Big Data and Cross-Document Coreference Resolution: Current State and Future Opportunities

Seyed-Mehdi-Reza Beheshti
Srikumar Venugopal
Seung Hwan Ryu
Boualem Benatallah
Wei Wang

University of New South Wales
Sydney 2052, Australia
{sbeheshti,srikumarv,seungr,boualem,weiw}@cse.unsw.edu.au

Technical Report
UNSW-CSE-TR-201333
November 2013

THE UNIVERSITY OF
NEW SOUTH WALES

SCIENTIA
MANU ET MENTE

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

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 ever-increasing amount of data depends critically upon Cross-Document Coreference Resolution (CDCR) - the task of identifying entity mentions across multiple documents that refer to the same underlying entity. Recently, document datasets of the order of peta-/tera-bytes has raised many challenges for performing effective CDCR such as scaling to large numbers of mentions and limited representational power. The problem of analysing such datasets is called "big data". The aim of this paper is to provide readers with an understanding of the central concepts, subtasks, and the current state-of-the-art in CDCR process. We provide assessment of existing tools/techniques for CDCR subtasks and highlight big data challenges in each of them to help readers identify important and outstanding issues for further investigation. Finally, we provide concluding remarks and discuss possible directions for future work.

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 normalised 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 [58]. In most cases, this activity concerns processing human language texts by means of natural language processing (NLP) [86].

Among various IE tasks, Cross-Document Coreference Resolution (CDCR) [57, 11] involves identifying equivalence classes for identifiable data elements, called entities, across multiple documents. In particular, CDCR is of critical importance for data quality and is fundamental for high-level information extraction and data integration, including semantic search, question answering, and knowledge base construction.

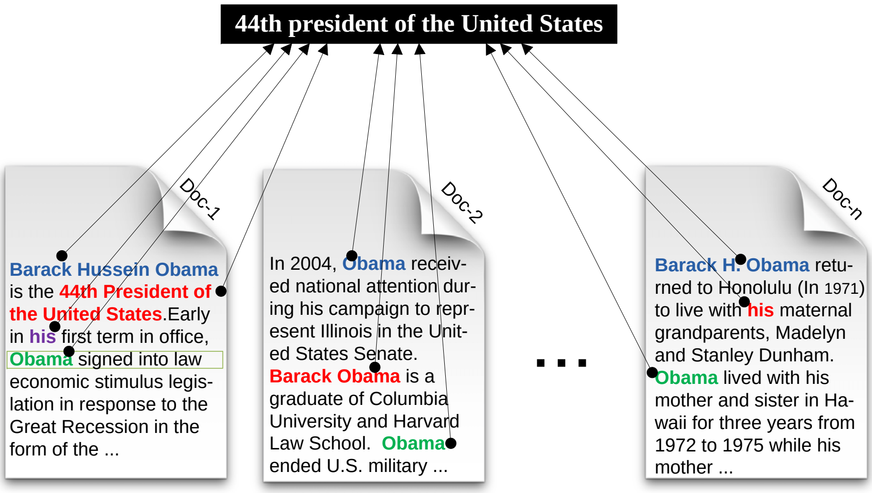
Traditional approaches to CDCR [57, 94] derive features from the context surrounding the appearance of an entity in a document (also called a "mention") and then apply clustering algorithms that can group similar or related entities across all documents. As we will soon discuss, these approaches aim for being exhaustive and grow exponentially in time with the increase in the number of documents.

Recently, a new stream of CDCR research [31, 69, 79, 81, 46] has focused on meeting the challenges of scaling CDCR techniques to deal with document collections sized of the order of terabytes and above. Popularly, dealing with such large-scale datasets has been termed as the "big data" problem. In this context, CDCR tasks may face various drawbacks including difficulties in clustering and grouping large numbers of entities and mentions across large datasets. Therefore, CDCR techniques need to be overhauled to meet such challenges.

To address these challenges, researchers have studied methods to scale CDCR subtasks such as computing similarity between pairs of entity mentions [65, 96], or even to replace pairwise approaches with more expressive and scalable alternatives [97, 94]. Recent publications [31, 69, 79, 81, 46] have reported on the usage of parallel and distributed architectures such as Apache Hadoop [95, 27] for supporting data-intensive applications which can be used to build scalable algorithms for pattern analysis and data mining.

Although these techniques represent the first steps to meeting big data challenges, CDCR tasks face various drawbacks in achieving a high quality coreference result (effectiveness) and performing the coreference resolution as fast as possible (efficiency) on large datasets. The aim of this paper is to provide readers with an understanding of the central concepts, subtasks, and the current state-of-the-art in CDCR process. We assess existing tools/techniques for CDCR subtasks and highlight big data challenges in each of them to help readers identify important and outstanding issues for further investigation. Finally, we provide concluding remarks and discuss possible directions for future work.

The remainder of this document is organized as follows. In Section 2, we introduce the CDCR process and its sub-tasks in detail. Sections 3 and 4 discuss the state-of-the-art in entity identification and entity classification. Section 5 discusses the challenges brought about by big data in CDCR. Section 6 present the state-of-the-art tools and techniques for CDCR. Section 7 presents our conclusions and a roadmap for the future. Finally, in the Appendix we discuss our experience in implementing a MapReduce-based CDCR software prototype to address challenges discussed in the paper.



Barack Hussein Obama is the 44th President of the United States.Early in his first term in office, Obama signed into law economic stimulus legislation in response to the Great Recession in the form of the ...

In 2004, Obama received national attention during his campaign to represent Illinois in the United States Senate. Barack Obama is a graduate of Columbia University and Harvard Law School. Obama ended U.S. military ...

Barack H. Obama returned to Honolulu (In 1971) to live with his maternal grandparents, Madelyn and Stanley Dunham. Obama lived with his mother and sister in Hawaii for three years from 1972 to 1975 while his mother ...

Figure 2.1: An *entity* (i.e. the person who serves as the 44th president of the United States) and its *entity mentions*, i.e. its true coreference resolutions.

2 CDCR Process and Evaluation Framework

2.1 Background and Preliminaries

CDCR approaches provide techniques for the identification of entity mentions in different documents that refer to the same underlying entity. In this context, 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”. Figure 2.1 illustrates a sample example of person name mentions from different documents and their coreference resolutions. Given a collection of mentions of entities extracted from millions 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 are in the same cluster and no other entities are included [81]. Mentions referring to the same entity are termed “co-referent”.

The current approach to cross-document (named) entity coreference resolution consists of two primary tasks [57, 97, 34, 73, 28]: entity identification and classification. Entity identification is the process of finding mentions of the entities of interest in documents and tie together those that are coreferent, while entity classification task involves deriving a classification and/or clustering technique that will separate data into categories, or classes, characterized by a distinct set of features. We discuss each in depth in the following subsections.

Entity Identification

Named-entity recognition [56, 48] (NER), also known as entity identification [62] and entity extraction [21, 2], refers to techniques that are 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, percentages, etc. There are numerous approaches and systems available for performing NER. For example, for traditional named entity recognition

(NER), the most popular publicly available systems are: OpenNLP NameFinder[1], Illinois NER system[2], Stanford NER system[3], and Lingpipe NER system[4].

Various steps are considered in this task including: (i) Format Analysis, in which document formats are analysed for formatting information in addition to textual content; (ii) Tokeniser, where text is segmented into tokens, e.g., words, numbers, and punctuation; (iii) Gazetteer, where the type and scope of the information is categorized; and (iv) Grammar, where linguistic grammar-based techniques as well as statistical models are used to extract more entities. The output of entity identification task will be a set of named entities, extracted from a set of documents. Numerous approaches, techniques, and tools to extracting entities from individual documents have been described in the literature and will be discussed in depth in the next section.

Entity Classification

Entity classification task involves deriving a classification and/or clustering technique that will separate data into categories, or classes, characterized by a distinct set of features. To achieve this, extracted entities and mentions (from the entity identification task) are assigned a metric based on the likeness of their meaning or semantic content.

Various machine learning techniques have modeled the problem of entity coreference as a collection of decisions between mention pairs [97]. Prior to entity pairing, various *features* may be extracted to annotate entities and their mentions. Figure 2.2 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. Then, the similarity scores for a pair of entities are computed using appropriate similarity functions for each type of feature (e.g., character-, document-, or metadata-level features).

The next step in entity classification task is to determine whether pairs of entities are co-referent or not. For example, in the sentence "Mary said she would help me", *she* and *Mary* most likely refer to the same person or group, in which case they are co-referent. Several filtering steps can be applied to entity pairs to eliminate those pairs that have little chance of being deemed co-referent. Various supervised and/or unsupervised classification/clustering techniques (over a set of training examples) can be used to classify related entities. For example, generative classifiers (e.g., Hidden Markov model), discriminative classifiers (e.g., Support Vector Machine (SVM) or maximum entropy model), or decision tree techniques can be used to separate a set of featurized paired entities into two possible classes - coreferent or not-coreferent.

3 Entity Identification: State-of-the-Art

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. NER is a key part of 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.

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

[2] http://cogcomp.cs.illinois.edu/demo/ner/?id=8

[3] http://www-nlp.stanford.edu/software/CRF-NER.shtml

[4] http://alias-i.com/lingpipe/demos/tutorial/ne/read-me.html

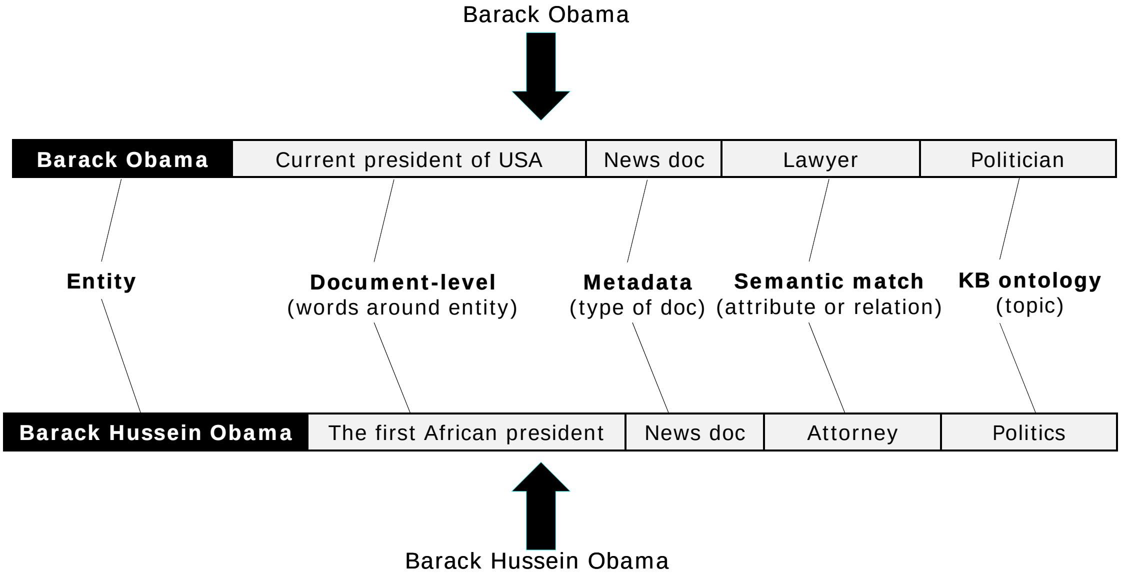


Figure 2.2: A simple example for calculating various featurization classes for the pair of entities (‘Barack Obama’ , ‘Barack Hussein Obama’).

A basic entity identification task can be defined as follows:

Let $\{t_1, t_2, t_3, ..., t_n\}$ *be a sequence of entity types denoted by* T *and let* $\{w_1, w_2, w_3, ..., w_n\}$ *be a sequence of words denoted by* W*, then the identification task can be defined as ‘given some* W*, find the best* T*’.*

In particular, entity identification consists of three subtasks: 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 unannotated 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[1]:

```
<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

[1] 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.

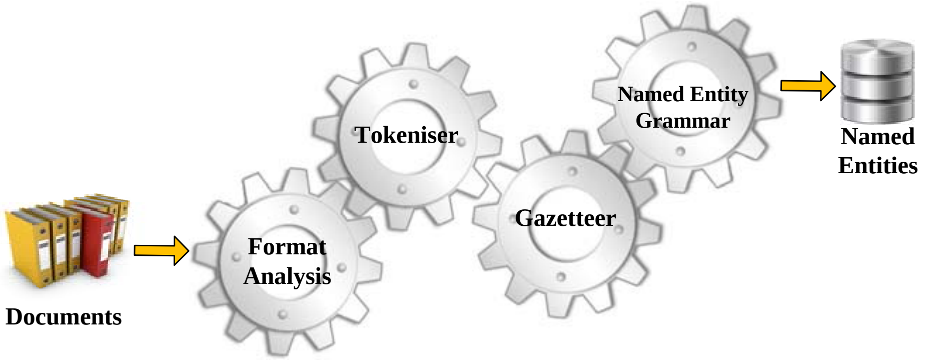


Figure 3.1: A simplified process for NER tasks.

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 DARPA (Defense Advanced Research Projects Agency) 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 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 challenges, the state of the art approaches to entity extraction proposed four primary steps [21, 56, 62, 15]: Format Analysis, Tokeniser, Gazetteer, Grammar. Figure 3.1 illustrates a simplified process for the NER task. Following is brief description of these steps:

Format Analysis. Many document formats contain formatting information in addition to textual content. For example, HTML documents contain HTML tags specifying formatting information such as new line starts, bold emphasis, and font size or style. The first step, format analysis, 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. 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. This module is responsible for categorizing the type and scope of the information presented. In particular, a gazetteer is a geographical dictionary or directory, an important reference for information about places and place names. It typically contains information concerning the geographical makeup of a country, region, or continent as well as the social statistics and physical features, such as mountains, waterways, or roads. As an output, this module will generate 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. Handcrafted 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 Entity Classification: State-of-the-Art

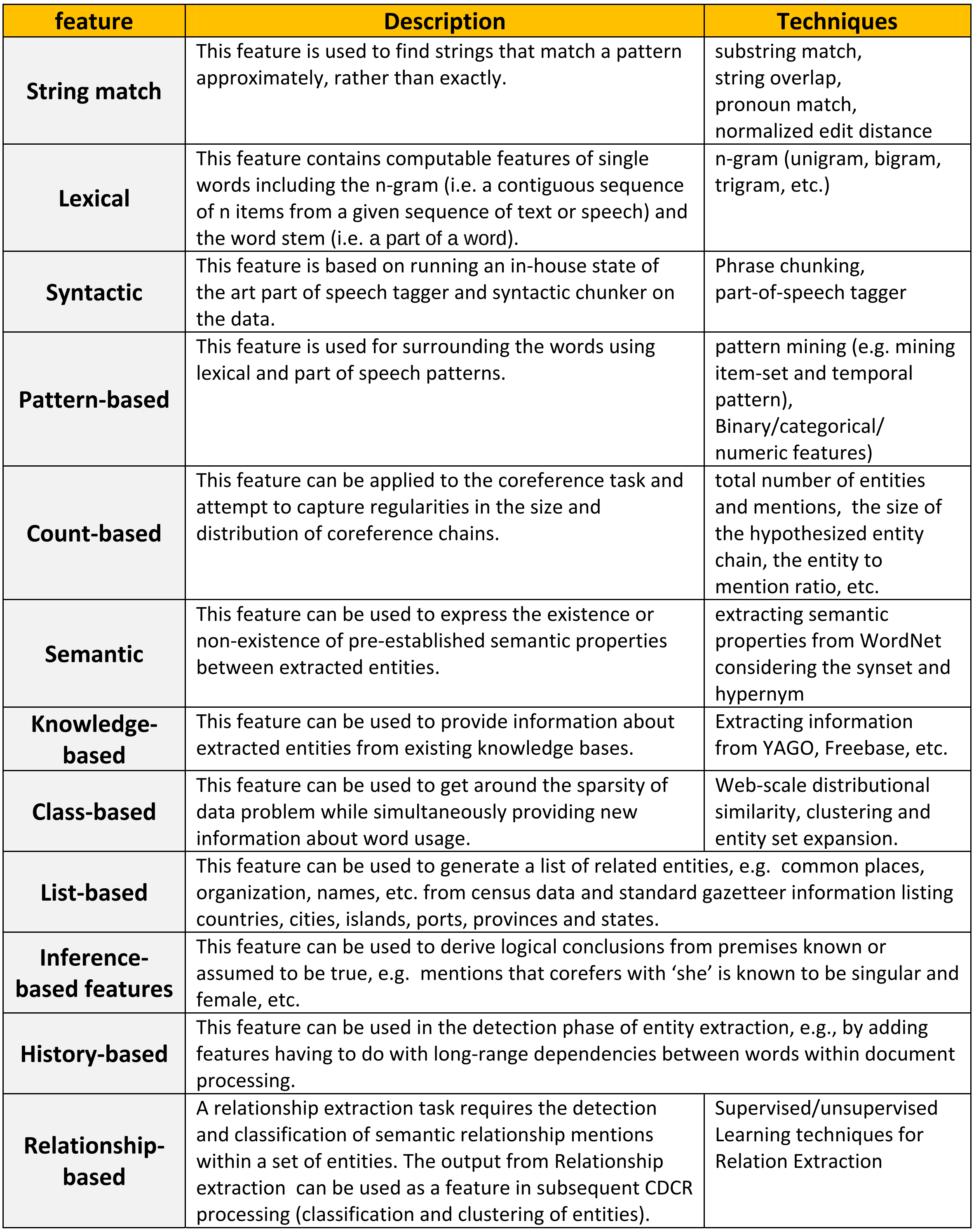
The classification step is responsible for determining whether pairs of entities are co-referent or not. To achieve this, extracted named entities in the entity identification phase should be compared by applying various features to pair of entities. Such features can be divided into various classes such as string match [51, 81, 20, 96], lexical [21, 9], syntactic [84, 90], pattern-based [25], count-based [25, 55, 72], semantic [45, 25], knowledge-based [25, 19, 63], class-based [75, 69, 97, 31, 79], list-/inference-/history-based [25], and relationship-based [38, 45] features. Table 4.1 illustrates the various classes of features, their description, and the state-of-the-art approaches.

Recently, linked data [16] has 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 DBpedia [8], freebase [18], WikiTaxonomy [71], and YAGO [88] have constructed huge knowledge bases (KBs) of entities, their semantic classes, and relationships among entities [93]. These systems can be used to enrich the entities with additional features and consequently to improve the effectiveness of the results. As an example, YAGO contains information harvested from Wikipedia, and linked to WordNet thesaurus [60] as a semantic backbone, and having more than two million entities (e.g., people, organizations, and cities) and 20 million facts about these entities.

4.1 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. In the classification step, similarity functions play a critical role in dealing with data differences caused by various reasons, such as misspellings, typographical errors, incomplete

Table 4.1: Various classes of features.



feature	Description	Techniques
String match	This feature is used to find strings that match a pattern approximately, rather than exactly.	substring match, string overlap, pronoun match, normalized edit distance
Lexical	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 or speech) and the word stem (i.e. a part of a word).	n-gram (unigram, bigram, trigram, etc.)
Syntactic	This feature is based on running an in-house state of the art part of speech tagger and syntactic chunker on the data.	Phrase chunking, part-of-speech tagger
Pattern-based	This feature is used for surrounding the words using lexical and part of speech patterns.	pattern mining (e.g. mining item-set and temporal pattern), Binary/categorical/ numeric features)
Count-based	This feature can be applied to the coreference task and attempt to capture regularities in the size and distribution of coreference chains.	total number of entities and mentions, the size of the hypothesized entity chain, the entity to mention ratio, etc.
Semantic	This feature can be used to express the existence or non-existence of pre-established semantic properties between extracted entities.	extracting semantic properties from WordNet considering the synset and hypernym
Knowledge-based	This feature can be used to provide information about extracted entities from existing knowledge bases.	Extracting information from YAGO, Freebase, etc.
Class-based	This feature can be used to get around the sparsity of data problem while simultaneously providing new information about word usage.	Web-scale distributional similarity, clustering and entity set expansion.
List-based	This feature 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	This feature 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	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 words within document processing.	
Relationship-based	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 (classification and clustering of entities).	Supervised/unsupervised Learning techniques for Relation Extraction

information, lack of standard formats, and so on. For example, personal name mentions may refer to a 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). For example, they are used for comparing strings (e.g., edit distance and its variations, Jaccard similarity, and tf/idf based cosine functions), for numeric values (e.g., Hamming distance and relative distance), for phonetic encoding (e.g., Soundex and NYSIIS), for images (e.g., Earth Mover Distance), and so on. 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 [43] 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. These functions (e.g., edit distance, Jaro, or Jaro-Winkler) consider characters and their positions within strings to estimate the similarity [92]. Following we describe set of character-based functions.

Edit distance: The edit distance between two strings is measured, based on the smallest number of edit operations (insertions, deletions, and substitutions) required to transform one string to the other. Each of the edit operations has a cost value (e.g., 1). For example, the edit distance between "window" and "widow" is 1 since deleting the letter "n" in "window" will convert the first string into the second. The edit distance function [64] is expensive or less accurate for measuring the similarity between long strings (e.g., document or message contents). It is likely to be suitable for comparing short strings (e.g., document titles) capturing typographical errors or abbreviations [30].

Jaro or Jaro-Winkler: The Jaro function computes the string similarity by considering the number of common characters and transposed characters. Common characters are ones that emerge in both strings within half the length of the shorter string [29]. Transposed characters are ones that are non-matching when comparing common characters of the same position in both strings. The Jaro-Winkler function improves the Jaro function by using the length of the longest common prefix between two strings. These functions are likely to work well for comparing short strings (e.g., personal names).

Q-grams: Given a string, q-grams are substrings in which each consists of q characters of the string [49]. For example, q-grams (q= 2) of "susan" are: 'su', 'us', 'sa', and 'an'. The similarity function computes the number of common q-grams in two strings and divides the number by either the minimum, average, or maximum number of q-grams of the strings [23]. If strings have multiple words and tokens of the strings are ordered differently in another strings, the similarity function might be more effective than the other character-based functions, such as edit distance or jaro function [23].

Token-based Functions. These 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 [47]. The following are some token-based functions:

Consonants	Number	Consonants	Number
b, f, p, v	1	l	4
c, g, j, k, q, s, x, z	2	m, n	5
d, t	3	r	6

Table 4.2: Soundex encoding table.

Jaccard: Jaccard 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.

TF/IDF: This function computes the closeness by converting two strings into unit vectors and measuring the angle between the vectors. In some situations, word frequency is important as in information retrieval applications that give more weight to rare words than on frequent words. In such cases, this function is likely to work better than the functions (e.g., Jaccard similarity) that are insensitive to the word frequency.

Phonetic Similarity Functions. These functions describe how two strings are phonetically similar to each other in order to compute the string similarity. Some examples are as follows:

Soundex [44], one of the best known phonetic functions, converts a string into a code according to an encoding table. A soundex code is comprised of one character and three numbers. The Soundex code is generated as follows: (i) Keep the first letter of a string and ignore all other occurrences of vowels (a, e, i, o, u) and h, w, y; (ii) Replace consonants with numbers according to Table 4.2; (iii) Code two consecutive letters as a single number; and (iv) Pad with 0 if there are less than three numbers. For example, using the soundex encoding table, both "daniel" and "damiel" return the same soundex code "d540".

Phonex/Phonix [50] is an alternative function to Soundex, which was designed to improve the encoding quality by preprocessing names based on their pronunciations. Phonix [37], an extension of Phonex, uses more than one hundred rules on groups of characters [23]. The rules are applied to only some parts of names, e.g., the beginning, middle or end of names.

Double Metaphone [70] 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.

Relative Distance: The relative distance is used for comparing numeric attributes x and y (e.g., price, weight, size): $R(x, y) = \frac{|x-y|}{max\{x,y\}}$.

Insurance Type	Car	Motorbike	Home	Travel
Car	1			
Motorbike	0.7	1		
Home	0	0	1	
Travel	0	0	0.3	1

Table 4.3: Similarity scores between two insurance types.

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 [29].

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. In the process during which date or time values are converted to a uniform representation, separator characters like ‘-’, ‘/’, ‘:’ are removed from the values. To determine the similarity between these converted values, we could use numeric similarity functions (e.g., absolute difference) by considering them as numeric values or character-based similarity functions (e.g., edit distance) by considering them as string values.

Similarity Functions for Categorical Data

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. Alternatively, in [61], the authors presented an approach that measures the similarity between two categorical values, based on user inputs. For example, Table 4.3 shows the user-defined similarity scores between any two insurance types. This method can give more detailed scores between categorical data, rather than giving only two scores ‘0’ or ‘1’, although some effort is needed to provide user-defined similarity scores in advance.

Even though there is no similarity between any two feature values, further comparisons can be made because of *semantic* relationships between them. For example, consider two feature strings “Richard” and “Dick” of person entities. Although normal string comparison functions may fail to see the similarity, the strings still can be considered as similar to each other, if we keep the information that the latter is an alias for the former.

4.2 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.

To achieve this goal, it is important to identify all references within the same document (i.e. within document coreference resolution). 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. Some techniques (e.g. [73, 41]) 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. The proposed method is designed to support both entity disambiguation and name variation operating in a streaming setting, in which documents are processed one at a time and only once.

The state-of-the-art in clustering has been discussed in previous publications [39, 80, 62]. Many of these approaches rely on mention (string) matching, syntactic features, and linguistic resources like English WordNet [87]. The classic works on clustering [10, 39] adapted the Vector Space Model (VSM[1]) or deployed different information retrieval techniques for entity disambiguation and clustering. Such works showed that clustering documents by their domain specific attributes such as domain genre will affect the effectiveness of cross-document coreferencing.

Some extensions to VSM for for cross-document coreference clustering have been proposed in [54, 22]. Furthermore, supervised approaches [17], Semi-supervised approaches [6], and and unsupervised approaches [32] have used clustering to group together different nominal referring to the same entity. In particular, approaches to cross document coreference resolution have first constructed a vector space representation derived from local (or global) contexts of entity mentions in documents and then performed some form of clustering on these vectors. Most of these approaches focused on disambiguating personal names.

Another line of related work, e.g. [35, 57] added a discriminative pairwise mention classifier to a VSM-like model. For example, Mayfield et al. [57], 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. [2] proposed a clique-based clustering method based upon a distributional approach which allows to extract, analyze and discover highly relevant information for corpus specific NEs annotation. Another line of related work [40, 3] proposed techniques for clustering text mentions across documents and languages simultaneously. Such techniques may produce cross-lingual entity clusters. Some later work [66, 7] relies on the use of extremely large corpora which allow very precise, but sparse features. For example, Ni et al. [66] enhanced the open-domain classification and clustering of named entity using linked data approaches.

Dynamic clustering approach [9] follows the method in which set of points are observed from a potentially infinite set X, one at a time, in order to maintain a fixed number of clusters while minimizing the maximum cluster radius (i.e. the radius of the smallest ball containing all points of the cluster). This approach 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.

[1] Vector space model or term vector model is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers

5 CDCR and Big Data

The ability to harness the ever increasing amounts of data will enable us to understand what is happening in the world. In this context, big data enables the two main characteristics to come together: (i) big data for massive amounts of detailed information; and (ii) advanced analytics including artificial intelligence, natural language processing, data mining, statistics and so on. Generating huge metadata for imbuing the data with additional semantics will form part of the big data challenges in CDCR. 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. More specifically, entities and their mentions may have massive amounts of detailed information which need to be analyzed over time.

Big data has raised various challenges in different tasks of information extraction, including in CDCR. In the entity identification phase, entity extraction subtasks such as format analysis, tokeniser, gazetteer, and grammar would have to be applied to huge number of documents. This is challenging as, in terms of scalability, entity extraction outputs more data than it takes. For example, as illustrated in Table 6.5, only 6600 documents provide more than two million entities. In contrast, the English Gigaword dataset contains more that nine million documents and will produce orders of magnitude more information.

Currently, the dominant methods for co-reference measure compatibility between pairs of mentions. These suffer from a number of drawbacks including difficulties scaling to large numbers of mentions and limited representational power [97]. For example, as illustrated in Table 6.5, for more than 30,000 extracted named entities, around 900 million pairs can be generated. In particular, in terms of scalability, pairwise entity comparison will become exponential across documents.

Recent research [97, 94, 65, 96] have studied methods that measure 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 and limited representational power. For example, Wick et al. [97] 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. [94] proposed an approach to integrated inference for entity extraction and coreference based on conditionally-trained undirected graphical models. Luo et al. [52] 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. [96] proposed a discriminatively-trained model that jointly performs coreference resolution and canonicalization, enabling features over hypothesized entities.

Finally, in the 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. To address these challenges, and to effectively classify and cluster these gigantic number of entities and pairs, parallel and distributed architectures have become popular. MapReduce [27] is a distributed computing framework introduced by Google with the goal of simplifying the process of distributed data analysis.

The MapReduce programming model consists of two functions called Map and Reduce. Data are distributed as key-value pairs on which the Map function computes a different set of intermediate key and value pairs. An intermediate Shuffle phase groups the values around common intermediate keys. The Reduce function then performs computation on the lists of values with the same key. The resulting set of key-value pairs from the reducers is the final output. Apache Hadoop [95] is the most popular, open source implementation of MapReduce that provides a distributed file system (i.e., HDFS[1]) and also, a high level language for data analysis, i.e., Pig[2]. Hadoop [95] can be used to build scalable algorithms for pattern analysis and data mining. This has been demonstrated by recent research [31, 69, 79, 81, 46] that have used MapReduce [27] for processing huge amounts of documents in a massively parallel way.

Elsayed et al. [31] 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 [36] in which each term is associated with a list of document identifiers for documents that contain it and the associated term weight; and (ii) Pairwise Similarity, where the MapReduce mapper generates key tuples corresponding to pairs of document IDs in the postings in which the key tuples will be associated with the product of the corresponding term weights.

A scalable MapReduce-based implementation based on distributional similarity have been proposed in [69], where the approach followed a generalized sparse-matrix multiplication algorithm [78]. The MapReduce plan uses the Map step to start $M * N$ Map tasks in parallel, each caching $1/M_{th}$ part of A as an inverted index and streaming $1/N_{th}$ part of B through it. In this approach, there is a need to process each part of A for N times, and each part of B is processed M times.

A multi-pass graph-based clustering approach to large scale named-entity disambiguation have been proposed in [79]. The proposed MapReduce-based algorithm is capable of dealing with an arbitrarily high number of entities types is able to handle unbalanced data distributions while producing correct clusters both from dominant and non-dominant entities. Algorithms will be applied to constructed clusters for assigning small clusters to big clusters, merging small clusters, and merging big and medium clusters. 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.

A distributed inference that uses parallelism to enable large scale processing have been proposed in [81]. 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. [46] 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. Moreover, it provides several load balancing strategies in combination with its blocking techniques to achieve balanced workloads across all employed nodes of the cluster.

6 CDCR Tools and Techniques Evaluation

6.1 Evaluation Dimensions

Cross-Document Coreference Resolution (CDCR) is the task of identifying entity mentions (e.g., persons, organizations or locations) across multiple documents that refer to the same underlying entity. An important problem in this task is how to evaluate a system's performance.

There are two requirements that should be lie at the heart of CDCR task: (i) effectiveness, which concerns with achieving a high quality coreference result. For the evaluation of accuracy,

[1] http://hadoop.apache.org/

[2] http://pig.apache.org/

well-known measures such as *precision* (the fraction of retrieved instances that are relevant) and *recall* (the fraction of relevant instances that are retrieved) [77] can be used; and (ii) efficiency, that concerns 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 [53]:

discriminativity: which is 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;

interpretability: which emphasis 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 correct for a system, we would expect that the vast majority of mentions are in right entities or coreference chains;

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) [77] can be used. As the complementary to precision/recall, some approaches such as link-based F-measure [91], 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 [67] count the number of false-alarm (the number of miss) 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) [53] 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 [10], 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[1] weighted vectors and 'cosine similarity'[2] 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.

[1] tf/idf, term frequency/inverse document frequency, is a numerical statistic which reflects how important a word is to a document in a collection or corpus.

[2] Cosine similarity is a measure of similarity which can be used to compare entities/documents in text mining. In addition, it is used to measure cohesion within clusters in the field of data mining.

6.2 Datastes

Measuring the effectiveness of CDCR task on large corpuses 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 few number of ground-truths.

Few publications [1, 26, 11] introduced manually-labeled, small datasets containing high ambiguity, which make it very 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[3] [82] which includes more than 40 million total disambiguated mentions over 3 million entities within around 10 million documents. Other examples of automatically labeled large datasets includes [68, 39, 83, 85]. Following we provide more details about TAC-KBP, John Smith, ACE, reACE, English Gigaword, and Google's Wikilinks datasets.

John Smith corpus [11]. 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 New York Times. The relatively of common name 'John Smith' used to find documents that were about different individuals in the news.

ACE (2008) corpus [33]. The most recent Automatic Content Extraction (ACE) evaluation took place in 2008, where the dataset includes approximately 10,000 documents from several genres (predominantly newswire). As the result of ACE participation (participants were expected to cluster person and organization entities across the entire collection), a selected set of about 400 documents were annotated and used to evaluate the system performance.

reACE [42]. The 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 to which the Edinburgh Regularized ACE (reACE) mark-up has been applied. In order to provide a sufficient level of ambiguity and reasonable ground-truth, the dataset includes annotation for: (1) a refactored version of the original data to a common XML document type; (2) linguistic information from LT-TTT (a system for tokenizing text and adding markup) and MINIPAR (an English parser); and (3) a normalized version of the original RE markup that complies with a shared notion of what constitutes a relation across domains. Similar to ACE and John Smith corpus, this dataset contains few annotated documents and cannot be used to evaluate the efficiency of big-data approaches in CDCR.

English Gigaword is a comprehensive archive of newswire text data that has been acquired over several years by the LDC 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 but can be used to assess the efficiency of the CDCR approaches.

Google's Wikilinks Corpus [82] This dataset comprises of 40 million mentions over 3 million entities gathered using an automatic method based on finding hyperlinks to Wikipedia from

[3] http://googleresearch.blogspot.com.au/2013/03/learning-from-big-data-40-million.html

a web crawl and using anchor text as mentions [82]. The Google search index has been used to discover the mentions that belong to the English language. The dataset provides the URLs of all the pages that contain labeled mentions, the actual mentions (i.e., the anchor text), the target Wikipedia link (entity label), and the byte offsets of the links on the page. Similar to Wikilinks, the *TAC-KBP corpus* [59] links entity mentions to corresponding Wikipedia derived knowledge base nodes, focusing on ambiguous person, organization, and geo-political entities mentioned in newswire, and required systems to cope with name variation and name disambiguation. The dataset contains over 1.2 million documents, primarily newswire.

6.3 Tools for Entity Identification and their evaluation

In this section, we assess a set of named entity extraction systems including: OpenNLP, Stanford-NLP, LingPipe, Supersense tagger, AFNER, and AlchemyAPI. Table 6.1 illustrates a set of Information Extraction tools and their applications. Table 6.2 depicted CDCR tasks and the tools that can be leveraged in each phase. 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 as well as process texts in a common language: English has been the selected language for this assessment. Following is a brief description of the selected tools.

Stanford-NLP [4], is an integrated suite of natural language processing 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. The model is dependent on the language and entity type it was trained for and offers a number of pre-trained name finder models that are trained on various freely available corpora.

OpenNLP [5], 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. The OpenNLP Name Finder can detect named entities and numbers in text. The Name Finder needs a model to be able to detect entities. The model is dependent on the language and entity type it was trained for. The OpenNLP projects offers a number of pre-trained name finder models that are trained on various freely available corpora. The OpenNLP engine reads the text content and leverages the sentence detector and name finder tools bundled with statistical models trained to detect occurrences of named entities.

The OpenNLP tools are statistical NLP tools including a sentence boundary detector, a tokenizer, a POS tagger, a phrase chunker, a sentence parser, a name finder and a coreference resolver. The tools are based on maximum entropy models. The OpenNLP tools can be used as standalone (in which the output will be a single text format) or as plugins with other Java frameworks including UIMA (in which the output will be in XML metadata Interchange (XMI) format). It is possible to pipe output from one OpenNLP tool into the next, e.g., from the sentence detector into the tokenizer.

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

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

Table 6.1: List of existing Information Extraction tools and their applications.

Tool	Application
OpenNLP (http://opennlp.apache.org/)	Supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, and parsing.
UIMA (http://uima.apache.org/)	Entity detection, language identification, language specific segmentation, sentence boundary detection.
Mahout (http://mahout.apache.org/)	scalable machine learning libraries including: Collaborative Filtering, User and Item based recommenders, K-Means, Fuzzy K-Means clustering, Mean Shift clustering, Dirichlet process clustering, Latent Dirichlet Allocation, Singular value decomposition, Parallel Frequent Pattern mining, Complementary Naive Bayes classifier etc.
Behemoth (https://github.com/DigitalPebble/behemoth)	Behemoth does not implement any NLP or Machine Learning components as such but it simplifies the deployment of and serves as a 'large-scale glueware' for existing resources such as Apache UIMA and OpenNLP.
Tika (http://tika.apache.org/)	Detects and extracts metadata and structured text content from various documents using existing parser libraries.
Solr (http://lucene.apache.org/solr/)	powerful full-text search, hit highlighting, faceted search, dynamic clustering, database integration, rich document (e.g., Word, PDF) handling, and geospatial search. It can be used for Entity Pairs Filtering.
uimaFIT	Simplifies the use of Apache UIMA. (http://code.google.com/p/uimafit/)
u-compare (http://u-compare.org/)	U-Compare is an integrated text mining/natural language processing system based on the UIMA Framework.
Duke (http://code.google.com/p/duke/)	Used for deduplication (or entity resolution, or record linkage) which is the process of identifying and merging records judged to represent the same real-world entity.
Lingpipe (http://alias-i.com/lingpipe)	Named Entity Recognition
Stanford NLP (http://nlp.stanford.edu/software/index.shtml)	Natural Language Processing including, Named Entity Recognizer, Word Segmenter, Classifier, EnglishTokenizer, Temporal Tagger (SUTime), and more.
Stanford NER	Named Entity Recognition (http://www-nlp.stanford.edu/ software/ CRF-NER.shtml)
Opencalais (http://www.opencalais.com/)	Extracts semantic information from web pages in a format that can be used on the semantic web.
Dedoop	MapReduce-based entity resolution of large datasets. (http://dl.acm.org/citation.cfm?id=2367527)
Illinois NER	Named Entity Recognition (http://cogcomp.cs.illinois.edu/demo/ner/?id=8)
BBN IdentiFinder Text Suite	Named Entity Recognition (http://bbn.com/technology/speech/identifinder)
Wikipedia Miner (http://www.nzdl.org/wikification/)	Extract entity from Wikipedia.
Entity Matching for Big Data (EMBD)	Efficient application of machine learning models for entity / ontology matching. (http://dbs.uni-leipzig.de/en/research/projects/large_scale_object_matching)
AlchemyAPI	Named Entity Recognition (http://www.alchemyapi.com/api/entity/)
ClearForest SWS (http://sws.clearforest.com/)	It allows the analysis of English texts and the identification of ENAMEX types, in addition to some other types such as products, currencies, etc.
Annie (http://gate.ac.uk/sale/tao/tao.pdf)	An entity extraction module incorporated in the GATE framework. It is open-source and under a GNU license, developed at the University of Sheffield
Freeling	An open source tool with GNU license that may be used as an API or independently for Named Entity Recognition. (http://nlp.lsi.upc.edu/web/index.php?option=com_content&task=view&id=54)
Afner	An open-source NERC tool, under GNU license for Named Entity Recognition. (http://aclweb.org/anthology-new/U/U06/U06-1009.pdf)
Supersense Tagger	An open-source Named Entity Ttagger with a version 2.0 Apache license. (http://www.loa.istc.cnr.it/Papers/mcya_emnlp06.pdf)
TextPro tools suite	offers various NLP functionalities interconnected in a pipeline order. It is under a (GNU license) (http://textpro.fbk.eu/download-textpro.html)
YooName (http://cogprints.org/5025/)	Named Entity Recognition which using a predefined classification of nine types of NEs (person, organization, location, miscellanea, facility, product, event, natural element and unit) and 100 subtypes.
SecondString	Similarity Computation, it supports following algorithms such as ED, q-grams, jaccard, cosine, and TFIDF. (http://secondstring.sourceforge.net/)
SimMetrics	Similarity Computation, it supports following algorithms such as ED, q-grams, jaccard, cosine, and TFIDF. (http://sourceforge.net/projects/simmetrics/)
Weka	Collection of machine learning algorithms such as SVM and decision tree. (http://www.cs.waikato.ac.nz/ml/weka/)
FEBRL	Collection of machine learning algorithms such as SVM and decision tree. (http://datamining.anu.edu.au/software/febrl/febrldoc/)

Table 6.2: CDCR tasks and the tools which can be leveraged in each phase.

	CDCR Phases				
	Intra-Document Processing	Entity Pairs Filtering	Featurization	Classification	Clustering
Tools	OpenNLP	Solr	Mahout	SecondString	Stanford NLP (Classifier)
	UIMA (uimaFIT/Behemoth/ u-compare)	Dedoop	Duke	SimMetrics	OpenNLP
	Tika	BBN IdentiFinder	Opencalais	Weka	Mahout
	Lingpipe		Dedoop	FEBRL	BBN IdentiFinder
	Stanford NLP/NER		Wikipedia Miner	Stanford NLP (Classifier)	
	Opencalais		EMBD	OpenNLP	
	Dedoop			Mahout	
	Illinois NER				
	BBN IdentiFinder				
	Wikipedia Miner				
	EMBD				
	AlchemyAPI				
	ClearForest SWS				
	Annie				
	Freeling				
	Afner				
	Supersense Tagger				
	TextPro tools suite				
	YooName				

The OpenNLP sentence detector is based on the approach proposed in [76]. One obvious drawback in the classification approach is that it cannot identify sentence boundaries where there is no marker. Next step is the statistical tagging. The statistical approach to tagging is to treat it as a multi-way classification problem. The OpenNLP POS (Part of speech) tagger is based on the approach proposed in [74]. After the OpenNLP tagger was developed, Toutanova and Manning [89] proposed approaches for improving the accuracy of maximum entropy taggers. The Stanford-NLP POS (Part of speech) is based on this latter work.

Chunking (also known as partial parsing) creates very shallow trees representing simple, flat phrase structure (mainly noun phrases). The basic approach in chunking is to exploit the work already done by the POS tagger in order to identify simple phrases by recognizing sequences of POS tags. The OpenNLP tools include a chunker, which uses a maximum entropy model to recognize patterns in the POS tags made by the OpenNLP tagger. Stanford NLP does not provide chunker. The Stanford parser is actually a set of alternative parsing algorithm and statistical models. It was developed in order to compare and evaluate different techniques.

LingPipe [6], is a toolkit for processing text using computational linguistics. LingPipe 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. Moreover, it includes online training (learn-a-little, tag-a-little) and character encoding-sensitive I/O. It offers a user interface and various demos through which it is possible to test texts. We used the latest release of LingPipe, LingPipe 4.1.0, in the assessment.

Supersense Tagger [7], is designed for the semantic tagging of nouns and verbs based on WordNet categories which includes set of named entities such as persons, organizations, locations, temporal expressions and quantities. It is based on automatic learning, offering three different models for application: CONLL, WSJ and WNSS. The Supersense-CONLL have been used in our evaluation.

AFNER [8], is a package for named entity recognition. AFNER uses regular expressions to find simple case named entities such as simple dates, times, speeds, etc. Moreover, it supports finding the parts of text matching listed named entities. The regular expression and list matches are then used in a '*maximum entropy*'[9] based classifier. Features relating to individual tokens (including list and regular expression matches) as well as contextual features are used. It also allows the addition of lists and regular expressions, as well as the training of new models. It is by default capable of recognizing persons' names, organizations, locations, miscellanies, monetary quantities, and dates in English texts.

AlchemyAPI [10], utilizes natural language processing technology and machine learning algorithms to analyze content, extracting semantic meta-data: information about people, places, companies, topics, facts and relationships, authors, languages, and more. API endpoints are provided for performing content analysis on Internet-accessible web pages, posted HTML or text

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

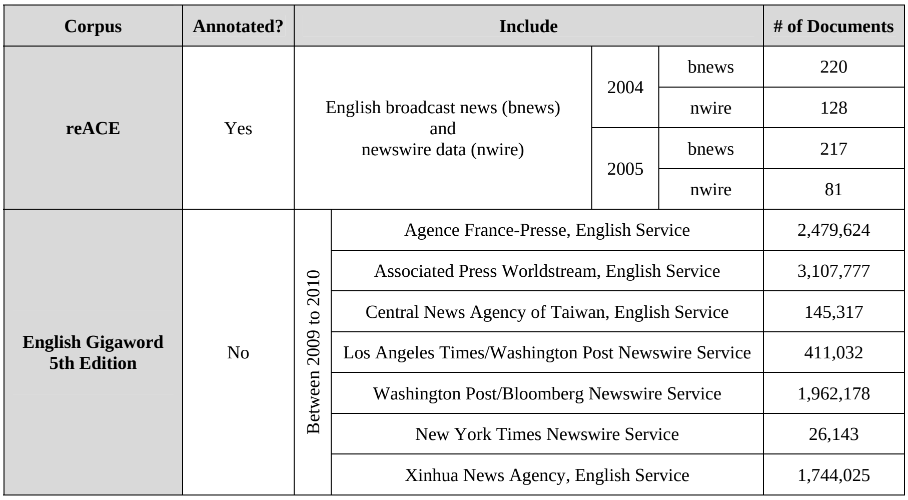
[7] https://sites.google.com/site/massiciara/

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

[9] Maximum entropy is a probability distribution estimation technique widely used for a variety of natural language tasks, such as language modeling, part-of-speech tagging, and text segmentation. Maximum entropy can be used for text classification by estimating the conditional distribution of the class variable given the document.

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

Table 6.3: Main characteristics of the datasets.



Corpus	Annotated?	Include			# of Documents
reACE	Yes	English broadcast news (bnews) and newswire data (nwire)	2004	bnews	220
				nwire	128
			2005	bnews	217
				nwire	81
English Gigaword 5th Edition	No	Between 2009 to 2010	Agence France-Presse, English Service		2,479,624
			Associated Press Worldstream, English Service		3,107,777
			Central News Agency of Taiwan, English Service		145,317
			Los Angeles Times/Washington Post Newswire Service		411,032
			Washington Post/Bloomberg Newswire Service		1,962,178
			New York Times Newswire Service		26,143
			Xinhua News Agency, English Service		1,744,025

content. It supports multiple languages and offers comprehensive disambiguation capabilities solutions. Moreover, it can be used to identify positive, negative and neutral sentiment within HTML pages and text documents/contents as well as for extracting document-level sentiment, user-targeted sentiment, entity-level sentiment, and keyword-level sentiment.

Analysis and Methodology

In this assessment, we use reAce (to evaluate the effectiveness of the results) and English Gigaword (to evaluate the efficiency of the results) datasets. These datasets have been discussed in Section 6.2. Table 6.3 illustrates the main characteristics of these datasets. The data analysis has been realized having focused on comparison of results obtained by the tools, for entities found in the test corpus: set of documents in the English Gigaword corpus has been used in order to evaluate the behavior of the tools. In particular, we used part of the Agence France-Presse, English Service (`afp_eng`), as part of English Gigaword corpus, that has a total of 492 words, distributed in 5 documents and 21 paragraphs in which more than 60 occurrences of various types of entities have been accumulated. The assessment only consider the names of persons, locations and organizations. These entity types were distributed in various phrases in the corpus with different typography, where entities in a tool could neither totally coincide in number nor in semantic with their equivalent entities in other tools. Consequently, we adopted the corpus for every tool.

The data analysis has been realized having focused on the comparison of results obtained by the tools, for entities found in the test corpus. For the evaluation of accuracy, we use the well-known measures of precision and recall [77]. 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:

$$\text{Precision} = \frac{number-of-currect-entities-found}{total-number-of-entities-extracted}$$

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.

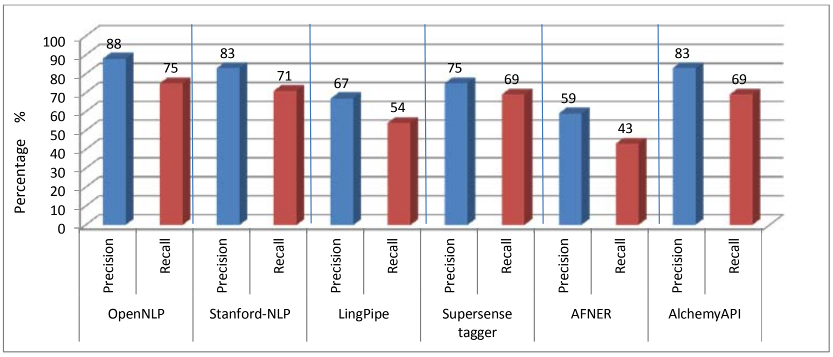


Figure 6.1: Precision-Recall in entity identification.

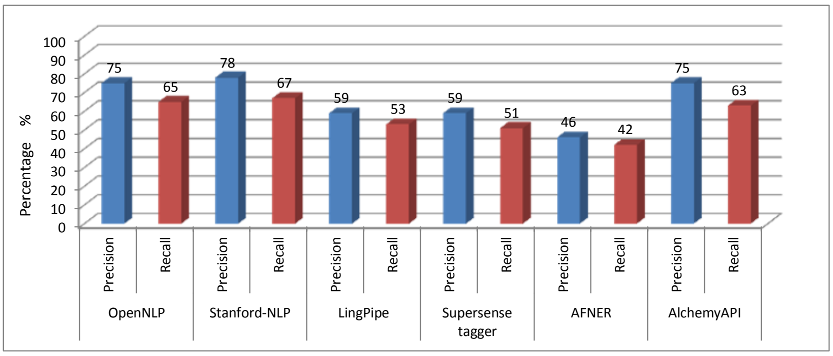


Figure 6.2: Precision-Recall in entity classification.

$$\text{Recall} = \frac{number-of-currect-entities-found}{total-number-of-correct-entities-that-should-be-ound}$$

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 [77]. The evaluation has been realized through distinct measures of precision and recall based on: (i) identification of the entities and false-positives[11] in the identification; (ii) classification of entities; and (iii) classification by NE types that each tool recognizes.

Figure 6.2 illustrates the precision-recall for entity classification. Notice that, entity classification is the process of classifying entities based on their type (i.e., recognized by the tools) and is different from coreference classification (see Section 4). Given that classification is a process that depends on the identification of entities, the f-measure in identification is always superior to that of the classification. In particular, F-measure is the harmonic mean of precision and recall:

$$\text{F-measure}=2.\frac{precision.recall}{precision+recall}$$

Figure 6.3 illustrates the F-measure in entity identification and classification. Comparing to the precision-recall for entity identification and classification, it is generally observed that the values are similar for the F-measure in entity identification and classification.

[11] In statistics, a false positive is the incorrect rejection of a true null hypothesis, which may lead one to conclude that a thing exists when really it doesn't. For example, that a named entity is of a specific type, e.g. Person, when the entity is not of that type.

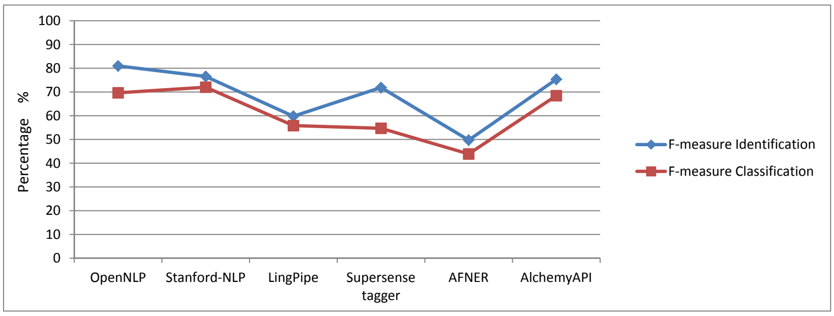


Figure 6.3: F-measure in entity identification and classification.

We took a further analysis in account for 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 achieve this, in Table 6.4 we calculate the number of persons, locations and organizations and the f-measure for them. In this experiment, we didn't 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 types.

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 6.4 allows us to observe some differences to the global analysis. 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 Tools for Entity Classification and their evaluation

The classification step compares pairs of entities, in which each entity is augmented with several features extracted from documents in the featurization step, and then determines whether these pairs of entities are coreferent or not. This step consists of two consecutive tasks (in Figure 6.4): *similarity computation* and *coreference decision*. 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 classification: 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 combined score is above a given threshold are considered as "coreferent"; and (ii) Machine learning-based classification: 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 use the following open-source packages:

- **SecondString and SimMetrics:** *SecondString*[12] and *SimMetrics*[13] are open-source

[12] http://secondstring.sourceforge.net
[13] http://sourceforge.net/projects/simmetrics/

Table 6.4: Results by entity type.

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

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.

- **Weka:** *Weka* [98] 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 and Methodology

In this assessment, we use reAce (to evaluate the effectiveness of the results) and English Gigaword (to evaluate the efficiency of the results) datasets. These datasets have been discussed in Section 2.1. Figure 6.5 shows the characteristics of those two datasets, which indicate for each dataset the types of extracted entities, the number of involved entities, the number of available feature attributes, the number of entity pairs, and so on. Figure 6.6 shows some example person entities (including metadata such as document identifier, type, and title) from the two datasets.

We measured the overall performance with both of *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

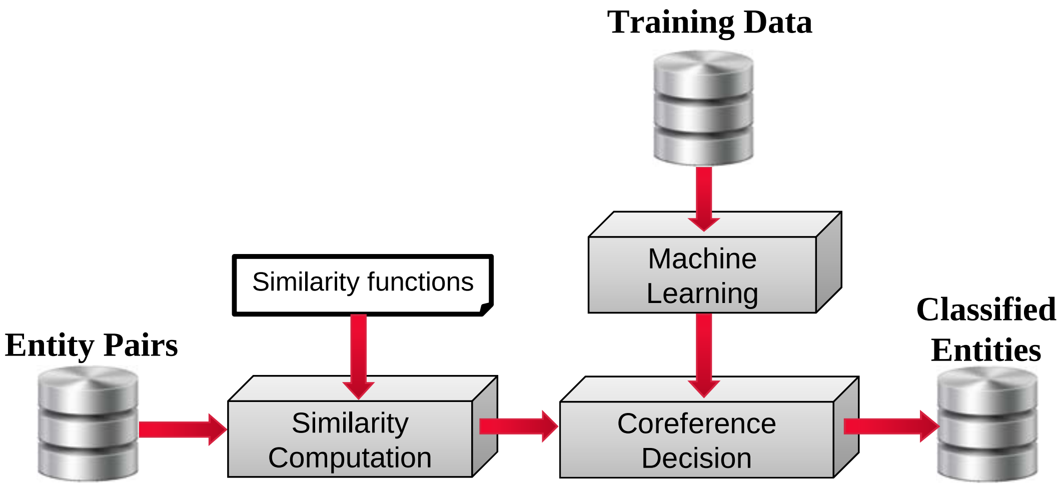
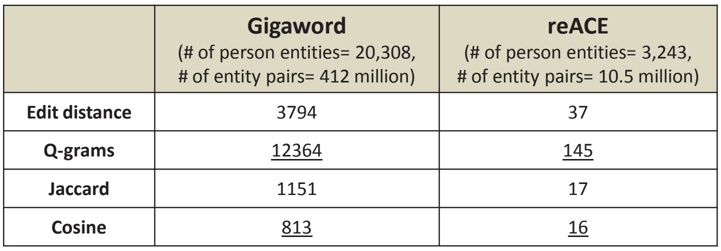


Figure 6.4: Coreference classification process.



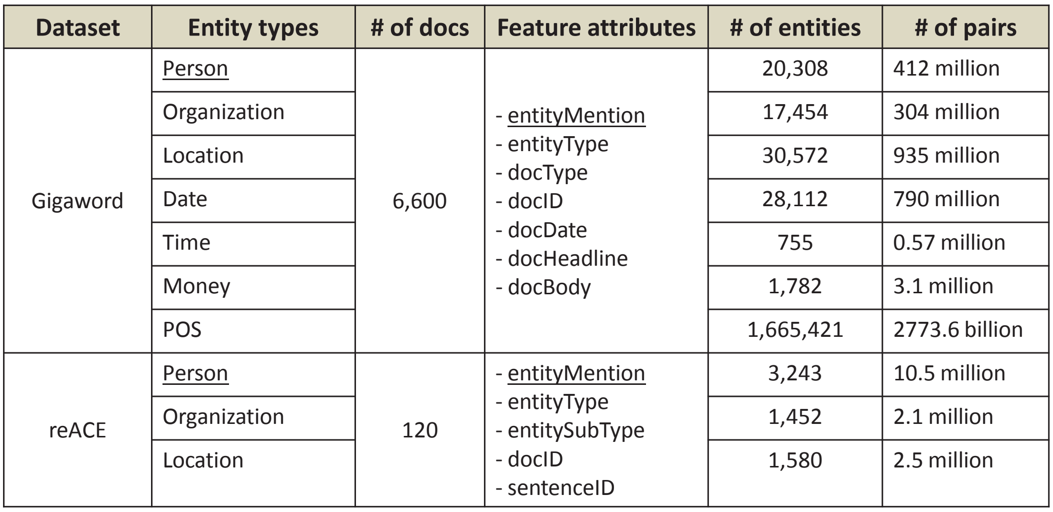
	Gigaword (# of person entities= 20,308, # of entity pairs= 412 million)	reACE (# of person entities= 3,243, # of entity pairs= 10.5 million)
Edit distance	3794	37
Q-grams	12364	145
Jaccard	1151	17
Cosine	813	16

Table 6.5: Execution times (in seconds) for the two datasets. The smallest and largest values are underlined.

negatives (missing results).

- Precision= $\frac{TP}{TP+FP}$;
- Recall= $\frac{TP}{TP+FN}$;
- F-measure= $\frac{2*Precision*Recall}{Precision+Recall}$;

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 compared feature attributes using various similarity functions. Figure 6.5 indicates that several feature attributes could be used for the coreference decision (e.g., entity mention, docType, docDate, docHeadline, and docBody for the "Gigaword" dataset). In addition, we used the following four string similarity functions: `edit distance`, `Q-grams`, `jaccard`, and `cosine` functions. Here, edit distance and Q-grams are character-based functions while jaccard and cosine functions are token-based 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 one feature attribute (i.e., 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



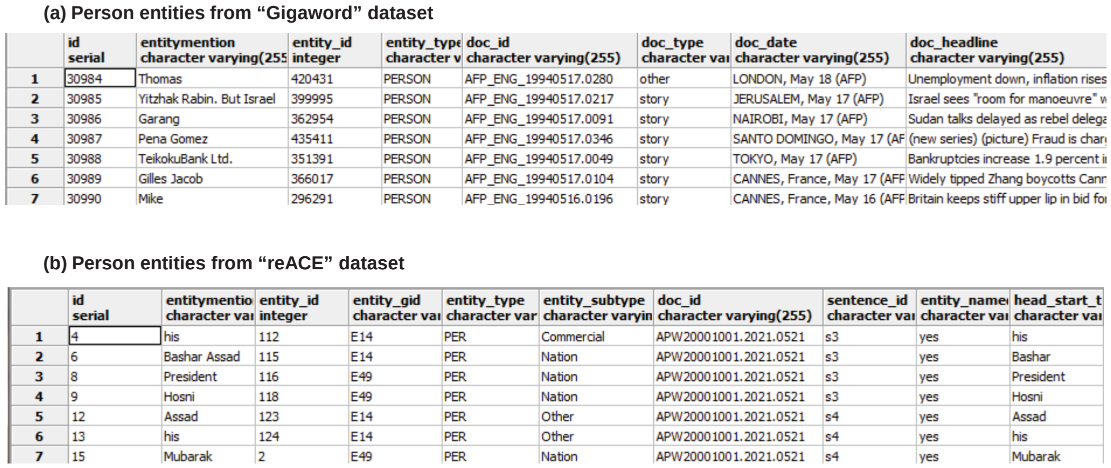
Dataset	Entity types	# of docs	Feature attributes	# of entities	# of pairs
Gigaword	Person	6,600	- entityMention - entityType - docType - docID - docDate - docHeadline - docBody	20,308	412 million
	Organization			17,454	304 million
	Location			30,572	935 million
	Date			28,112	790 million
	Time			755	0.57 million
	Money			1,782	3.1 million
	POS			1,665,421	2773.6 billion
reACE	Person	120	- entityMention - entityType - entitySubType - docID - sentenceID	3,243	10.5 million
	Organization			1,452	2.1 million
	Location			1,580	2.5 million

Figure 6.5: Characteristics of datasets. The entity type and the feature attribute, which are considered in the evaluation, are underlined.

the entity mention feature.

Table 6.5 lists the execution times taken for making coreference decisions by comparing `person` entities of the two datasets. The table shows significant differences between the applied similarity functions. The token-based functions (`Jaccard` and `cosine`) achieved fast execution time, when compared to the character-based functions (`edit distance` and `Q-grams`). 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 one. When comparing 20,308 entities (the number of entity pairs is 412 millions) from "Gigaword" dataset, an execution time of 12,364 seconds is needed with the `Q-grams` function while an execution time of 813 seconds is needed with the `cosine` function.

Figure 6.7 shows the coreference quality (precision, recall, and F-measure) results for the "reACE" dataset with different similarity functions. The top half shows the results obtained by applying the character-based functions on just one single feature attribute of "reACE" dataset, namely `person name` entity mention. Among the character-based functions, the `Q-grams` function (average precision: 0.87) worked better than the `edit distance` function (average precision: 0.80). The bottom half shows 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 functions, the coreference decision using the `Q-grams` function performed best while the one using the `edit distance` performed worst. The reason is that, if person names have multiple tokens and the tokens of the names are ordered differently in the other names, the `Q-grams` function could be more effective than the other character-based function, such as `edit distance`. All the functions performed reasonably well in terms of precisions, but they all suffered from very low recall, which means they missed many true coreferent entity pairs that should be contained in the returned results.



(a) Person entities from “Gigaword” dataset

	id serial	entitymention character varying(255	entity_id integer	entity_type character v	doc_id character varying(255)	doc_type character vai	doc_date character varying(255)	doc_headline 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 char
5	30988	TeikokuBank Ltd.	351391	PERSON	AFP_ENG_19940517.0049	story	TOKYO, May 17 (AFP)	Bankruptcies increase 1.9 percent i
6	30989	Gilles Jacob	366017	PERSON	AFP_ENG_19940517.0104	story	CANNES, France, May 17 (AFF	Widely tipped Zhang boycotts Cann
7	30990	Mike	296291	PERSON	AFP_ENG_19940516.0196	story	CANNES, France, May 16 (AFF	Britain keeps stiff upper lip in bid fo

(b) Person entities from “reACE” dataset

	id serial	entitymentio character vai	entity_id integer	entity_gid character vai	entity_type character var	entity_subtype character varyin	doc_id character varying(255)	sentence_id character vai	entity_name character vai	head_start_t character vai
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

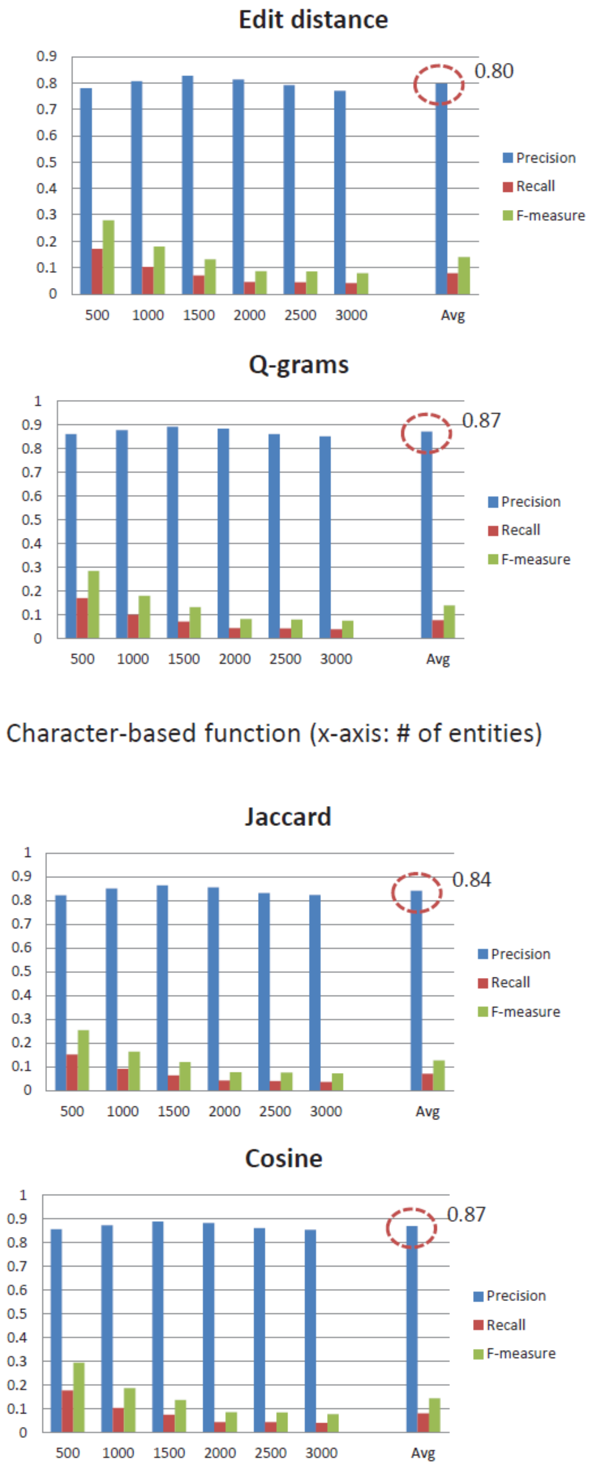
Figure 6.6: Example person entities from two datasets.

7 Conclusions and Future Work

In this paper we discussed the central concepts, subtasks, and the current state-of-the-art in Cross-Document Coreference Resolution (CDCR) process. We provided assessment of existing tools/techniques for CDCR subtasks and highlight big-data challenges in each of them to help readers identify important and outstanding issues for further investigation. Finally, we provide concluding remarks and discuss possible directions for future work. 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 significant research directions in this area.

Entity Extraction and the Big Data. The entity extraction task outputs more data than it takes. Millions of documents can be used as an input to this task, and billions 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. For example, as depicted in table 6.5 in Section 4, the evaluation results on the effectiveness shows that the recall can be very poor, compared with the precision. There is a strong need for improving the recall results by exploiting more useful features and applying appropriate similarity functions to those features. In this context, various load balancing techniques can be used to optimize the performance of MapReduce in extracting entities from huge number of documents. Moreover, 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 Billions Extracted Entities. For 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 eliminates those pairs that have little chance of being deemed co-referent. Similar to entity extraction phase, generating a knowledge-base from existing Linked Data sys-



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

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

Figure 6.7: Evaluation results with the four different similarity functions (threshold= 0.5).

tems may facilitate the featurization step.

Classification of Billions Entity Pairs. 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 has different similarity threshold, 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) which feature attributes to be used for classification; (ii) which similarity functions to be used for the chosen feature attributes; and (iii) which threshold is suitable for the classification decision.

Clustering of Billions 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. Challenges here include: (i) assigning each cluster to a global entity. 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'. To achieve, Linked Data systems can be used to help identifying the entities; and (ii) when co-referent text mentions appear in different languages, standard entity clustering techniques cannot be easily applied.

Bibliography

[1] Luisa Abentivogli, Christian Girardi, and Emanuele Pianta. Creating a gold standard for person crossdocument coreference resolution in italian news. In *The Workshop Programme*, page 19, 2008.

[2] J. Ah-Pine and G. Jacquet. Clique-based clustering for improving named entity recognition systems. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 51–59. Association for Computational Linguistics, 2009.

[3] E. Aktolga, M.A. Cartright, and J. Allan. Cross-document cross-lingual coreference retrieval. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1359–1360. ACM, 2008.

[4] Mohammad Allahbakhsh, Aleksandar Ignjatovic, Boualem Benatallah, Seyed-Mehdi-Reza Beheshti, Elisa Bertino, and Norman Foo. Reputation management in crowdsourcing systems. In *CollaborateCom*, pages 664–671, 2012.

[5] Mohammad Allahbakhsh, Aleksandar Ignjatovic, Boualem Benatallah, Seyed-Mehdi-Reza Beheshti, Elisa Bertino, and Norman Foo. Collusion detection in online rating systems. In *APWeb*, pages 196–207, 2013.

[6] R.K. Ando and T. Zhang. A high-performance semi-supervised learning method for text chunking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 1–9. Association for Computational Linguistics, 2005.

[7] G. Attardi, S.D. Rossi, and M. Simi. Tanl-1: coreference resolution by parse analysis and similarity clustering. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 108–111. Association for Computational Linguistics, 2010.

[8] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In *ISWC/ASWC*, pages 722–735, 2007.

[9] E. Baengtson and D. Roth. Understanding the value of features for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Association for Computational Linguistics, 2008.

[10] A. Bagga and B. Baldwin. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–6, 1998.

[11] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 79–85, 1998.

[12] Seyed-Mehdi-Reza Beheshti, Boualem Benatallah, and Hamid R. Motahari Nezhad. Enabling the analysis of cross-cutting aspects in ad-hoc processes. In *CAiSE*, pages 51–67, 2013.

[13] Seyed-Mehdi-Reza Beheshti, Boualem Benatallah, Hamid R. Motahari Nezhad, and Mohammad Allahbakhsh. A framework and a language for on-line analytical processing on graphs. In *WISE*, pages 213–227, 2012.

[14] Seyed-Mehdi-Reza Beheshti, Boualem Benatallah, Hamid R. Motahari Nezhad, and Sherif Sakr. A query language for analyzing business processes execution. In *BPM*, pages 281–297, 2011.

[15] O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S.E. Whang, and J. Widom. Swoosh: a generic approach to entity resolution. *The VLDB Journal*, 18(1):255–276, 2009.

[16] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

[17] W.J. Black, F. Rinaldi, and D. Mowatt. Facile: Description of the ne system used for muc-7. In *Proceedings of the 7th Message Understanding Conference*, 1998.

[18] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*, pages 1247–1250, 2008.

[19] V. Bryl, C. Giuliano, L. Serafini, and K. Tymoshenko. Using background knowledge to support coreference resolution. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010), August*, 2010.

[20] C. Chen and V. Ng. Combining the best of two worlds: A hybrid approach to multilingual coreference resolution. *EMNLP-CoNLL 2012*, page 56, 2012.

[21] H.H. Chen, Y.W. Ding, and S.C. Tsai. Named entity extraction for information retrieval. *Computer Processing of Oriental Languages*, 12(1):75–85, 1998.

[22] Y. Chen and J. Martin. Towards robust unsupervised personal name disambiguation. *Proceedings of EMNLP and CoNLL*, pages 190–198, 2007.

[23] Peter Christen. A comparison of personal name matching: Techniques and practical issues. In *ICDM Workshops*, pages 290–294, 2006.

[24] Paul Compton, Boualem Benatallah, Julien Vayssière, Lucio Menzel, Hartmut Vogler, et al. An incremental knowledge acquisition method for improving duplicate invoices detection. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pages 1415–1418. IEEE, 2009.

[25] H. Daumé III and D. Marcu. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 97–104. Association for Computational Linguistics, 2005.

[26] David Day, Janet Hitzeman, Michael L Wick, Keith Crouch, and Massimo Poesio. A corpus for cross-document co-reference. In *LREC*, 2008.

[27] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.

[28] C. Dozier and T. Zielund. Cross-document co-reference resolution applications for people in the legal domain. In *42nd Annual Meeting of the Association for Computational Linguistics, Reference Resolution Workshop, Barcelona, Spain*, 2004.

[29] Mohamed G. Elfeky, Ahmed K. Elmagarmid, and Vassilios S. Verykios. Tailor: A record linkage tool box. In *ICDE*, pages 17–28, 2002.

[30] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.

[31] T. Elsayed, J. Lin, and D.W. Oard. Pairwise document similarity in large collections with mapreduce. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 265–268. Association for Computational Linguistics, 2008.

[32] M. Elsner, E. Charniak, and M. Johnson. Structured generative models for unsupervised named-entity clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 164–172. Association for Computational Linguistics, 2009.

[33] Automatic Content Extraction. Evaluation plan (ace08). *Proceedings of the ACE*, pages 1–3, 2008.

[34] T. Finin, Z. Syed, J. Mayfield, P. McNamee, and C. Piatko. Using wikitology for cross-document entity coreference resolution. In *Proceedings of the AAAI Spring Symposium on Learning by Reading and Learning to Read*, pages 29–35. AAAI Press, 2009.

[35] M.B. Fleischman and E. Hovy. Multi-document person name resolution. In *Proceedings of 42nd Annual Meeting of the Association for Computational Linguistics (ACL), Reference Resolution Workshop*, pages 66–82, 2004.

[36] W.B. Frakes and R. Baeza-Yates. Information retrieval: data structures and algorithms. 1992.

[37] T. N. Gadd. Phoenix: the algorithm. *Program: Autom. Libr. Inf. Syst.*, 24(4):363–369, September 1990.

[38] Cory B Giles and Jonathan D Wren. Large-scale directional relationship extraction and resolution. *BMC bioinformatics*, 9(Suppl 9):S11, 2008.

[39] Chung Heong Gooi and James Allan. Cross-document coreference on a large scale corpus. In *HLT-NAACL*, pages 9–16, 2004.

[40] S. Green, N. Andrews, M.R. Gormley, M. Dredze, and C.D. Manning. Entity clustering across languages. In *NAACL*, pages 591–599. Association for Computational Linguistics, 2011.

[41] Spence Green, Nicholas Andrews, Matthew R Gormley, Mark Dredze, and Christopher D Manning. Entity clustering across languages. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 60–69. Association for Computational Linguistics, 2012.

[42] Ben Hachey, Claire Grover, and Richard Tobin. Datasets for generic relation extraction*. *Natural Language Engineering*, 18(1):21–59, 2012.

[43] Patrick A. V. Hall and Geoff R. Dowling. Approximate string matching. *ACM Comput. Surv.*, 12:381–402, December 1980.

[44] David O. Holmes and M. Catherine McCabe. Improving precision and recall for soundex retrieval. In *ITCC*, 2002.

[45] N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22. Association for Computational Linguistics, 2004.

[46] L. Kolb, A. Thor, and E. Rahm. Dedoop: efficient deduplication with hadoop. *Proceedings of the VLDB Endowment*, 5(12):1878–1881, 2012.

[47] Hanna Köpcke, Andreas Thor, and Erhard Rahm. Learning-based approaches for matching web data entities. *IEEE Internet Computing*, 14(4):23–31, 2010.

[48] Robert Krovetz, Paul Deane, and Nitin Madnani. The web is not a person, berners-lee is not an organization, and african-americans are not locations: an analysis of the performance of named-entity recognition. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, MWE '11, pages 57–64, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[49] Karen Kukich. Techniques for automatically correcting words in text. *ACM Comput. Surv.*, 24(4):377–439, 1992.

[50] A. Lait and B. Randell. An assessment of name matching algorithms. Technical report, Department of Computer Science, University of Newcastle upon Tyne, 1993.

[51] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34. Association for Computational Linguistics, 2011.

[52] X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 135. Association for Computational Linguistics, 2004.

[53] Xiaoqiang Luo. On coreference resolution performance metrics. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32. Association for Computational Linguistics, 2005.

[54] G.S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 33–40. Association for Computational Linguistics, 2003.

[55] L. Màrquez, M. Recasens, and E. Sapena. Coreference resolution: An empirical study based on semeval-2010 shared task 1. *Language Resources and Evaluation*, pages 1–34, 2012.

[56] Mnica Marrero, Sonia Snchez-Cuadrado, Jorge Morato Lara, and George Andreadakis. Evaluation of named entity extraction systems. *Advances in Computational Linguistics. Research in Computing Science*, 41:47–58, 2009.

[57] James Mayfield, David Alexander, Bonnie J. Dorr, Jason Eisner, Tamer Elsayed, Tim Finin, Clayton Fink, Marjorie Freedman, Nikesh Garera, Paul McNamee, Saif Mohammad, Douglas W. Oard, Christine D. Piatko, Asad B. Sayeed, Zareen Syed, Ralph M. Weischedel, Tan Xu, and David Yarowsky. Cross-document coreference resolution: A key technology for learning by reading. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, pages 65–70, 2009.

[58] A. McCallum. Information extraction: Distilling structured data from unstructured text. *Queue*, 3(9):48–57, November 2005.

[59] Paul McNamee and Hoa Trang Dang. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, 2009.

[60] George A. Miller and Christiane Fellbaum. Wordnet then and now. *Language Resources and Evaluation*, 41(2):209–214, 2007.

[61] Anderberg M.R. *Cluster analysis for applications.* Academic Press, 21973004.

[62] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[63] V. Nastase, M. Strube, B. Börschinger, C. Zirn, and A. Elghafari. Wikinet: A very large scale multi-lingual concept network. In *Proc. of LREC*, volume 10, 2010.

[64] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48, 1970.

[65] V. Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411. Association for Computational Linguistics, 2010.

[66] Y. Ni, L. Zhang, Z. Qiu, and C. Wang. Enhancing the open-domain classification of named entity using linked open data. *The Semantic Web–ISWC 2010*, pages 566–581, 2010.

[67] US NIST. The ace 2003 evaluation plan. *US National Institute for Standards and Technology (NIST)*, pages 2003–08, 2003.

[68] Cheng Niu, Wei Li, and Rohini K Srihari. Weakly supervised learning for cross-document person name disambiguation supported by information extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 597. Association for Computational Linguistics, 2004.

[69] P. Pantel, E. Crestan, A. Borkovsky, A.M. Popescu, and V. Vyas. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 938–947. Association for Computational Linguistics, 2009.

[70] L. Philips. The double-metaphone search algorithm. In *C/C++ User's Journal*, 2000.

[71] Simone Paolo Ponzetto and Michael Strube. Deriving a large-scale taxonomy from wikipedia. In *AAAI*, pages 1440–1445, 2007.

[72] M.R. Potau. *Coreference: Theory, Annotation, Resolution and Evaluation.* PhD thesis, University of Southern California, 2010.

[73] D. Rao, P. McNamee, and M. Dredze. Streaming cross document entity coreference resolution. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1050–1058. Association for Computational Linguistics, 2010.

[74] Adwait Ratnaparkhi. A linear observed time statistical parser based on maximum entropy models. *CoRR*, cmp-lg/9706014, 1997.

[75] D. Ravichandran, P. Pantel, and E. Hovy. Randomized algorithms and nlp: Using locality sensitive hash functions for high speed noun clustering. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 43, page 622, 2005.

[76] Jeffrey C. Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *ANLP*, pages 16–19, 1997.

[77] G. Salton and M.J. McGill. Introduction to modern information retrieval. 1986.

[78] S. Sarawagi and A. Kirpal. Efficient set joins on similarity predicates. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 743–754. ACM, 2004.

[79] L. Sarmento, A. Kehlenbeck, E. Oliveira, and L. Ungar. An approach to web-scale named-entity disambiguation. *Machine Learning and Data Mining in Pattern Recognition*, pages 689–703, 2009.

[80] S. Sekine and E. Ranchhod. *Named entities: recognition, classification and use*, volume 19. John Benjamins Publishing Company, 2009.

[81] S. Singh, A. Subramanya, F. Pereira, and A. McCallum. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 793–803. Association for Computational Linguistics, 2011.

[82] Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015, 2012.

[83] Sameer Singh, Michael L. Wick, and Andrew McCallum. Distantly labeling data for large scale cross-document coreference. *CoRR*, abs/1005.4298, 2010.

[84] W. Skut and T. Brants. Chunk tagger-statistical recognition of noun phrases. *arXiv preprint cmp-lg/9807007*, 1998.

[85] Valentin I Spitkovsky and Angel X Chang. A cross-lingual dictionary for english wikipedia concepts. In *LREC*, pages 3168–3175, 2012.

[86] P Spyns. Natural language processing. *Methods of information in medicine*, 35(4):285–301, 1996.

[87] M.M. Stark and R.F. Riesenfeld. Wordnet: An electronic lexical database. In *Proceedings of 11th Eurographics Workshop on Rendering*. Citeseer, 1998.

[88] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *WWW*, pages 697–706, 2007.

[89] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL*, 2003.

[90] Y. Tsuruoka, Y. Tateishi, J.D. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii. Developing a robust part-of-speech tagger for biomedical text. *Advances in informatics*, pages 382–392, 2005.

[91] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*, pages 45–52. Association for Computational Linguistics, 1995.

[92] Jiannan Wang, Guoliang Li, and Jianhua Feng. Fast-join: An efficient method for fuzzy token matching based string similarity join. In *ICDE*, pages 458–469, 2011.

[93] Gerhard Weikum, Johannes Hoffart, Ndapandula Nakashole, Marc Spaniol, Fabian M. Suchanek, and Mohamed Amir Yosef. Big data methods for computational linguistics. *IEEE Data Eng. Bull.*, 35(3):46–64, 2012.

[94] B. Wellner, A. McCallum, F. Peng, and M. Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 593–601. AUAI Press, 2004.

[95] T. White. *Hadoop: The Definitive Guide*. O'Reilly Media, original edition, June 2009.

[96] M. Wick, A. Culotta, K. Rohanimanesh, and A. McCallum. An entity based model for coreference resolution. In *SIAM International Conference on Data Mining (SDM)*, 2009.

[97] Michael Wick, Sameer Singh, and Andrew McCallum. A discriminative hierarchical model for fast coreference at large scale. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 379–388, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[98] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

Appendix: MapReduce-based Software Prototype

To address the big data challenges discussed in this paper, we have developed a MapReduce-based software prototype to identify huge number of entity mentions (e.g., persons, organizations or locations) across huge amount of multiple documents that refer to the same underlying entity, i.e., the process of cross-document coreference resolution (CDCR). Figures 7.1 and 7.2 illustrate the overall MapReduce architecture for this process. As illustrated in this figure, specified workflows are automatically translated into MapReduce jobs for parallel execution on different Hadoop clusters. In particular, five MapReduce jobs are specified: Entity Extraction, Entity Partitioning, Entity Matching, Classification, and Clustering.

- **MR Job1:** Entity Extraction. In this phase we use UNIMA and OpenNLP to extract named entities. Figure 7.4 illustrates details of this MapReduce job, where set of documents will be feed as input into the Hadoop File System (HDFS). Afterward, set of mappers will prepare documents for tokenization. Finally, using UIMA and OpenNLP, set of extracted named entities and some related metadata such as the document ID which the entity has been extracted from, the type/sub-type of the entity, and document timestamp will be generated as the output of the first MapReduce phase. Entities and associated attributes will be stored into distributed database (Apache Hbase[1], i.e., the Hadoop database which is a distributed scalable big data store) and can be considered as the knowledge base (KB).

- **MR Job2:** Entity Partitioning. As mentioned earlier, pairwise entity comparison can become exponential and very time consuming. To avoid this, we partition named entities according to their type and subtype. For example, all entities typed as 'person' and sub-typed as 'politician' will be stored in the same partition. Each partition will be send to different (MapReduce) mappers to generate candidate entity pairs.

- **MR Job3:** Entity Matching. After partitioning the entities, the third MapRedcue job will be responsible for pairing entities in each constructed partition. Figure 7.5 illustrates details of this MapReduce job, where duplicate pairs will be identified, the importance of the selected entities to a document in the corpus will be identified, and pairwise similarity degree will be calculated for each pair. As a result set of candidate pair entities will be generated and feed into next MapReduce job.

- **MR Job4:** Classification. Candidate pair entities generated in previous MapReduce job, will be feed to this MapReduce job, where entity pairs with similarity degree (i.e., a real number between 0 and 1) more than, or equal to, 0.5 will be considered as *coreferent* otherwise they will be considered as *not-coreferent*.

- **MR Job5:** Clustering. Set of coreferent entities generated in previous MapReduce job, will be feed into this MapReduce job. Figure 7.2 illustrates more details of these steps and the MapReduce mappers and reducers. In particular, set of supervised and/or unsupervised algorithms can be used to analyze coreferent entities, recognize patterns among them, and classify them in different clusters. We used decision tree like algorithm (see Section 4 for details) to classify coreferent entities.

[1] http://hbase.apache.org/

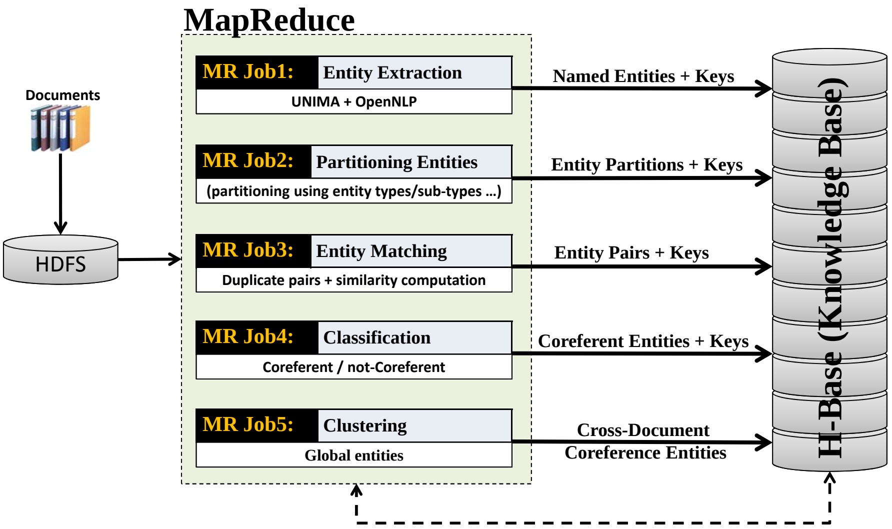


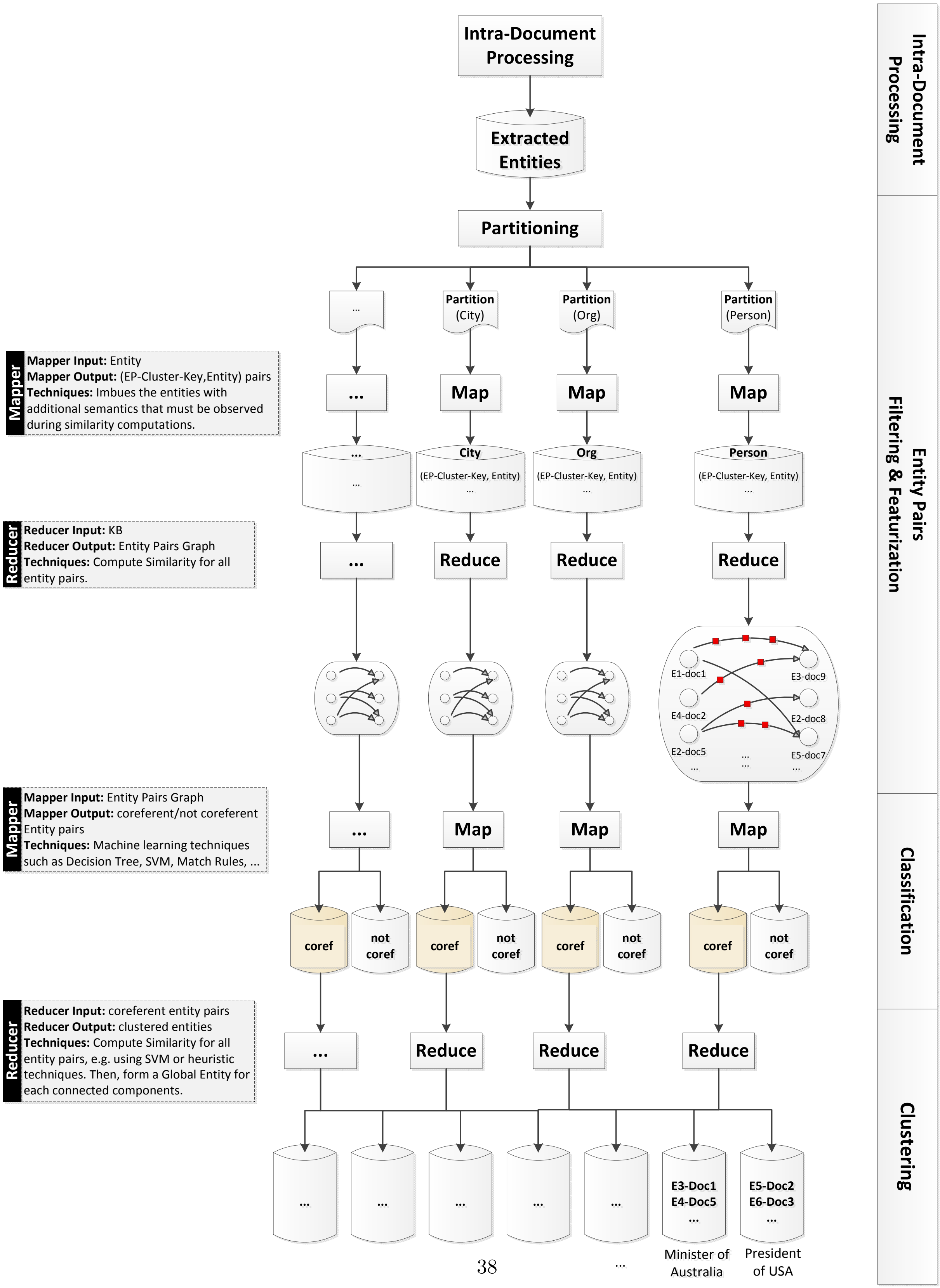
Figure 7.1: CDCR MapReduce Process.

7.1 Software Prototype Evaluation

We evaluated the performance of the software prototype on a cloud system having: (i) One Head Node, having one QuadCore 2.33 GHz processor, 8 GB RAM, 140 GB storage, and with 15TB RAID6 disk array attached; and (ii) Two Identical Blade Servers, having four QuadCore 2.4 GHz processor, 96 GB ram, 1.2 TB storage which configured as a private cloud using OpenStack. We configured the cloud using Apache Hadoop 1.0.4 and ran MapReduce CDCR over clusters of 1, 2 and 4 4-core virtual-machines.

We used Gigaword dataset (see Table 6.3) for evaluating the software prototype. Figure 7.3 illustrates the execution times and the scalability evaluation. As depicted in the figure, we divided each dataset into regular number of documents and ran the experiment for each of them. The evaluation shows the viability and efficiency of using MapReduce in cross-document coreference resolution process. Table 7.1 illustrates the result of experiments over Gigaword dataset including samples of different number of documents, number of extracted entites from each sample, number of entity pairs for each group of extracted entities, and number of coreferent clusters for classified coreferent entities. Table 7.3 illustrates a sample of extracted named entities from Gigaword dataset including the entity type, document ID which the entity has been extracted from, and some metadata about the document such as type of the document (e.g., sport and history) and document timestamp. Table 7.4 illustrates a sample of paired entities and their similarity degree. And Table 7.2 illustrates a sample of coreferent clusters generated from the coreferent entities.

Table 7.3 illustrates a sample of extracted named entities from Gigaword dataset including the entity type, document ID which the entity has been extracted from, and some metadata about the document such as type of the document (e.g., sport and history) and document timestamp. Table 7.4 illustrates a sample of paired entities and their similarity degree. And Table 7.2 illustrates a sample of coreferent clusters generated from the coreferent entities.



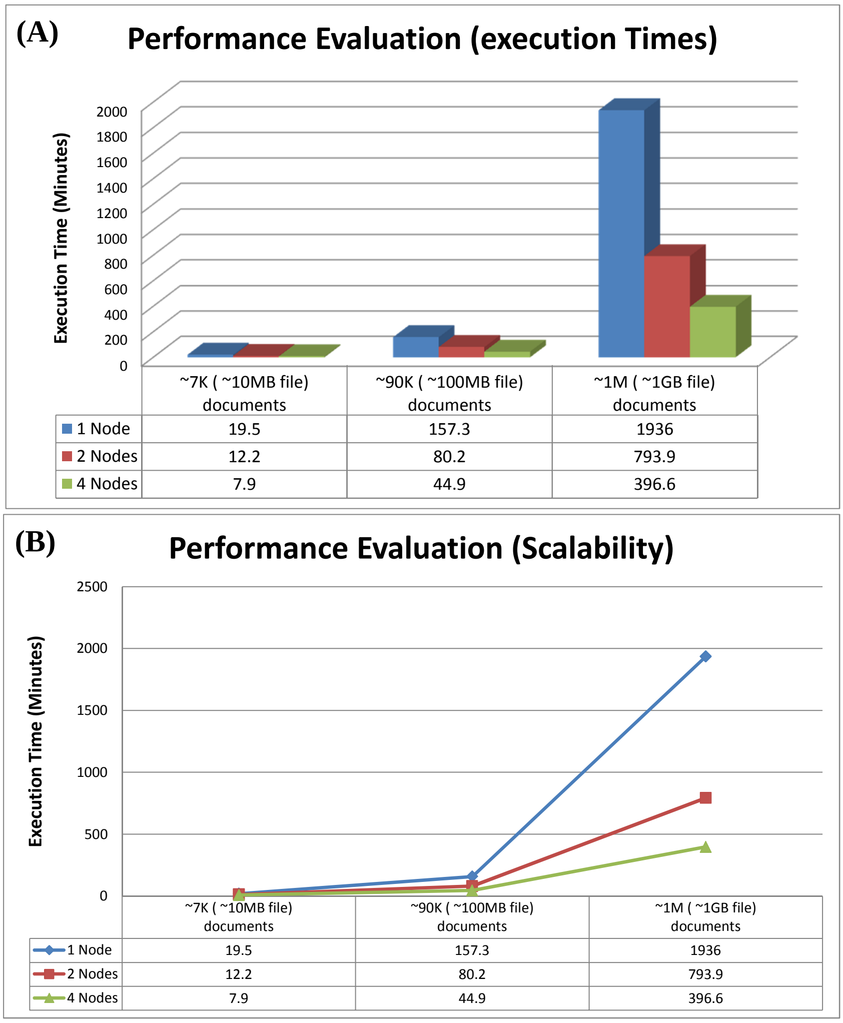
Mapper Input: Entity
Mapper Output: (EP-Cluster-Key,Entity) pairs
Techniques: Imbues the entities with additional semantics that must be observed during similarity computations.

Reducer Input: KB
Reducer Output: Entity Pairs Graph
Techniques: Compute Similarity for all entity pairs.

Mapper Input: Entity Pairs Graph
Mapper Output: coreferent/not coreferent Entity pairs
Techniques: Machine learning techniques such as Decision Tree, SVM, Match Rules, ...

Reducer Input: coreferent entity pairs
Reducer Output: clustered entities
Techniques: Compute Similarity for all entity pairs, e.g. using SVM or heuristic techniques. Then, form a Global Entity for each connected components.

Figure 7.2: CDCR MapReduce Process.



	~7K (~10MB file) documents	~90K (~100MB file) documents	~1M (~1GB file) documents
🟦 1 Node	19.5	157.3	1936
🟥 2 Nodes	12.2	80.2	793.9
🟩 4 Nodes	7.9	44.9	396.6

	~7K (~10MB file) documents	~90K (~100MB file) documents	~1M (~1GB file) documents
◆ 1 Node	19.5	157.3	1936
■ 2 Nodes	12.2	80.2	793.9
▲ 4 Nodes	7.9	44.9	396.6

Figure 7.3: Software prototype evaluation: (A) execution times; and (B) Scalability.

As an ongoing work, we plan to use a graph-based clustering approach in the second mapper in Figure 7.1. We will use our previous work, i.e. a Map-Reduce enabled graph processing engine [14, 13, 12], to model the entities and the relationships among them as graphs. We will use On-Line Analytical Processing on Graphs [13] to create new relationship between entities mentions by calculating the similarity between them. We plan to employ the weighted support vectors and find the shortest path between two mentions in the graph to facilitate the classification and clustering of entities and their mentions across documents. Moreover, we plan to leverage crowdsourcing techniques [24, 5, 4] for incremental and end-user-centered knowledge acquisition to improve the classification of paired entities.

Table 7.1: The result of experiments over Gigaword dataset including samples of different number of documents, number of extracted entities from each sample, number of entity pairs for each group of extracted entities, and number of coreferent clusters for classified coreferent entities.

Number of Documents	Number of Extracted Entities	Number of Pairs	Number of Coreferent Clusters
~7K (~10MB file)	68,532	382,102	3,823
~90K (~100MB file)	660,877	2,125,666	29,275
~1M (~1GB file)	5,652,005	16,722,466	86,186

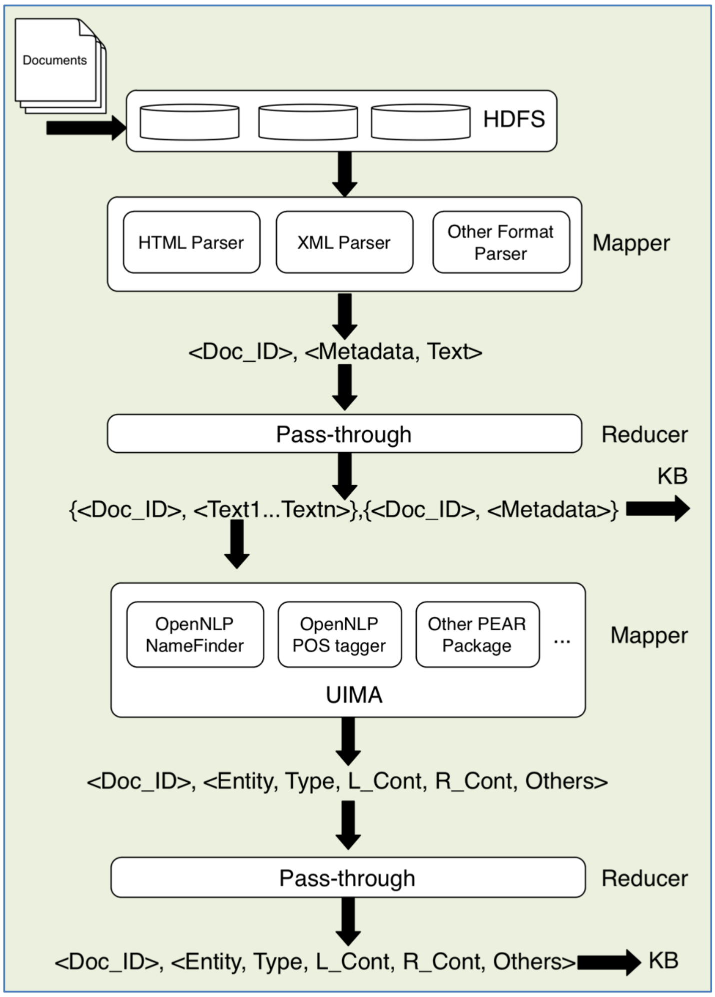

Documents
HDFS
HTML Parser
XML Parser
Other Format Parser
Mapper
<Doc_ID>, <Metadata, Text>
Pass-through
Reducer
KB
{<Doc_ID>, <Text1...Textn>},{<Doc_ID>, <Metadata>}
OpenNLP NameFinder
OpenNLP POS tagger
Other PEAR Package
...
Mapper
UIMA
<Doc_ID>, <Entity, Type, L_Cont, R_Cont, Others>
Pass-through
Reducer
<Doc_ID>, <Entity, Type, L_Cont, R_Cont, Others>
KB

Figure 7.4: Entity extraction task in the MapReduce-based software prototype.

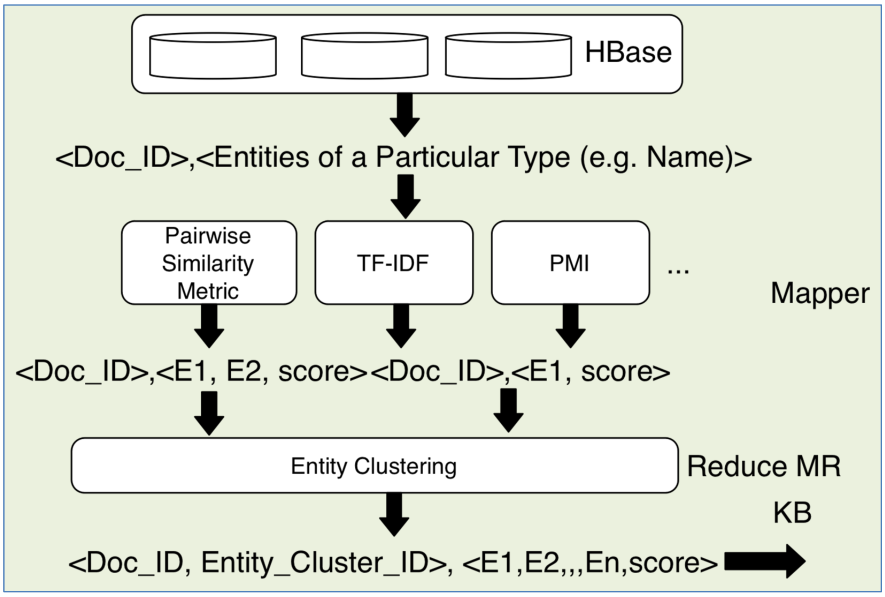
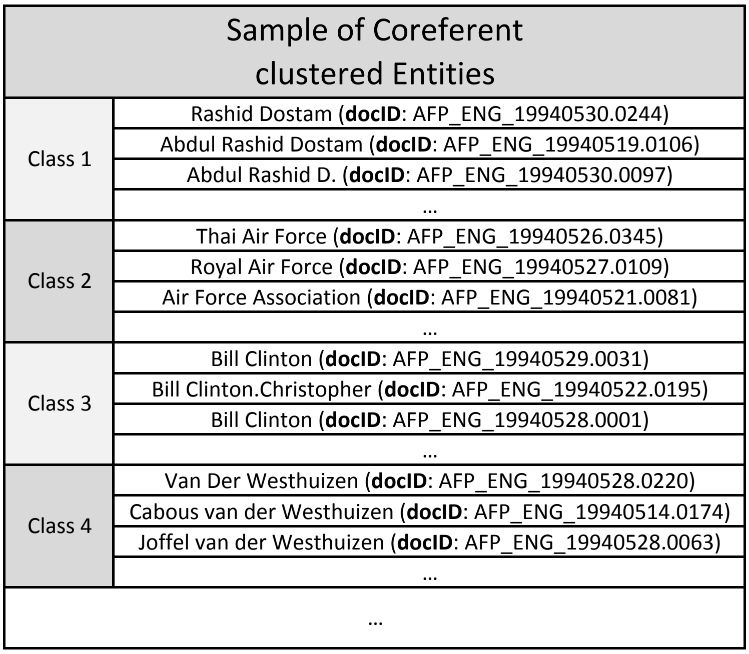

HBase
<Doc_ID>,<Entities of a Particular Type (e.g. Name)>
Pairwise Similarity Metric
TF-IDF
PMI
...
Mapper
<Doc_ID>,<E1, E2, score><Doc_ID>,<E1, score>
Entity Clustering
Reduce MR
KB
<Doc_ID, Entity_Cluster_ID>, <E1,E2,,,En,score>

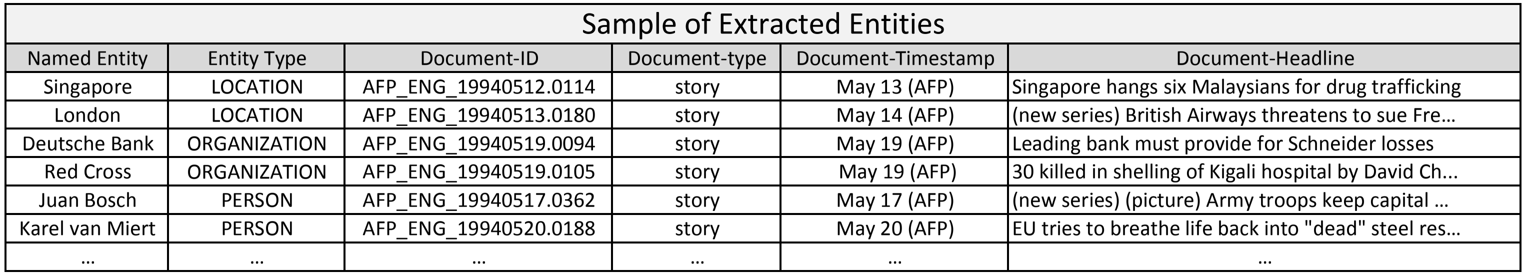
Figure 7.5: Entity clustering task in the MapReduce-based software prototype.

Table 7.2: Sample of coreferent clusters generated from coreferent entities in the MapReduce-based software prototype.



Sample of Coreferent clustered Entities	
Class 1	Rashid Dostam (**docID**: AFP_ENG_19940530.0244)
	Abdul Rashid Dostam (**docID**: AFP_ENG_19940519.0106)
	Abdul Rashid D. (**docID**: AFP_ENG_19940530.0097)
	…
Class 2	Thai Air Force (**docID**: AFP_ENG_19940526.0345)
	Royal Air Force (**docID**: AFP_ENG_19940527.0109)
	Air Force Association (**docID**: AFP_ENG_19940521.0081)
	…
Class 3	Bill Clinton (**docID**: AFP_ENG_19940529.0031)
	Bill Clinton.Christopher (**docID**: AFP_ENG_19940522.0195)
	Bill Clinton (**docID**: AFP_ENG_19940528.0001)
	…
Class 4	Van Der Westhuizen (**docID**: AFP_ENG_19940528.0220)
	Cabous van der Westhuizen (**docID**: AFP_ENG_19940514.0174)
	Joffel van der Westhuizen (**docID**: AFP_ENG_19940528.0063)
	…
…	

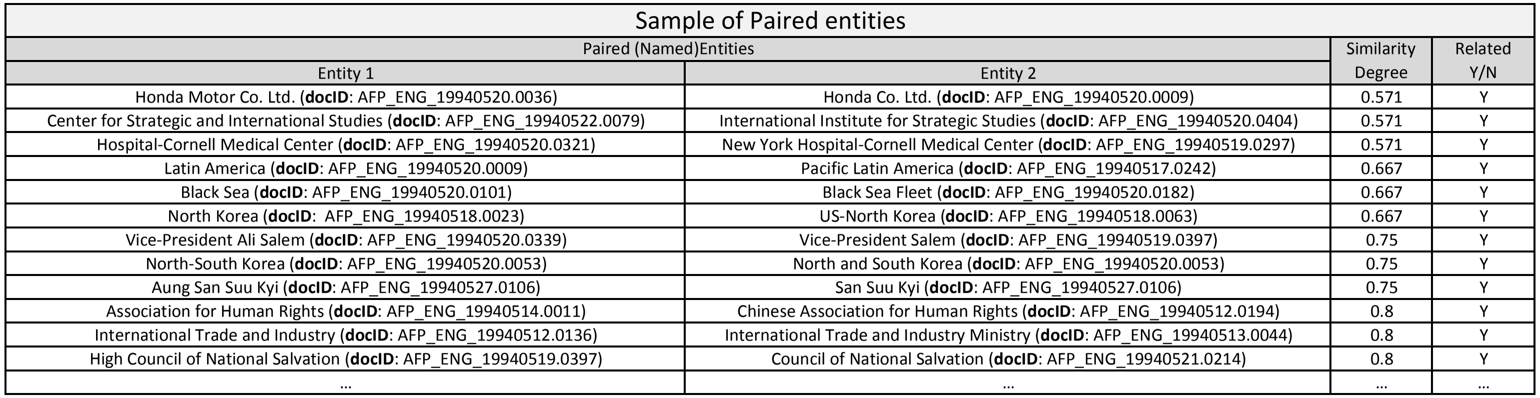
Table 7.3: Sample of extracted named entities from Gigaword dataset including the entity type, document ID which the entity has been extracted from, and some metadata about the document such as type of the document (e.g., sport and history) and document timestamp in the MapReduce-based software prototype.



Sample of Extracted Entities

Named Entity	Entity Type	Document-ID	Document-type	Document-Timestamp	Document-Headline
Singapore	LOCATION	AFP_ENG_19940512.0114	story	May 13 (AFP)	Singapore hangs six Malaysians for drug trafficking
London	LOCATION	AFP_ENG_19940513.0180	story	May 14 (AFP)	(new series) British Airways threatens to sue Fre...
Deutsche Bank	ORGANIZATION	AFP_ENG_19940519.0094	story	May 19 (AFP)	Leading bank must provide for Schneider losses
Red Cross	ORGANIZATION	AFP_ENG_19940519.0105	story	May 19 (AFP)	30 killed in shelling of Kigali hospital by David Ch...
Juan Bosch	PERSON	AFP_ENG_19940517.0362	story	May 17 (AFP)	(new series) (picture) Army troops keep capital ...
Karel van Miert	PERSON	AFP_ENG_19940520.0188	story	May 20 (AFP)	EU tries to breathe life back into "dead" steel res...
...	...	...	...	...	...

Table 7.4: Sample of paired entities and their similarity degree in the MapReduce-based software prototype.



Sample of Paired entities

Paired (Named)Entities		Similarity	Related
Entity 1	Entity 2	Degree	Y/N
Honda Motor Co. Ltd. (**docID**: AFP_ENG_19940520.0036)	Honda Co. Ltd. (**docID**: AFP_ENG_19940520.0009)	0.571	Y
Center for Strategic and International Studies (**docID**: AFP_ENG_19940522.0079)	International Institute for Strategic Studies (**docID**: AFP_ENG_19940520.0404)	0.571	Y
Hospital-Cornell Medical Center (**docID**: AFP_ENG_19940520.0321)	New York Hospital-Cornell Medical Center (**docID**: AFP_ENG_19940519.0297)	0.571	Y
Latin America (**docID**: AFP_ENG_19940520.0009)	Pacific Latin America (**docID**: AFP_ENG_19940517.0242)	0.667	Y
Black Sea (**docID**: AFP_ENG_19940520.0101)	Black Sea Fleet (**docID**: AFP_ENG_19940520.0182)	0.667	Y
North Korea (**docID**: AFP_ENG_19940518.0023)	US-North Korea (**docID**: AFP_ENG_19940518.0063)	0.667	Y
Vice-President Ali Salem (**docID**: AFP_ENG_19940520.0339)	Vice-President Salem (**docID**: AFP_ENG_19940519.0397)	0.75	Y
North-South Korea (**docID**: AFP_ENG_19940520.0053)	North and South Korea (**docID**: AFP_ENG_19940520.0053)	0.75	Y
Aung San Suu Kyi (**docID**: AFP_ENG_19940527.0106)	San Suu Kyi (**docID**: AFP_ENG_19940527.0106)	0.75	Y
Association for Human Rights (**docID**: AFP_ENG_19940514.0011)	Chinese Association for Human Rights (**docID**: AFP_ENG_19940512.0194)	0.8	Y
International Trade and Industry (**docID**: AFP_ENG_19940512.0136)	International Trade and Industry Ministry (**docID**: AFP_ENG_19940513.0044)	0.8	Y
High Council of National Salvation (**docID**: AFP_ENG_19940519.0397)	Council of National Salvation (**docID**: AFP_ENG_19940521.0214)	0.8	Y
...	...	...	...