deemed co-referent. Similar to entity extraction phase, generating a knowledge-base from existing Linked Data systems may facilitate the featurization step. Various machine learning over a set of training examples can be used to classify the pairs as either co-referent or not co-referent. Different approaches have different similarity thresholds, where entity pairs with a similarity above the upper classification threshold are classified as matches, pairs with a combined value below the lower threshold are classified as non-matches, and those entity pairs that have a matching weight between the two classification thresholds are classified as possible matches. This task is challenging as we need to investigate how different configurations could have an impact on the effectiveness and efficiency of coreference classification. Three characteristics can be considered for this configuration: (i) feature attributes to be used for classification; (ii) similarity functions to be used for the chosen feature attributes; and (iii) threshold to be suitable for the classification decision.

Clustering of (cross document) co-referent entities Once the individual pairs are classified, they must be clustered to ensure that all mentions of the same entity are placed in the same equivalence class. Standard entity clustering systems commonly rely on mention (string) matching, syntactic features, and linguistic resources like English WordNet. In this context, assigning each cluster to a global entity would be challenging. For example, the cluster including "Obama, B. Obama, B.H. Obama, Barack Obama, Barack H. Obam, etc" should be considered as mentions of the global entity 'President of the United State'. Linked Data systems can be used to help identifying the entities. Also it is important to consider that, when the co-referent text mentions appear in different languages, standard clustering techniques cannot be easily applied.

7 Conclusions, research limitations and future work

In this paper we presented a systematic review of the state of the art challenges of and solutions to CDCR approaches. We discussed the central concepts, subtasks, and the current state-of-the-art in CDCR process. We provided assessment of existing tools/techniques for CDCR subtasks and highlight challenges in each of them to help readers identify important and outstanding issues for further investigation. We believe that this is an important research area, which will attract a lot of attention in the research community. In the following, we summarize research limitations and significant research directions in this area.

7.1 Research limitations

The quality of this SLR was ensured by developing and reviewing research protocol following the guidelines of conducting SLRs [17]. The research protocol contained research questions, search strategies, inclusion and exclusion criteria, data extraction form, and literature synthesis approach to be used in our review. The research protocol helped us to overcome bias in the study selection process. We ensured that the search strings were appropriately derived from research questions. Moreover, to reduce the potential bias in the selection of the studies, the study selection was done in multiple stages. The references of the included and excluded studies during each selection stage were maintained in EndNote libraries. Accuracy and consistency during the review process are usually based on a common understanding among the authors; misunderstandings can result in biased effects. One of the main limitations of the review could be the possibility of bias in study selection. To help ensure that the selection process was as much unbiased as possible, we developed detailed guidelines in the review protocol prior to the start of the review. It should be considered that bias in selection process could be a threat to internal validity [17]. To ensure the validity, the searches were performed on multiple databases to get relevant publications. In this context, the validity of data selection and its representation to address research questions is referred as Construct Validity [17]. The research protocol was developed in order to minimize the potential threat to construct validity. The research question, inclusion and exclusion criteria, search strings used for searches and data extraction strategy ensured consistent data extraction process and valid results.

7.2 Future work

Coreference resolution on non-text and multimedia data Most of the work in Coreference Resolution has been focused on text documents, however, the new sources (e.g. social media) generate a large volume of non-text and multimedia data on a continuous basis. For example it is possible to share short messages (including text and multimedia data) using Twitter or share video and photography content using YouTube and Instagram. In order for NLP systems to make a successful transition to these new sources, it is critical for coreference resolution systems to also work on non-text and multimedia data. In this context, coreference can use the information extracted from non-text and multimedia data to connect and summarize documents across bodies of human knowledge. An interesting application could be to use coreference resolution to discover the connections among entities and objects in the Open Data (e.g. social, natural, and information systems such as World-Wide Web and social networks). For example, it would be interesting to discover the hidden knowledge in the relationships among entities in Twitter, Instagram, and News data. This may enable to form an intelligence picture from the Open Data sources around a topic of interest, group related entities (text and non-text) around that topic, find paths among entities, and use all these information for the follow on analysis. This is important as the production of knowledge from Open Data is seen by many organizations as an increasingly important capability that can complement the traditional intelligence sources.

Integration of existing machine learning and natural language processing algorithms into bigdata platforms The explosion of unstructured data has created an information challenge for many organizations. Information extraction, and at its core machine learning and natural language processing algorithms, can be used to analyze these large volumes of text data. Batch-oriented systems, such as Hadoop MapReduce, have been highly successful in implementing large-scale data-intensive applications. However, these systems are not suitable for iterative machine learning and NLP algorithms, as these systems are built around an acyclic data flow model which is not the case in interactive applications and real-time operations. Apache Spark [15] has been introduced to support these types of applications while retaining the scalability and fault tolerance of MapReduce. Although Spark took the first step to address this challenge, there is a need for a declarative framework capable of efficiently integrating and supporting a broad range of machine learning and NLP tasks that require iteration.

Knowledge graph for CDCR A knowledge graph (KG) typically consists of a set of concepts organized into a taxonomy, instances for each concept, and relationships among the concepts. For example, Google-KG[17] is a graph of popular concepts and

[17] http://www.google.com/insidesearch/features/search/knowledge.html.

instances on the Web, such as places, people, actors, politicians, and many more. The 'Knowledge Graph for CDCR' may contain concepts and instances specifically for the CDCR process. This domain specific knowledge graph may include CDCR specific operations (e.g. APIs for entity extraction, coreference resolution, similarity computation, clustering, etc), external training datasets (e.g. YAGO, freebase, DBpedia, and reACE), documents, and the relationship among them. Construction of this knowledge graph is challenging and requires proper analysis of existing machine learning, NLP techniques, and knowledge bases.

High-level declarative approaches to assist users Both novices and specialists need assistance in navigating the space of possible CDCR processes. Consider a high-level declarative approach that helps a user with the exploration of the space of valid CDCR processes. Such an approach can take advantage of the taxonomy of CDCR techniques, e.g. the one that is proposed in this paper, which defines the various techniques and their properties. This approach can determine characteristics of, for example, input documents (edited texts such as news articles and/or non-edited texts such as Twitter text) and uses the taxonomy to search for combination of stages and algorithms that are valid for producing the desired result from the given input documents. In particular, an intelligent assistant can be designed for assisting users to specify the desired tradeoffs between accuracy, speed, etc and then determining which CDCR processes are appropriate: there are many possible choices for each CDCR stage and only some combinations could be valid for a specific case scenario, e.g. classifying tweets (in Twitter) based on the similarity among discovered cross-document coreference entities extracted from tweets.

Acknowledgements We Acknowledge the Data to Decisions CRC (D2D CRC), the Cooperative Research Centres Programme and the Defence Systems Innovation Centre (DSIC) for funding this research.

References

1. McCallum A (2005) Information extraction: distilling structured data from unstructured text. ACM Queue 3(9):48–57
2. Crouch R, van den Berg MH, Salvetti F, Thione GL, Ahn D (2014) Coreference resolution in an ambiguity-sensitive natural language processing system. Google Patent 8,712,758
3. Bagga A, Baldwin B (1998) Entity-based cross-document coreferencing using the vector space model. In: COLING-ACL, pp 79-85
4. Dutta S, Weikum G (2015) Cross-document co-reference resolution using sample-based clustering with knowledge enrichment. Trans Assoc Comput Linguist 3:15–28
5. Mayfield J et al (2009) Cross-document coreference resolution: a key technology for learning by reading. In: AAAI'09, pp 65-70
6. Vincent Ng, Cardie C (2002) Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp 104-111
7. Wellner B et al (2004) An integrated, conditional model of information extraction and coreference with application to citation matching. In: UAI'04, pp 593-601. AUAI Press
8. Singhal A (2012) Introducing the knowledge graph: things, not strings. Official Google Blog
9. Elsayed T, Lin JJ, Oard DW (2008) Pairwise document similarity in large collections with mapreduce. In: ACL (short papers), pp 265-268
10. Kolb L, Thor A, Rahm E (2012) Dedoop: efficient deduplication with hadoop. Proc VLDB Endow 5(12):1878–1881

11. Pantel P, Crestan E, Borkovsky A, Popescu AM, Vyas V (2009) Web-scale distributional similarity and entity set expansion. In: EMNLP, pp 938-947
12. Sarmento L, Kehlenbeck A, Oliveira EC, Ungar LH (2009) An approach to web-scale named-entity disambiguation. In: MLDM, pp 689-703
13. Singh S, Subramanya A, Pereira FCN, McCallum A (2011) Large-scale cross-document coreference using distributed inference and hierarchical models. In: ACL, pp 793-803
14. Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. Commun. ACM 51(1):107–113
15. Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I (2010) Spark: cluster computing with working sets. In: USENIX'10, pp 10-10
16. Barnawi A, Batarfi O, Beheshti SMR, Elshawi R, Nouri R, Sakr S (2014) On characterizing the performance of distributed graph computation platforms. In: TPCTC
17. Keele S (2007) Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, EBSE Technical Report EBSE-2007-01
18. Cornolti M, Ferragina P, Ciaramita M (2013) A framework for benchmarking entity-annotation systems. In: WWW'13, pp 249-260
19. Bollacker KD, Evans C, Paritosh P, Sturge T, Taylor J (2008) Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD Conference, pp 1247-1250
20. Suchanek FM, Kasneci G, Weikum G (2007) Yago: a core of semantic knowledge. In: WWW, pp 697-706
21. Ah-Pine J, Jacquet G (2009) Clique-based clustering for improving named entity recognition systems. In: EACL, pp 51-59
22. Attardi G, Rossi SD, Simi M (2010) Tanl-1: coreference resolution by parse analysis and similarity clustering. In: SemEval'10, pp 108-111
23. Bengtson E, Roth D (2008) Understanding the value of features for coreference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08, pp 294-303
24. Bryl V, Giuliano C, Serafini L, Tymoshenko K (2010) Using background knowledge to support coreference resolution. In: ECAI, pp 759-764
25. Chen C, Ng V (2012) Combining the best of two worlds: a hybrid approach to multilingual coreference resolution. EMNLP-CoNLL, p 56
26. Chen H-H, Ding Y-W, Tsai S-C (1998) Named entity extraction for information retrieval. Comput Process Orient Lang 12(1):75–85
27. Elsner M, Charniak E, Johnson M (2009) Structured generative models for unsupervised named-entity clustering. In: HLT-NAACL, pp 164-172
28. Luo X (2005) On coreference resolution performance metrics. In: HLT'05, pp 25-32
29. Màrquez L, Recasens M, Sapena E (2013) Coreference resolution: an empirical study based on semeval-2010 shared task 1. Lang Resour Eval 47(3):661–694
30. Luisa B, Christian G, Emanuele P (2008) Creating a gold standard for person crossdocument coreference resolution in italian news. In: The Workshop Programme, p 19
31. Bizer C, Heath T, Berners-Lee T (2009) Linked data—the story so far. Int J Semant Web Inf Syst 5(3):1–22
32. Daumé III H, Marcu D (2005) A large-scale exploration of effective global features for a joint entity detection and tracking model. In: HLTNLP'05, pp 97-104
33. Green S, Andrews N, Gormley MR, Dredze M, Manning CD (2012) Entity clustering across languages. In: HLT-NAACL, pp 60-69
34. Köpcke H, Thor A, Rahm E (2010) Learning-based approaches for matching web data entities. IEEE Internet Comput 14(4):23–31
35. Ni Y, Zhang L, Qiu Z, Wang C (2010) Enhancing the open-domain classification of named entity using linked open data. Int Semantic Web Conf 1:566–581
36. Niu C, Li W, Srihari RK (2004) Weakly supervised learning for cross-document person name disambiguation supported by information extraction. In: ACL'04, USA
37. Singh S, Wick ML, McCallum A (2010) Distantly labeling data for large scale cross-document coreference. CoRR. arXiv:1005.4298
38. Sleeman j, Finin T (2013) Entity type recognition for heterogeneous semantic graphs. In: Semantics for Big Data, AAAI Technical Report FS-13-04

39. Wang J, Li G, Feng J (2011) Fast-join: an efficient method for fuzzy token matching based string similarity join. In: ICDE, pp 458-469
40. Wick ML, Culotta A, Rohanimanesh K, McCallum A (2009) An entity based model for coreference resolution. In: SDM, pp 365-376
41. Zheng J, Vilnis L, Singh S, Choi J, McCallum A (2013) Dynamic knowledge-base alignment for coreference resolution. In: CoNLL'13, pp 153-162
42. Ando RK, Zhang T (2005) A high-performance semi-supervised learning method for text chunking. In: Proceedings of the 43rd annual meeting on association for computational linguistics, pp 1-9
43. Bagga A, Baldwin B (1998) Algorithms for scoring coreference chains. Int Conf Lang Resour Eval Workshop Linguist Coreference 1:563–566
44. Black W, Rinaldi F, Mowatt D (1998) Facile: description of the ne system used for muc-7. In: Proceedings of Message Uunderstanding Conference (MUC)-7
45. Chen Y, Martin J (2007) Towards robust unsupervised personal name disambiguation. In: EMNLP-CoNLL, pp 190-198
46. Fleischman M, Hovy E (2004) Multi-document person name resolution. In: ACL, pp 66-82
47. Giles CB, Wren JD (2008) Large-scale directional relationship extraction and resolution. BMC Bioinform 9(S-9)
48. Gooi CH, Allan J (2004) Cross-document coreference on a large scale corpus. In: HLT-NAACL, pp 9-16
49. Hall PA, Dowling GR (1980) Approximate string matching. ACM Comput Surv 12(4):381–402
50. Holmes DO, McCabe MC (2002) Improving precision and recall for soundex retrieval. In: ITCC, pp 22-27
51. Kambhatla N (2004) Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: ACL'04, ACLdemo '04
52. Karaboga D, Ozturk C (2011) A novel clustering approach: artificial bee colony (abc) algorithm. Appl Soft Comput 11(1):652–657
53. Luo X, Ittycheriah A, Jing H, Kambhatla N, Roukos S (2004) A mention-synchronous coreference resolution algorithm based on the bell tree. In: ACL, pp 135-142
54. Vincent Ng (2010) Supervised noun phrase coreference research: the first fifteen years. In: ACL
55. Randell L (1993) An assessment of name matching algorithms. Technical reports 550, Department of Computer Science, University of Newcastle upon Tyne
56. Rao D, McNamee P, Dredze M (2010) Streaming cross document entity coreference resolution. In: COLING (Posters), pp 1050-1058
57. Ravichandran D, Pantel P, Hovy EH (2005) Randomized algorithms and nlp: using locality sensitive hash functions for high speed noun clustering. In: ACL
58. Sarawagi S, Kirpal A (2004) Efficient set joins on similarity predicates. In: SIGMOD Conference, pp 743-754
59. Tsuruoka Y et al (2005) Developing a robust part-of-speech tagger for biomedical text. In: Panhellenic Conference on Informatics, pp 382-392
60. Vilain M, Burger J, Aberdeen J, Connolly D, Hirschman L (1995) A model-theoretic coreference scoring scheme. In: MUC6'95, pp 45-52. USA
61. Wick M, Singh S, McCallum A (2012) A discriminative hierarchical model for fast coreference at large scale. In: ACL'12, pp 379-388
62. Anderberg MR (1973) Cluster analysis for applications. Academic Press, New York
63. Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives ZG (2007) Dbpedia: a nucleus for a web of open data. In: ISWC/ASWC, pp 722-735
64. Benjelloun O, Garcia-Molina H, Menestrina D, Qi S, Whang SE, Widom J (2009) Swoosh: a generic approach to entity resolution. VLDB J 18(1):255-276
65. Day D, Hitzeman J, Wick ML, Crouch K, Poesio M (2008) A corpus for cross-document co-reference. In: LREC
66. Elfeky MG, Elmagarmid AK, Verykios VS (2002) Tailor: a record linkage toolbox. In: Data Engineering. Proceedings 18th International Conference on. IEEE, pp 17-28
67. Finkel JR, Grenager T, Manning C (2005) Incorporating non-local information into information extraction systems by gibbs sampling. In: ACL'05, pp 363-370
68. Hachey B, Grover C, Tobin R (2012) Datasets for generic relation extraction. Nat Lang Eng 18(1):21–59
69. Lee H, Peirsman Y, Chang , Chambers N, Surdeanu M, Jurafsky D (2011) Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In: CONLL'11

70. Miller GA (1995) Wordnet: a lexical database for english. Commun ACM 38(11):39–41
71. Miller GA, Fellbaum C (2007) Wordnet then and now. Lang Resour Eval 41(2):209–214
72. Nastase V, Strube M, Boerschinger B, Zirn C, Elghafari A (2010) A very large scale multi-lingual concept network. In: LREC, Wikinet
73. Philips L (2000) The double-metaphone search algorithm. C/C++ User's J 18(6):38-43
74. Ponzetto SP, Strube M (2007) Deriving a large-scale taxonomy from wikipedia. In: AAAI, pp 1440-1445
75. Singh S et al (2012) Wikilinks: a large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015. University of Massachusetts, Amherst
76. Spitkovsky VI, Chang AX (2012) A cross-lingual dictionary for english wikipedia concepts. In: LREC, pp 3168-3175
77. Nadeau D, Sekine S (2007) A survey of named entity recognition and classification. Lingvisticae Investigationes 30(1):3–26
78. Sekine S, Ranchhod E (2009) Named entities: recognition, classification and use, vol 19. John Benjamins Publishing Company, The Netherlands
79. Skut W, Brants T (1998) Chunk tagger–statistical recognition of noun phrases. CoRR. arXiv:9807007 [cmp-lg]
80. Witten IH, Frank E (1999) Data mining: practical machine learning tools and techniques with Java Implementations. Morgan Kaufmann, USA
81. Weikum G, Hoffart J, Nakashole N, Spaniol M, Suchanek F, Yosef M (2012) Big data methods for computational linguistics. IEEE Data Eng Bull 35(3):46–64
82. Riddle WE (1984) The magic number eighteen plus or minus three: a study of software technology maturation. ACM SIGSOFT Softw Eng Note 9(2):21–37
83. Cruzes DS, Dyba T (2011) Recommended steps for thematic synthesis in software engineering. In: Empirical Software Engineering and Measurement (ESEM), pp 275-284. IEEE
84. Marrero M, Sanchez-Cuadrado S, Morato J, Andreadakis Y (2009) Evaluation of named entity extraction systems. Adv Comput Linguistics 41:47–58
85. Mousavi H, Kerr D, Iseli M, Zaniolo C (2014) Mining semantic structures from syntactic structures in free text documents. In: ICSC'14, pp 84-91. IEEE
86. Rahman A, Ng V (2011) Coreference resolution with world knowledge. In: ACL, pp 814-824
87. SMR Beheshti, Motahari Nezhad HR, Benatallah B (2012) Temporal provenance model (tpm): model and query language. CoRR. arXiv:1211.5009
88. Tasdemir K, Merényi E (2011) A validity index for prototype-based clustering of data sets with complex cluster structures. IEEE Trans 41(4):1039–1053
89. Estivill-Castro V, Houle ME (2001)Robust distance-based clustering with applications to spatial data mining. Algorithmica 30(2):216-242
90. Vincent Ng (2008) Unsupervised models for coreference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp 640-649
91. Olston C, Reed B, Srivastava U, Kumar R, Tomkins A (2008) Pig latin: a not-so-foreign language for data processing. In: SIGMOD'08. ACM, pp 1099-1110
92. Frakes WB, Baeza-Yates R (eds) (1992) Information retrieval: data structures and algorithms. Prentice-Hall Inc, Upper Saddle River
93. Nist Ac (2008) Extraction automatic content: Evaluation plan (ace08). In: Proceedings of the ACE, pp 1-3
94. McNamee P, Dang H (2009) Overview of the TAC 2009 knowledge base population track. In: Proc. Text Analysis Conference (TAC) Workshop
95. Salton G, McGill M (1984) Introduction to Modern Information Retrieval. McGraw-Hill Book Company, New York
96. US NIST (2003) The ace 2003 evaluation plan. US National Institute for Standards and Technology (NIST), pp 2003-2008
97. Ciaramita M, Altun Y (2006) Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In: EMNLP, pp 594-602
98. Van Zaanen M, Mollá D et al (2007) A named entity recogniser for question answering. Pacific Association for Computational Linguistics
99. Beheshti SMR et al (2013) Big data and cross-document coreference resolution: current state and future opportunities. CoRR. arXiv:1311.3987