Efficient Knowledge Management for Named Entities from Text

Sourav Dutta

Dissertation
for Obtaining the Degree of

Doctor of Engineering (*Dr.-Ing.*)
Faculty Of Mathematics and Computer Science
SAARLAND UNIVERSITY

Saarbrücken, Germany
September, 2016

Degree Colloquium

Dean	Prof. Dr. rer. nat. Frank-Olaf Schreyer
Date	09 March, 2017
Place	Saarbrücken, Germany

Examination Board

Chairman	Prof. Dr. Dietrich Klakow
Supervisor and First Reviewer	Prof. Dr.-Ing. Gerhard Weikum
Second Reviewer	Prof. Dr. techn. Wolfgang Nejdl
Third Reviewer	Prof. Dr.-Ing. Klaus Berberich
Scientific Assistant	Dr. Rishiraj Saha Roy

Abstract

The evolution of search from keywords to entities has necessitated the efficient harvesting and management of entity-centric information for constructing knowledge bases catering to various applications such as semantic search, question answering, and information retrieval. The vast amounts of natural language texts available across diverse domains on the Web provide rich sources for discovering facts about named entities such as people, places, and organizations.

A key challenge, in this regard, entails the need for precise identification and disambiguation of entities across documents for extraction of attributes/relations and their proper representation in knowledge bases. Additionally, the applicability of such repositories not only involves the quality and accuracy of the stored information, but also storage management and query processing efficiency. This dissertation aims to tackle the above problems by presenting efficient approaches for entity-centric knowledge acquisition from texts and its representation in knowledge repositories.

This dissertation presents a robust approach for identifying text phrases pertaining to the same named entity across huge corpora, and their disambiguation to canonical entities present in a knowledge base, by using enriched semantic contexts and link validation encapsulated in a hierarchical clustering framework. This work further presents language and consistency features for classification models to compute the credibility of obtained textual facts, ensuring quality of the extracted information. Finally, an encoding algorithm, using frequent term detection and improved data locality, to represent entities for enhanced knowledge base storage and query performance is presented.

Kurzfassung

Die Weiterentwicklung der Schlagwortsuche hin zu Entitäten erfordert die effiziente Sammeln und Verwalten von Informationen zur Konstruktion von Wissensbasen. Diese Wissensbasen sind die Grundlage für verschiedene Anwendungen, wie etwa semantische Suche, maschinelle Beantwortung von Fragen oder Informationsrückgewinnung. Die großen Mengen von Texten in natürlicher Sprache, die über verschiedene Domänen hinweg im World Wide Web verfügbar sind, bietet dabei eine reiche InformationsQuelle zum Auffinden von Fakten über benannte Entitäten wie Personen, Orte, und Organisationen.

Wesentliche Herausforderungen in dieser Hinsicht sind die genaue, dokumentübergreifende Identifikation und Disambiguierung verteilten Entitäten, die zur Extraktion von Attributen/Relationen benötigt werden sowie deren geeignete Repräsentation in Wissensbasen. Die Anwendbarkeit solcher Wissensbasen hängt dabei nicht nur von der Qualität und Genauigkeit der gespeicherten Informationen ab, sondern auch von der Speicherplatzverwaltung und der effizienten Bearbeitung von Anfragen. Ziel dieser Dissertation ist die Auseinandersetzung mit den zuvor beschriebenen Problemen. Dazu werden effiziente Verfahren zur entitätszentrischen Wissensaquise aus Texten und deren Repräsentation in Wissensbasen vorgestellt.

Diese Dissertation stellt ein robustes Verfahren zur Identifikation von Textphrasen vor, die sich auf die gleiche Entität beziehen und über große Korpora verteilt sein können. Außerdem wird die Disambiguierung und Kanonisierung von Entitäten mittels eines hierarchischen Clusteranalyseverfahrens beschrieben, das auf angereichertem semantischem Kontext und Linkvalidierung beruht. Einen weiteren Teil dieser Arbeit stellen Sprach- und Konsistenzmerkmale dar, die in Klassifizerungsmodellen verwendet werden, um die Glaubwürdigkeit von Fakten zu bestimmen und somit die Qualität der extrahierten Informationen zu gewährleisten. Schlussendlich wird ein Kodierungsalgorithmus zur Repräsentation von Entitäten präsentiert, der auf der Erkennung von häufigen Termen und einer verbesserten Datenlokalität beruht, um dadurch eine bessere Speicherplatzverwaltung und Abfrageperformanz der Wissensbasis zu ermöglichen.

Summary

Entity attributes and their relationships are valuable assets for various modern day applications such as semantic search, question answering, and other information extraction tasks. The huge amount of natural language texts across diverse domains on the Web, such as Wikipedia, news articles, blogs, and forums, provide rich sources for harnessing entity and event related information. The harvested knowledge is then succinctly represented, typically as a labeled graph structure, by large knowledge bases such as YAGO, DBPedia, Freebase, etc., forming the backbone of Web-scale applications. Crucial challenges in the construction of such knowledge repositories not only involve scalability and robustness to tackle petabytes of text data, but also accurate detection and disambiguation of the discovered entities for proper representation, credibility of obtained facts to ensure quality, and superior query performance of the knowledge stores for real-time interaction. This dissertation makes the following contributions for efficient knowledge management for named entities, such as people, locations, and organizations, extracted from text.

Cross-document entity co-reference resolution involves the identification of textual expressions referring to the same entity, possibly with different surface forms, present across documents within a corpus. The linguistic diversity among documents, scalability for tackling large data volumes, and the presence of multiple entities with similar representations further complicates the task at hand. Existing approaches combining text snippets to capture context, external feature inclusion, and vector-space similarity for clustering, tend to be computationally expensive with limited performance accuracy, and suffer in terms of robustness. The proposed $CROCS$ method addresses the problems by intelligent construction of *semantic summaries* better capturing entity context by using texts, co-occurring mentions, and external features encapsulated within a sampling-based hierarchical clustering framework for enhanced performance quality and robustness. Experimental analysis on large real-life news corpora demonstrates significant improvements in co-reference resolution accuracy (even for sparse "long tail" entities) and run-time for $CROCS$ over state-of-the-art methods.

Entity linking corresponds to the disambiguation of discovered named entities and linking them to pertinent entries, if present, in a knowledge base. It enables the dynamic growth of repositories with emerging facts about already known entities, or the incorporation of newly extracted entities currently absent in the knowledge base.

State-of-the-art methodologies exploit context overlap, semantic relatedness, similarity to Wikipedia articles, or coherence based graph algorithms. However, they tend to disregard the interaction synergies between entity contexts present across documents, leading to performance degradation and poor disambiguation of emerging or sparsely represented entities. This dissertation proposes the $C3EL$ framework for *jointly* performing co-reference resolution and entity linking by iteratively harnessing enhanced mention contexts and external features to enable global propagation of information for precisely linking known entities and identifying previously unknown emerging entities. Empirical results on large text datasets show significant improvements in entity co-reference resolution and linking accuracies, particularly for "long tail" entities.

Text credibility analysis measures the veracity of information presented within a text snippet, thereby lending credibility to the facts extracted from such sources to be incorporated in knowledge bases. The presence of deceptive entity attributes and relationships within a knowledge store would severely impact its applicability in real-life scenarios. Previous works consider language model and user-centric features to classify text as truthful or otherwise. However, these approaches tend to utilize overly domain-specific information and fail to provide evidence as to why a text should not be considered credible. This work presents a credibility analysis framework by using limited information for constructing *language models*, *user-sentiment features*, and *consistency features* to classify text snippets as credible or not. Further, the approach not only provides observed inconsistencies as *evidence* for the classification decision, but is also shown to be robust and domain-independent. Evaluations on user reviews from real-world forums depict enhanced credibility classification accuracy of the framework over existing approaches.

Encoding of knowledge bases forms a key step for efficient storage of the underlying graphical structure representing the knowledge base, wherein a dictionary mapping the original entity strings to numerical identifiers (ID) is constructed for downstream index construction and query processing by an RDF engine. Current RDF engines generally employ appearance order or hash based ID assignment to terms, and hence fail to capture semantic relationships across entities for possible join query optimizations. Moreover, such strategies also incur high storage and load costs from sub-optimal compression due to possible large ID assignment to highly frequent terms. This dissertation proposes the $KOGNAC$ framework for efficient encoding of knowledge base terms based on approximate *frequent item detection* (enabling index compression) and *ontological similarity* based term clustering to assign close IDs to semantically related terms (capturing data locality) to enhance join query performance. Integration of $KOGNAC$ into state-of-the-art RDF engines and column-stores demonstrates improved storage requirements, query runtime, and I/O scan costs on various benchmarks.

Keywords: *Cross Document Co-Reference Resolution, Named Entity Linking, Knowledge Base, Credibility, RDF Term Encoding, Classification Models, Clustering, Semantic Context, Ontology, RDF Engines, Text*

Zusammenfassung

Entitätsattribute und deren Beziehungen sind wichtige Komponenten für moderne Anwendungen, wie etwa semantische Suche, maschinelle Beantwortung von Fragen und anderen Informationsextraktionsaufgaben. Die immense Menge an Texten in natürlicher Sprache, beispielsweise in der Wikipedia, in Nachrichtenartikeln, Blogs und Foren, die im World Wide Web über mehrere Domänen hinweg verfügbar ist, bietet eine reiche Quelle, um entitäts- und ereignisbezogene Informationen zu gewinnen. Große Wissensbasen, wie zum Beispiel YAGO, DBPedia, Freebase, etc., repräsentieren das gesammelte Wissen üblicherweise als beschriftete Graphstruktur und bilden das Rückgrat für "Web-Scale"-Systeme. Wichtige Herausforderungen bei der Konstruktion solcher Wissensbasen beinhalten nicht nur die Skalierbarkeit und Robustheit von Verfahren, um Petabytes an Text zu verarbeiten, sondern auch die genaue Erkennung und Disambiguierung von erkannten Entitäten, um diese geeignet zu repräsentieren. Darüber hinaus muss die Plausibilität von Fakten in Betracht gezogen werden, um die Qualität einer Wissensbasis zu gewährleisten. Schlussendlich ist auch eine hervorragende Abfrageperformanz entscheidend, um mit einer Wissensbasis in Echtzeit zu interagieren. Diese Dissertation macht dabei die folgenden Beiträge zum effizienten Wissensmanagement von benannten Entitäten, wie zum Beispiel Personen, Orte, und Organisationen, die aus Text extrahiert wurden.

Dokumentübergreifende Entitätskoreferenz-Auflösung beinhaltet die Identifikation von sprachlichen Ausdrücken, die sich auf die gleiche Entität beziehen, jedoch möglicherweise in verschiedenen Ausdrucksweisen vorliegen und über mehrere Dokumente innerhalb eines Korpus verteilt sind. Die linguistische Diversität von zwischen Dokumenten, die nötige Skalierbarkeit, um große Datenvolumen zu handhaben, und das Auftreten von mehreren Entitäten mit ähnlichen Repräsentationen sind dabei erschwerende Faktoren. Existierende Verfahren kombinieren Textausschnitte zur Erfassung von Kontextinformationen, externe Merkmale, und vektorraumbasierte Ähnlichkeitsmaße zur Clusteranalyse. Solche Verfahren sind jedoch oft nicht robust, äußerst rechenintensiv und haben eine geringe Genauigkeit. Die vorgeschlagene $CROCS$-Methode adressiert diese Probleme mittels einer intelligenten Konstruktion von *semantischen Zusammenfassungen*, die den Kontext der Entitäten besser darstellen, indem sie Informationen aus Text, dem gleichzeitigen Vorkommen von Entitäten, sowie externe Merkmale in einem stichprobenbasierten, hierarchischen Clusteranalyseverfahren

bündeln. Experimente mit einem aus Nachrichtentexten bestehenden Korpus demonstrieren signifikante Verbesserungen in der Auflösung von Koreferenzen (sogar für nicht häufig auftretende Entitäten) und der Laufzeit von $CROCS$ gegenüber anderen aktuellen Methoden.

Entitätsverlinkung bezieht sich auf die Disambiguierung von gefundenen benannten Entitäten und das Verlinken derer mit Einträgen aus einer Wissensbasis. Solche Verlinkungen ermöglichen ein dynamisches Wachsen von Wissensbasen entweder mit neu aufkommenden Fakten über schon bekannte Entitäten oder mit neu extrahierten Entitäten, die noch nicht in der Wissensbasis vorhanden sind. Aktuelle Methoden nutzen Überlappungen zwischen Kontextinformationen von Entitäten, semantische Beziehungsmaße, Ähnlichkeit zu Wikipediaartikeln oder kohärenzbasierte Graphalgorithmen. Solche Methoden tendieren jedoch dazu, die Interaktionssynergien zwischen Entitäskontexten zu ignorieren, die sich über mehrere Dokumente verteilen. Dies resultiert in einem Performanzverlust und schlechter Disambiguierung von neu aufkommenden und seltenen Entitäten. Diese Dissertation stellt den $C3EL$ Ansatz vor, welcher Koreferenz-Auflösung und Entitätsverlinkung in *Kombination* durchführt, indem sich bessere Kontextinformationen und externe Merkmale iterativ zu Nutze gemacht werden, um eine globale Informationsausbreitung zum genauen Verlinken von bekannten Entitäten und Identifikation von zuvor unbekannten und neu aufkommenden Entitäten zu ermöglichen. Empirische Resultate mit großen Textdatensätzen zeigen signifikante Verbesserungen der Entitätskoreferenz-Auflösung und eine höhere Genauigkeit bei der Entitätsverlinkung speziell bei seltenen, sogenannten “long tail” Entitäten.

Glaubwürdigkeitsanalysen von Texten untersuchen die Glaubhaftigkeit von Information, die in einem Textausschnitt vorliegen, und beurteilen somit die Glaubwürdigkeit der daraus extrahierten Fakten, die in eine Wissensbasis übernommen werden sollen. Die Übernahme von irreführenden Entitätsattributen oder -relationen würde die Anwendbarkeit von Wissensbasen negativ beeinflussen und einschränken. Zur Klassifikation eines Textes als wahrheitsgetreu oder nicht ziehen vorherige Arbeiten in diesem Gebiet Sprachmodelle und benutzerspezifische Merkmale in Betracht. Sie tendieren dabei aber zu stark dazu, domänenspezifische Informationen einzusetzen, und scheitern daran, Evidenzen bereitzustellen, die begründen, wieso ein Textausschnitt glaubwürdig ist. Diese Arbeit stellt ein Verfahren vor, das lediglich eine begrenzte Menge an Information zur Konstruktion von *Sprachmodellen, Sentimentmerkmalen* (“Sentiment” für Empfindung, Gefühl), und *Konsistenzmerkmalen* benötigt, um damit Textausschnitte als glaubhaft oder nicht zu klassifizieren. Des Weiteren bietet dieser Ansatz nicht nur beobachtete Inkonsistenzen als *Evidenz* für die Klassifikationsentscheidung an, sondern ist auch robust und domänenunabhängig. Evaluationen auf Basis von Benutzerbeurteilungen mit realen Forendaten zeigen verbesserte Klassifikationsergebnisse im Vergleich zu existierenden Verfahren.

Die Kodierung von Wissensbasen ist ein entscheidender Schritt zum effizienten Speichern der zugrunde liegenden Graphstrukturen, die eine Wissensbasis repräsentieren. Zur Anfragebearbeitung und nachgelagerten Indexkonstruktion durch eine RDF-Datenbank wird ein Katalog konstruiert, der die ursprünglichen Entitätszeichenketten auf numerische Identifikatoren (ID) abbildet. Aktuelle RDF-Datenbanken benutzen dafür die Reihenfolge des Erscheinens oder hashbasierte Zuweisungen zur ID-Vergabe an Terme. Sie scheitern aber daran, semantische Beziehungen zwischen Entitäten zu erfassen, die eine verbesserte Optimierung von JOIN-Anfragen zulassen würden. Des Weiteren leiden solche Strategien an höheren Speicherplatzanforderungen und höherer Last aufgrund suboptimaler Kompression durch die Vergabe zu großer IDs an häufig auftretende Terme. Zur effizienten Kodierung von Termen in Wissensbasen stellt diese Dissertation das $KOGNAC$ System vor. Basierend auf einer approximierten *Erkennung von häufigen Termen* und *Termclustern ontologisch ähnlicher Terme* werden nahe beieinander gelegene IDs an semantisch verwandte Terme vergeben, um dadurch die Performanz von JOIN-Anfragen zu verbessern. Die Integration von $KOGNAC$ in aktuelle RDF- und spaltenorientierte Datenbanken hat in mehreren Experimenten verbesserte Speicherplatzanforderungen, kürzere Anfragelaufzeiten und geringere E/A Kosten gezeigt.

Stichwörter:

Dokumentübergreifende Entitätskoreferenz-Auflösung, Benannte Entitätsverlinkung, Wissensbasen, Glaubwürdigkeit, RDF Term Kodierung, Klassifikationsmodelle, Clustern, Semantischer Kontext, Ontologie, RDF-Datenbank, Textverarbeitung

Acknowledgements

I would like to extend my heartfelt thanks and regards to my supervisor *Prof. Dr-Ing. Gerhard Weikum* for his enormous support and guidance throughout the period of my doctoral studies. Without his invaluable insights, continuous counsel, and the encouragement to pursue a steady research direction, this work would certainly not have been possible.

I am immensely grateful to the *International Max Planck Research School for Computer Science (IMPRS-CS)* and the *Max Planck Institute for Informatics (MPI-INF)*, Germany for providing a superb and stimulating research environment enabling me to focus primarily on my Ph.D. studies, and also a strong platform for evolving as a researcher.

I would also like to thank Google Inc., for funding my studies through the *Google European Doctoral Fellowship*.

My colleagues and friends at the Databases and Information Systems group at Max Planck Institute ensured a cheerful atmosphere and much needed motivation during stressful time-periods — a big thank you to all! A special thanks to Mr. Patrick Ernst for helping me with the translation for the German portions of this dissertation.

I would also like to thank my parents for their unwavering support in all my decisions at all times. Most importantly, thank you so much Nikita for sharing your life with me and always reassuring and standing with me; before, during my Ph.D., and forever — a journey to cherish.

Table of Contents

1 Introduction

The area of Information Extraction (IE) involves the task of extracting information from unstructured or semi-structured data from vast amounts of digital information expressed in natural language texts. The proliferation of knowledge sharing across domains in the form of Web pages, text documents, blogs, news articles, and communities like Wikipedia has opened rich avenues to systematically harvest and represent entity related attributes and relationships as machine-readable knowledge repositories. The advent of knowledge bases such as DBPedia, YAGO, and Freebase, has enabled the precise representation of obtained facts and subsequent concise querying catering to modern-day applications like *semantic search*, *question answering*, and *machine translation* to name a few. As a consequence, the intelligent understanding of natural language and the expressed sentiments by machines is necessitated for query reporting, analytics, and automated discovery and acquisition of further knowledge. To this end, this dissertation discusses the related state-of-the-art methods and research challenges, and proposes new efficient algorithms to tackle the inherent problems in entity-centric knowledge harvesting and management from text documents.

1.1 Motivation

Identification of *named entities* such as persons, locations, organizations, events, products, etc., along with their senses from texts forms the basic building block for the extraction of facts for construction of knowledge stores [Weikum and Theobald, 2010]. Given the linguistic diversity across documents and varied surface forms of entities (such as acronyms and aliases), disambiguation of mentions to proper entities is crucial to obtain a holistic view of the information related to an entity. The problem of *cross-document co-reference resolution* (CCR) to identify and link textual mentions denoting the same entity across documents, provides a primary module for information extraction. For example, consider the following text snippets obtained from two documents to form the input to an IE pipeline.

T1: Hugh Jackman was thrilled to play the role of Wolverine.

T2: Hugh was accompanied by daughter Eva on the movie sets.

Named entity recognition (NER) techniques are initially used to extract the mentions (phrases referring to named entities) present in the texts, i.e., `Hugh Jackman`, `Wolverine`, `Hugh`, and `Eva`. The ensuing CCR procedure should identify that the mentions `Hugh Jackman` and `Hugh` refer to the same entity – hence facts obtained from the text snippets pertain to the same person. However, the sheer amount of such available data along with the unstructured and noisy nature, poses a challenge in terms of computational efficiency. The presence of possibly large numbers of entities with the same or similar name (e.g., "Hugh") further complicates the precise identification and disambiguation of co-referent mentions across documents.

The generation of new entities and relationships for already known entities necessitates the periodic maintenance of knowledge bases to reflect the dynamic world. To enable proper knowledge aggregation, an important aspect is to determine if the identified entity is already present in the knowledge base, referred to as the task of *named entity linking* (NEL). Although disambiguation can be performed based on prior popularity, context similarity, and semantic coherence of entities, such approaches fail to capture sparsely represented entities or newly emerging entities currently absent in the knowledge repository. In our example, the *out-of-knowledge-base entity* "Eva" would thus be incorrectly linked to some other entity, leading to mis-representation of the father-daughter relationship (obtained from the text) between "Hugh" and "Eva". Hence, a good disambiguation method needs to sift the "long tail" entities for proper and accurate representation of extracted facts.

The proliferation of platforms for user expressivity such as social media and blogs has led to the surge of available facts pertaining to entities. Unfortunately, such sources are increasingly prone to non-credible, erroneous, and biased data and viewpoints. Extraction and incorporation of possibly deceptive information would thus impair the quality of knowledge repositories. This further necessitates not only the proper detection and disambiguation of entities, but also efficient approaches for credibility analysis of the textual information presented and the extracted facts therefrom.

For exposition, considering the example entity `Hugh Jackman`, assume attributes and relationships such as $actor$, $bornIn\ Australia$, $playedIn\ XMen$, and $marriedTo\ Deborra-Furness$, to have been extracted from the text documents. The harnessed information is then intrinsically represented as a labeled graph structure with the entities forming the nodes (labeled with attributes) and the relations among entities denoted as edges (labeled with relations). The underlying graphical structure of knowledge bases is generally cast into triplets using the *Resource Description Framework* (RDF)

format for efficient management by RDF engines [Neumann and Weikum, 2008] or column stores [Sidirourgos *et al.*, 2008] and SPARQL (and its extensions) based query processing [Harbi *et al.*, 2015]. Succinct representation of entities for proper storage management and optimized query processing is thus pivotal for entity oriented analytics [Suchanek and Weikum, 2014] and complex question answering like `actor Hugh Jackman married to ?` or `Which Australian actor has the well-known screen alias Logan?`. This enables knowledge repositories to cater to modern applications like Google Knowledge Graph search [Singhal, 2012], IBM Watson Question Answering system [IBM, 2012], etc., for natural language and keyword based *semantic search*.

Entity-centric knowledge acquisition and subsequent representation of extracted facts and relations as machine readable knowledge bases undergo several life-cycle stages:

1. *collect and build* a knowledge base of facts/attributes pertaining to entities and their inter-relationships from natural language texts and other Web sources;
2. *validate* the obtained facts to remove possible discrepancies and errors;
3. *efficiently manage* the collected information for proper storage and enhanced query answering performance; and
4. *expand and update* the knowledge repository with new facts and relations coming into existence.

The next section discusses the associated challenges for entity based knowledge extraction and management.

1.2 Challenges

Scalable Entity Resolution. Efficient state-of-the-art methods for entity co-reference resolution (CR) within documents use syntactic and linguistic features [Lee *et al.*, 2013] or multi-phase sieves [Lee *et al.*, 2011] to capture the similarities between mentions. These methods have been shown to provide high accuracy. Further, cross-document CR (CCR) approaches have been proposed using vector-space cosine similarity to compute similarities between mention contexts. However, such methods are computationally expensive involving pair-wise mention comparisons, rendering them impractical for CCR tasks at Web scale. Moreover, the distinction between multiple entities with the same textual representation (e.g., several persons with name “Hugh”) or of the same entities with different surface forms (e.g., name “Robert” abbreviated to “Bob”) requires precise representation of the mention attributes and contexts across documents. Text based similarity measures alone are unable to capture the contextual and semantic similarities across documents and suffer from *scalability* issues.

High Quality Entity Disambiguation. The process of disambiguating and linking a discovered entity to an entry already present in a knowledge base, a variant of the record-linking task, involves the semantic relatedness, situational context and feature

similarities between the current entity and the candidates entities in the KB. Procedures for correct mapping of surface strings to unique entities based on context coherence and syntactic similarity suffer from the lack of information propagation across documents leading to performance degradation in the face of noise and linguistic diversity.

Incompleteness or Long Tail of Entities. The dynamic nature of information generation entails the emergence of new entities and/or relationships for already known entities. The presence of sparsely represented entities or novel entities currently absent from knowledge repositories aggravates the performance of entity co-reference resolution and linking strategies based on rules or prior knowledge. Precise disambiguation of such new or emerging entities poses significant challenges.

Information Veracity. Although Web documents such as news articles, forums, blogs, etc., provide rich sources of information, they inherently suffer from noise and more importantly from possibly biased, misrepresented, or non-credible statements – severely limiting the purpose and applicability of knowledge repositories. Human curated domains such as Wikipedia partially solve the problem, but fail to represent all possible entities and relationships. Hence, there is an immense need for automated approaches to assess information credibility for detecting possible candidate facts as discrepant and/or biased. Existing strategies adopting bigram language models and user activity based features exhibit limited accuracy and tend to be limited to specific domains.

Succinct Representation of Knowledge Bases. The real-time characteristics of Web applications harnessing knowledge bases requires efficient loading and query processing performance of such repositories. Knowledge bases, in the form of labeled graph structures, tend to optimize storage, querying, and I/O costs. Existing frameworks for storing and indexing knowledge bases by RDF engines or column stores (with various compression techniques) ignore input data distribution and data locality (semantic similarity), thus are expensive in terms of computation and storage.

1.3 Contributions

This dissertation addresses the aforementioned challenges in the domain of knowledge acquisition and management from natural language texts, thus advancing the state-of-the-art by proposing the following novel methodologies.

• **CROCS: Robust entity co-reference resolution across documents –** CROCS provides an improved framework for cross-document co-reference resolution geared for Web scale and “long tail” entity mentions. The performance enhancement stems from the construction of summary snippets to capture entity contexts from documents, use of external features for enrichment, and entity context similarity computation,

judiciously combined within a hierarchical clustering architecture (Chapter 3). Additionally, robustness and correctness for sparsely represented entities is achieved by selective feature inclusion and sampling strategies. The proposed method was published in TACL 2015 [Dutta and Weikum, 2015a].

• **C3EL: Framework for joint entity co-reference resolution and linking –** Although, interactions between co-reference resolution and entity disambiguation have been shown to be beneficial, such frameworks do not exist for tackling the problem across documents. C3EL presents a novel framework for jointly tackling CCR and NEL for extending the coverage of entity-centric information available. In this context, we propose iterative steps of co-reference resolution and disambiguation to enable global entity context propagation for improved performance on both tasks (Chapter 4). The adoption of modules from the CROCS approach and external entity linking systems, coupled with link validation step and information feedback, achieves a higher accuracy for both the tasks compared to existing methods. C3EL demonstrates significant gains, especially in detecting out-of-knowledge base entities. The work was presented at EMNLP 2015 [Dutta and Weikum, 2015b].

• **Credibility analysis of texts with limited information –** This work proposes a credibility framework utilizing language, user, and text sentiment based features for constructing consistency features to score and classify the veracity of information present in natural language texts. In this context, our method uses limited information in contrast to existing approaches employing large number of features (difficult to obtain in real-life scenarios), to deliver better classification accuracy and domain-independence with limited re-training (Chapter 5). Furthermore, possible inconsistencies present in the texts are extracted as evidence for the non-credible classification decision, a novel feature in the field of credibility analysis research. The results were presented at ECML-PKDD 2016 [Mukherjee *et al.*, 2016].

• **KOGNAC: Efficient encoding of knowledge bases –** Entity attributes and relationships present in knowledge bases are typically represented as graphs and are generally stored and queried using RDF engines. For efficient storage and computation, modern RDF engines and column stores encode entity strings by assigning numerical identifiers based on appearance order or hash based methods. KOGNAC proposes a new encoding strategy for improving the indexing and storage requirements, query performance, and I/O scan costs in RDF engines (Chapter 6). The mapping of entity strings to identifiers based on approximate frequency estimation for assigning smaller identifiers to common entities enables better compression in the framework. Further, the use of ontological distance based entity semantic similarity measure enhances the index data locality for improved join query runtime. Integration of the proposed approach with state-of-the-art RDF engines is observed to significantly enhance query performance for large knowledge bases. The KOGNAC framework was presented at IJCAI 2016 [Urbani *et al.*, 2016].

1.4 Organization

The remainder of this dissertation is organized as follows. Chapter 2 introduces the related tasks and problems pertaining to entity co-reference resolution and linking, credibility analysis of textual snippets, and storage of knowledge bases for *knowledge management of named entities from texts*. It further discusses state-of-the-art approaches in this domain and their limitations, which we aim to alleviate in this dissertation. Chapter 3 presents an efficient algorithm to perform cross-document entity co-reference resolution at Web-scale for robustly dealing with sparsely represented or emerging entities. Chapter 4 proposes a novel joint framework for interleaved entity co-reference resolution and linking with iterative knowledge propagation, specifically to cope with out-of-knowledge base entities. Chapter 5 discusses language and user based consistency models to compute the credibility of text snippets, for enriching the quality of knowledge repositories. Chapter 6 puts forth an intelligent encoding strategy, based on term frequency estimation and semantic similarity, for RDF style storage and query performance efficiency of knowledge bases. Chapter 7 finally concludes the dissertation summarizing the contributions and possible directions of future work, followed by a list of references to existing methodologies.

2 Background & Related Work

This chapter provides an overview of state-of-the-art approaches in Information Extraction for automated knowledge acquisition from natural language texts.

2.1 Named Entity Recognition and Classification

Identifying information units such as persons, locations, organizations, events, and numeric expressions from unstructured texts forms the basis of entity discovery and relation extraction and is referred to as the task of *Named Entity Recognition* (NER). Automated approaches involving the detection of single or multi-word expressions denoting a real world entity for NER across vast heterogeneous digital data is thus an essential task for Information Extraction.

Classical methods for NER relied majorly on hand-crafted rules based on linguistic features from words, like punctuations, capitalizations, morphological representations, dictionary based pattern matches, and grammatical rules [Mikheev *et al.*, 1999; Cohen and Sarawagi, 2004]. The use of document level term statistics [Thielen, 1995; Da Silva *et al.*, 2004] to detect mention boundaries pertaining to named entities was also proposed across various domains and languages. The emergence of major scientific events and communities such as MUC-6, ACE, CoNLL, LREC, etc., have provided impetus to the expanse of research on natural language processing (NLP).

Linguistic rules obtained from dependency parser and multi-sieve strategies [Lee *et al.*, 2011] have been proposed to efficiently tackle the problem of NER. A typical approach in unsupervised methodologies involves clustering of named entities based on context and semantic patterns using hyponyms and hypernyms [Evans, 2003] from lexical resources such as WordNet. Semi-supervised techniques like *bootstrapping* involve the use of domain specific seed words to initiate the learning process. Regular expression based patterns and syntactic relations along with feature inclusion from external repositories such as Wikipedia [Cucchiarelli and Velardi, 2001] and user generated

forums and blogs [Liu *et al.*, 2011] have been shown to provide accurate contextual evidences in text to detect entity mentions. Such semi-supervised learning methods aimed to combine sparse training data with large collections of additional unlabeled data (e.g., unlabeled Web pages). With the growth in human-annotated corpora, supervised learning techniques such as Hidden Markov Models (HMM) [Bikel *et al.*, 1997], Decision Trees [Sekine and Nobata, 2004], Support Vector Machines (SVM) [Asahara and Matsumoto, 2003], and Conditional Random Fields (CRF) [McCallum and Li, 2003] were used for NER by learning disambiguation rules based on discriminative feature spaces. Another fundamental building block from machine learning involve models for joint segmentation and labeling. Semi-Markov structured classifier for NER and subsequent normalization using semantic indexing was proposed for joint NER and normalization of entities [Leaman and Lu, 2016]. A comprehensive survey on NER techniques and their evaluations can be found in [Nadeau and Sekine, 2007].

The use of entity semantic classes to identify entities via multi-label classification for unstructured text was proposed in [Ling and Weld, 2012]. Recent methods explored the viability of neural network embeddings, leveraging lexicon based information to obtain state-of-the-art results for the task of NER [Passos *et al.*, 2014]. The use of conditional random fields (CRF) and long short term memory (LSTM) for neural network embedding based architecture was put forth in [Lample *et al.*, 2016]. Ranking approaches to combine entity recognition and their linking to KB by merging candidate mentions from both NER and linking have been shown to provide significant error reduction [Sil and Yates, 2013]. A probabilistic graphical model based approach to jointly perform NER and disambiguation capturing mention span and KB mapping was presented in [Nguyen *et al.*, 2016].

Named Entity Classification (NEC), a closely related task, involves the tagging of discovered entities into rigid designated object types such as person, date, location, etc., based on predominant entity type classes. Each of the types can further be sub-divided into fine-grained subtypes, e.g., location can be further categorized as city, state, and country [Fleischman and Hovy, 2002]. Clustering of discovered terms based on context and attribute similarities [Da Silva *et al.*, 2004; Etzioni *et al.*, 2005], along with dictionary based approaches have been proposed to provide *POS tags* to the entities, thereby allowing for separate downstream processing [Ratinov and Roth, 2009]. An active research issue in NLP is to extend this line of analysis into semantic role labeling (SRL) [Moschitti *et al.*, 2008] for mapping sentences to knowledge representations. A hierarchical multi-type classification of named entities based on gazetteer features and joint evidence was presented by [Yosef *et al.*, 2012]. To handle new or emerging entities, discovering and semantically typing such entities based on an integer linear program formulation was shown to achieve high performance [Nakashole *et al.*, 2013]. A collection of recent NER and NEC methods and their multi-lingual applications can be found in [Sekine and Ranchhod, 2016].

2.2 Co-Reference Resolution

Identifying and cross-linking named entities in Web contents is at the heart of information acquisition for knowledge-enhanced search, recommendations, and analytics. Among various IE tasks, extracting actionable intelligence from an ever-increasing amount of data depends critically upon Coreference Resolution (CR) for identifying noun phrases occurring within a document that refer to the same real-world entity. This enables the consolidation of information pertaining to a particular entity for effective knowledge base population.

Traditional CR methods are based on rules or supervised learning using different kinds of linguistic features like syntactic paths between mentions, the distance between mentions, and their semantic compatibility as derived from co-occurrences in news and Web corpora [Haghighi and Klein, 2009; Lee *et al.*, 2013], for identifying the best antecedent (preceding name or phrase) for a given mention (name, phrase, or pronoun). To harness context features, first-order probabilistic models over mention clusters were proposed in [Culotta *et al.*, 2007]. The use of hierarchical Dirichlet process to obtain the referent mentions for unsupervised CR approaches was presented by [Haghighi and Klein, 2007], which was later extended to a fully generative Bayesian model [Ng, 2008]. Syntactic features derived from deep parsing of sentences and noun group parsing, with semantic features obtained by mapping mentions to background knowledge resources such as Wikipedia and YAGO were used for CR in [Rahman and Ng, 2011a; Ratinov and Roth, 2012] to identify entity candidates. Recent methods adopted the paradigm of multi-phase sieves, applying a cascade of rules to narrow down the choice of antecedents for a mention [Haghighi and Klein, 2009; Raghunathan *et al.*, 2010]. Cluster-ranking and ensemble based methods [Rahman and Ng, 2011b] extended this paradigm for connecting mentions with a preceding mention cluster by incrementally expanding groups of mentions, and exploiting relatedness features derived from semantic types and Wikipedia category names. The closely related task of person name disambiguation deals with only person names, title, nicknames, and surface forms variations [Chen and Martin, 2007]. An overview of CR methods was given by [Clark and González-Brenes, 2008; Ng, 2010].

The task of CR, when extended to process multiple documents across corpora is known as *Cross-Document Co-reference Resolution* (CCR), and involves the computation of equivalence classes of entity mentions across documents. CCR can essentially be viewed as a clustering problem, using a similarity metric between mention features. However, standard clustering lacks awareness of entity contexts and interdependencies and suffer from high computational complexity. Early work on CCR such as [Bagga and Baldwin, 1998; Gooi and Allan, 2004] used IR-style similarity measures (tf-idf cosine, KL divergence, etc.) over features like the ones used in intra-document CR. Baron and Freedman [Baron and Freedman, 2008] proposed a CCR method involving full clustering coupled with statistical learning of parameters. Several CCR methods

have harnessed co-occurring entity mentions, especially for the task of disambiguating person names [Mann and Yarowsky, 2003; Niu *et al.*, 2004; Chen and Martin, 2007; Baron and Freedman, 2008]. However, these methods do not utilize knowledge bases, but use information extraction (IE) methods on the input corpus; thus facing substantial noise due to quality variance on stylistically diverse documents like Web articles.

More recent works such as [Singh *et al.*, 2010] are based on probabilistic graphical models for jointly learning the mappings of all mentions into equivalence classes. A large scale distributed inference mechanism based on Markov chain Monte Carlo method to introduce sub-entity and super-entity variables representing clusters for distributing or collecting entities was presented by [Singh *et al.*, 2011]. To obtain richer semantic features, additional knowledge resources such as Wikipedia and FrameNet corpora have also been considered in the context of CCR [Suchanek *et al.*, 2007; Baker, 2012]. In general, CCR methods tend to "featurize" entities by context patterns around it and compute the similarity between entity pairs. The entity mentions are then grouped into clusters such that the similarities among members within the same cluster are maximal while similarities across data members from different clusters are minimal. An orthogonal approach on CCR based on the use of latent features derived from matrix factorizations combined with parameter-free graph clustering was recently proposed in [Ngomo *et al.*, 2014]. The use of other knowledge contexts such as *cross-media* features for CCR was also shown to be efficient in practice [Zhang *et al.*, 2015]. A wide range of features capturing evidence for both entity merging and arguing against merging, can significantly improve machine learning-based CCR as reported in [Mayfield *et al.*, 2009]. The use of external features to develop global learning model to enhance CCR performance was presented in [Ng, 2016]. An interesting review and evaluation of existing CCR techniques has been presented in [Beheshti *et al.*, 2016].

2.3 Fact Extraction

A key process in the creation of knowledge bases involves models and algorithms for transforming Web pages, text sources, and other unstructured data into explicit facts and relationships across the discovered entities as unary, binary, or higher arity relations. In general, *fact extraction* aims at extraction of triples in the form of $\langle subject, predicate, object \rangle$ (SPO) denoting unary/binary relations. As such, methods for harvesting relational facts can be categorized into different paradigms such as rule-based (for semi-structured data), pattern and statistics based (for natural language texts), or joint inferencing based learning for combining best of both worlds.

Rule-based strategies exploit regularities in the structures of digital resources, centered around regular expressions over DOM trees [Kushmerick *et al.*, 1997; Sahuguet and Azavant, 2001; Arasu and Garcia-Molina, 2003]. Textual patterns based on natural language rules such as the occurrence of noun phrases as subjects or objects related

to verb phrases have been used for clustering approaches to identify relationships among entities [Hindle, 1990]. Part-of-Speech enriched lexico-syntactic patterns like Hearst patterns [Hearst, 1992] and their extensions capturing predefined relation types have been shown to achieve high precision for obtaining facts from free text. The *duality of facts and patterns* [Brin, 1998] iteratively enables the automatic identification of linguistic patterns and subsequent detection of fact candidates, based on a small start set of *seed facts*. The extension of such models by employing machine learning techniques such as Hidden Markov Model (HMM) and classifiers to arrive at good extraction rules have been pursued by systems such as RoadRunner [Crescenzi and Mecca, 2004] and SEAL [Wang and Cohen, 2009]. The use of attribute-value pair templates from external knowledge sources such as Wikipedia infoboxes have also been proposed for fact extraction during construction of knowledge bases like DBPedia and YAGO, with careful noise elimination based on frequency or learning techniques [Weld *et al.*, 2008]. The use of distant entity labels enabling proper fact segmentation for fact extraction was presented in [Sutton and McCallum, 2004]. Under the ambit of OpenIE, the use of Wikipedia as data source for enhanced fact extraction was studied by [Wu and Weld, 2010]. Fact gathering in *declarative IE* combining the process of extraction and inferencing encapsulated fact using datalogs was also presented [Shen *et al.*, 2007]. Subsequently, query languages such as AQL [Reiss *et al.*, 2008] were developed for interfacing with such fact stores based on pattern matching and dictionary lookups. To manage the uncertainty of extracted fact by rule based systems, [Michelakis *et al.*, 2009] proposed the use of parametric exponential model to capture the interaction between compositional rules, while the refinement of text based rules using ranking approaches in the context of data provenance was presented by [Liu *et al.*, 2010].

Statistical analysis such as frequency, confidence, etc., for pattern co-occurrences, and consistency constraints for reducing errors, with the use of structured ontological sources to boost the scope and precision of pattern-based fact extraction techniques have also been proposed [Udrea *et al.*, 2007; Zhu *et al.*, 2009]. Extensions in this domain for the use of lexical dependency parsers, proximity, entity disambiguation, and incorporation of negative patterns [Suchanek *et al.*, 2006; Bunescu and Mooney, 2007] have been shown to enhance robustness and expressivity. Distant supervision approaches for relation extraction to tackle the lack of labeled data was explored in [Mintz *et al.*, 2009]. Scalable knowledge harvesting based on n-gram-itemset for obtaining richer patterns using constraint reasoning was depicted by [Nakashole *et al.*, 2011].

Amalgamation of precision oriented rule-based methods and recall oriented pattern-based approaches was proposed by [Lin and Pantel, 2001]. Pattern based methods were further augmented with joint reasoning by the use of Markov Logic Networks [Domingos *et al.*, 2007] and other relational learning methods for grounding rules against base facts and newly extracted candidate facts. This, in combination with patterns from existing knowledge bases enabled the learning of new facts, patterns, and constraints with high precision and efficiency [Suchanek, 2008]. The union of patterns, word dis-

ambiguation, and ontological reasoning was shown to produce high performance even in scenarios of unstructured documents [Suchanek *et al.*, 2009]. A tutorial to automatically construct and maintain a comprehensive knowledge base was presented in [Weikum and Theobald, 2010].

The paradigm of *Open IE* aims at the extraction of entity attributes expressed in verb form from Web pages, rather than based on predefined canonical relations. The use of POS tags to locate verbs and the use of dependency parsers [Mausam *et al.*, 2012] or language clauses [Corro and Gemulla, 2013] thereafter, allowed the identification of subjects and objects connected to the verb phrases. The use of nominal attributes for capturing patterns and ontologies was studied in [Yahya *et al.*, 2014]. The process of OpenIE enables the extraction of large number of meaningful facts, however suffer from noise due to the absence of consistency reasoning on the extracted data.

2.4 Fact Verification

The deluge of digital information available on the Web provides rich resources for extraction of entity attributes and relationships. However, the credibility of such sources and the extracted facts remains a major concern for assessing the quality of the obtained data. As such, the presence of noisy, biased, or deceptive claims tend to severely impact the performance of modern knowledge bases, and calls for efficient approaches to verify the extracted information. Traditional methods involving dedicated human fact-checkers or crowd-sourcing approaches based on cognitive heuristics such as reputation, endorsement, and consistency [Metzger and Flanagin, 2013] suffer from expensive computations, corroboration of multiple sources, and possible absence of contextual information. Hence, automated methods for characterizing the credibility of information have broadly considered classification strategies based on linguistic, user activity, and psychological features.

Distributional difference in the wordings between authentic and fake texts to learn latent topics and word-level features for training SVM classifiers was proposed in [Mihalcea and Strapparava, 2009; Ott *et al.*, 2011; Ott *et al.*, 2013]. Additionally, linguistic features such as *text sentiment* [Yoo and Gretzel, 2009], *readability score* [Hu *et al.*, 2012], *textual coherence* [Mihalcea and Strapparava, 2009], and rules based on *Probabilistic Context Free Grammar* (PCFG) [Feng *et al.*, 2012] have also been studied. Entity and user based features to capture background information like name, description, history, location, etc., [Lim *et al.*, 2010; Wang *et al.*, 2011; Mukherjee *et al.*, 2012; Mukherjee *et al.*, 2013b; Rahman *et al.*, 2015] were employed to train regression models on extracted features to classify facts as credible or deceptive. Similar works for fact checking on social media also considered ad-hoc features like extreme sentiments, user activity (number of posts, friends etc.), length, deviation from community mean, burstiness, and simple language features like content similarity, presence of literals,

numerals, capitalizations, and POS tags, for learning classification models. Supervised approaches using user sentiment, POS tags, and entity types from sentences to train random forest classifiers obtaining a ranking score for credibility analysis was studied by [Hassan *et al.*, 2015]. Wu et al. [Wu *et al.*, 2014] provided an interesting approach for fact checking and completion based on reverse engineering by "perturbing" and re-formulation the extracted facts as queries to observe their effects on the obtained results. The effect of majority opinion and their propagational pattern for analyzing the perception of truthfulness of facts in social media was studied in [Li and Sakamoto, 2015]. However, such methods are rarely generalizable and provide no concrete evidence as to why a relation is deemed non-credible.

2.5 Entity Disambiguation and Linking

The dynamic nature of digital contents available on the Web leads to the discovery of new entities or the emergence of new relationships about already known entities, that needs to be extracted and updated in the existing knowledge bases (KB) for precisely modeling the real world. A crucial challenge in this regard is to distinguish between different entities having the same surface forms (e.g., multiple persons with the same name "John") and/or between same entities represented in different forms (e.g., acronyms, aliases, etc.) – referred to as the task of *Named Entity Disambiguation* (NED). The presence of noise and context based word-sense ambiguity further complicates the proper alignment of mentions to real-life entities. For example, the mention "apple" might refer to either the fruit or the company depending on the textual context.

The related task of *Wikification* involves the disambiguation of all words and phrases having a corresponding Wikipedia article, catering to a broader scope for disambiguating common nouns and phrases onto concepts, while NED restricts itself to mentions denoting individual entities. Although Wikification enables more words of a text to be associated with their meaning, useful for downstream applications and joint inference methods, the absence of all word senses from Wikipedia limits its applicability. The task of *Named Entity Linking* (NEL) offers a broader scope than NED, by not only disambiguating and mapping entity mentions to canonical correct entries in a KB if present, but also dealing with the case when there is no entry in the reference KB and marking such entities as absent (i.e., link to $null$) [Alhelbawy and Gaizauskas, 2014].

Naïve approaches involved defining a similarity measure between the surface forms of entity strings and possible target entities by using traditional measures such as Jaccard similarity, edit distance, and Jaro-Winkler distance, operating on flexible notions of token representation (e.g., stemmed words, POS-tagged words, N-grams, etc.) [Koudas *et al.*, 2006] and contexts. Machine learning techniques were employed to consider the joint disambiguation of multiple entities with mutual re-inforcement [Singla and Domingos, 2006]. However, such strategies not only fail to capture semantic similarities

between mentions, but also disregards the presence of existing knowledge sources.

Initial works using semantic similarity for NEL proposed the disambiguation of named entities to Wikipedia articles using textual context similarity, word-category association, and HTML redirect links [Bunescu and Paşca, 2006; Cucerzan, 2007]. The incorporation of disambiguation confidence along with the notion of semantic similarity of entities to corresponding Wikipedia pages was studied in [Milne and Witten, 2008]. A supervised learning approach for obtaining a similarity prior and modeling the pairwise coherence of entity candidates as a probabilistic factor graph (with entity-entity relatedness measure) was used for heuristically solving an integer linear program (ILP) formulation in [Kulkarni *et al.*, 2009]. External lexical and encyclopedic resources were also used to capture semantic similarity across candidate entities [Poesio *et al.*, 2008]. Further, ontological distance similarity based NEL have also been studied [Hassell *et al.*, 2006]. The combination of WordNet and Wikipedia to leverage both semantic knowledge and taxonomy was proposed in [Shen *et al.*, 2012]. A tabular comparison of several NEL methodologies and setup can be found in [Hoffart, 2015].

Extending the context and semantic similarity between mentions to incorporate the global coherence within a document in an iterative manner, [Ratinov *et al.*, 2011] disambiguated mentions independently for an optimization step taking into account the semantic relatedness, disambiguation confidence, and link-likelihood of a phrase to train a classifier. An ensemble of classifiers to improve NEL accuracy was shown to exhibit high accuracy in [Zuo *et al.*, 2014]. A scalable Gibbs sampling based approach for probabilistic entity linking was also proposed [Houlsby and Ciaramita, 2014]. A novel phrase-unigram language model to efficiently capture high-order dependencies in lexical features from noisy free-text documents was shown to significantly improve "long tail" entity linking [Jin *et al.*, 2014].

Graph models have also been proposed for collective approaches to model entity inter-dependencies for disambiguation based on relationship contexts. The use of local dependency and semantic relatedness between mentions and the candidate entities to construct a referent graph for collective inference of the correct reference node in the graph was presented in [Han *et al.*, 2011]. The problem of NEL was cast as dense subgraph finding, and the use of strongly interconnected components based on Wikipedia link structures and content similarities to obtain a coherence weighted graph for candidate entity generation was studied by [Hoffart *et al.*, 2011]. Representing all possible entity candidates as nodes and their initial association confidences as edges, the use of PageRank algorithm for candidate ranking was recently shown to be quite effective [Alhelbawy and Gaizauskas, 2014] in practice. In fact, collective ranking and collaborative ranking among candidate entities to ensure semantic similarity for NEL was recently proposed [Chen and Ji, 2011; Zhao *et al.*, 2016]. Re-ranking of candidate Wikipedia entities based on mutual relationship between the entities of a document was incorporated in SemLinker to offer a more robust disambiguation [Charton *et al.*,

2014]. A detailed overview of current NEL methods, challenges and their evaluations was presented in [Hachey *et al.*, 2013; Shen *et al.*, 2015].

2.6 Knowledge Base Construction

Knowledge Bases (KB) provide a coherent integration of knowledge extracted from diverse Web sources into structured machine-readable format for varied applications, and has received tremendous interest with the emergence of repositories such as YAGO, NELL, DBPedia, Google Knowledge Graph, etc., constructed from Wikipedia, Web, and/or specific domains. The process of *Knowledge Base Construction* (KBC) incorporates procedures for entity and mention discovery, fact and relationship extraction, and veracity of information collected, along with updates pertaining to the emergence of new entities and facts.

Traditional knowledge sources such as *Cyc* [Lenat, 1995] and *WordNet* [Fellbaum, 1998] were created by manually compiling lexical and word-sense information, yielding high-quality repositories on intensional knowledge containing general concepts, semantic classes, and relationships within a taxonomical structure. However, such resources were labor intensive and lacked extensional knowledge about individual entities and their relationships. Construction of KB using co-operative strategies across documents across domains was initially proposed in [Mansell, 2002]. A moderated approach to factor in expert opinion into knowledge bases was presented in [van Ast *et al.*, 2004].

The emergence of automatically constructed KB of facts about named entities, their semantic classes, and their mutual relationships containing millions of entities and billions of facts, forms the backbone of modern Web of Linked Data [Heath and Bizer, 2011]. Initial works for enhanced extraction and knowledge management activity cycle was presented by [Sterner, 2000; Matthee and Viktor, 2001]. Automated approaches adopt diverse methodologies such as: use of semi-structured Wikipedia infoboxes to extract entity attributes and WordNet for taxonomy structure (e.g., YAGO [Suchanek *et al.*, 2007]), statistical learning and inferencing based on linguistic and dependency path features (e.g., DeepDive [Niu *et al.*, 2012]), coupled learning (e.g., NELL [Lao *et al.*, 2011]), probabilistic approaches using distributed inference algorithm with graph construction and hypergraph sampling [Niepert *et al.*, 2012], or a combination of natural language processing tools (e.g., KELVIN [McNamee *et al.*, 2013]). A brief survey of recent algorithms and trends in knowledge base construction can be found in [Ji and Grishman, 2011; Suchanek *et al.*, 2012].

Knowledge bases are most easily represented as a labeled graph structure, where the entities denote vertices and the edges model the relationships between the entities. KBs are generally stored as $\langle subject, predicate, object \rangle$ (SPO) triples according to the *Resource Description Framework* (RDF) data model [Klyne and Carroll, 2006] with SPARQL

based query interface. RDF engines such as RDF-3X [Neumann and Weikum, 2008] and TriAD [Gurajada *et al.*, 2014] then implement the index structures and dictionaries to represent the graph. However, existing RDF engines focus on individual triples rather than providing a graph oriented representation of the data, and thus the implementation of core graph-based query tasks are time-inefficient. To this end, several general purpose graph databases and column-stores such as MonetDB [Sidirourgos *et al.*, 2008] and Neo4j [Robinson *et al.*, 2015], have been proposed to represent the entities and relationships as attribute labeled multigraphs in native data structures. An introductory survey of RDF engines and graph databases with performance comparisons on benchmark queries has been presented in [Abreu *et al.*, 2013].

RDF Representation: The storage of RDF terms, representing the KB entities and relationships, in their original format is both space and compute inefficient since these are typically long strings (e.g., URIs). As such, all existing approaches *encode* the URIs and literals in a judicious manner, typically by mapping them to fixed-length integer IDs, with the original strings retrieved only during execution.

The fastest method to perform such dictionary encoding uses hashcodes of the terms as numerical IDs, but suffer from hash collisions wherein different terms receive the same ID. Systems adopting hash-based approaches (e.g., 4Store [Harris *et al.*, 2009] and Stardog (goo.gl/zSalQl) address this problem by calculating lengthy hash codes with cryptographic functions like SHA-1 to practically remove the possibility of collisions. Unfortunately, this strategy makes the encoding process rather slow since cryptographic functions are expensive to compute and the resulting IDs are unnecessarily long.

Most of the current large-scale RDF engines follow a different approach where the IDs are assigned with counters. TripleBit [Yuan *et al.*, 2013], for example, splits the dictionary table into two tables, one for the predicates and the other for the objects and subjects, and assigns new IDs when new terms appear in the input. Sesame [Broekstra *et al.*, 2002], another very popular RDF store, encodes all the namespaces in a separate data structure and encodes the terms using a 32 bit counter. RDF-3X [Neumann and Weikum, 2008] – perhaps the most scalable among current RDF engines – adopts a different solution by sorting the terms and stores the mapping using a traditional disk-based B-Tree. Recent methods like TriAD [Gurajada *et al.*, 2014] employ graph partitioning techniques for clustering together related terms for term ID assignment. Scalable parallel approaches using MapReduce for dictionary encoding have also been proposed [Urbani *et al.*, 2013].

In the remainder of this dissertation, additional background concepts and existing approaches specific to the different stages of knowledge harvesting have also been presented within the ambit of the pertaining chapters.

3 Efficient Cross-document Co-reference Resolution

Acquisition of relevant and associated information pertaining to an entity involves the task of *Cross-Document Co-Reference Resolution* (CCR) for precise identification of mentions, present in texts within a corpus, that refer to the same real-life entity, enabling the construction of semantic resources such as Knowledge Bases (KB). However, the enormous amount of documents involved in Web-scale corpora imposes a prohibitive computational complexity on existing CCR approaches.

This chapter presents an efficient and scalable approach for CCR across documents within huge collections from diverse sources, thereby handling Web-scale corpora. It harnesses enriched mention contexts by considering co-occurring mentions as well as distant features from external knowledge sources. The proposed method adopts a sampling-based hierarchical clustering procedure to provide an efficient CCR approach alleviating the above problem. Experimental results on benchmark datasets show our proposed algorithm to outperform state-of-the-art methodologies for CCR.

3.1 Introduction

Motivation. Knowledge bases such as Google Knowledge Graph, DBPedia, and YAGO to name a few, provide rich semantic resources containing huge collections of entities such as people, places, celebrities, organizations, movies, and events, along with their properties and inter-relationships. The construction of such huge repositories involves information extraction from multiple documents across varied sources for individual entities forming the knowledge base entries. This enables efficient performance for applications involving entity search, semantic search, question answering, and others.

Perhaps the most important value-adding component in this setting is provided by the *recognition* and *disambiguation* of named entities in Web and user contents to aggregate meaningful facts pertaining to the corresponding entity only, sifting out other entities with same or similar surface forms.

The initial stage of Information Extraction deals with *Named Entity Recognition* (NER) for identification of entity *mentions* within a text and their classification into coarse-grained semantic types (person, location, etc.) [Finkel *et al.*, 2005; Nadeau and Sekine, 2007; Ratinov and Roth, 2009]. It involves segmentation of token sequences to obtain mention boundaries, and subsequent mapping of relevant token spans to predefined entity categories. For example, NER on the text: `Einstein won the Nobel Prize.` identifies the mentions "Einstein" and "Nobel Prize" and marks them as *person* and *misc* type, respectively.

Subsequently, the task of *co-reference resolution (CR)* (see, e.g., [Haghighi and Klein, 2009; Ng, 2010; Lee *et al.*, 2013]) identifies all mentions in a given text that refer to the same entity, including anaphors such as "the president's wife", "the first lady", or "she". This task when extended to process an entire corpus is then known as *cross-document co-reference resolution (CCR)* [Bagga and Baldwin, 1998; Singh *et al.*, 2011]. It takes as input a set of documents with mentions, and computes as output a set of equivalence classes over the entity mentions. A related but different problem of *Named Entity Disambiguation* (NED) (see, e.g., [Cucerzan, 2007; Milne and Witten, 2008; Cornolti *et al.*, 2013]) maps a mention string (e.g., a person name like "Bolt" or a noun phrase like "lightning bolt") onto its proper entity if present in a KB (e.g., the sprinter Usain Bolt). Observe that, CR does not involve mapping the mentions to the entities of a KB. Unlike NED, CCR can be used to identify long tail or emerging entities that are not currently captured in the KB or are merely present in very sparse form. However, the task is plagued by *popularity-based* priors for frequent and well-known entities.

The CCR task – computing equivalence classes across documents – can essentially be cast as a clustering problem using a similarity metric between mentions with contextual and semantic features characterizing an entity within its surrounding text. However, standard clustering (e.g., k-means or EM variants, CLUTO, etc.) lacks awareness of the transitivity of co-reference equivalence classes and suffers from knowledge requirement of cluster model dimensions. Note that CCR cannot be addressed by simply applying existing local CR techniques to a "super-document" concatenating all documents in the corpus. As, within a document, identical mentions typically refer to the same entity, while in different documents, identical mentions can have different meanings. Although a cross-document view gives the opportunity to spot joint cues from different contexts for an entity, documents vary in their styles of referring to entities and merely combining the local co-reference chains into a super-group might lead to substantial noise introduction. In addition, CR methods are not designed for scaling to huge "super-documents" comprising millions of web pages or news articles.

Problem Statement. In this chapter, we aim to overcome the above limitations by proposing a CCR method that makes rich use of distant KB features, considers transitivity, and is not only computationally efficient for Web-scale corpora but also provides accurate detection of co-referring mentions and identification of long tail entities.

3.1.1 Approach and Contributions

In this chapter, we efficiently tackle the CCR problem by proposing the $CROCS$ *(*CRO*ss-document* C*o-reference reS*$olution$*)* framework adopting *unsupervised hierarchical clustering* by repeated bisection using spectral clustering or graph partitioning. $CROCS$ considers *enhanced mention context* from co-occurring mentions along with rich features from external knowledge bases and provides *scalability* by using a transitivity-aware sampling-based hierarchical clustering approach.

To this end, the major novel components of $CROCS$ involve,

– $CROCS$ in addition to textual context, harnesses semantic features derived from external KBs by constructing a notion of *semantic summaries* (*semsum*) for representing intra-document co-reference chains. Further, to incorporate enriched KB labels as features for the co-referring mentions, we also consider co-occurring mentions belonging to other entities and utilize their features to expand the scope of mention context involved in the CCR procedure.
For example, consider the text: `Michelle lived in the White House and backed Barack despite his affairs.` containing 3 mention groups: {"Michelle"}, {"Barack"}, and {"White House"}. Merely obtaining distant KB features for the first mention group, the sparse information leads to high ambiguity, e.g., may refer to the German singer Michelle. But by also obtaining features from KB for "White House" (co-occurring mention), we obtain much stronger cues towards the correct solution.

– $CROCS$ adopts a bisection based clustering method and invokes it repeatedly in a top-down hierarchical procedure with an information-theoretic stopping criterion for cluster splitting. We escape the quadratic run-time complexity for pair-wise similarity computations by using a *sampling technique* for the spectral eigenspace decomposition or for graph partitioning, inspired by the recent work of [Krishnamurty *et al.*, 2012; Wauthier *et al.*, 2012] on active clustering techniques. Similarity computations between mention groups are performed on-demand for dynamically selected samples, alleviating the scalability issues in state-of-the-art CCR techniques.

The above features are intelligently combined within the framework to obtain equivalent classes of co-referring mentions across documents within the input corpus with high accuracy. In a nutshell, the novel contributions of this chapter are as follows:

1. $CROCS$ framework for CCR using sample-based spectral clustering or graph partitioning embedded in a hierarchical bisection process (Section 3.3);
2. *semsum*, a method incorporating distant KB features by also considering the coupling between co-occurring mentions in different co-reference chains for *knowledge enrichment* to capture global mention context (Section 3.5);
3. *context similarity based active hierarchical clustering* for scalability and unsupervised selection of model parameters (Sections 3.6 and 3.7); and
4. experimental evaluation with benchmark corpora demonstrating substantial gains over prior methods in accuracy and run-time (Section 3.8).

3.2 Related Work

Co-reference Resolution (CR): CR methods, for co-references within a document, are generally based on rules or supervised learning using different kinds of linguistic features like syntactic paths between mentions, and their semantic compatibility as derived from co-occurrences in news and Web corpora [Haghighi and Klein, 2009; Lee *et al.*, 2013]. Existing intra-document CR methods combine syntactic with semantic features for identifying the best antecedent for a given mention. Recent methods adopt the paradigm of *multi-phase sieves*, applying a cascade of rules to narrow down the choice of antecedents for a mention (e.g., [Haghighi and Klein, 2009; Raghunathan *et al.*, 2010; Ratinov and Roth, 2012]). The cluster-ranking family of methods (e.g., [Rahman and Ng, 2011b]) extends this paradigm for connecting mentions with a cluster of preceding mentions. Some methods additionally use distant labels from knowledge bases and incrementally expand groups of mentions by exploiting relatedness features derived from Wikipedia categories [Rahman and Ng, 2011a; Ratinov and Roth, 2012].

Distant Knowledge Labels for CR: To obtain semantic features, additional knowledge resources such as Wikipedia, YAGO ontology, and FrameNet corpus have been considered [Suchanek *et al.*, 2007; Rahman and Ng, 2011a; Baker, 2012]. To identify the entity candidate(s) that a mention (group) should use for distant supervision, CR methods such as [Ratinov and Roth, 2012; Lee *et al.*, 2013] use matching heuristics based on the given mention alone to identify a single entity or all matching entities with confidence above some threshold. Zheng et al. [Zheng *et al.*, 2013] generalizes this by maintaining a ranked list of entities for distant labeling, as mention groups are updated. Unlike $CROCS$, prior methods utilize only the candidates for the given mention (group) itself while distant knowledge features for co-occurring mentions are not considered.

Cross-Document CR (CCR): Early works [Gooi and Allan, 2004] on CCR, introduced by [Bagga and Baldwin, 1998], used IR-style similarity measures (tf$\times$idf cosine [Salton and Buckley, 1988], KL divergence [Kullback and Leibler, 1951; Kullback, 1987], etc.) on features, similar to intra-document CR. Recent works such as [Culotta *et al.*, 2007; Singh *et al.*, 2010; Singh *et al.*, 2011] are based on probabilistic graphical models for jointly learning the mappings of all mentions into equivalence classes. The features for this learning task are essentially like the ones in local CR. A more light-weight online method by [Rao *et al.*, 2010] performs well on large benchmark corpora. It is based on a streaming clustering algorithm, which incrementally adds mentions to clusters or merges mention groups into single clusters, and has linear time complexity; albeit with inferior clustering quality compared to advanced methods like spectral clustering. Several CCR methods have harnessed co-occurring entity mentions, especially for the task of disambiguating person names [Mann and Yarowsky, 2003; Niu *et al.*, 2004]. Probabilistic graphical models like Markov Logic networks [Richardson and Domingos, 2006; Domingos *et al.*, 2007; Domingos and Lowd, 2009] or factor graphs [Loeliger, 2004; Koller and Friedman, 2009] take into consideration constraints such as transitivity,

while spectral clustering methods [von Luxburg, 2007] implicitly consider transitivity in the underlying eigenspace decomposition, but suffer from high computational complexity. In particular, all methods need to precompute features for the data points and similarity values between all pairs of data points. The latter may be alleviated by pruning heuristics, but only at the risk of degrading the output quality. A brief survey on CCR methods is presented in [Beheshti *et al.*, 2016]. $CROCS$ provides a framework incorporating co-occurring mention context from texts, and external KB feature to reduce the effect of surface form variational noise and improve the accuracy of CCR.

Active Clustering: Clustering models and algorithms based on spectral decomposition (for identifying co-referring mentions in our case) was provided in [von Luxburg, 2007]. Approximation algorithms, based on the k-means technique and random projections, reducing the $O(n^3)$ time complexity to $O(k^3)+O(kn)$ where k is the number of clusters, were also proposed in [Yan *et al.*, 2009]. Graph partitioning provides an alternate approach to group together possibly co-referring mentions based on edge weights modeled by context similarity. A detailed study of related computational methods and applications have been presented in [Kernighan and Lin, 1970; Buluc *et al.*, 2013].

In CCR, the number of clusters (truly distinct entities) can be huge and typically unknown leading to scalability issues; hence [Shamir and Tishby, 2011; Krishnamurty *et al.*, 2012; Wauthier *et al.*, 2012] developed an active spectral clustering approach, wherein the expensive clustering step is based on small randomly selected data samples and other data points are merely "folded in". The term "active" refers to the active learning flavor of choosing the samples (notwithstanding that these methods mostly adopt uniform random sampling). $CROCS$ adopts this approach by improving the initial seed-selection procedure and also applies active clustering to graph partitioning.

3.3 CCR Computational Framework

The $CROCS$ model assumes an input set of text documents $D = \{d_1, d_2, \ldots\}$, with a markup of the entity mentions present in the documents, i.e., $M = \{m_{11}, m_{12}, \ldots, m_{21}, m_{22}, \ldots\}$, where $m_{ij} \in d_i$. As output, $CROCS$ computes an equivalence relation over M with equivalence classes C_l, where $C_l \cap C_n = \emptyset$ (for $l \neq n$) and $\bigcup_l C_l = M$, i.e., respecting the *exhaustive* and *mutual exclusion* properties. Since, the number of desired classes is apriori unknown – it needs to be determined by the algorithm.

The $CROCS$ framework majorly consists of *four* operational stages:

1. **Intra-document Co-Reference Resolution**: Given an input corpus of text documents, D with mentions M identified, we initially perform *intra*-document co-reference resolution using existing approaches, to obtain co-referent mention-chains (groups) within each input document (Section 3.4).
2. **Knowledge Enrichment:** For each of the local mention groups ($\{m_{ij}\}$) obtained

in the previous step, we combine the sentences within the text containing the mentions contributing to the group, $sentence(m_{ij})$. Analogously the sentences for co-occurring mentions of $\{m_{ij}\}$, present in $sentence(m_{ij})$ are also aggregated. We then extract *keywords* (for querying) to determine the best matching entity in an external KB and retrieve relevant distant features. We term this feature set corresponding to $\{m_{ij}\}$ as its *semantic summary* or *semsum* (Section 3.5).

3. **Similarity Computation:** We compute similarity scores between mention groups based on the features extracted from the *semsums* above. These are computed on-demand, and only for a sampled subset of mentions during clustering, thereby avoiding the quadratic computation cost (Section 3.6).

4. **Sampling-based Clustering:** We perform active spectral clustering or balanced graph partitioning, using the similarity metric among the mention groups, in a hierarchical fashion to compute the *cross*-document co-reference equivalence classes of mentions (Section 3.7).

We next describe the different computational modules of $CROCS$ in details.

3.4 Intra-Document Co-Reference Resolution

$CROCS$ initially pre-processes the input documents to cast them into plain text from native HTML format using standard tools like Boilerpipe (code.google.com/p/boilerpipe/) or jsoup (www.jsoup.org). It then uses the *Stanford CoreNLP* tool suite (nlp.stanford.edu/software/) to detect and mark the mentions and anaphors present in the text. The identified mentions are then tagged with coarse-grained lexical types (e.g., person, organization, location) by the Stanford NER Tagger [Finkel *et al.*, 2005]. This forms the input to the intra-document co-reference resolution (CR) step, wherein we use the state-of-the-art open-source CR tool based on multi-pass sieve algorithm from Stanford to compute the local co-reference mention chains [Raghunathan *et al.*, 2010; Lee *et al.*, 2011; Lee *et al.*, 2013]. The tagged texts and the local co-reference chains are then passed to the second stage.

Observe that, the local CR step may produce errors (e.g., incorrect chaining of mentions or omissions) which propagate to the later stages. Our experiments later show that $CROCS$ is robust and produces high-quality output even with moderate errors encountered during the local-CR stage. Later in Chapter 4, we discuss an approach to alleviate the effects of such errors.

3.5 Knowledge Enrichment

The *knowledge enrichment* phase starts with the local mention co-reference chains and mention type tags per document as input, and constructs the *semantic summary*

representing the context of each mention chain present within individual documents.

Formally, a mention m is a text string at a particular position within a document and belonging to a mention group $M(m)$ consisting of all equivalent mentions, with the same surface form (at different positions) or different strings (aliases, anaphoras, etc.). For a given mention group M, the *basic semsum* of M, $S_{basic}(M)$, is constructed by aggregating the document sentences containing a reference to the mentions m present in M. Formally,

$$S_{basic}(M) = \cup \{t \in sentence(m) | m \in M(m)\}$$

where t are extracted text tokens (words or phrases after stop word removal) and $sentence(m)$ is the sentence in which mention m occurs. Note that $S_{basic}(M)$ is a bag of tokens, as different mentions in $M(m)$ can obtain the same tokens or labels and there could be multiple occurrences of the same mention string in $M(m)$ anyway.

Similarly, tokens extracted from the sentences of *co-occurring mentions*, i.e., mentions present in $S_{basic}(M)$ other than those in the chain M, are also appended to the basic *semsum* of M to form the *extended scope*; and key-phrases are extracted for querying to an external knowledge base (KB).

For our text example in Section 3.1.1, assume that mention group (chains) phrases `{Michelle, she, first lady}` and `{the president's wife, first lady}` to have been obtained from different documents. To assess whether these two chains should be combined, i.e., they both refer to the same entity, we compute semantic features by tapping into knowledge bases (KB). Specifically, we harness labels and properties from Freebase (www.freebase.com) entries, for possible matching entities, to enrich the features of a mention group. The KB features then form a part of the *semantic summary* or *semsum* for each of the local mention group. Context derived from the constructed *semsum* are later used to compare different mention groups via a similarity measure (described in Section 3.6).

Prior works on CR (e.g., [Rahman and Ng, 2011a; Ratinov and Roth, 2012; Hajishirzi *et al.*, 2013]) and NED (e.g., [Cucerzan, 2007; Milne and Witten, 2008; Ratinov *et al.*, 2011; Hoffart *et al.*, 2011]) have considered such form of distant features. $CROCS$ extends these previous methods by also considering distant features for *co-occurring* mention groups, and not just the group at hand. We now introduce a general *framework for knowledge enrichment* in our CCR setting.

3.5.1 Enrichment Strategies

The strategies for knowledge enrichment (KE) involve decision making along the following dimensions:

– **Target:** items (single mentions, local mention groups, or global mention groups across documents) for which semantic features are obtained.
– **Source:** the resource from where semantic features are extracted. Existing methods consider a variety of choices: (i) input corpora, (ii) external text corpus, e.g., Wikipedia, and (iii) knowledge bases such as Freebase, DBPedia [Bizer *et al.*, 2009], or YAGO [Suchanek *et al.*, 2007].
– **Scope:** the neighborhood of the target considered for enrichment. It can either be restricted to the target itself or can consider co-occurring items (other mention groups connected to the target).
– **Match:** involves mapping the target to one or more relevant items in the source, and can involve simple name queries to full-fledged NED based on relevance or score confidence.

Existing methods generally consider individual mentions or local mention groups as target. Extended scopes like co-occurring entities based on automatic NER and IE techniques have been proposed [Mann and Yarowsky, 2003; Niu *et al.*, 2004; Chen and Martin, 2007; Baron and Freedman, 2008], but use only the input corpus as the enrichment source. Recent methods [Rahman and Ng, 2011a; Ratinov and Roth, 2012; Hajishirzi *et al.*, 2013; Zheng *et al.*, 2013] harness KBs, but consider only local mention groups. Also, these methods rely on high-quality NED for mapping mentions to KB entries. In contrast, $CROCS$ considers extended scopes that include mention groups along with co-occurring mention groups when tapping into KB. We make only weak assumptions on matching mentions against KB entities, by filtering on confidence and merely treating *semsum* as features rather than relying on perfectly mapped entities. Specifically, our $CROCS$ method handles the four dimensions of knowledge enrichment as follows:

Enrichment Target: We use per-document mention groups, after the local CR step, as target. In principle, we could repeat the enrichment during the iterations of the CCR algorithm. However, as $CROCS$ performs top-down splitting of groups rather than bottom-up merging, there is no added value. Assuming mentions m_{ij} and m_{xy} to corefer (obtained from the Stanford toolkit), we represent it as $m_{ij} \overset{\text{coref}}{\longleftrightarrow} m_{xy}$. Formally, for a mention group M,

$$KE_{target}[M] := \{m_{ij} \mid (\forall j \neq k)\ m_{ij} \overset{\text{coref}}{\longleftrightarrow} m_{ik} \wedge m_{i*} \in d_i\} \quad \text{(co-referring mention chain)}$$

Enrichment Source: We include all the sentences of a mention group in its *semsum*, thus drawing on the input document itself. The main enrichment harnesses entity-structured KBs like Freebase or YAGO by querying them with phrases derived from the mention groups' summaries. The features that are extracted from the best-matching entity include semantic types or categories (e.g., "politician", "award nominee"), alias names (e.g., "Michelle Robinson"), titles (e.g., "First Lady of the United States") and gender of people. These features are appended to the *semsums* and form the core of a

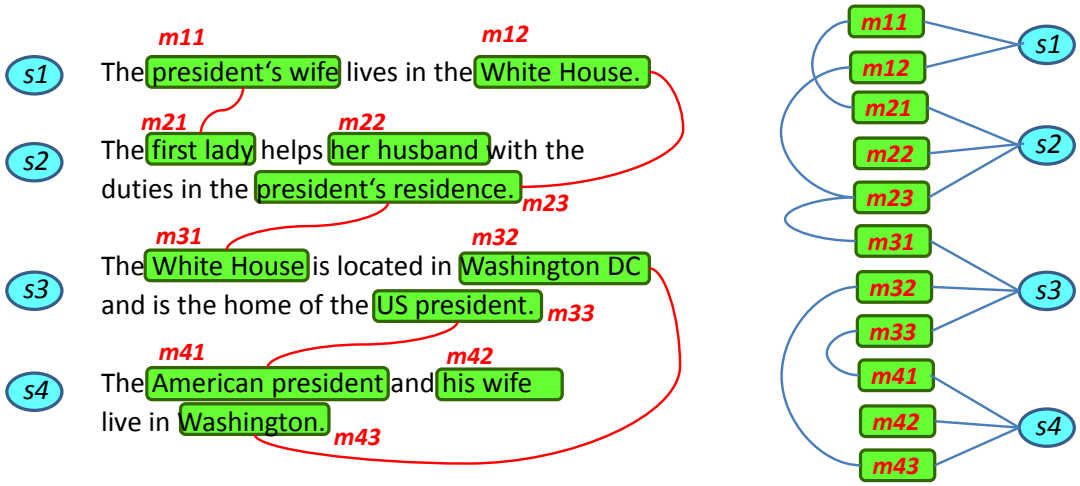

Figure 3.1 – Bi-partiteness of local mention groups for enrichment in $CROCS$.

mention group's semantic summary. Explicitly, for mention group M we have,

$$KE_{source}[M] := \{sentence,\ s \mid (\forall m \in M)\ s \text{ contains } m\} \bigcup \{KB\ feature,\ f \mid f \in \text{KB entity } E \text{ matched with } M\}$$

Enrichment Scope: $CROCS$ includes co-occurring mention groups as additional targets for semantic features. Consider the four example sentences in Figure 3.1, where the local CR is supposed to find four mention groups as shown. The mentions and the sentences in which they occur are represented as a bipartite graph depicting their connections (right side of Figure 3.1). Consider the mention group of "president's wife" (m_{11}) and "first lady" (m_{21}). Together with their immediate sentence neighbors in the bipartite graph, these mentions form what we call the *basic scope* for knowledge enrichment, i.e., $\{m_{11}, s_1, m_{21}, s_2\}$.

The sentences of this mention group contain other mentions which can be in mention groups spanning further sentences. We utilize this co-occurrence as additional cues for characterizing the mention group at hand. The union of the current scope with that of all the two-hop neighbors in the bipartite graph form the *extended scope*. For the group $\{m_{11}, s_1, m_{21}, s_2\}$, the two-hop mention neighbors are $\{m_{12}, m_{22}, m_{23}, m_{31}\}$. Hence, we include the scopes of these groups, the mentions and sentences, yielding the extended scope $\{m_{11}, s_1, m_{21}, s_2, m_{22}, m_{23}, m_{31}, s_3\}$. Thus, the enrichment scope for a mention group M in $CROCS$ is,

$$KE_{scope}[M] := \{m \mid m \in M\} \cup \{m' \mid m' \in sentences(m)\}$$

The complete process of *knowledge enrichment* in $CROCS$ thus involves the incorporation of the concatenated mention contexts obtained, i.e., $KE_{source} \cup KE_{scope} \cup KE_{target}$. Formally, the *extended semsum*, $S_{extended}(M)$ of mention group M is as:

$$S_{extended}(M) = S_{basic}(M) \cup \left(\bigcup_{m'} (S_{basic}(m') \mid \exists s : m' \in s \wedge m' \in S_{basic}(M)) \right)$$

where s represents a sentence in which m' occurs. In principle, we could consider even more aggressive expansions, like 4-hop neighbors or transitive closures. However, our experiments show that the 2-hop extension is a sweet spot that gains substantial benefits over the basic scope.

Enrichment Matching: For each local mention group, $CROCS$ first inspects the *coarse-grained types* (person, organization, location) as determined by the Stanford NER Tagger. We consider pronouns to derive additional cues for person mentions. If all tags in a group agree, we mark the group by this tag; otherwise the group as a whole is not type-tagged.

To match a mention group against a KB entity, we trigger a phrase query comprising tagged phrases from the mention group to the KB interface [1]. We remove non-informative words from the phrases, dropping articles and stop-words. For example, the first mention group, $\{m_{11}, m_{21}\}$ in Figure 3.1 leads to the query `"president wife first lady"`. The query results are filtered by matching the result type-tag with the type tag of the mention group. For the extended scope, we construct analogous queries for the co-occurring mentions: `"White House US president residence"` and `"husband"` in the example. The results are processed as follows.

We primarily rely on the KB service itself to rank the matching entities by confidence and/or relevance/importance. We simply accept the top-ranked entity and its KB properties, and extend the *semsum* by appending such distant features. This is also done for the co-occurring mention groups, leading to the enrichment of the *extended scope* of the original mention group considered.

To avoid dependency on the ranking of the KB, we can alternatively obtain the top-k results for each query and also the KB's confidence for the entity matching. We then re-rank the candidates by our similarity measures and prune out candidates with low confidence. We introduce a *confidence threshold*, θ, such that all candidates having a matching confidence below the threshold are ignored, i.e., the mention group is then disregarded for such enrichment of *semsum* construction. This makes extended scope robust to noise. For example, the mention group $\{husband\}$ having low confidence would likely degrade the semsum's quality and is thus dropped. Algorithm 3.1 shows the pseudo-code for constructing the *semsums*.

1. For example, gate.d5.mpi-inf.mpg.de/webyagospotlx/WebInterface or www.freebase.com/query

Algorithm 3.1: Extended Knowledge Enrichment in $CROCS$

Require: Text T, Mention groups M, Knowledge base KB, KB Match Threshold θ
Ensure: *semsum* for each mention group in G

1: **for** each mention group, $M \in G$ **do**
2: `Basic Scope:` $semsum_M \leftarrow$ sentences from T containing mentions m in M
3: `Extended Scope:` Append context of 2-hop co-occurring mentions (from bipartite graph) to $semsum_M$
4: Extract phrases from $semsum_M$ for query generation to KB
5: `Match:` Retrieve highest ranked KB result entity e with match confidence
6: **if** match confidence of $e > \theta$ **then**
7: Extract set of features for e, L_e from KB
8: `Enriched Scope:` Add extracted KB features L_e in $semsum_M$ ($S_{extended}(M)$) for mention group M
9: **end if**
10: **end for**
11: Output $semsum_M$ for all $M \in G$

3.5.2 Feature Vector Construction

The enriched *semsums* of the mention groups thus obtained comprise mention sentences, bags of phrases, and extracted KB features. For the example mention group $\{m_{11}, m_{21}\}$, we include the sentences $\{s_1, s_2, s_3\}$ during the extended-scope enrichment, and obtain phrases from the KB like, "Michelle Obama", "First Lady of United States", "capital of the United States", etc. to be appended to the extended *semsum*.

$CROCS$ next casts each *semsum* into two forms, (i) a *bag of words*, and (ii) a *bag of key-phrases*, and uses both for constructing a feature vector to compute the context similarity among the different mention groups.

3.6 Similarity Computation

$CROCS$ compares the mention groups by a similarity measure to infer whether they denote the same entity or not, i.e., co-refer. The similarity is based on the feature vectors of mention groups (constructed as in Section 3.5) obtained from the *semsum*. Each feature in a mention group's vector is weighted using IR-style measures according to the bag-of-words (BoW) model or the key-phrases (KP) model for the *semsums*. Empirically, the best approach is a mixture of both the words and key-phrases model, which is employed by $CROCS$. Similarity comparisons are computed on-demand and only for a small sampled set of mention groups, as required during the hierarchical clustering procedure (see Section 3.7).

The similarity between two mentions groups G_1, G_2 is then computed by a linear

combination as,

$$sim(G_1, G_2) = \alpha \times sim_{BoW}(G_1, G_2) + (1 - \alpha) \times sim_{KP}(G_1, G_2)$$

where α is a tunable hyper-parameter. Whenever two mention groups have a high similarity score, they are to be combined (referring to the same entity) and their feature vectors are concatenated by computing a bag union of their words and/or phrases, and then recomputing the feature weights respectively. Without loss of generality, our default setting is $\alpha = 0.5$.

3.6.1 Bag-of-Words Model (BoW)

For this model, we compute the term frequency, $tf(w)$ for each word w in the *semsums*, and also the inverse document frequency, $idf(w)$, of the word across all *semsums* (i.e., all mention groups from all input documents). The weight of w is then obtained as, $wgt(w) = tf(w) \times idf(w)$. As the *semsums* are short, we use the simple product rather than dampening tf values or other variations. Alternatively, more advanced IR weighting models such as Okapi BM25 [Jones *et al.*, 2000] or statistical language models can be used. However, the classical $tf \times idf$ measure [Salton, 1989] works quite well, and the similarity of two feature vectors are computed by their *cosine distance*.

3.6.2 Key-phrases Model (KP)

The key-phrases of a mention group are obtained by extracting proper names, titles, alias names, locations, and organization, from its *semsum*. Similar to the BoW model, $CROCS$ employs $tf \times idf$ style weights for the key-phrases-based features.

For computing the similarity of key-phrases between two mention groups G_1 and G_2, $CROCS$ matches the key-phrases of G_1 in the *semsum* of G_2, and vice versa. However, entire phrases rarely match exactly. For example, the key-phrase `"US President"` match only partially in the text `"President of the US"`. To consider such partial matches and reward both high overlap of words and short distances between matching words (i.e., match locality), we adopt the scoring model of [Taneva *et al.*, 2011], wherein the score for partial match of a key-phrase p in text x is given by,

$$S(p|x) = \frac{\#\ match\ words}{length\ of\ cov(p|x)} \left(\frac{\sum_{w \in cov(p)} wgt(w)}{\sum_{w \in p} wgt(w)} \right)^{1+\gamma}$$

where the *cover* (cov) of p in x is the shortest word span in text x containing all the words of p intersecting with x (with a bound of 10-20 words). For the example above, the cover of p = `"US President"` in the text x is `"President of the US"` (both words of p matching with cover length 4 in x). The parameter γ, $(0 < \gamma < 1)$ serves to tune

the progression of penalizing missing words. In our experiments, γ was set to 0.5 and stopwords such as "a", "the", etc., were removed with only keywords being considered.

For mention groups G_1 and G_2, we thus compute their *key-phrase* similarity as,

$$sim(G_1|G_2) = \sum_{p \in KP(G_1)} wgt(p) \times S(p|semsum's(G_2))$$

Finally, we resolve the asymmetry in the similarity measure, due to the ordering of the two groups, by considering the maximum similarity, as,

$$sim(G_1, G_2) = max\{sim(G_1|G_2), sim(G_2|G_1)\}$$

3.7 Hierarchical Clustering

The final stage of $CROCS$ takes the mention groups and the *semsums* as input. It performs a *top-down hierarchical bisection* process, based on the similarity scores among entities (mention groups), to cluster together co-referring mention groups at each splitting level.

Initially all mention groups are placed in a single cluster, and are then recursively split until a stopping criterion finalizes a cluster as leaf. At each level, cluster splitting is performed by using either *spectral clustering* [von Luxburg, 2007] or *balanced graph partitioning* [Karypis and Kumar, 1998]. Both these methods implicitly consider transitivity, which is essential as the equivalence classes of mentions should be transitively closed. The challenge of this seemingly simple procedure lies in,
(i) judiciously choosing and optimizing the model selection and stopping criteria; and
(ii) reducing the computational cost.

The latter is crucial as spectral clustering has cubic complexity, graph partitioning heuristics are compute expensive, and CCR (unlike CR) needs to cope with Web-scale inputs consisting of millions of documents and entities.

Clustering is invoked for each of the coarse-grained entity types (as obtained from Stanford NER tagger, e.g., people, places, and organizations) separately. The benefit is twofold: gaining efficiency and improving accuracy, as two different entity types would not co-refer in reality. However, the risk is that two differently tagged mention groups might actually refer to the same entity, with at least one tag being incorrect. Our experiments show that the benefits clearly outweigh this risk. Without loss of generality, we only consider chains that are tagged into one of the above types, and other co-reference chains are ignored.

We now discuss the different clustering approaches utilized in $CROCS$.

3.7.1 Active Spectral Clustering

Spectral clustering [von Luxburg, 2007] uses the eigenspace decomposition of the data similarity graph's Laplacian matrix to compute graph partitions as clusters based on eigen vectors. $CROCS$ adopts the recently proposed *Active Spectral Clustering* technique [Krishnamurty *et al.*, 2012; Wauthier *et al.*, 2012], which approximates the total eigenspace of a Laplacian matrix with a small subset of *sampled data points* (mention groups in $CROCS$). For n data points and sample size s in the order of $O(\log n)$, this technique significantly reduces the cost of spectral clustering from $O(n^3)$ to $O(\log^3 n)$ (with bounded error).

$CROCS$ initializes each bisection step by selecting s mention groups from a cluster and computes all the pair-wise similarities among the sampled groups. Spectral clustering is then performed on this subset to obtain a split into two clusters. The non-sampled mention groups are assigned to the closest cluster in terms of average distance to cluster centroids based on the k-means clustering approach. The children clusters are iteratively split further at next levels until the stopping criterion fires (discussed in Section 3.7.4).

3.7.2 Balanced Graph Partitioning

Balanced graph partitioning assigns the vertices of a graph into components of nearly the same size having few edges across components. The problem is NP-complete, and several approximation algorithms have been proposed [Buluc *et al.*, 2013]. $CROCS$ uses the *METIS* software (glaros.dtc.umn.edu/gkhome/metis/metis/overview) to obtain mention group partitioning at each level of the hierarchical clustering.

The underlying mention similarity graph is constructed by sampling s mention groups, and the similarities among them are represented as edge weights. For mention groups not selected in the sample, similarities to only the s sample points are computed and corresponding edges created. The graph is then partitioned using the multi-level recursive procedure of METIS [Karypis and Kumar, 1998] to minimize the edge-cuts thereby partitioning dissimilar mention groups. Similar to the spectral clustering approach, each graph component is hierarchically partitioned (into two parts) at each level until the stopping criterion is reached.

3.7.3 Specifics of Clustering

Active spectral clustering [Krishnamurty *et al.*, 2012] and graph partitioning uses random sampling, chooses the number of final clusters, k based on eigengap/graph-cut, and enforces a balancing constraint for the k clusters to be of similar sizes. However, $CROCS$ judiciously deviates from the design of [Krishnamurty *et al.*, 2012] as:

– *Model selection:* We choose a fixed number of partitions k at each cluster-splitting step of the hierarchical process. Specifically, we use a small k value, typically $k = 2$. This approach avoids prior selection of model dimension parameters, and allows the stopping criterion to decide the final number of clusters.
– *Form of graph cut:* $CROCS$ uses balanced normalized cut for graph partitioning [Karypis and Kumar, 1998]. However, unbalanced cluster sizes with several singleton clusters (having only one mention group) might be formed [Recasens *et al.*, 2013]. In our CCR setting, this is actually a natural outcome as many long tail entities occur only once in the corpus. Such mention groups significantly differ in semantic and contextual features compared to the other mention groups. Hence, singleton cluster mentions have low similarity score (based on *semsum*) with other mentions groups. This translates to low edge weights in the underlying similarity graph structure (between mentions), thus forming favorable candidates in the initial phases of cluster splitting using minimum edge-cut based graph partitioning. Therefore, $CROCS$ inherently incorporates early partition (during the clustering phase) of such possibly singleton mention clusters from the "main data", thereby helping in de-noising and efficiency.
– *Sampling:* Instead of sampling data points uniformly at random, we use biased sampling similar to initialization used in k-means++ clustering [Arthur and Vassilvitskii, 2007]. Starting with a random point, we add points to the sample set such that their average similarity to the already included points is minimized, thus maximizing the diversity among the samples.

3.7.4 Stopping Criterion

The unsupervised sample-based hierarchical clustering process in $CROCS$ operates without any prior knowledge of the number of clusters (entities) present in the corpus. We use the *Bayesian Information Criteria* (BIC) [Schwarz, 1978; Hourdakis *et al.*, 2010] as the *stopping criterion* to decide whether a cluster should be further split or finalized.

BIC is a Bayesian variant of the *Minimum Description Length* (MDL) principle [Grünwald, 2007], assuming the points in a cluster to be Gaussian distributed. The BIC score of a cluster C with s (sampled) cluster data points, x_i and cluster centroid $\bar{C}$ is,

$$BIC(C) = s \log_2 \Big(\sum_{i=1,\cdots,s} (x_i - \bar{C})^2 \Big) \; + \; \log_2 s$$

The BIC score for a set of clusters is the micro-averaged BIC of the clusters. $CROCS$ splits a cluster C into sub-clusters $C_1, \ldots, C_k$ iff the combined BIC value of the children is greater than that of the parent, else C is marked as leaf.

3.8 Experimental Evaluation

In this section, we empirically evaluate the performance of the proposed $CROCS$ framework and also suitably tune the operating parameters involved.

Benchmark Datasets: We performed experiments with the following three publicly available benchmarking datasets, thereby comparing the performance of $CROCS$ against state-of-the-art baselines under various input characteristics.

– ***John Smith* corpus**: provides the classical benchmark for CCR [Bagga and Baldwin, 1998] comprising 197 articles selected from the New York Times. It includes mentions of 35 different "John Smith" person entities across documents. However, all mentions pertaining to John Smith within a document refer to the same person. This provides a small-scale but demanding setting for CCR, as most John Smiths are long tail entities unknown to Wikipedia or any KB.

– **WePS-2 collection**: consists of a set of $4,500$ Web pages used in the *Web People Search 2* competition [Artiles *et al.*, 2009]. The corpus comprises the top 150 Web search results (using Yahoo! search (as of 2008)) for 30 different people obtained from Wikipedia, ACL'08, and US Census, covering both prominent entities (e.g., Ivan Titov, computer science researcher) and long tail entities (e.g., Ivan Titov, actor).

– **New York Times (NYT) archive**: comprises a set of nearly 1.8 million news article from the archives of the newspaper [Sandhaus, 2008] extracted between January 1987 and June 2007. We randomly selected $220,000$ articles from the time range of January 1, 2004 through June 19, 2007 containing about 3.7 million mentions, leading to nearly 1.6 million local mention chains after the initial intra-document CR pre-processing step.

In our experiments, we consider only mentions of person entities as this is the most predominant and demanding class of entities in the datasets, with different individuals sharing the same name or with similar surface forms. The John Smith and WePS-2 datasets have explicit ground truth annotations, while the NYT contains editorial annotations for entities present in each article. For knowledge enrichment, we used Freebase; although during sensitivity studies we explore alternative setups with YAGO.

Evaluation Measures: We use established measures to assess the output quality of the CCR methods, as:

– B^3 **F1 score** [Bagga and Baldwin, 1998]: it measures the $F1$ score as a harmonic mean of *precision* and *recall* of the final equivalence classes of mention groups. Precision is defined as the ratio of the number of correctly reported co-references (for each mention/entity) to the total number of reported co-reference chains; while recall computes the fraction of actual co-references (in the gold standard) that were correctly identified and reported. Both the final precision and recall are computed by averaging the values over all the mention groups.

– ϕ_3**-CEAF score** [Luo, 2005]: this provides an alternate way of computing precision, recall, and $F1$ scores using the best 1-to-1 mapping between the final mention

equivalence classes obtained and those in the ground truth annotations. The best mapping of ground-truth to the output classes are computed by pairing an output equivalence class to a gold standard class that exhibits the highest mention overlap.

All experiments were conducted on a 4 core Intel i5 2.50 GHz processor with 8 GB RAM running Ubuntu 12.04 LTS OS.

3.8.1 Parameter Tuning

In this section, we discuss the appropriate setting of the operational parameters for the $CROCS$ framework. The use of external KB features extracted (for mention groups) forms an integral part in the working of $CROCS$, and is represented by the choice of the enrichment *threshold*, θ. Given an input corpus, we now discuss the tuning of θ based on splitting the available data into *training* and *testing* subsets for the datasets.

We randomly partition the input data into 3 parts (assumed to be labeled as A, B, and C). One of the data parts is considered as the training data while the other two parts then provide the test dataset. Using the gold annotations of the training dataset, we empirically learn the value of θ that provides the best B^3 F1 score for CCR, using a simple *line search* strategy. Initially, θ is set to 1 (i.e., no KB feature usage) and is subsequently decreased using 0.01 as the step size for each of the learning phase iterations. As soon as the performance of $CROCS$ is seen to degrade (compared to the previous iteration), the procedure is terminated and the previous value of θ is considered as the corresponding learnt parameter value. The final results reported are averaged over 3 independent runs, each considering different data partitions (among A, B, and C) as the training data.

Although more advanced learning algorithms might be used, this simple approach is observed to work well in our settings. Learning of the θ value might converge to a local maximum, or may be distorted due to presence of noise in the training data – however, we later show (in Section 3.8.5) that the performance of $CROCS$ is robust to small variations of θ. We now study the performance of $CROCS$ on the different corpora.

3.8.2 John-Smith Corpus: Long-Tail Entities

Table 3.1 compares $CROCS$ with two state-of-the-art methods (based on stream clustering and inferencing) achieving the best published results for this dataset. 66 randomly selected documents were used as the training set (while the rest 131 formed the test set) and the subsequent θ value learnt (as described in Section 3.8.1) was 0.96.

Since the corpus contained mostly long tail entities not present in any KB (only *5-6* of the 35 different John Smith's are in Wikipedia, e.g., the explorer John Smith), the KB matches were too unreliable and led to the introduction of noise. Hence, a high value

Table 3.1 – B^3 F1 results on John Smith dataset for $CROCS$.

Method	P (%)	R (%)	F1 (%)
$CROCS$	78.54	72.12	**75.21**
Stream (Rao, 2010)	84.7	59.2	69.7
Inference (Singh, 2011)	-	-	66.4

Table 3.2 – B^3 F1 results on WePS-2 dataset for $CROCS$.

Method	P (%)	R (%)	F1 (%)
$CROCS$	85.3	81.75	**83.48**
PolyUHK (Artiles, 2009)	87	79	82
UVA_1 (Artiles, 2009)	85	80	81

of θ was obtained (i.e., KB features mostly disregarded).

$CROCS$ (using sample-based spectral clustering) was seen to achieve an *F1 score* of 75.21%, while the *Stream* [Rao *et al.*, 2010] and *Inference* [Singh *et al.*, 2011] methods attained only 69.7% and 66.4% F1 scores respectively. $CROCS$ also has a high ϕ_3-CEAF score of 69.89% exhibiting substantial gains over prior methods. The runtime of $CROCS$ was quite low, taking only around 6 seconds. We observed that the novel notion of representing mention chains by *semsums* with extended scope (mentions and co-occurring mention groups) proved essential for outperforming the existing methods (see Section 3.8.6).

3.8.3 WePS-2 Corpus: Web Contents

We next compared the sampling spectral clustering based $CROCS$ on the WePS-2 corpus against the best methods reported in the WePS-2 task [Artiles *et al.*, 2009]. We empirically obtained the KB match parameter $\theta = 0.68$ according to the train-test setup described earlier (with 1500 training documents).

$CROCS$ achieves a B^3 based $F1$ score of 83.48% and a ϕ_3-CEAF score of 74.02% (see Table 3.2), providing an improvement of about 1.5 F1 score points over the reported state-of-the-art approaches. We observe that the gain is not as high as that for the John Smith dataset, since the WePS-2 corpus contains fewer ambiguous entity mentions and the documents are longer as well, thereby providing richer context for the mention chains. Thus, simpler methods are also seen to perform fairly well. The runtime of $CROCS$ on WePS-2 corpus was seen to be about 90 seconds.

3.8.4 New York Times Corpus: Web Scale

The previous two datasets, John Smith and WePS-2 are too small to assess the robustness and scalability of $CROCS$ for handling Web-scale datasets. We therefore evaluate $CROCS$ (with sample-based spectral clustering) on the huge New York Times news corpus. A random sample of 220,000 NYT news articles were sampled and following the parameter learning approach, the knowledge enrichment threshold θ was learnt to be 0.45 with around $73K$ training documents.

$CROCS$ achieved a B^3 $F1$ score of 59.17% (with $P = 56.18\%$ and $R = 62.49\%$) and a ϕ_3-CEAF score of 50.0%. Due to the sheer scale of the problem, no prior methods for CCR was evaluated for such large dataset. However, the factor graph based approach of [Singh *et al.*, 2010] does measure the mention co-reference accuracy for a small sample of 1,000 documents only. Accuracy in this context is defined as the ratio of document clusters assigned to an entity to the ground truth annotated mention clusters. Following a similar approach for comparison, we also randomly sampled 1,000 documents considering only mentions with multiple entity candidates. $CROCS$ was seen to achieve an accuracy of 81.71%, as compared to 69.9% for [Singh *et al.*, 2010].

As for run-time, $CROCS$ took around 14.3 hours to process around 150,000 news articles selected as the test corpus on the NYT articles – making it a viable approach for such huge datasets. We also compared the results against alternative algorithms within our framework as shown later in Section 3.8.6. Hence, we observe that $CROCS$ efficiently handles Web scale input data.

3.8.5 Sensitivity Studies

The $CROCS$ framework involves a number of tunable hyper-parameters for adjusting the precise performance of the components. In this section, we study the robustness of spectral clustering based $CROCS$ for varying operational settings.

Knowledge Enrichment Scope:

$CROCS$ supports several levels of knowledge enrichment for the construction of the mention group *semsums* as:
(i) including only sentences of a mention group (i.e., disregard KB – no enrichment);
(ii) using distant KB labels for the given mention group only (basic scope); and
(iii) adding distant KB labels for co-occurring mention groups (extended scope).
We compared these configurations among each other and also against a state-of-the-art NED method alone. Specifically, we used the AIDA [Hoffart *et al.*, 2011] open-source software (github.com/yago-naga/aida) for performing NED, and mentions mapped to the same KB entity were considered to co-refer.

Table 3.3 – B^3 F1 (%) scores for $CROCS$ enrichment variants.

$CROCS$ **configuration**	**WePS-2**	**NYT**
Sentences only	50.35	39.52
Basic Scope	64.14	53.88
Extended Scope	**83.48**	**59.71**
NED baseline	61.25	59.62

Table 3.4 – B^3 F1 scores (%) for different choices of θ in $CROCS$.

Dataset	θ						
	0.0	0.25	0.5	0.65	0.75	0.9	1.0
WePS-2	76.9	77.3	82.4	83.9	75.7	68.9	63.5
NYT	60.5	61.5	62.2	62.2	60.0	52.1	48.4

We use the trained value of θ obtained previously (for the respective datasets) for constructing the *basic and extended scope* of *semsum* in $CROCS$, and report the best B^3 F1 scores. Note that the *Sentences only* and *NED* configurations are independent of the choice of θ value. The results are shown in Table 3.3.

Since, real-life Web articles contain a mixture of prominent entities, ambiguous names, and long tail entities, the sole reliance on NED for CCR fares poorly. Further, the mention context is better captured by considering co-occurring mentions (providing additional cues) as compared to simply using the mention sentences only. In fact, the proposed extended scope for *semsum* construction produces superior results compared to other models, with nearly 20 F1 points improvement on WePS-2 dataset.

Knowledge Enrichment Matching Threshold:

To study the influence of different degrees of distant KB feature extraction, we varied the enrichment matching threshold θ from 0.0 (accept all KB matches) to 1.0 (no import from KB). The John Smith dataset, largely containing long tail entities, uses $\theta \approx 1$ (trained value), and operates on *semsums* containing practically no feature inclusion from external KB. Hence for this corpus, we only considered the scenario where the KB is completely disregarded (i.e., $\theta = 1$), to obtain a B^3 F1 score of 76.47%.

For the other two datasets, the B^3 F1 results for varying θ are shown in Table 3.4. We observe that KB features help the CCR process and the best results are obtained for θ between 0.6 and 0.7. We observe that the exact choice of θ is not a sensitive issue, and any choice between 0.25 and 0.75 yields fairly good F1 scores (within 10% of the empirically optimal F1 results). Hence, $CROCS$ is robust to parameter tuning.

We observe that the trained value of θ (obtained previously) for both the WePS-2 and

Table 3.5 – θ error sensitivity of $CROCS$.

Dataset	θ used	P (%)	R (%)	F1 (%)
WePS-2	0.45	83.46	80.21	81.9
NYT	0.68	59.42	64.2	61.8

Table 3.6 – B^3 F1 scores (%) for different number of sub-clusters k in $CROCS$.

Dataset	k=2	k=3	k=4	k=5
John Smith	**76.47**	73.24	65.29	60.7
WePS-2	**83.92**	82.61	78.37	73.19
NYT	**62.24**	59.34	52.60	46.64

the NYT datasets are close to the optimal setting as seen from Table 3.4 and provide nearly similar F1 score performance. Therefore, we set $\theta = 0.65$ and consider the entire input corpora as test set for the remainder of our experiments.

To re-confirm the robustness of $CROCS$ to θ value ranges, we use the KB threshold trained on WePS-2 dataset, and test it on the NYT dataset (and vice versa). From Table 3.5, it is interesting to observe that $CROCS$ achieves comparable performance to that obtained when θ is learnt from the corresponding dataset, thereby demonstrating robustness even in presence of errors during the θ learning phase.

Clustering Model Hyper-Parameters:

We study the effect of varying k, the number of sub-clusters invoked for the bisection procedure at each level of the hierarchical clustering procedure in $CROCS$. By default, this is set to 2 (i.e., bisection) in our empirical settings. Table 3.6 shows the B^3 F1 scores obtained for different choices of k, for our three datasets (with $\theta = 1.0$ for John Smith and $\theta = 0.65$ for the other two datasets). We observe that $k = 2$ performs best in all the cases. The output quality is seen to monotonically drop with increase in k, as such aggressive partitioning forces even similar mention groups to form separate clusters, thereby degrading performance. On the other hand, bisection allows the hierarchical process to adjust the model selection parameters at the global level based on the information-theoretic stopping criterion.

Alternative KB Usage:

In our experimental setting, KB features (of best matching entity) for knowledge enrichment of *semsums* in $CROCS$ are extracted from Freebase. To assess the impact of dependency on the knowledge base used for feature extraction, alternative experiments on the WePS-2 and NYT datasets were conducted by using YAGO.

We obtain all approximate matches for a mention group and rank them based on the

Table 3.7 – $CROCS$ B^3 F1 scores with Freebase versus YAGO.

Dataset	Freebase			YAGO		
	P (%)	*R (%)*	*F1 (%)*	*P (%)*	*R (%)*	*F1 (%)*
WePS-2	86.3	82.1	83.9	86.6	82.5	84.0
NYT	59.8	64.9	62.2	61.3	60.8	61.0

Table 3.8 – Accuracy and scalability of various algorithms embedded in $CROCS$.

Dataset	Clustering Method	B^3 measure			ϕ_3 (%) measure	Run-time
		P (%)	*R (%)*	*F1 (%)*		
John Smith	**Spectral clustering**	79.6	80.1	79.85	73.52	8.11 sec
	k-means clustering	71.27	83.83	77.04	71.94	8.01 sec
	Balanced graph partition	75.83	79.56	77.65	70.63	7.83 sec
	Sampled k-means	63.57	65.52	64.53	59.61	5.12 sec
	Sampled spectral clustering	79.53	73.64	76.47	70.25	6.5 sec
	Sampled graph partitioning	71.42	77.83	74.49	68.36	6.86 sec
WePS-2	**Spectral clustering**	88.2	85.61	86.88	77.91	331 sec
	k-means clustering	85.7	84.01	84.85	76.45	296.56 sec
	Balanced graph partition	86.56	82.78	84.63	77.73	324.64 sec
	Sampled k-means	72.67	68.56	70.56	66.92	72 sec
	Sampled spectral clustering	86.2	82.11	83.92	74.7	85.8 sec
	Sampled graph partitioning	85.3	82.2	83.72	74.5	83.65 sec
New York Times	**k-means clustering***	39.34*	49.17*	43.72*	31.45*	>20 hrs
	Sampled k-means	40.45	45.34	42.76	40.61	17.8 hrs
	Sampled spectral clustering	59.78	64.92	62.24	51.02	19.6 hrs
	Sampled graph partitioning	61.45	62.71	62.07	50.88	19.7 hrs

* results after run terminated at 20 hrs (∼5% mentions processed)

key-phrase similarity (see Section 3.6) between the extended *semsums* of the mention group and the extracted features for the matched entity from the YAGO $hasLabel$ property and infoboxes in Wikipedia pages associated with the $sameAs$ link. The results obtained (shown in Table 3.7) depict similar performance to that obtained by using Freebase; hence portraying no preference of $CROCS$ to any particular KB.

3.8.6 Algorithmic Variants

The $CROCS$ framework supports a variety of algorithmic building blocks, most notably, different clustering methods (e.g., k-means) or graph partitioning for the bisection step, and most importantly, sampling-based methods versus methods that fully process all data points. In this section, we provide the comparative results for such scenarios on the three different datasets as tabulated in Table 3.8.

For the John Smith corpus (with $\theta = 1.0$), all algorithms except the sample-based k-means were seen to achieve similar performances in accuracy and runtime. The

best method was the full-fledged spectral clustering approach, with about 2% F1 score improvement in co-reference accuracy.

With the WePS-2 dataset, we obtain a similar picture with respect to output quality of the algorithms. However, this dataset is large enough to bring out the run-time differences. Sampling-based methods, including $CROCS$, were about $4\times$ faster than their full-fledged counterparts, albeit with a meager loss of about 2% in F1 score.

The NYT dataset finally portrays the scenario on huge datasets. Here, only the sample-based methods were able to run to completion, while all the full-fledged methods were terminated after 20 hours. The fastest of them, the simple k-means method, had processed only about 5% of the data at this point (needing about 400 hours on extrapolation). In contrast, $CROCS$, using sample-based spectral clustering or graph partitioning, needed about 19.6 hours with decent accuracy for the 220,000 documents. The sampling-based k-means competitor was slightly faster (17.8 hours), but lost dramatically on output quality (due to loss of transitivity information): with only about 42% F1 score compared to 62% F1 score for $CROCS$ with sampled spectral clustering.

Discussion: Hence, from the above experimental evaluations, we observe that $CROCS$ is indeed well designed for accurate and scalable sampling-based CCR, specifically for distinguishing "long tail" entities, whereas other simpler methods like k-means, lacking transitivity awareness, fail to deliver good output quality.

3.9 Summary

The presented $CROCS$ framework for cross-document co-reference resolution (CCR) provides an efficient and scalable approach for obtaining the equivalence classes of entity mentions present in the documents of an input corpus. $CROCS$ constructs *knowledge enriched semsum* for the individual mention groups by harnessing features based on the mention context from text, co-occurring mentions, and distant semantic labels from external KBs. It performs sampling-based spectral clustering or graph partitioning in a hierarchical bisection process to obtain the mention equivalence classes, thereby avoiding model-selection parameters and high cost of clustering or partitioning. $CROCS$ performs significantly better than state-of-the-art baselines, mitigating the problems of Web scale CCR and long tail entity identification.

4 Joint Entity Co-reference Resolution and Linking

The maintenance of Knowledge Bases (KB) in the face of newly emerging entities and/or extraction of new relationships among already discovered entities from new Web documents, raises the dual problem of not only identifying co-referring mentions across documents (CCR), but also the precise mapping of an entity to its corresponding entry in an existing KB (if present). This enables the extraction and incorporation of fresh information pertaining to the entity. The procedure of mapping a discovered entity to a KB entry is referred to as *Named Entity Linking* (NEL), and involves the technical difficulty of detecting whether an entity is already known or a new entity, i.e., whether the entity is present in the KB.

This chapter presents a novel framework for jointly modeling cross-document co-reference resolution (CCR) and linking of named entities (NEL) to entries within a KB. The proposed interleaved iterative CCR and NEL approach harnesses enhanced mention contexts from input texts for improving co-reference resolution and uses link validation step during entity linking to efficiently identify long tail entities absent in the KB. Empirical evaluations on large-scale benchmark datasets exhibit significant accuracy improvements for our algorithm over state-of-the-art approaches for both the CCR and NEL procedures – tackling both the above problems.

4.1 Introduction

Motivation. The advent of large knowledge bases like DBPedia, YAGO, and Freebase, containing huge collections of entities (e.g., people, places, and organizations) along with their attributes and relationships cater to myriad of modern smart applications like search, analytics, recommendations, and question answering. The major task that arises in both the KB construction process and entity-centric applications involves precise *recognition* and *resolution* of entities distributed across Web pages, news articles, and social media for precise information extraction (as discussed in Chapter 3).

However, the dynamics of a changing world generates huge volumes of new/updated Web pages and articles every day, providing new information about entities, their attributes, and relations. This leads to the continuous update task for knowledge bases to capture the real world, and involves not only the discovery of emerging entities/relationships, but also the evolution of already known entities present in KBs. In this regard, the newly extracted entities further need to be disambiguated and *linked*, if present, to entries in KBs for possible information addition. The challenge in such scenarios is the precise decision as to whether the current entity is *known* or represents a *new* entity, with high accuracy.

The problem of *Named Entity Recognition* (NER) deals with the identification of entity *mentions* in a text and their classification into coarse-grained semantic types (person, location, etc.) [Finkel *et al.*, 2005; Nadeau and Sekine, 2007; Ratinov and Roth, 2009], and involves token segmentation for mention boundaries detection, and mapping them to pre-defined categories. Entity *Co-reference Resolution* (CR) [Haghighi and Klein, 2010; Ng, 2010; Lee *et al.*, 2013] is essentially a clustering task to identify mentions (and anaphors) within a document referring to the same entity, thus computing equivalence classes or *mention groups*. When CR is extended to an entire text corpus, in order to generate equivalence classes of co-referring mentions across documents, the task is known as *Cross-document Co-reference Resolution* (CCR) [Bagga and Baldwin, 1998; Culotta *et al.*, 2007; Singh *et al.*, 2011] (see Chapter 3.1). However, neither CR nor CCR links mention groups to corresponding KB entities and represent both *in-KB* and *out-of-KB* entities (e.g., long tail or emerging entities absent in KB) in the same way.

Named Entity Linking (NEL), on the other hand, involves the *disambiguation* of textual mentions, based on context and semantic information, and their mapping to proper *entities within a KB* [Bunescu and Paşca, 2006; Cucerzan, 2007; Milne and Witten, 2008; Hoffart *et al.*, 2011; Ratinov *et al.*, 2011; Cornolti *et al.*, 2013]. *Named Entity Disambiguation (NED)* and "Wikification" are often used to denote the same task. The latter may be more broadly used, though, to include the disambiguation of common nouns and phrases onto concepts, whereas NED restricts itself to noun phrases that denote individual entities. NEL methods often harness the semantic similarity between mentions and entities for contextualization and coherence disambiguation among candidates [Milne and Witten, 2008; Kulkarni *et al.*, 2009; Hoffart *et al.*, 2011; Ratinov *et al.*, 2011]. However, in the absence of CR mention groups, NEL has limited context and is bound to miss out on certain kinds of difficult cases.

For example, in the text, `Albert Einstein won the Nobel Prize`, NER detects the mentions "Einstein" and "Nobel Prize" and marks their types as *person* and *misc* respectively. The CCR procedure identifies mentions `Albert Einstein` and `Nobel laureate Einstein` (if present in some other texts) to refer to the same entity *physicist Albert Einstein*, and to be different from the mention `Hans Albert Einstein`. Finally, "Einstein" is linked to the entry *German physicist Albert Einstein* in a KB by NEL.

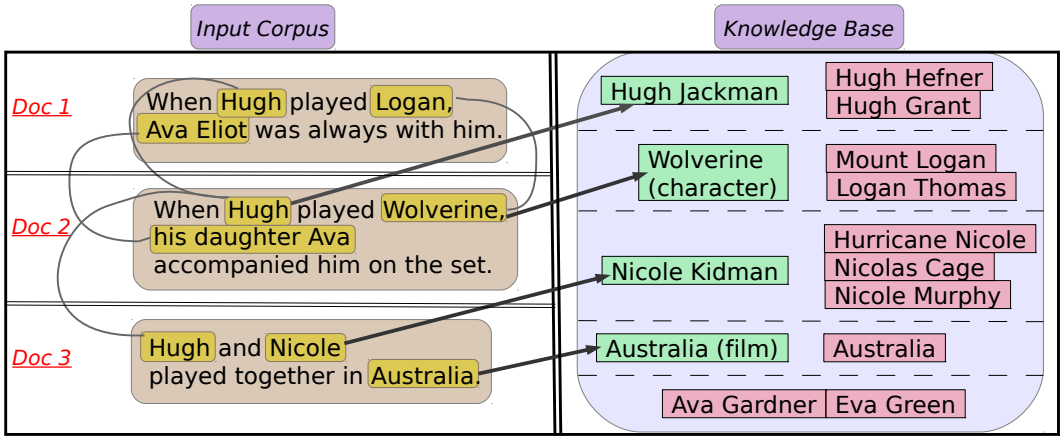

Figure 4.1 – Joint CCR-NEL example for $C3EL$ *(Green KB entries connected via arrows denote the correct entity linkage for the mention co-reference groups; while the* red *ones represent alternative incorrect candidates with similar surface forms).*

Problem Statement. Although NER, CR, CCR and NEL involve closely related tasks and their tighter integration has been shown to be promising [Cheng and Roth, 2013; Zheng *et al.*, 2013], they have mostly been explored in isolation. Recently, several *joint models* have been proposed for *CR-NER* [Haghighi and Klein, 2010; Singh *et al.*, 2013], *CR-NEL* [Hajishirzi *et al.*, 2013], and *NER-CR-NEL* [Durrett and Klein, 2014]. However, no method exists for jointly handling CCR and NEL for large text corpora. In this chapter, we aim to resolve the above shortcoming by proposing a combined *CCR-NEL* framework enabling information propagation across both the procedures, thereby improving the performances of both the CCR and NEL tasks along with precise identification of out-of-KB entities (linking to $null$).

4.1.1 Approach and Contributions

This chapter proposes the novel $C3EL$ *(*C*ross-doc*C*ument* C*o-reference resolution and* $Entity\ Linking$*)* framework for jointly modeling cross-document co-reference resolution (CCR) and linkage of mention groups to entities in a knowledge base (NEL). We next describe a toy example to outline the algorithmic approach in $C3EL$.

Example: To illustrate the potential synergies between CCR and NEL, consider the three documents in Figure 4.1 containing nine mentions (on the left) with candidate entities from a KB (on the right). CCR operating alone would likely miss the co-reference relation between `Logan` (in Doc. 1) and its alias `Wolverine` (in Doc. 2), leaving NEL with the difficult task of disambiguating "Logan" in a document with sparse and highly ambiguous context. On the other hand, NEL alone would likely map `Australia` (in Doc. 3) to the country (and not the movie) based on *prior popularity*, and could easily choose the wrong link for mention "Hugh". Moreover, the presence of `Ava Eliot`, an

out-of-KB mention, complicates the task.

However, if we could more freely interleave CCR and NEL and iterate them several times, we would have much stronger cues. An initial NEL step for the easiest mention, namely "Wolverine", maps it to the character of X-Men movies. This indicates that the three "Hugh" mentions could all be referring to the same actor, and are thus easily merged into a co-reference group during the CCR phase. Further, *knowledge enrichment* (as in Chapter 3.5) incorporates aliases of "Wolverine" into the context, thus assisting CCR of "Logan". We now have enough cues for NEL to choose the right entity for the "Hugh" mention group, which in turn enables the proper mapping of "Australia" to the movie (using context of actor and movie from co-occurring mentions). Finally, it becomes clear that mentions "Ava Eliot" and "daughter Ava" should be merged into the same group and represented as an out-of-KB entity mapped to $null$.

The above example clearly demonstrates that interleaving CCR and NEL is highly beneficial. However, appropriate choices for the ordering of CCR and NEL steps are usually not obvious at all. The proposed $C3EL$ algorithm solves this problem: automatically determining an efficient interleaving of CCR and NEL as discussed in the next sections.

To this end, the novel algorithmic components of $C3EL$ are:
- $C3EL$ iteratively aggregates intermediate information obtained from alternating steps of CCR and NEL, thus forming a *feedback loop* for propagating mention features and entity knowledge. Intuitively, co-referring mentions obtained via CCR generate global context for improved NEL performance, while mentions linked to KB entities (by NEL) provide distant semantic features as additional cues for CCR.
- $C3EL$ couples several building blocks like unsupervised hierarchical clustering, *context summaries* for mentions, and distant KB features during entities co-reference resolution step, drawing inspiration from the CCR-only method of Chapter 3.
- Mention linking to the KB (NEL) is performed using distant knowledge and co-occurring mentions, along with *link validation* based on the confidence and feature similarity for accurate detection of out-of-KB entities.

The above salient features are coupled together to jointly tackle the tasks of CCR and NEL. In a nutshell, the major contributions of this chapter are:

1. $C3EL$ framework for joint computation of cross-document co-reference resolution (CCR) and entity linking to a KB (NEL), based on propagating information across iterative CCR and NEL steps (Section 4.3);
2. techniques for considering co-occurring mentions in *context summaries* and for harnessing context-based keywords for *link validation* in NEL, improving accuracy on out-of-KB entities (Sections 4.5.1 and 4.5.2); and
3. experimental evaluation with two different huge corpora based on news articles and on web pages, demonstrating substantial gains for both CCR and NEL over state-of-the-art methods (Section 4.7).

4.2 Related Work

Co-reference Resolution (CR): Traditional intra-document CR methods involve syntactic and semantic feature combination for identifying the best antecedent (preceding name or phrase) for a mention. CR methods employ rules or supervised learning techniques based on linguistic features such as syntactic paths and mention distances to assess semantic compatibility [Haghighi and Klein, 2009; Raghunathan *et al.*, 2010; Rahman and Ng, 2011b], while syntactic features are derived by deep parsing of sentences and noun group parsing. Semantic features from background knowledge resources like encyclopedia were used in [Daumé III and Marcu, 2005; Ponzetto and Strube, 2006; Ng, 2007]. The use of Wikipedia and structured knowledge bases (such as YAGO) to obtain mention-type relation and fine-grained attributes was explored by [Haghighi and Klein, 2009; Rahman and Ng, 2011a]. An overview of CR methods is given in [Ng, 2010]. Recent methods involve the use of multi-phase sieve, applying a cascade of rules for narrowing down the antecedent candidates for a mention [Raghunathan *et al.*, 2010]. Cluster ranking functions have also been proposed [Rahman and Ng, 2011b; Zheng *et al.*, 2013] to extend this paradigm for incrementally expanding and merging mention groups with preceding candidate clusters using relatedness features [Ratinov and Roth, 2012] and distant knowledge inclusion [Durrett and Klein, 2013].

Distant Knowledge Labels: For obtaining semantic features, additional knowledge resources such as Wikipedia, YAGO, and FrameNet have been considered [Rahman and Ng, 2011a; Baker, 2012]. CR methods with confidence-thresholds were proposed in [Ratinov and Roth, 2012; Lee *et al.*, 2013], and [Zheng *et al.*, 2013] generalized these techniques by ranking the matching entities for distant labeling. Such prior methods utilize distance labels of the current mention and considers all matching mentions making the procedure expensive. On the other hand, $C3EL$ extracts distant features for the strongly matching (best) candidate alone, reducing the performance overhead.

Cross-Document CR (CCR): Early approaches towards CCR involved the use contextual information from input documents for IR-style similarity measures (e.g., tf$\times$idf score, KL divergence, etc.) over textual features [Bagga and Baldwin, 1998; Gooi and Allan, 2004]. Probabilistic graphical models jointly learning the mappings of mentions to equivalent classes (co-referring mentions) using features similar to CR techniques were studied in [Culotta *et al.*, 2007; Singh *et al.*, 2010; Singh *et al.*, 2011], while a clustering approach coupled with statistical learning of parameters was presented in [Baron and Freedman, 2008]. However, such methods fail to cope with large corpora, and hence a "light-weight" streaming variant of CCR was introduced by [Rao *et al.*, 2010]. Co-occurring mentions context have been harnessed for disambiguating person names for CR in [Mann and Yarowsky, 2003; Niu *et al.*, 2004; Chen and Martin, 2007; Baron and Freedman, 2008]. However, these methods do not use KB and depend on information extraction (IE) methods, witnessing substantial noise due to IE quality variance. Additional works in this domain have been described in Chapter 3.2.

Named Entity Linking (NEL): Named entity resolution and linking stems from SemTag [Dill *et al.*, 2003], and similar frameworks like GLOW, WikipediaMiner, AIDA, and others [Milne and Witten, 2008; Ratinov *et al.*, 2011]. A collection of entity disambiguation models was presented in [Kulkarni *et al.*, 2009]. Additional NEL approaches utilize the notion of semantic similarity of entities to corresponding Wikipedia pages [Milne and Witten, 2008], while co-referent mention graph construction modeling mention co-occurrences and context similarity from outgoing hyperlinks in Wikipedia was used by [Hoffart *et al.*, 2011]. An integer linear programming (ILP) formulation also based on Wikipedia page similarities was presented in [Ratinov *et al.*, 2011]. However, none of these methods involve the incorporation of CR results for NEL. The first study on the benefits of CR for NEL was by [Ratinov and Roth, 2012]; but a joint model was not proposed, instead attributes from Wikipedia categories were used as features. An overview and evaluation of several NEL methods was presented in [Hachey *et al.*, 2013].

Joint Models: Jointly solving CR for entities and events utilizing cluster construction based on feature semantic dependencies was devised in [Lee *et al.*, 2012]. The use of CR as a pre-processing step for subsequent NEL procedure using an ILP formulation was also proposed by [Cheng and Roth, 2013]. Recently, [Hajishirzi *et al.*, 2013] proposed a joint model for CR and NEL using the Stanford multi-pass cluster update CR system with automatic linking of mentions to Wikipedia. An integrated belief propagation-based framework for CR, NER, and relation extraction was developed in [Singh *et al.*, 2013]. Subsequently, the model was enhanced by the use of structured conditional random fields to solve CR, NER, and NEL in combination [Durrett and Klein, 2014]. Other works involving joint formulation of NER and NEL use uncertainty of mention boundaries along with segmentation information extracted from Wikipedia [Sil and Yates, 2013]. However, to the best of our knowledge, this work provides the first approach to jointly tackle CCR and NEL across documents in an entire corpus.

4.3 Joint CCR-NEL Framework

The proposed $C3EL$ framework, similar to the $CROCS$ model (as described in Chapter 3.3), assumes an input corpus of text document $D = \{d_1, d_2, \cdots\}$, with a markup of entity mentions present in the documents, i.e., $M = \{m_{11}, m_{12}, \cdots, m_{21}, m_{22}, \cdots\}$ with $m_{ij} \in d_i$. As output, it not only computes the equivalence classes of co-referring mentions (CCR), but also links the classes to corresponding KB entries if present (NEL), else links them to $null$. Formally, $C3EL$ aims to *jointly compute*:

– CCR: equivalence relations over M to obtain equivalence classes C_l of co-referring mentions. Formally, for corefering mentions m_{ij} and m_{xy} (i.e., $m_{ij} \overset{\text{coref}}{\longleftrightarrow} m_{xy}$),

$$C_l = \{m_{ij} \mid m_{ij} \in C_l \text{ and } m_{ij} \overset{\text{coref}}{\longleftrightarrow} m_{xy}\} \qquad [\forall m_{ij}, m_{xy} \in_{i \neq x \wedge j \neq y} C_l], \text{ and}$$
$$C_l \cap_{l \neq n} C_n = \emptyset \; and \; \cup_l C_l = M$$

– NEL: linking of classes, C_i to entities present in KB or map it to $null$ if absent; i.e.,

$$C_i \mapsto \begin{cases} E_i & \text{, if entity } E_i \in KB \\ \emptyset & \text{, otherwise} \end{cases}$$

We demonstrate that our interleaved CCR-NEL method enables global semantic propagation across document boundaries for precise distinction of popular and long tail entities (well/sparsely represented in KB) from new/emerging entities (absent in KB).

To this end, the $C3EL$ framework consists of *three* major algorithmic stages:

1. **Pre-Processing:** The input corpus of text documents, D are processed with markup of the mentions, M, and construction of intra-document co-referent mention chains. The mention chains are then represented by a *context summary* characterizing the mention context obtained from the input text (Section 4.4).
2. **Interleaved NEL and CCR:** The local mention chains are iteratively processed by the CCR and NEL stages, incorporating external KB features and link validation for proper co-reference resolution across documents and linking to KB entries (Section 4.5).
3. **Finalization:** Possible association for "orphan" or missed mention groups are computed based on context similarity to alleviate cascading mention detection/omission and other errors within the iterative stage (Section 4.6).

We next discuss the individual working components of $C3EL$ in details.

4.4 Pre-Processing Stage

Similar to the initial pre-processing steps as described in Chapter 3.4, HTML documents in the input corpus D are first transformed into plain text using standard tools like jsoup (jsoup.org). Recognition and markup of mentions and anaphors present in the texts are performed using the *Stanford CoreNLP toolkit* (nlp.stanford.edu), and a coarse-grained lexical type for each mention chain (e.g., person, location, organization) is obtained from the Stanford NER Tagger [Finkel *et al.*, 2005]. The multi-pass sieve algorithm for single-document CR [Raghunathan *et al.*, 2010; Lee *et al.*, 2011; Lee *et al.*, 2013] is then used to compute the mention co-reference chains within each document (intra-document co-reference resolution), and a *head mention*, h is chosen for each of the extracted mention groups (chains). The head mention is typically represented by the most explicit denotation of the entity (e.g., person's full name with title, location name with country, etc.).

The local CR procedure might introduce errors in mention boundary detection or mention omission which might impact the performance of subsequent stages in $C3EL$. However, later in Section 4.6, we propose to mitigate the effect of such inconsistencies based on mention context similarity.

Given the local co-referent mention chains, for each of the mention groups $M(h)$, $C3EL$ then captures the textual context of the mentions by constructing a *context summary*, $\mathbb{CS}[M(h)]$, using the following features:

- **Mention Sentences** – all sentences within a document containing a reference to mentions m belonging to the group $M(h)$ are aggregated; and
- **Co-occurrence Mention Sentences** – all sentences pertaining to other mention groups, M', such that any mention of M' co-occurs in any of the sentences of $M(h)$ (as obtained above), are also concatenated.

Formally, for each mention group $M(h)$, let $S(M(h)) = \{sentence(m_j) \,|m_j \in M(h)\}$ represent the set of extracted *sentences* for $M(h)$, where $sentence(m_j)$ denotes the sentences in which the mention m_j occurs. Also, let the co-occurring mention set of $M(h)$ be denoted by $Co(M(h)) = \{m' \,|m' \in S(M(h)) \wedge m' \notin M(h)\}$ for finding the *co-occurrence mention sentences*. The context summary of $M(h)$ is then defined as:

$$\mathbb{CS}[M(h)] = S(M(h)) \cup \left(\bigcup_{m' \in Co(M(h))} S(m') \right)$$

Observe that, the above constructed *context summary* for mention groups are similar to the *extended scope* summaries of $CROCS$ (Chapter 3.5). However, the context summaries in $C3EL$ intentionally do not initially include any distant KB features for mentions (i.e., knowledge enrichment – which is added only during later stages) to minimize potential noise inclusion from overly speculative mappings to KB entities at this initial stage with limited information.

4.5 Interleaved NEL and CCR Approach

After the preliminary CR step on each document and the construction of context summaries for the mention groups, $C3EL$ now performs an initial NEL step for each of the mention groups $M(h)$, using the extracted summaries $\mathbb{CS}[M(h)]$ as inputs. From the NEL stage, it obtains: (i) the best matching entity, (ii) the confidence of the match, and (iii) the corresponding entity Wikipedia page. Off-the-shelf NEL softwares (like WikipediaMiner [Milne and Witten, 2013] or Illinois-Wikifier [Cheng and Roth, 2013]) [1] can be used for mention-entity mapping based on the prior popularity of the named-entities (from the KB), and textual similarity between $\mathbb{CS}[M(h)]$ (context of the mention group) and the entity descriptions in KB.

For each mention group $M(h)$, the entity link obtained (from NEL) is then "validated" using a similarity measure between features from the *context summary*, $\mathbb{CS}[M(h)]$ (including co-occurring mentions) and distant KB labels of the linked entity – forming

1. obtained from github.com/dnmilne/wikipediaminer/wiki/About-wikipedia-miner and cogcomp.cs.illinois.edu/page/software_view/Wikifier respectively.

the *link validation* procedure of $C3EL$. This explicit use of co-occurring mentions' contexts ($S(Co(M_i))$) helps to better identify out-of-KB entities compared to direct full-fledged NEL using the entire input text (shown in Section 4.7). Also the use of NEL on $\mathbb{CS}[M(h)]$ alone, makes $C3EL$ "light-weighted".

The mappings between the mention groups and KB entries obtained during NEL, are then classified, on the basis of their linking confidence scores, into *Strong Evidence* (SE), *Weak Evidence* (WE), and *No Evidence* (NE) classes. For mention groups placed in SE, the KB features (obtained previously) are appended to their context summaries, while mentions strongly linked to same KB entities are considered to be co-referring and hence grouped together with concatenated summaries (performing implicit CCR). Hierarchical clustering based CCR (as in Chapter 3) is then performed on the mention groups to obtain co-referring mention clusters with concatenated context summaries. This process of interleaved NEL and CCR is then repeated.

Considering our example (in Figure 4.1), we now outline the iterative steps of $C3EL$.
I. During the **first iteration**, the NEL and CCR stages performed in $C3EL$ are as:
– **NEL:** The initial NEL step maps the unambiguous mentions, `Wolverine` to the X-Men movie character and `Australia` to the country, with high confidence. Using context similarity (involving movie and actor) the linking of Wolverine is accepted, and owing to the high confidence of match, it is added to the SE class. We now *enrich* the context of this "strongly" linked mention by appending external KB features (e.g., alias "Logan" in this case) to the context summary.
However, link validation fails for "Australia" as there is very low similarity between the mention context features (e.g., Hugh, Wolverine, etc.) and the distant KB labels extracted from the linked Wikipedia page (e.g., Commonwealth, population, etc.). Thus the link is dropped and the mention is added to the NE class for further processing. On the other hand, the three "Hugh" mentions exhibit low NEL confidence due to the high ambiguity of this first name and are therefore classified into WE. The remaining mentions have extremely low NEL confidence (due to sparse contextual information) and are added to NE, as shown in Figure 4.2.
– **CCR:** The entities in the SE class are considered to be properly linked to corresponding KB entries. The WE and NE classes are then fed separately to the CCR procedure. Based on the *context summary* similarities between the mention groups, $C3EL$ performs hierarchical clustering to group together co-referring mentions. In our example, "Logan" is now connected to "Wolverine" (due to similarity based on external KB features obtained during NEL), and the three "Hugh" mentions (in WE class) are seen to be clustered together with the individual mention context summaries concatenated. This merging of the summaries strengthens the captured contexts, which propagates across document boundaries. Due to lack of sufficient contextual information, the other mentions form singleton clusters after the CCR procedure as shown in Figure 4.2. This concludes the first combined iteration of $C3EL$.
II. The results obtained at the end of iteration I are provided as inputs to the **second**

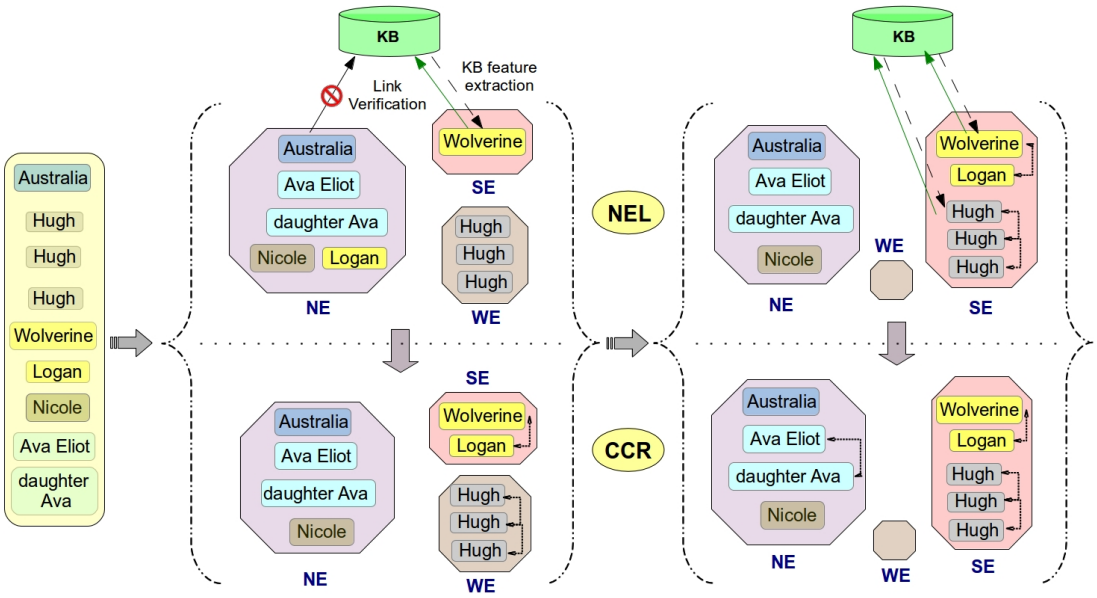

Figure 4.2 – Classification of mentions during iterative NEL and CCR in $C3EL$.

iteration, forming a feedback-loop, and are processed as:

– **NEL:** The enhanced *context summary* across the documents for the "Hugh" mention group in WE now provides definitive cues for correctly mapping it to the actor Hugh Jackman in KB with a high confidence and successful link validation, and is thus moved to the SE class. Context enrichment is performed by the addition of external KB features into the group's context summary (Figure 4.2).

– **CCR:** The ensuing CCR step groups together "Ava Eliot" and "Ava" (in class NE) based on the co-occurrence context similarity induced by the co-referring Hugh mentions.

III. During the next iteration, the subsequent NEL stage now correctly links "Australia" to the movie and "Nicole" to the actress by using CCR-generated mention-group contexts based on co-occurring mentions. On the other hand, the "Ava" mention group remains unlinked (or linked to $null$) due to the link validation procedure and is hence accurately identified as an out-of-KB entity. The final CCR step, in this case, becomes redundant as all mention groups have been disambiguated and linked.

Note that, this process of alternating CCR and NEL is repeated until all mention groups are strongly linked to KB entities, or no changes are observed in the different classes, i.e., state of equilibrium. The NEL and CCR procedures within each iteration step of $C3EL$ are performed separately on the different mention type-tags (like PER, and LOC), since different mention types rarely co-refer in practice. In fact, the framework not only performs significantly well in "difficult" scenarios (as shown) with highly ambiguous and context dependent mentions, but intuitively can also efficiently handle other text scenarios such as tweets and news headlines, where well-known entities are presented with limited context (assuming readers' prior knowledge). We next present the internal working details of the NEL and CCR stages of $C3EL$.

4.5.1 Named-Entity Linking (NEL) Module

The NEL procedure of $C3EL$ helps in disambiguation of mentions to corresponding entities in a knowledge base, specifically to the YAGO knowledge base (yago-knowledge.org) in our setting. During each iteration, NEL is performed on the small context summaries ($\mathbb{CS}[M(h)]$) of the mention groups based on named entity popularity statistics and context coherence, to obtain the best matching entity, the match confidence score, and the corresponding Wikipedia page (from $sameAs$ link in YAGO). Assume a mention group $M(i)$ to have been mapped to an entity e_i with a confidence score of $\phi(M(i), e_i)$.

The efficiency of NEL in the $C3EL$ framework for properly discerning out-of-KB entities stems from the following two-fold novelty:

A. Link Validation: The obtained candidate entities during NEL are not only pruned based on the match confidence (as reported by the employed NEL procedure), but are also validated by similarity measures between the original entity context (from its summary) and the candidate entity features (from KB or other sources).

Hence, once a mention group (e.g., Hugh) is linked to a candidate KB entity during the NEL stage, we extract *distant KB labels* such as semantic types or categories (e.g., actor), title (e.g., Golden Globe winner), alias (e.g., Wolverine), location, and gender (for person) from the infoboxes in the candidate entity's Wikipedia page. The similarity of these features to keywords obtained from the context summary of the entity, $\mathbb{CS}[M(i)]$, is computed using IR-style term frequencies within a document (tf) and inverse document frequencies within the corpus (idf). We utilize the bag-of-words model based $tf \times idf$-weighted cosine similarity measure. If the similarity score is above a threshold, τ, the NEL result is accepted, i.e., mention group $M(i)$ is considered to be linked to KB entity e_i; otherwise the linking obtained is discarded – thus avoiding noisy linkage of sparse mentions to prominent KB entries. This subtle introduction of *controlled distant supervision* within the $C3EL$ framework enables efficient detection of emerging and/or out-of-KB mentions, while the inclusion of KB features for strongly linked mention groups provide possible cues to later CCR steps.

For example, as described earlier, the NEL procedure initially (incorrectly) maps mention "Australia" to the corresponding country entity in a KB based on popularity and prior (in the example of Figure 4.2). However, the proposed link validation step fails (and the linking is rejected) in this case, as there exists very low similarity between the mention context features (e.g., Hugh, Wolverine, etc.) and the distant KB labels extracted from the candidate entity Wikipedia page (e.g., Commonwealth, population, etc.). This enables proper NEL (to the correct but sparsely represented movie entity in KB) during later stages of $C3EL$ with enhanced global context across documents.

B. Classification: To sift out well-known and long tail entities from emerging ones, and prevent "noisy" interactions among the contexts of in-KB and out-of-KB mentions

(with similar surface forms), each mention group $M(i)$ (linked to entity e_i with score $\phi(M(i), e_i)$ in NEL) is classified into 3 classes by 2 threshold parameters, δ_s and δ_w, as:

– *Strong Evidence* (SE): For $\phi(M(i), e_i) \geq \delta_s$, the mention group $M(i)$ is considered to exhibit high linkage confidence with e_i and is thus placed in the SE class. If two or more mentions in SE are independently mapped to the same KB entity, they co-refer transitively and are hence grouped together with their context summaries merged (implicit CCR). Distant KB features for mentions in SE are extracted and appended to $\mathbb{CS}[M(i)]$, providing additional cues for later steps.
– *Weak Evidence* (WE): Mention groups with $\delta_w \leq \phi(M(i), e_i) < \delta_s$ are placed in this class, and mostly represent long tail in-KB entities (sparsely represented in KB) with limited semantic information (for detection) but might also be new/emerging entities absent from KB.
– *No Evidence* (NE): When $\phi(M(i), e_i) < \delta_w$, it represents mentions groups that have not been mapped to any KB entity due to near-zero match confidence or failed link validation step during the NEL procedure. These entities are most likely to be sparse or out-of-KB and are thus allocated to this class.

The mention groups within the classes are then separately processed by $C3EL$.

4.5.2 Cross-Document CR (CCR) Module

The CCR stage of $C3EL$ adopts the sampling-based hierarchical clustering approach using context similarity measure of $CROCS$, as described in Chapters 3.6 and 3.7, to obtain co-referring mention clusters. We now provide a brief description of the working of the CCR step in $C3EL$.

A. Similarity Measure: To infer whether two mention groups represent the same entity and co-refer, the similarity between the context summaries are computed based on:
(i) classical *tf-idf* weighted bag-of-words cosine distance to match context related words and key-phrases; and
(ii) partial-match scores of multi-word key-phrases in bounded text windows proposed in [Taneva *et al.*, 2011] to reward match overlap and locality.

The context summaries (with stop-words removed) are initially interpreted by two different language models, such as, (i) bag of words, and (ii) bag of key-phrases, to extract feature vectors (such as named mentions, dates, and quantity) for similarity computation. Finally, a linear mixture model combining the *bag-of-words* (BoW) and *key-phrases* (KP) similarity is used to assign feature weights using the *tf-idf* measure, to characterize the mention group similarities.

B. Hierarchical Clustering: $C3EL$ also adopts the active clustering technique proposed by [Krishnamurty *et al.*, 2012] and applies it to graph partitioning so as to compute a bisection-based hierarchical clustering for obtaining the equivalent classes

of co-referring mention groups (as described in Chapter 3.7.2).

Specifically, initially s mention groups are uniformly randomly sampled and their similarities to the other groups (using context summary) are computed. A similarity-weighted graph with the mention groups as nodes and edge weights representing mention-mention similarities is constructed. Bisection-based hierarchical *balanced min-edge-cut graph partitioning* [Buluc *et al.*, 2013] is performed, using the $METIS$ software [Karypis and Kumar, 1998] (obtained from glaros.dtc.umn.edu/gkhome/metis/metis/overview), to partition non-coreferent mentions groups. The *Bayesian Information Criterion* (BIC) [Schwarz, 1978; Hourdakis *et al.*, 2010], a Bayesian variant of Minimum Description Length [Grünwald, 2007], is used as the cluster split stopping criterion, and the context summaries within each final cluster are merged.

The combination of several CCR and NEL strategies within the proposed $C3EL$ framework enables it to efficiently process heterogeneous corpora that go beyond a single domain and style, catering to diverse information sources. In Section 4.7, we present extensive experimental results to validate the performance gain of $C3EL$ (in both CCR and NEL) due to the joint model and judicious selection of working principles.

4.6 Finalization Stage

The final stage of the $C3EL$ framework aims to alleviate propagated CR errors like erroneous mention boundary detection (in NER) and mention omissions in co-reference chain (in CR), leading to "phantom" or spurious sparse/out-of-KB entities. However, we use limited context similarity (provided by CCR), weak linking to entity (during NEL), and head mention name similarity in conjunction to facilitate probable correct mention disambiguation in such cases.

After completion of the iterative CCR-NEL phases of $C3EL$ (i.e., equilibrium state), for the remaining mention groups in WE or NE classes, we finally perform threshold based disambiguation of mention clusters using the context summaries. Hence, for each mention group $M(i) \in (WE \vee NE)$, we compute the following:
(1) Context summary similarities (as in Section 4.5.2) of $M(i)$ to all the other mention groups $M(j)$ present in SE is computed. We also use distance features for $M(i)$, in case it is weakly linked to a KB entities. This alleviates mention omission problems leading to ill-represented mention due to extremely limited context.
(2) The textual overlap between the mention group representatives between $M(i)$ and the mention groups in SE are computed. If there exists significant overlap of mention name along with a high degree of context similarity, it possibly signifies same or co-referring mentions. Spurious mentions generated by erroneous boundary detection can thus be handled.

Finally, $M(i)$ is concatenated with the best matching entity $M(k)$ (in SE), if the linear combination of the similarity scores obtained above is above a threshold θ; else $M(i)$ is marked as an out-of-KB entity (mapped to $null$) and is finalized in the NE class. The obtained mention groups represent the final equivalence classes of co-referring mentions across documents – capturing both in-KB entities (with links to the KB) in the SE class and out-of-KB entities (mapped to $null$) in the NE class. Algorithm 4.1 presents the pseudo-code outlining the detailed working of the entire $C3EL$ framework.

4.7 Experimental Evaluation

In this section, we empirically study the performance of $C3EL$ against various state-of-the-art methods, analyzing the individual gains in CCR and NEL due to joint modeling.

Datasets: To evaluate the approaches, we use the following 2 publicly available corpora:

– **EventCorefBank (ECB) corpus** [2] [Bejan and Harabagiu, 2010]**:** it contains 482 news and Web articles (classified into 43 topics) with a total of 5447 mentions corresponding to 1068 distinct named-entities. Entity co-reference annotations (across documents within each topic cluster) were provided by [Lee *et al.*, 2012], and we performed manual examination of the annotations for KB linking of the entities to Wikipedia entries, if present; thus providing ground truth for both CCR and NEL.

– **ClueWeb2009 FACC1 dataset** [3] [Gabrilovich *et al.*, 2013]**:** this provides machine automated entity-linkage annotations of the *ClueWeb09* corpus (circa 1 billion crawled Web pages) with Freebase entries [4]. The corpus contains many topical domains and highly diverse documents from news, movie reviews, people home pages, blogs and other social media posts. We randomly select $500K$ documents containing 4.64 million mentions associated with 1.29 million distinct entities to form our corpus. For NEL ground-truth construction, we link the entities to their Wikipedia pages (using Freebase's *"on the web"* property). Since no explicit annotations of inter-document entity co-references exists, we consider two mentions (in different documents) to co-refer if they are linked with the same Freebase entity.

Evaluation: To assess the output quality of $C3EL$ against the other methods, we use the following established metrics:

– B^3 $F1$ **score** [Bagga and Baldwin, 1998]: measures the $F1$ score as the harmonic mean of averaged *precision* and *recall* computed over all mention groups in the final equivalence classes. Precision (for a mention group) represents the ratio of the number of correctly reported co-references (or linking) to the actual number; while recall computes the fraction of the gold-standard annotations correctly identified.

– $\phi_3 - CEAF$ **score** [Luo, 2005]: provides an alternate $F1$ score computed as in the

2. obtained from faulty.washington.edu/bejan/data/ECB1.0.tar.gz
3. obtained from lemurproject.org/clueweb09/FACC1
4. Human analysis of a subset of the annotations generated revealed a precision of 80 – 85% [Gabrilovich *et al.*, 2013]

Algorithm 4.1: $C3EL\ Framework$ for Joint CCR and NEL

Require: Input corpora D and parameters δ_s *and* δ_w (for NEL match classification), τ (for link validation), and θ (for finalization)
Ensure: Equivalence classes, C of co-referring mention groups linked to KB entities or to $null$

Perform intra-document CR to obtain M, the set of co-referring mention groups
for each mention group $M(h) \in M$ **do**
  Pick representative mention h
  Construct *context summary*, $\mathbb{CS}[M(h)]$ with sentences and co-occurring mentions
end for
Create 3 category classes, $\mathbb{CL} = \{SE, WE, NE\}$
Initialize $SE := WE := \emptyset$, and $NE \leftarrow M$

while $(WE = NE = \emptyset) \vee$ (No change in $\mathbb{CL}$) **do**
  for each mention-group, $M(i) \in \{WE \cup NE\}$ **do**
    Perform NEL to obtain best matching entity e_i with confidence $\phi(M(i), e_i)$
    $Sim \leftarrow$ similarity between KB features of e_i and keywords in context-summary
    if $Sim \geq \tau$ **then**
      if $\phi(M(i), e_i) \geq \delta_s$ **then**
        SE $\leftarrow SE \cup M(i)$
        Link $M(i)$ to e_i and append KB features to context summary
      else if $\phi(M(i), e_i) \geq \delta_w$ **then**
        WE $\leftarrow WE \cup M(i)$
      else
        NE $\leftarrow NE \cup M(i)$
      end if
    else
      NEL results discarded, $M(i)$ mapped to $null$, and NE $\leftarrow NE \cup M(i)$
    end if
  end for
  Merge mentions in SE linked to same KB entities and concatenate context summary
  for mention-groups $\in [WE, NE]$ **do**
    Run CCR using hierarchical graph partitioning on mention similarity graph based on context summary similarities
  end for
  Merge mention groups and the summaries clustered together by CCR
end while

for mention groups, $M(i) \in WE$ **do**
  $S^i_j \leftarrow$ similarity of $M(i)$ to group $M(J)(\in SE)$, based on context summary similarity, representatives i and j, and KB features of match entity (e_i) if present
  if $S^i_j \geq \theta$ **then**
    $M(i) \leftarrow M(i) \cup M(j)$
  end if
end for

Output mention groups in $\mathbb{CL}$ as *equivalence classes*, C of co-referring mention groups linked to KB entities (or to $null$)

B^3F1 measure; but calculates precision and recall of mention groups using the best 1-to-1 mapping (i.e., mapping with maximum mention overlap) between the

Table 4.1 – Parameter tuning results for $C3EL$ on (a) CCR with δ_s, (b) NEL for out-of-KB entities with δ_w, and (c) CCR with τ.

	(a)					(b)					(c)				
Datasets	δ_s (B^3 F1 %)					δ_w (P %)					τ (B^3 F1 %)				
	0.01	*0.05*	*0.10*	*0.15*	*0.20*	*0.01*	*0.02*	*0.04*	*0.06*	*0.08*	*0.03*	*0.10*	*0.20*	*0.35*	*0.50*
ECB	79.3	82.2	84.2	83.5	81.0	73.1	75.3	77.3	78.7	78.4	76.9	81.2	81.2	81.1	79.2
ClueWeb	70.1	77.2	81.5	81.0	78.7	78.2	81.1	83.6	85.1	85.1	70.3	79.1	78.2	78.8	76.4

resultant equivalence classes and those in the ground truth. Normalization with the number of mentions for each of the resultant classes yields the ϕ_4-CEAF score.

We consider only the 3 most notable mention types: person (PER), location (LOC), and organization (ORG) – accounting for 99.7% of the total entities in ECB corpus (distributed as 45.3%, 32.4%, and 22.3% resp.), and 96.3% of ClueWeb09 corpus (with 52.6%, 29.2%, and 18.2% resp.) – as tagged by the Stanford toolkit. All experiments were run on a 4 core Intel i5 2.50 GHz processor with 8GB RAM and Ubuntu 12.04 LTS OS.

4.7.1 Parameter Tuning & Sensitivity Study

The proposed $C3EL$ framework involves various tunable operational parameters for link validation of entity linkage to KB and their subsequent classification into *confidence classes* (as described in Section 4.3) during the NEL step. In this section, we present the proper setting of the 3 parameters: confidence thresholds δ_s and δ_w, and link validation threshold, τ, tuned based on the *cross-validation* approach with *train* and *test* data subsets. Using the "gold annotations" of the train-set (kept at 30% of total data), the parameter values providing the best precision score are individually learnt for the datasets using *line search* with a small step size of 0.01. Although, such procedures might converge to a local maxima, or be affected by presence of the train-data noise, we later show $C3EL$ to robustly handle such scenarios.

In our experimental setup, we systematically vary the parameter values and observe its effects on $C3EL$ for the training data. With increase in δ_s, the number of mentions mapped to the *Strong Evidence* (SE) class decreases. This in turn limits the influx of external KB features, thus degrading CCR performance as observed in Table 4.1(a). While for low values of δ_s, even weak mention links are placed in SE, leading to a decrease in precision due to noisy KB feature inclusion. On the other hand, a high δ_w value increases the number of mentions placed in the NE class, while low values tends to accumulate mentions in the WE class. This adversely affects the detection of out-of-KB entities due to noise from other KB mentions with similar surface forms (refer Table 4.1(b)) during clustering in CCR step.

The effect of link validation parameter, τ on $C3EL$ has been shown in Table 4.1(c). Similar to the behavior induced by the classification threshold δ_s, we observe that a

Table 4.2 – CCR performance comparison on ECB for $C3EL$.

Approach	**P** (%)	**R** (%)	B^3 **F1** (%)	ϕ_3 **F1** (%)	ϕ_4 **F1** (%)
EECR	74.9	55.5	63.7	-	33.7
$CROCS$	73.11	75.28	74.18	67.35	-
$C3EL$	79.52	82.91	**81.18**	**73.89**	**53.3**

Table 4.3 – Gold Standard CCR results on ECB for $C3EL$.

Approach	**P** (%)	**R** (%)	B^3 **F1** (%)	ϕ_3 **F1** (%)
$CROCS_G$	79.9	83.33	81.58	74.11
$C3EL_G$	84.74	89.9	**87.24**	**80.5**

high τ value limits entity linking and possible KB feature inclusion due to increased rejection of KB linkage. While an extremely low value (near to zero) allows for noisy feature incorporation even for the weakly linked entities – both situations leading to lowered CCR efficiency. However, since τ prevents gross mis-alignment of mentions to KB entities, a wide range of small value $(0.1 - 0.35)$ is seen to provide comparable performance. Hence, for our remaining experiments, we set $\delta_s = 0.11$ and $\delta_w = 0.06$, as seen from the above table and also suggested in [Hoffart *et al.*, 2014]. The parameter τ is set to 0.1, and the similarity threshold for the finalization stage θ is set at $2 \times \delta_s = 0.22$.

4.7.2 CCR Performance Results

We initially benchmark the performance improvement in cross-document co-reference resolution (CCR) procedure by $C3EL$ against two competing approaches:
(1) sampling based hierarchical clustering method with enhanced mention context for CCR only, $CROCS$, as presented earlier in Chapter 3; and
(2) iterative joint entity-event CCR, $EECR$ proposed in [Lee *et al.*, 2012].

Table 4.2 tabulates the results obtained on the ECB dataset. We observe $C3EL$ to decisively outperform both the existing methods, providing a B^3 F1 score improvement of around 7% over $CROCS$ and 17% over EECR. We further attain around 6% $\phi_3 - CEAF$ F1 score enhancement over $CROCS$, and a significant 20% improved $\phi_4 - CEAF$ F1 score compared to EECR.

A. Gold Results: Errors introduced during the pre-processing stage of $C3EL$ (e.g., mention omission, tag mis-classification, intra-document CR errors, etc., by the Stanford CoreNLP toolkit) propagate to subsequent computing stages and might adversely impact the overall system performance of $C3EL$. To provide an unbiased viewpoint of the actual performance of $C3EL$, we manually provided "exact" or "gold" mentions boundaries, mention tags, and intra-document CR mention chains for the ECB corpus; thereby obtaining *gold performance results*. As shown in Table 4.3, we observe a 6% F1 points improvement (for both B^3 & ϕ_3-CEAF F1 score) in $C3EL$ compared to $CROCS$.

Table 4.4 – CCR results for $C3EL$ on ECB for different mention types.

Type	Approach	P (%)	R (%)	B^3 F1 (%)
PER	$CROCS_G$	71.8	74.15	72.96
	$C3EL_G$	84.85	82.73	**83.78**
LOC	$CROCS_G$	78.23	85.41	81.66
	$C3EL_G$	81.41	94.31	**87.29**
ORG	$CROCS_G$	85.73	87.89	86.8
	$C3EL_G$	88.52	91.82	**90.14**

Table 4.5 – CCR results on ClueWeb09-FACC1 for $C3EL$.

Approach	P (%)	R (%)	B^3 F1 (%)	ϕ_3 F1(%)
$CROCS$	68.66	70.96	69.79	62.85
$C3EL$	75.76	81.42	**78.49**	**74.13**

B. Mention Categorization: Person mention type (PER) provides the greatest challenge for CCR systems (compared to other types like LOC, and ORG) due to associated nicknames, titles, and varied surface forms (abbreviations, spellings, etc.). We thus evaluate the CCR performance of $C3EL$ (and compare it with $CROCS$) on the ECB data, with "exact" input mentions, for the different mention categories. Table 4.4 validates that our joint modeling provides better global information cues, reporting a B^3 F1 score enhancement of around 11% over $CROCS$ for the difficult PER mention type; along with improved results for the other mention types as well.

C. Large Data: To study the robustness of $C3EL$ and the effects of large datasets on CCR, we performed evaluations on the huge ClueWeb09-FACC1 dataset. Similar to the ECB dataset, $C3EL$ is seen to exhibit a B^3 F1 score improvement of nearly 10% and a ϕ_3-CEAF F1 improvement of 12% over $CROCS$ (as shown in Table 4.5).

The above experimental results showcase that the proposed interleaved iterative approach helps overcome the challenges faced in CCR by entity linkage and corresponding distant KB feature extraction; thereby improving the overall accuracy.

4.7.3 Named-Entity Linking (NEL) Results

We now benchmark the performance of named entity linking (NEL) procedure for $C3EL$ against state-of-the-art open-source AIDA software [Yosef *et al.*, 2011] (from github.com/yago-naga/aida). We separately inspect the precision of mention linking for *prominent entities* (in-KB) as well as *new/emerging* (out-of-KB) entities, and characterize the links as *Correct* ($\mathbb{C}$), *Incorrect* ($\mathbb{I}$), or *Unlinked* ($\mathbb{U}$). The results on the ECB corpus are reported in Table 4.6. $C3EL$ attains comparable performance ($\sim 85\%$ precision) to that of AIDA for well-known entity-mentions present in KB; albeit with a few mentions remaining unlinked due to our cautious *link validation* (using τ) ap-

Table 4.6 – NEL performance (%) comparison of $C3EL$ on ECB.

Approach	Within-KB			Out-of-KB		Overall
	$\mathbb{C}$	$\mathbb{I}$	$\mathbb{U}$	$\mathbb{C}$	$\mathbb{I}$	**P** (%)
AIDA	86.5	13.5	0.0	63.9	36.1	83.4
$C3EL$	85.4	14.4	0.2	**79.0**	21.0	**84.9**

Table 4.7 – NEL accuracy results (%) for $C3EL$ on ClueWeb09-FACC1.

Approach	Within-KB			Out-of-KB		Overall
	$\mathbb{C}$	$\mathbb{I}$	$\mathbb{U}$	$\mathbb{C}$	$\mathbb{I}$	**P** (%)
AIDA	88.5	10.6	1.0	69.6	30.4	84.6
$C3EL$	89.0	9.8	1.2	**83.7**	16.3	**88.1**

proach. However, the use of τ reduces aggressive KB linking to provide a significant 15% accuracy improvement (over AIDA) in the precise detection of new/emerging entities absent in KB. Overall, an 1.5% precision gain is observed by the joint formulation.

A. Large Data: The diverse nature of the web-scale ClueWeb09 dataset clearly portrays the performance gains in NEL procedure due to CCR generated information integration across documents as reported in Table 4.7. For entities present in the KB, $C3EL$ observes an accuracy improvement of 0.5% over AIDA, while attaining a statistically significant (p-value of < 0.01) 14% improvement in the detection of new/emerging entities absent in the KB, similar to that of the ECB dataset. For a total of nearly 1 million mentions, $C3EL$ provides around 4% overall performance gain over AIDA.

Using a bootstrap re-sampling t-test (as in [Durrett and Klein, 2014]), we observed high statistical significance ($p - value < 0.01$) for Out-of-KB and overall NEL performance of $C3EL$, whereas the difference for Within-KB NEL was not statistically significant. Coping with out-of-KB entities is essential for joint CCR-NEL, and an improved NEL performance using propagated semantic information from CCR along with link validation, enables highly efficient detection of new or emerging entities. Interestingly, $C3EL$ was mainly seen to suffer from co-reference resolution errors, while AIDA suffered from the lack of CR within its framework.

4.7.4 Comparison with Joint Models

Traditional CR methods fail to cope with the heterogeneity of mentions and contexts across multiple documents (depicting lower accuracy), and some form of clustering or joint reasoning over mention contexts across documents is thus necessary. These methods also suffer from quadratic or cubic (sometimes even exponential) computational complexity, and hence running CR-NEL together on a concatenated "super-document" works only for small corpora, and would be prohibitively expensive for large corpora, even in offline processing mode [Singh *et al.*, 2011].

Table 4.8 – Joint "Simulated" CR-NEL result comparison with $C3EL$ on ECB subset for (a) CCR, and (b) NEL.

(a)

Approach	P (%)	R (%)	B^3 F1 (%)
NECo	87.77	82.09	84.84
BER	88.30	86.53	87.41
$C3EL$	87.54	88.11	**87.82**

(b)

Approach	$\mathbb{C}$ (%)	$\mathbb{I}$ (%)	$\mathbb{U}$ (%)
NECo	89.13	10.87	0.0
BER	89.89	10.11	0.0
$C3EL$	**93.2**	**4.61**	2.19

However, to study the behavior of existing CR-NEL joint models under such "small" CCR environments, we compare $C3EL$ with the following methods:
(1) multi-sieve based $NECo$ [Hajishirzi *et al.*, 2013] [5]; and
(2) conditional random field based BER [Durrett and Klein, 2014] [6].

Three topic clusters from the ECB corpus with 3, 4, and 5 articles respectively were randomly uniformly selected, and the documents within each cluster were merged to form three "super-articles" (one per topic), forming a "simulated" CR setting. $NECo$ and BER were then used to perform CR and NEL on these 3 articles, and the results compared with that obtained by $C3EL$ with the original documents as inputs. We repeatedly sample 12 articles across 3 topic clusters, and execute the approaches to report the micro-averaged results across 5 independent runs.

From the results of Table 4.8(a), we observe that the algorithms exhibit comparable performance for entity co-reference resolution; thus validating our intuition that $C3EL$ enables propagation of global semantics due to the joint formulation. However, such CR methods using multi-sieves and CRF do not scale beyond a few documents, and require at least $4\times$ more run-time compared to $C3EL$. Hence, CCR cannot be efficiently tackled by simply employing CR methods on a "super-document".

However, the harnessing of non-local mention features (via CCR) and efficient detection of new mentions using link validation enables $C3EL$ to achieve a gain of around 5% in NEL compared to the others approaches as reported in Table 4.8(b). We observed a statistically significant improvements of $C3EL$ over both BER and NECo with a p-value of < 0.05, using the bootstrap re-sampling t-test.

To further study the effect of larger corpus in such scenarios, we sampled 25 documents (with co-referring mentions) from the ClueWeb09 dataset and performed analysis among the algorithms. As previously, we observed significant computational complexity for traditional CR methods when applied to CCR setting making them far slower ($6\times - 7\times$) than $C3EL$. Table 4.9 reports the CCR and NEL averaged results obtained across 5 independent runs. We attained comparable performance in CCR with around 3% improvement in NEL. All the algorithms are seen to achieve high NEL results due to the large presence of well-known (in-KB) entities.

5. obtained from cs.washington.edu/research-projects/nlp/neco
6. obtained from nlp.cs.berkeley.edu/projects/entity.shtml

Table 4.9 – Joint "Simulated" CR-NEL performance comparison of $C3EL$ on ClueWeb09 subset for (a) CCR, and (b) NEL.

(a)

Approach	P (%)	R (%)	B^3 F1 (%)
NECo	81.14	79.65	80.39
BER	84.36	83.01	83.68
$C3EL$	83.52	85.56	**84.53**

(b)

Approach	$\mathbb{C}$ (%)	$\mathbb{I}$ (%)	$\mathbb{U}$ (%)
NECo	94.71	5.29	0.0
BER	95.27	4.73	0.0
$C3EL$	**98.23**	**1.5**	0.27

Table 4.10 – CCR and NEL results (%) on ECB for different baseline variations of $C3EL$.

Baselines	CCR result			NEL results				
				Within-KB			*Out-of-KB*	
	P	R	B^3	$\mathbb{C}$	$\mathbb{I}$	$\mathbb{U}$	$\mathbb{C}$	$\mathbb{I}$
Ignored Mention Co-occurrence	72.5	74.4	73.4	80.2	19.6	0.2	74.4	25.6
Link Validation (τ) ignored	79.0	81.4	80.2	**85.5**	14.5	**0.0**	62.8	37.2
Removed NEL Classification	73.2	80.7	76.8	83.9	15.9	0.2	76.1	23.9
Distant KB feature dropped	68.9	73.1	70.9	82.8	17.0	0.2	**79.0**	**21.0**
$C3EL$	**79.5**	**82.9**	**81.18**	85.4	**14.4**	0.2	**79.0**	**21.0**

4.7.5 Algorithmic Baseline Study

We now explore the performance of several baseline variants of $C3EL$ ablating various system components. Tables 4.10 and 4.11 report the obtained results on ECB and ClueWeb09 respectively. Explicitly, we study the effects of the following modules:

- **Co-occurring Mentions:** Removal of co-occurrence mention contexts during the creation of mentions' *context summaries* reduces the semantic information content for disambiguation and hence adversely affects both NEL and CCR procedures. We thus observe a sharp decrease in CCR performance and also a degradation in NEL.
- **Link Validation:** Filtering of mention linking to KB entities using link validation step (with threshold τ) in $C3EL$ enables corroboration of mention context keywords with the linked entity features. This leads to enhanced detection of new or emerging entities by reducing induction of noise during the CCR phase. Removal of this process permits aggressive entity linking and introduces noise, affecting new/emerging entity detection. From the above tables, on removal of link validation step, we observe nearly 20% reduction of accuracy (on both datasets) in the identification of out-of-KB entities compared to $C3EL$.
- **NEL Categorization:** The differentiation of mentions into classes (during NEL) using mapping confidence to KB entity reduces the collusion of "strong" linked mentions with other "noisy" mention contexts. This reduces incorrect grouping of different mentions with similar surface forms and contexts, thereby improving the precision of CCR. Elimination of the classification approach is observed to degrade the CCR

Table 4.11 – CCR and NEL results (%) on ClueWeb09 for different baselines of $C3EL$.

Baselines	CCR result			NEL results				
				Within-KB			*Out-of-KB*	
	P	R	B^3 F1	$\mathbb{C}$	$\mathbb{I}$	$\mathbb{U}$	$\mathbb{C}$	$\mathbb{I}$
Ignored Mention Co-occurrence	69.3	72.2	70.7	83.8	14.6	1.6	80.6	19.4
Link Validation (τ) ignored	74.8	81.0	77.8	**88.9**	**10.1**	**1.0**	69.8	30.2
Removed NEL Classification	70.1	77.6	73.6	86.1	12.3	1.6	79.5	20.5
Distant KB feature dropped	66.4	72.9	69.5	85.4	13.0	1.6	**83.7**	**16.3**
$C3EL$	**75.8**	**81.4**	**78.5**	88.3	**10.1**	1.6	**83.7**	**16.3**

results, which in turn increases spurious entity linkage, decreasing NEL efficiency.

– **Distant KB features:** As observed by [Baker, 2012; Zheng *et al.*, 2013], extracted external KB features provide global and enhanced information cues promoting CR. We similarly observe CCR to attain the lowest F1 scores (compared to other baselines) when the KB features are ignored. This in turn affects the linking of (some) well-known entities due to reduced context, leading to incorrect or low confidence NEL. Since no feature inclusion is performed for out-of-KB mentions (due to failed link validation), no effect is observed for such entities.

Discussion: Hence, from the above empirical setup and evaluations, we observe that the joint CCR-NEL formulation in $C3EL$ encompassing multiple information sources (from source text and external KB) and noise filtering (by link validation) enables global information propagation across the iterations, thereby providing mutually enhanced performance for both CCR and NEL.

4.8 Summary

This chapter presented the novel $C3EL$ framework, the first approach for joint computation of cross-document co-reference resolution (CCR) and named entity linking (NEL). Our approach utilizes *context summaries* including co-occurring mention context and external KB features allowing for global context and feature propagation across documents and *link validation* for precise detection of out-of-KB entities. The iterative approach embedded in the interleaved CCR-NEL stages enables information feedback between CCR (providing corpus-wide information cues) and NEL (providing distant KB features) for enhanced performance of both CCR and NEL tasks, along with highly accurate new or emerging entity identification. Experimental results on large news and Web data demonstrate robustness and performance gains of our framework compared to existing methodologies.

5 Credibility of Entity-centric Texts

Knowledge base (KB) construction entails the efficient representation of extracted entity related facts and relationships. The quality of information pertaining to the entities, obtained from the input corpus thus plays a pivotal role in the overall applicability of the KB. In this setting, the precise categorization of entity-centric textual information as *credible* or *non-credible* provides a significant challenge, given the diversity and subtle introduction of possibly spam, irrelevant, biased, and fake information.

This chapter presents a novel language and temporal model based methodology to efficiently identify fact/information demonstrating low credibility. The proposed method harnesses features derived from texts, associated user/reader ratings and sentiments, and publication timestamps for leveraging classifiers to label extracted textual snippets as credible or not. Experimental results on large real-life datasets demonstrate significant classification accuracy improvements over state-of-the-art approaches.

5.1 Introduction

Motivation. The extraction of accurate and meaningful entity based facts and entity-entity relationships from news articles, blogs, and forums forms the bedrock of large knowledge base construction procedures and applicability to search related applications.

Given the vast amount of data generated across diverse domains and the popularity of social media, there has been an unfortunate increase in the proportion of *non-credible* documents and facts – either fake (aimed at promotion/demotion of entities), incompetent (irrelevant), biased (distorted), or for sensationalization. In fact, recent studies [1] found that majority web users tend to share news and information without reading or verifying article details. This serious trend is even more pronounced in

1. yackler.ca/blog/2016/07/09/scientists-say-giant-asteroid-hit-earth-next-week-causing-mass-devastation

potential customer-centric domains such as product, service, and travel review forums such as TripAdvisor, Yelp and Amazon, – wherein manipulative/deceptive item reviews amounted to nearly 20%, with a further 16% of users reviews deemed as "not-recommended" by Yelp [Luca and Zervas, 2015]. Hence, entity-centric facts extracted (and represented in KBs) from such sources might severely degrade the reliability of KB, necessitating the identification of potentially *non-credible* information.

Several approaches geared towards *fact checking* [Metzger and Flanagin, 2013; Wu *et al.*, 2014] have been proposed to alleviate the problem. In this work, we aim to assess the credibility of natural language texts before fact extraction, which can then be further combined with existing fact verification methodologies to provide a robust framework towards clean knowledge repository construction. Owing to the scarcity of credibility based labeled data in most domains, this work primarily focuses on the detection of deceptive review texts present in customer-oriented product/service review portals such as TripAdvisor and Yelp. We later show that the proposed method is domain-independent and hence can easily be transferred to other text based scenarios, thereby addressing the current lack of labeled training data.

Existing research on this topic has cast the problem of review credibility into a binary classification task: a review is either credible or deceptive. To this end, supervised and semi-supervised methods relying on features about users, entities, and activities have been proposed [Jindal and Liu, 2008]. However, information about user histories and activities are not always available in many scenarios, for example in cases of "long tail" items or users. On the other hand, language-based approaches [Mihalcea and Strapparava, 2009; Ott *et al.*, 2011; Ott *et al.*, 2013] consider word-level unigrams or bigrams as features to learn latent topic models and classifiers (e.g., [Li *et al.*, 2013]). User activity and their behavioral deviation from the mean/majority ratings have also been used by the industry [Mukherjee *et al.*, 2013a], but it tends to over-emphasize trusted long-term contributors and suppress outlier opinions. All these approaches employ several aggregated metadata, and are thus hardly viable for cross-domain adaptation and for new items with very few reviews – often by not so active users or newcomers in the community.

Problem Statement. In this chapter, we aim to efficiently detect *non-credible* entity review texts *with limited information* in the absence of rich data about user histories, community-wide correlations, and for "long tail" items (with sparse review texts and ratings), thereby providing domain-independence. Interestingly, prior methods shown to achieve high classification accuracy, do not provide any interpretable evidence as to why a certain review is classified as non-credible. Our goal is then to not only compute a *credibility score* for review texts but also to provide possibly *interpretable evidence* for explaining why certain reviews have been categorized as non-credible.

5.1.1 Approach and Contributions

The proposed method efficiently performs *credibility analysis* of entity reviews by exploiting *inconsistencies* across features derived from the user item review sentiments and the corresponding item ratings. Further, temporal "burst" features – where a number of extreme reviews are written within a short span of time – are also fed to Support Vector Machines for obtaining credibility scores for reviews and identifying possible causes leading to a review being categorized as deceptive.

To this end, the novel components of the proposed approach are:
- a classification model based on extracted feature vectors from limited item-user metadata across items and users, to compute review credibility score for detecting non-credible reviews.
- a novel notion of *interpretable evidence* for entity texts based on language models, sentiment, timestamp, and rating to possibly characterize as to why a review is deemed deceptive.

The above features are used to identify, score, and highlight inconsistencies that may appear between reviews, ratings, and the community's overall characterization of an item, for classifying item reviews as credible or otherwise. In a nutshell, the major contributions of this chapter are as follows:
- A novel *consistency model* for credibility analysis of reviews that works with limited information, with particular attention to "long tail" items, and offers interpretable evidence for reviews classified as non-credible (Section 5.3);
- investigate how credibility scores and the learnt model can be transferred across different domains and communities thereby addressing the scarcity of labeled training data (Section 5.3.4); and
- experimental evaluation on TripAdvisor and Yelp datasets to demonstrate the viability and advantages of the proposed method over state-of-the-art baselines in terms of classification accuracy and providing interpretable evidence (Section 5.4).

5.2 Related Work

Existing approaches for fake review and opinion spam detection primarily focused on two different aspects of the problem:

Linguistic Analysis [Mihalcea and Strapparava, 2009; Ott *et al.*, 2011; Ott *et al.*, 2013] – These approaches exploit the distributional difference in the wordings of authentic and manually-created fake reviews using word-level features to learn latent topic models and classifiers [Li *et al.*, 2014b]. However, the artificially created fake review datasets (by Amazon Mechanical Turks) for the studied tasks were shown to give away explicit features not dominant in real-world data. This was confirmed by a study on Yelp filtered reviews [Mukherjee *et al.*, 2013a], where the n-gram word-level language features along

with specific lexicons (e.g., LIWC psycholinguistic lexicon [Pennebaker *et al.*, 2001] and WordNet Affect [Strapparava and Valitutti, 2004]) performed poorly. Additionally, linguistic features such as *text sentiment* [Yoo and Gretzel, 2009], *readability score* (e.g., Automated readability index (ARI), Flesch reading ease, etc.) [Hu *et al.*, 2012], *textual coherence* [Mihalcea and Strapparava, 2009], and rules based on *Probabilistic Context Free Grammar* (PCFG) [Feng *et al.*, 2012] have also been studied.

Rating and Activity Analysis – In the absence of proper ground-truth data, prior works make simplistic assumptions about non-credibility, e.g., duplicates and near-duplicates are fake, and make use of *extensive* background information like brand name, item description, user history, IP addresses and location [Jindal and Liu, 2007; Jindal and Liu, 2008; Lim *et al.*, 2010; Wang *et al.*, 2011; Mukherjee *et al.*, 2012; Mukherjee *et al.*, 2013b; Mukherjee *et al.*, 2013a; Li *et al.*, 2014a; Rahman *et al.*, 2015] to train regression models on extracted features to classify reviews as credible or deceptive. Similar works in this area also consider ad-hoc features like extreme ratings, user activity (number of posts, friends etc.), review length, rating deviation from community mean, burstiness, and simple language features (like content similarity, presence of literals, numerals, capitalizations, and POS tags) for learning models. Although such approaches perform quite well in practice, the use of extensive and aggregated features limits their application to a broader domain due to lack of related information. In contrast to these works, our approach uses limited information about users and items to construct several consistency features harvested primarily from user ratings and review texts only, thereby catering to a broad domain of applications. Further, none of the existing approaches provide interpretation as to why a review should be deemed non-credible – which we aim to tackle in this chapter.

Learning to Rank – Supervised models have also been developed to rank items from constructed item feature vectors [Liu, 2009]. Such techniques optimize measures like Discounted Cumulative Gain, Kendall-Tau, and Reciprocal Rank to generate item rankings similar to the training data based on the feature vectors. As such, a related area of study involves the re-ranking of items according to their proper rating, wherein the "credible" reviews are implicitly gauged based on the constructed feature vectors.

5.3 Entity Review Text Credibility Analysis

This section presents the *language* and *behavioral* models of the proposed framework for constructing the consistency features vectors for training a Support Vector Machine (SVM) to classify the input texts as *credible* or *deceptive*. To this end, we primarily focus on entity-centric review texts obtained from product/service based consumer portals.

5.3.1 Language Model

Previous works [Mihalcea and Strapparava, 2009; Ott *et al.*, 2011; Ott *et al.*, 2013; Chen and Chen, 2015] in linguistic analysis explore distributional difference in the wordings between deceptive and authentic reviews. In general, authentic reviews tend to have more *sensorial and concrete language* than deceptive reviews, with higher usage of nouns, adjectives, prepositions, determiners, and coordinating conjunctions; whereas deceptive reviews were shown to use more verbs, adverbs, and superlatives manifested in exaggeration for imaginary writing. Ott et al. [Ott *et al.*, 2011; Ott *et al.*, 2013] found that authentic hotel reviews are more specific about spatial configurations (small room, low ceiling, etc.) and aspects like location, amenities, and cost; whereas deceptive reviews focus on aspects external to the item being reviewed (like traffic jam, children, business, and vacation). Extreme opinions were also found to be dominant in deceptive reviews to assert stances, whereas authentic reviews have a more balanced view analyzing the item on several aspects.

In order to explicitly capture such distributional difference in the language model of credible and non-credible reviews at word-level, we capture unigram and bigram language features shown to outperform other fine-grained linguistic features using psycholinguistic features (e.g., LIWC lexicon) and Part-of-Speech tags [Ott *et al.*, 2011]. Specifically, the bigram and unigram features depicting context-dependent information performed quite well. Further, lexicons from WordNet Affect were used to capture fine-grained emotional dimensions (like anger, hatred, and confidence) present in reviews. The presence or absence of such words were used as features in the model, and all the features were length normalized, retaining punctuations (like '!') and capitalization, as non-credible reviews manifesting exaggeration tend to over-use the latter (e.g., "the hotel was AWESOME !!!").

Language-based feature vector: Consider a vocabulary V of unique unigrams and bigrams in the corpus (after removing stop words). For each token type $f_i \in V$ and each review d_j, we compute the presence/absence of words, w_{ij}, of type f_i occurring in d_j, thus constructing a feature vector $F^L(d_j) = \langle w_{ij} = I(w_{ij} = f_i) \;/\; length(d_j) \rangle, \forall i$, with $I(.)$ denoting an indicator function.

Sentiment-based feature: To characterize the overall polarity of a review, the user opinion sentiment on an item extracted from the review texts are further provided as features for training the classification model. This approach helps to identify possible inconsistencies between reviews and rating, e.g., a poor rating with no discussion or inadvertent mistakes by users to gauge the rating model (i.e., a rating of 1.0 is good or bad). It also highlights stark differences in review sentiments from the majority, thereby identifying likely candidates for *spam* or *non-credible* review. For example, given review snippets like "the hotel offers free wi-fi", we aim to find the sentiment polarities. Interestingly, although the unigram "free" does not have a polarity of its

own, in the above example "free" in conjunction with "wi-fi" expresses a positive sentiment of a service being offered without charge. The hope is that although "free" does not have an individual polarity, it appears in the neighborhood of words (from language model) that have known polarities (from external lexicons). The proposed model adopts the Joint Sentiment Topic Model approach (JST) [Lin and He, 2009; Mukherjee *et al.*, 2014] to discover the expressed polarities for review snippets extracted from the text. As such, the review texts "free wi-fi" and "internet without extra charge" should ideally be mapped to a single cluster with similar polarities using their co-occurrence with similar words with positive polarities.

JST assumes each review (document), d to be associated with a multinomial distribution θ_d over word clusters z and sentiment labels l with a symmetric Dirichlet prior, where $\theta_d(z,l)$ denotes the probability of sentiment polarity l for cluster z in document d. Thus, in the generative process, a sentiment label l is added to word w from a document-specific word distribution π_d with a symmetric Dirichlet prior. Formally, given a set of reviews $\langle D \rangle$ written by users $\langle U \rangle$ on a set of items $\langle I \rangle$, each word (drawn from vocabulary V) in review d (denoted by sequence of words $\{w_1, w_2, \cdots\}$) is assigned a sentiment label from $l = \{l_1, l_2, \cdots\}$. The obtained word-sentiment distribution for each review is then provided as a classification feature for model training.

5.3.2 Behavioral Model

Based on earlier works [Jindal and Liu, 2007; Jindal and Liu, 2008; Lim *et al.*, 2010] on review spam, we harness *user-dependent models* for detecting user-preferences and biases towards product review and rating. However, extensive information about users is not always available, especially for newcomers and not so active users. Further spammers tend to open multiple fake accounts for malicious activities and use such accounts sparsely to avoid detection [Mukherjee *et al.*, 2012; Mukherjee *et al.*, 2013b]. Since this chapter proposes a robust credibility detection framework also geared for "long tail" users and items, instead of relying on extensive user history, simple proxies for user activity easier to aggregate from the community have been used as:

1. **User Posts:** number of posts written by the user in the community – users active in the community provide credibility based on "track record".
2. **Review Length:** length of the reviews – longer reviews tend to frequently go off-topic with high emotional digression.
3. **User Rating Behavior:** absolute deviation of the review rating from the mean and median rating of the user to other items, as well as the first three moments of the user rating distribution, capturing typical user *rating behavior* across items.
4. **Item Rating Pattern:** absolute deviation of an item rating from the mean and median rating obtained from other users captures the extent to which the user disagrees with other users about the item quality; the first three moments of the

item rating distribution captures the general item rating pattern.

5. **User Friends:** number of friends of the user – users with greater communication circle tend to demonstrate more credibility.
6. **User Check-in:** if the user has actually used the product to provide reviews – first hand experience of the user adds to the review credibility.
7. **Elite:** elite status of the user, capturing higher credibility of user's opinions.
8. **Review helpfulness:** number of helpfulness votes received by the user post – captures the quality of user postings.

To model credibility for domain-specific scenarios where even more limited information is available, the above behavioral features are partitioned into two components:
(a) $Activity^{-}$ using features $[1-4]$ that can be obtained straightforward from the tuple $\langle userId, itemId, review, rating\rangle$ and are easily available across domains even for "long tail" items and newcomers; and
(b) $Activity^{+}$ using all the listed features, possibly requiring additional information (for features $[5-8]$) not always available or takes long time to aggregate.

Behavior-based feature vector: For each review d_j by user u_k, using the above dimensions, a behavioral feature vector $\langle F^B(d_j)\rangle$ is constructed for the classification task.

5.3.3 Consistency Model

In order to provide *interpretable* evidence as to why a review text has been deemed *non-credible*, the proposed model detects the following *inconsistencies* based on an item review and rating for credibility analysis. The reasoning as to why the text is likely deceptive, not only enables the end users to decide on the relevance of the review but also provides insights into the language/behavioral pattern in such deceptive reviews for further improvements in their detection.

1. **User Review – Rating:** The user-assigned rating for an item should be consistent with the opinion or sentiment expressed in the corresponding review text. For example, a user is unlikely to give a poor rating to an item simultaneously with a positive opinionated review about the item. Such scenarios would indicate the presence of inadvertent errors or credibility issues of the review-rating.

The sentiment expressed in word level n-grams of a review text is obtained from JST and lexicons, and the inferred rating distribution $\pi_d^{'}$ (with dimension L) of a review d consisting of a sequence of words $\{w\}$ and learned Φ (word cluster-label distribution obtained from JST) is computed. For each word, the sentiment label that jointly maximizes the word cluster-label distribution is considered, and then aggregated over all words present in the text. The absolute deviation (across L dimensions) between the actual user rating π_d, and estimated rating $\pi_d^{'}$ forms a component of the overall classification feature vector.

2. **User Rating Behavior:** Previous works [Ott *et al.*, 2011; Hu *et al.*, 2012; Sun *et al.*, 2013] on opinion spam found that deceptive reviews tend to have overtly positive or overtly negative opinions. Hence, the rating distribution for such users tends to depict mass concentration at extreme ratings – providing interesting cues as to the credibility of such review-ratings. Therefore, the user rating distribution is also considered as a component of the overall feature vector.

3. **Temporal Burst:** This characteristic is typically observed in *group spamming*, wherein a number of reviews are posted targeting an item in a very short span of time. Considering a set of reviews $\{d_j\}$ at timepoints $\{t_j\}$ posted for a *specific* item, the presence of temporal burstiness of review d_i within a window of timepoints (e.g., number of reviews posted within a month) for the given item can be obtained as $\left(\sum_{j,j\neq i} \frac{1}{1+e^{t_i - t_j}}\right)$. Here, an exponential decay attribute is considered to model the temporal proximity of reviews, thereby capturing the presence of burst within the considered timeframe window. The presence/absence of such burstiness is then used as a feature for our classification model.

4. **User Review Deviation:** In general, the description of the item outlined in a user review should not differ much from that of the majority, or from the features described in the product summary. For example, if majority says the "hotel offers free wi-fi", and a user review says "internet is charged" – this presents a possible inconsistency. To capture such discrepancies for detecting possible deceptive reviews, the deviation between the language/sentiment models obtained from a user review and the majority reviews about an item is provided as a feature to SVM for classification. Here, we exploit the fact that non-credible reviews generally form only a small fraction of all the reviews on an item. In our current setting, the Jensen-Shannon divergence, a symmetric and smoothed version of the Kullback-Leibler divergence is used as the deviation measure to depict how much the distribution in a given review diverges from the general opinion of other people about the item.

Consistency-based feature vector: For each review d_j, the above *consistency features* are computed and represented as feature vector $\langle F^T(d_j)\rangle$ for the classification task.

5.3.4 Credibility Detection Framework

The different models (discussed previously) are intelligently constructed from a varied set of signals and are cast as feature vectors for credibility related application tasks as:

Credible Review Classification: The main task of the current work focuses on *classifying* or identifying reviews as *credible* or not. For each review d_j by user u_k, a joint feature vector $F(d_j) = F^L(d_j) \cup F^B(d_j) \cup F^T(d_j)$ is constructed, and SVM [Cortes and Vapnik, 1995] is used for classification of the reviews. SVM maps the input (using kernels) to a high dimensional space, and constructs a hyperplane to separate the

two categories of elements. Although there can be an infinite number of such hyperplanes possible, SVM constructs the one with the largest functional margin given by the distance of the nearest point to the hyperplane on each side of it. New points are mapped to the same space and classified to a category based on which side of the hyperplane it lies. A linear kernel was seen to perform better than other kernels like polynomial, radial basis, and sigmoid for the classification. Hence, we use the L_2 regularized L_2 loss SVM with dual formulation from the LibLinear package (obtained from csie.ntu.edu.tw/cjlin/liblinear) [Fan *et al.*, 2008] with other default parameters. We report the classification accuracy with 10-fold cross-validation on ground-truth annotated data from TripAdvisor (synthetic dataset from Turks) and Yelp (marked as "not-recommended").

Credibility Detection under Domain Transfer

A typical and pressing issue in credibility analysis task is the scarcity of labeled training data, leading to the development and performance analysis of novel methods for tackling the credibility detection problem. Existing methods demonstrating high accuracy on benchmark datasets tend to be domain-dependent (usage of domain-dependent features) and are hence unable to be transferred to other applications. In this section, we aim to alleviate this problem and showcase that the credibility model proposed in this chapter is fairly domain-independent and can be easily re-trained to provide highly accurate performance even when trained and tested on different domains. Specifically, we discuss how the labels from Yelp Spam Filter (considered to be the industry standard) can be used to re-train our model (based on the feature vectors previously constructed) for applicability to other communities.

Although, in principle, the trained model M_{Yelp} on Yelp can be directly used to filter out non-credible reviews in other domains, transferring the learned model to other scenarios assumes the learnt weights of *features* to be analogous to that in the new application, and hence encounters the following issues:

– The word and label language distributions of Yelp reviews pertaining to food and restaurants are different from that of other communities such as Amazon involving software, consumer electronics, etc., and hence the previously learnt feature weights from Yelp cannot be directly used, as the latent dimensions are different.
– Additionally, specific metadata like check-in, user-friends, and elite-status might be missing for other domains.

However, the learnt weights for the following features can be directly used:

– Certain unigrams and bigrams, especially depicting opinion, occurring in both domains.
– Behavioral features like user and item rating patterns, review count and length, and usefulness votes.
– Temporal burstiness, as a unary feature, of the review.

– Deviation features derived from *domain-specific* review sentiment distribution obtained using the JST model:
  1. Deviation (with dimension L) of the user assigned rating from that inferred from review content.
  2. Distribution (with dimension L) of positive and negative sentiment as expressed in the review.
  3. Divergence, as a unary feature, of the sentiment label distribution in the review from the aggregated distribution over other reviews on a given item.

The above components, common across domains, can be used to re-train the model M_{Yelp} from Yelp to remove the non-contributing features. A direct transfer of the model weights from Yelp assumes the distribution of credible to non-credible reviews and corresponding feature importance to be the same in both domains, which is not necessarily true. In order to boost certain features to better identify non-credible reviews pertaining to diverse text sources, the *soft margin parameter* C in the SVM is tuned by using *C-SVM* [Chen *et al.*, 2004], with slack variables, that optimizes:

$$min_{\vec{w},b,\xi_i \geq 0} \ \frac{1}{2}\vec{w}^T\vec{w} + C^+ \sum_{y_i=+1} \xi_i + C^- \sum_{y_i=-1} \xi_i$$
$$\text{subject to } \forall\{(\vec{x_i}, y_i)\}, y_i(\vec{w}^T\vec{x}_i + b) \geq 1 - \xi_i$$

C^+ and C^- are regularization parameters for positive and negative class (credible and deceptive), respectively, and provide a trade off as to how wide the margin can be made by moving around certain points which incurs a penalty of $\{C\xi_i\}$. A high value of C^-, for instance, places a large penalty for mis-classifying instances from the negative class, and therefore boosts certain features from that class. As the value of C^- increases, the model starts classifying more reviews as non-credible. In the worse case, all the reviews of an item are classified as non-credible, leading to the aggregated item rating being zero. The procedure of cross-validation can then be performed to find the optimal value of C^- by varying it across a *validation set*. With increase in C^-, more non-credible reviews are filtered out, after which it stabilizes, providing the parameter for re-training the learnt model on the new domain.

Discussion: In this context, it can be observed that the proposed model is amenable to re-training and hence can easily be adapted for credibility detection of textual information across varied domains, such as news article, blogs, and other text documents wherefrom harvesting entity-centric facts and relations for knowledge base management forms the main focus of this dissertation. The increase in modern online new articles, blogs, and other documents posted and/or shared in online communities, has enabled the engagement of users in such forums (in terms of up/down votes, comments, etc.). As such, extraction of language and behavioral based features from the author as well as the participating users enables the construction of our model features for credibility analysis. Further, external trusted knowledge sources such as Wikipedia

Table 5.1 – Dataset statistics for credible review classification. (Yelp* denotes balanced dataset obtained by random sampling)

Dataset	Non-Credible Reviews	Credible Reviews	Items	Users
TripAdvisor	800	800	20	-
Yelp	5169	37,500	273	24,769
Yelp*	5169	5169	151	7898

can be tapped in to incorporate information divergence as additional classification feature for such domains. Hence, the classification model can be adapted for diverse domains, enabling the extraction of precise and accurate information for harvesting and representation in large knowledge bases.

5.4 Experimental Evaluation

This section empirically studies the performance of the proposed credibility detection framework on real datasets and compares it to several state-of-the-art baselines.

Datasets and Ground-Truth: To evaluate the performance accuracy of the competing methodologies in detecting non-credible reviews, we consider the following three real-life datasets with available ground-truth information. Table 5.1 reports the characteristics of the datasets.

– **TripAdvisor Dataset** [Ott *et al.*, 2011; Ott *et al.*, 2013]: consists of 1600 reviews from TripAdvisor with positive (5 star) and negative (1 star) sentiment – containing 20 credible and 20 non-credible reviews for *each* of 20 most popular hotels in Chicago. Reviews crawled from the online portal of TripAdvisor were marked as *credible*, whereas the *non-credible* ones were generated by users in Amazon Mechanical Turk. This balanced dataset only contains the review text and positive/negative item ratings with corresponding hotel names, with no other information on users or items – providing *limited information* for credibility analysis.

– **Yelp Dataset**: consists of $37.5K$ recommended (i.e., *credible*) reviews, and $5K$ non-recommended (i.e., *non-credible*) reviews as annotated by the Yelp filtering algorithm, on 273 restaurants in Chicago. For each review, the following information tuple is extracted to construct the features of the different analysis models for classification: $\langle userId, itemId, timestamp, rating, review, metadata \rangle$ (as discussed in Section 5.3), where meta-data consists of user activity information as outlined in Section 5.3.2. The reviews marked as "not recommended" by the Yelp spam filter are considered to be the ground-truth for comparing the accuracy for credible review detection for our proposed model. The Yelp spam filter presumably relies on linguistic, behavioral, and social networking features as studied in [Mukherjee *et al.*, 2013a].

Comparison Baselines: We use the following state-of-the-art baselines (given the full set of features that fit with the approaches) for comparing the credibility detection accuracy performance of the proposed model with competing approaches.

– *Language Model Baselines:* We benchmark our model against several existing language-based approaches as:
(1) The unigram and bigram language model presented in [Ott *et al.*, 2011; Ott *et al.*, 2013] have been shown to outperform other approaches using psycholinguistic features, part-of-speech tags, information gain, etc., and forms the best language based approach for review credibility classification. In contrast, our proposed model enriches the above unigram/bigram model by using length normalization, distributional deviation features, consistency features, and sentiments.
(2) The recently proposed *doc-to-vec* approach based on Neural Networks, overcomes the weakness of bag-of-words models by taking the context of words into account to learn a vector representation for documents [Le and Mikolov, 2014].
(3) In addition, several other features such as readability (ARI) and review sentiment scores [Hu *et al.*, 2012] have been proposed under the hypothesis that writing styles would be random because of diverse customer background, whereas deceptive reviews and group spamming would deviate from such styles. ARI measures the reader's ability to comprehend a text and is measured as a function of the total number of characters, words, and sentences present, while review sentiment tries to capture the fraction of occurrences of positive/negative sentiment words to the total number of such words used.

– *Activity & Rating Baselines:* Given the tuple $\langle userId, itemId, rating, review, metadata \rangle$, several user and item related features have been extracted for using activity and rating behavioral features of users for classification as proposed in [Jindal and Liu, 2007; Jindal and Liu, 2008; Lim *et al.*, 2010; Wang *et al.*, 2011; Mukherjee *et al.*, 2012; Mukherjee *et al.*, 2013a; Mukherjee *et al.*, 2013b; Li *et al.*, 2014a]. Further, the number of helpful feedbacks, review title length, review rating, use of brand names, percent of positive and negative sentiments, average rating, and rating deviation have also been harnessed as features. Rahman et al. [Rahman *et al.*, 2015] recently proposed the use of user check-in and user elite status information as additional features. The above aggregated feature space forms the state-of-the-art methodology (also presumed to be used by industry) for review credibility detection.

Parameter Initialization: The use of sentiment-aware language model forms an important aspect for the credibility detection in our framework – enabling the identification of review-rating inconsistencies as well as user deviation within the community. In our experimental setup, the sentiment lexicon from [Hu and Liu, 2004] consisting of 2006 positive and 4783 negative polarity bearing words is used to initialize the review text based label-word distribution prior to inference. The number of word clusters considered by the JST procedure was set at 20 for Yelp with the review sentiment labels $L = \{+1, -1\}$ (for positive/negative reviews) initialized randomly. The symmetric Dirichlet priors for training/inference stages were set to default values.

Table 5.2 – Credible review classification accuracy with 10-fold cross validation. (TripAdvisor dataset contains only review texts and no user/activity information)

Models	Features	TripAdvisor	Yelp*
Deep Learning	Doc2Vec	69.56	64.84
	Doc2Vec + ARI + Sentiment	76.62	65.01
Activity & Rating	Activity+Rating	-	74.68
	Activity+Rating+Elite+Check-in	-	79.43
Language	Unigram + Bigram	88.37	73.63
	Consistency	80.12	76.5
Behavioral	Activity Model^{-}	-	80.24
	Activity Model^{+}	-	86.35
Aggregated	N-gram + Consistency	**89.25**	79.72
	N-gram + Activity^{-}	-	82.84
	N-gram + Activity^{+}	-	88.44
	N-gram + Consistency + Activity^{-}	-	86.58
	N-gram + Consistency + Activity^{+}	-	**91.09**

5.4.1 Review Credibility Classification

This section studies the performance of the various approaches for the task of credible review classification, i.e., distinguishing a *credible* review from a *non-credible* one. For review credibility classification, inherently a binary classification task, we consider a balanced dataset containing equal proportion of data from each of the two classes (credible and non-credible) in the current experimental setup. The TripAdvisor review dataset is already balanced by construction, while for the Yelp dataset, an equal number of credible and non-credible reviews from the annotated dataset (recommended and not recommended) provided by Yelp is constructed by uniform and random sampling (to obtain the balanced Yelp* dataset). Observe that for the reported results, the TripAdvisor dataset had *only* review texts, and hence the rating and user/activity models could *not* be used.

Table 5.2 shows the 10-fold cross validation accuracy results for the different models on the two datasets. We observe that our proposed model using the various *consistency and behavioral features* exhibit significant classification accuracy improvements on the datasets over the best performing existing baselines. Specifically, we observe,

Language Model: The bigram language model is observed to perform very well on the TripAdvisor dataset due to the setting of the task. The Amazon Mechanical Turk workers were tasked with writing fake reviews (for this dataset) with the guideline of knowing all the hotel amenities in its website before writing reviews, and hence the consistency features are seen to provide marginal improvements (1%) in combination with the language model, due to the absence of stark contradictions or mismatches in item reviews and ratings.

Table 5.3 – Top n-grams (by feature weights) for credibility classification.

Credible Reviews	Non-Credible Reviews
not, also, really, just, like, get, perfect, little, good, one, space, pretty, can, everything, come_back, still, us, right, definitely, enough, much, super, free, around, delicious, no, fresh, big, favorite, lot, selection, sure, friendly, way, dish, since, huge, etc, menu, large, easy, last, room, guests, find, location, time, probably, helpful, great, now, something, two, nice, small, better, sweet, though, loved, happy, love, anything, actually, home	dirty, mediocre, charged, customer_service, signature_lounge, view_city, nice_place, hotel_staff, good_service, never_go, overpriced, several_times, wait_staff, signature_room, outstanding, establishment, architecture_foundation, will_not, long, waste, food_great, glamour_closet, glamour, food_service, love_place, terrible, great_place, never, wonderful, atmosphere, signature, bill, will_never, good_food, management, great_food, money, worst, horrible, manager, service, rude

On the other hand, the real-world Yelp dataset exhibits more noise. As such, the bigram language model and doc-to-vec do not perform as well compared the previous dataset; and neither does the consistency model in isolation. However, for all the component models and features put together, a significant performance improvement (around 8%) is attained. Incorporation of writing style using only the ARI and sentiment measures improves performance of doc-to-vec approach on the TripAdvisor dataset, but not significantly in the real-world Yelp data.

Table 5.3 shows the a snapshot of the top unigrams and bigrams contributing to the language feature space in the *joint model* for credibility classification – given by the feature weights of the SVM. We find that the credible reviews contain a mix of function and content words, balanced opinions, with the highly contributing features being mostly unigrams. Whereas, non-credible reviews contain extreme opinions, less function words and more sophisticated content words – consisting of a lot of signature bigrams – possible to catch the readers' attention.

Behavioral Model: In general, the standalone *activity based model*, currently the industry standard, is observe to perform extremely well (refer to Table 5.2), given the annotations (in Yelp) are obtained from similar classification models. Interestingly, the combination of our proposed language and consistency features further improves the performance by around 5% classification accuracy. Additional meta-data like the user elite and check-in status is seen to improve the performance of activity based baselines – however, such features might typically not be available for "long tail" items or newcomers in the community. In such scenarios, the proposed model using limited information (*N-gram+Consistency+Activity*$^-$) performs better than the activity baselines using fine-grained information about items (like brand description) and user behavior. Incorporating additional user features (*Activity*$^+$) further boosts the performance of the framework.

Consistency Features: In order to explore the effectiveness of the consistency features, we perform ablation tests as shown in Table 5.2. The removal of the consistency model from the aggregated model significant degraded the accuracy performance by $3-4\%$ for the Yelp* dataset, with slight impact for the TripAdvisor dataset. The consistency model, although not as efficient in isolation, provides important value addition in conjunction with other features to improve the overall model performance.

In this chapter, we aim to provide *evidences* as to why a review text should be deemed non-credible based on extracted discrepancies. Table 5.4 shows a snapshot of the non-credible reviews obtained by the framework, with corresponding (in)consistency features for the Yelp. Further, to depict the robustness of the proposed framework across different domains, we re-train the model (as discussed previously) on a subset of Amazon review dataset containing around $149K$ reviews from nearly $117K$ users for $25K$ items across three item categories, namely Consumer Electronics, Software, and Sports. Due to the absence of ground-truth non-credible reviews, the model was used to highlight the inconsistencies found, thus identifying possibly deceptive reviews as shown in Table 5.4. Such reviews can then be considered as automated annotated data or for further refinement by human involvement. We observe that the ratings of deceptive reviews do not corroborate with the textual description, contain irrelevant reviews and rating of item, contradict majority users opinions, review sentiment-rating inconsistency, expressing extreme opinions without explanation, depicting temporal "burst" in ratings, etc. In principle, such evidences can be further investigated to detect other anomalous phenomena like group-spamming (one of the principal indicators is temporal burst), leading to an enhanced credibility detection framework.

Discussion: Hence, we observe that the proposed credibility detection framework is extremely efficient for accurate detection of *non-credible review text* by harnessing language, sentiment, user behavior, and consistency features for SVM based classification. We also depict that the framework is domain-independent and robust across various applications, outperforming other existing approaches.

5.5 Summary

This chapter presented a novel consistency model using limited information for detecting non-credible reviews. Our approach makes use of different *consistency features* from *language and behavioral models* derived from user text, sentiments, and ratings to not only attain high classification accuracy over state-of-the-art methods, but also provides possible *evidences* to explain the assessments – a novel feature absent in the present literature. We also showcase that the model is amenable to domain transfer and adaptation, overcoming the limitation of existing works using fine-grained information unavailable in certain scenarios. Experimental results on large real-life datasets demonstrate the robustness and performance gains for the proposed framework.

Table 5.4 – Snapshot of example non-credible reviews with inconsistencies. (The reviews from Yelp were also flagged as "not-recommended" by the Yelp Spam Filter.)

Consist. Features	**Yelp** Review & [Rating]	**Amazon** Review & [Rating]
user review – rating *(promotion/demotion)*:	never been inside James. never checked in. never visited bar. yet, one of my favorite hotels in Chicago. James has dog friendly area. my dog loves it there. [5★]	Excellant product-alarm zone, technical support is almost non-existent because of this i will look to another product. this is unacceptible. [4★]
user review – item description *(deviation from community)*:	internet is charged in a 300 dollar hotel! [3★]	The book Amazon offers is a joke! All it provides is the forward which is not written by Kalanithi. I don't have any sample of HIS writing to know if it appeals. [1★]
extreme user rating *(no explanation)*:	GREAT!!!i give 5 stars!!!Keep it up. [5★]	GREAT. This camera takes pictures. [1★]
temporal bursts *(between 3/14/2012 - 4/18/2012)*:	Dan's apartment was beautiful and a great downtown location... [5★] I highly recommend working with Dan and NSRA... [5★] Dan is super friendly, demonstrating that he was confident... [5★] my condo listing with no activity, Dan really stepped in... [5★]	

6 Efficient RDF Encoding of Knowledge Bases

Knowledge Bases (KBs), providing invaluable semantic resources comprising entities, attributes, and their inter-relationships. Such huge repositories are modeled using an underlying *labeled graphical structure* wherein the vertices represent entities and the relationships among the entities (vertices) are constructed as edges. The ensuing representation is then typically cast into triples using the traditional *RDF format*. However, efficient term encoding for improving RDF-style storage to enhance downstream storage, I/O, and query performance of the RDF engine has grossly been overlooked.

This chapter proposes the $KOGNAC$ algorithm for scalable and efficient dictionary encoding of huge KBs. The key aspects of our algorithm involve distinction of the frequent and infrequent terms in KBs, for enabling different encoding strategies based on data streaming approximate methods and clustering utilizing taxonomy or ontological schema. Experimental evaluations of our methodology (in combination with state-of-the-art RDF engines) on real-life large datasets exhibit significantly performance improvements on various parameters, such as query runtime, RAM usage, I/O, and cost of scans.

6.1 Introduction

Motivation. Knowledge bases have not only been built in academic projects like DBpedia [Bizer *et al.*, 2009] and YAGO [Suchanek *et al.*, 2007], but are also used by leading organizations like Google, Microsoft, etc., for modeling entity-entity relationships to support natural language based queries, user-centric Internet services, as well as mission-critical data analytics.

Modern knowledge bases can be visualized as entity-entity relationship graphs, with entities representing labeled nodes and the relations as labeled edges between them. These graphs are generally represented using the *Resource Description Framework* (RDF) data model [Klyne and Carroll, 2006], in which the KB corresponds to a finite set

of $\langle subject, predicate, object \rangle$ (SPO) triples, where the S and O terms are URIs, blank nodes, or literal values [Klyne and Carroll, 2006].

Efficient and scalable real-time querying on huge KBs with billions of RDF triples have necessitated intelligent KB representation catering to storage requirement, load time, disk access, query join and optimization, and varied other performance parameters. In concept, KBs can be stored and managed using a variety of platforms, like RDF engines [Neumann and Weikum, 2008; Yuan *et al.*, 2013; Gurajada *et al.*, 2014], relational column stores [Sidirourgos *et al.*, 2008], NoSQL key-value stores [Erling and Mikhailov, 2009; Oracle, 2011], or graph database systems [Robinson *et al.*, 2015].

In this context, the storage of RDF terms in their original format is both space and process inefficient since these are typically long strings. As such, all existing approaches *encode* the URIs and literals in a judicious manner, typically by mapping them to fixed-length integer IDs, with the original strings retrieved only during execution. These IDs are then further compressed by the database engine during initial load time (with techniques such as run-length encoding and Ziv-Lempel), reducing the storage footprint and allowing for $10\times$ or better compression of index lists, thus accelerating scans and other in-memory operations.

Objectives. Modern KBs are typically queried using the W3C SPARQL language [Harris *et al.*, 2013]. Currently, the impact that different ID mappings have on more advanced operations (like query joins, index compression, data locality, special operators, etc.) is less well studied, and their influence on bulk update performance has largely been disregarded.

Current RDF engines generally employ four types of encodings: (1) order-based, (2) hash-based, (3) syntactic, or (4) based on coordinates. Most RDF engines use *dictionary encoding* for assigning numeric IDs to terms based on their *appearance ordering*, thereby implicitly assuming similar data to be serialized closely. Term-hashing based approaches not only disregard any possible co-relation among the terms, but also do not consider term frequencies, possibly leading to sub-optimal encoding with frequent terms being assigned larger IDs. On the other hand, pre-sorting the SPO triples lexicographically to achieve better locality has been shown to perform well in most cases. Unfortunately, this heuristic breaks when the string similarities do not follow such ordering. Relational engines optionally using memory addresses of dedicated string storage data structures as numerical IDs, reflect the physical storage locality rather than the term semantic locality present in KB. Such methods are also prohibitively expensive [Boncz *et al.*, 2005; Shannon and Benninger, 2014; Harbi *et al.*, 2015] and thus do not support the real-time characteristics of most applications. Typical use of term grouping in such methods is driven by limited co-relational and syntactic criteria, failing to capture subtle relations within the cluster members.

Ideally, the encoding of URIs and literals of RDF triples into numerical IDs should satisfy the following desiderata:

– *Encoding and Scan:* The integer-valued IDs assigned should consider the *skew* in the term frequencies in the KG, and assign smaller IDs to frequent terms, in order to facilitate efficient down-stream compression (by the storage engine). This not only provides storage space reduction (usually in memory), but also speed-up for sequential data scans (for reduced data size and better L3 cache efficiency).
– *Joins:* For more advanced query access patterns, particularly for join operations, *data locality* should be preserved as much as possible by the encoding. That is, terms that are often accessed together should have close ID assignment in order to further reduce disk and index access [Pacher *et al.*, 2011; Harbi *et al.*, 2015].
– *Bulk Loading:* It is often crucial to quickly load billions of triples, for example, when a KB is required as background knowledge for new analytic applications, or for append-only bulk update operations. Thus, the encoding process should support parallelism as much as possible for better scale-up.

Problem Statement. Interestingly, none of the existing approaches for KB encoding performs well along all three dimensions of the above desiderata. Assigning consecutive or pseudo-random identifiers leads to good compression, but exhibits poor locality for joins and is sub-optimal in exploiting term skew. Syntactic clustering is a compromise along the three objectives, but yet is not robust enough to handle the cases where similarity cannot be extracted from the syntax. In this chapter, we aim to alleviate the above challenges by generating a *dictionary* for encoding RDF URIs and literals for large KBs such that all the desiderata, namely better index compression, load time, data locality, and query performance, are well satisfied.

6.1.1 Approach and Contributions

This chapter proposes the $KOGNAC$ algorithm *(KnOwledge Graph eNcoding And Compression)* for efficient KB encoding based on the distinction of frequent and infrequent terms and a judicious combination of statistical and semantic techniques to capture the semantic similarities between terms. Further, it has the advantage that it is independent from index compression techniques or actual RDF engine implementation details, since the output is an intelligent mapping from term strings to corresponding IDs, facilitating compression, data locality, and query operations.

To this end, the novel components in $KOGNAC$ are:

– *Skewness* in term frequency distribution is detected by use of parallel streaming approaches and subsequently, frequent terms are encoded differently to facilitate high downstream compression by the RDF engine.
– To improve *data locality* for join access patterns, $KOGNAC$ computes *semantic relatedness* between infrequent terms by hierarchically clustering them into ontological classes (using frequent association mining for sparse data), and mapping terms

in the same cluster to consecutive IDs.

We intelligently integrate the above techniques to propose a full-fledged efficient encoding framework, which can be generically applied to different RDF triple-store systems. In a nutshell, the contributions of this chapter are as follows:

– $KOGNAC$, a generic approach using a novel combinations of techniques for efficient term encoding for large KB storage in combination with different RDF engines (Section 6.3);
– novel combination of parallel data streaming algorithms for accurate detection of term skew, leading to smaller ID assignment to frequent terms enabling better index compression for the RDF store (Section 6.4);
– clustering approach using a mixture of ontological hierarchy and/or frequent class association mining to construct a class taxonomy for grouping infrequent terms and close ranged ID assignment to improve *data locality* (Section 6.5); and
– a comprehensive experimental evaluation on varied synthetic and real-life datasets and queries by integrated $KOGNAC$ with four RDF systems, to demonstrate significant performance gains, over state-of-the-art approaches, in compression, load and run times, query execution, and I/O access (Section 6.6).

6.2 Related Work

Background. The RDF data model represents the input data as a collection of triples of terms that can be either URIs, blank nodes, or literals [Klyne and Carroll, 2006]. Let U be the set of URIs, B a set of blank nodes, and L the set of literals, such that $U \cap B \cap L = \emptyset$. Each RDF triple then belongs to the set $\{(U \cup B) \times U \times (U \cup B \cup L)\}$, usually depicted as $\langle subject, predicate, object \rangle$ (SPO). A set of such RDF triples can then be represented by a directed, labeled multigraph $G = (V_G, E_G)$, where terms denote vertices V_G, and triples represent edges E_G.

Assume I to be an input set of RDF triples representing a knowledge base, with the set $D = U \cup B \cup L$ defining the *domain* of I. We introduce three functions $s, p, o = I \rightarrow D$ to retrieve the corresponding individual members from the triples. That is, function s returns the first term of the triple (i.e., the *subject*), p returns the *predicate*, while o returns the *object*. Further, we define the *vocabulary* set of I as $V = \{\forall d \in D, \exists t \in I \mid s(t) = d \vee p(t) = d \vee o(t) = d\}$. In RDF graphs, V is a set of long strings, which are inefficient to process in their raw form, and are therefore typically encoded using shorter numerical IDs.

Definition 1. *A dictionary encoding algorithm aims at encoding V by assigning numerical IDs (assume in $\mathbb{N}$) to each member of V using a bijection $\theta = V \rightarrow \mathbb{N}$ to return a dictionary table $T \subset \mathbb{N} \times V$.*

The goal of $KOGNAC$ is to calculate a "good" θ which leads to better performance.

Table 6.1 – Loading and dictionary encoding time in RDF engines for LUBM dataset.

System	Loading Time	% Dictionary Encoding Time
RDF-3X	42m42s	26%
TripleBit	28m43s	71%
Sesame	2h37m47s	21%

Once θ is calculated, it can be applied to I to load the encoded database in the RDF store. Note, the bijection condition does not imply a numerical ID to uniquely map to one term, but only that the mapping is unique within each field of the triples. Thus we can assign different numbers to the same term in case it appears in different fields of the triples, as adopted by *TripleBit* [Yuan *et al.*, 2013].

We map our three desiderata into two major problem domains:

(1) *Skew and Locality:* A judicious encoding scheme should not only provide smaller IDs to frequently occurring items (to assist index compression) but also assign closer IDs to semantically related terms, thereby enabling *data locality* in the stored indices. Improved encoding quality would thereby reduce I/O operations and memory consumption. Henceforth, we refer to this criteria as $encoded\ locality$.

(2) *Scalability and Runtime:* An efficient encoding scheme should also be fast enough to quickly encode large knowledge bases. This requirement is important because dictionary encoding takes a non-negligible part of the loading time for current RDF stores (RDF-3X [Neumann and Weikum, 2008], TripleBit [Yuan *et al.*, 2013], and Sesame [Broekstra *et al.*, 2002]). As an example, the encoding operation sometimes takes up to 71% of the total load-time, for the LUBM dataset [Guo *et al.*, 2005] with about 100 million RDF triples, in TripleBit [Yuan *et al.*, 2013] (as reported in Table 6.1). On larger inputs, this can easily translate to few hours of computation. Therefore, a faster encoding algorithm can have a visible effect on the total loading time of the system.

Addressing the above "twin challenges" precisely forms the goal of our proposed encoding algorithm, $KOGNAC$.

State-of-the-art approaches. RDF provides the current de-facto method for conceptual modeling of relations between resources, and is denoted as $\langle subject, predicate, object \rangle$ triples [Klyne and Carroll, 2006]. Thus a collection of RDF triples intrinsically represents a labeled, directed multi-graph [Hayes, 2004]. RDF Schema [Brickley and Guha, 2004] (RDFS), a standardized extension of RDF provides an additional vocabulary for the description of taxonomies and properties, and is widely used to list the type objects and the relationships connecting them.

For efficient management of RDF graphs, several RDF engines have been proposed, such as TripleBit [Yuan *et al.*, 2013], RDF-3X [Neumann and Weikum, 2008], TriAD [Gu-

rajada *et al.*, 2014], and Sesame [Broekstra *et al.*, 2002]. Further, column-store based approaches (such as MonetDB [Sidirourgos *et al.*, 2008] and Virtuoso [Erling and Mikhailov, 2009]) as well as extensions to traditional DBMS systems for RDF have been studied [Alexander and Ravada, 2005; Erling and Mikhailov, 2009; Bornea *et al.*, 2013]. A survey on RDF stores can be obtained in [Sakr and Al-Naymat, 2010].

RDF Encoding. For compute efficiency, RDF terms – typically strings – are encoding into numerical IDs during the pre-processing stages in RDF engines. Most SPARQL engines (like TripleBit [Yuan *et al.*, 2013], TriAD [Gurajada *et al.*, 2014], etc.,) use *dictionary encoding* and assign numeric IDs to terms based on *appearance ordering*, i.e., using consecutive or pseudo-random numbers for incoming triples; thereby implicitly assuming similar data to be serialized closely. On the other hand, term-hashing based IDs used in 4Store [Harris *et al.*, 2009] disregards any possible co-relation among terms. In fact, both approaches do not consider term frequencies leading to sub-optimal encoding with frequent terms possibly assigned larger IDs.

RDF-3X [Neumann and Weikum, 2008], one of the fastest single-machine RDF storage engines, pre-sorts the SPO triples lexicographically and then assigns consecutive integers, thus achieving better locality and encoding for its clustered indices. A similar parallelized approach is also followed by [Urbani *et al.*, 2013]. Unfortunately, this heuristic breaks when the string similarity does not follow lexicographic ordering for data fetched from Web crawls, extracted in parallel, or where it is particularly preprocessing (e.g., duplicate removal). Further, such dissimilarities occur frequently (e.g., via subdomain usage in URIs), or may even be imposed by political decisions (e.g., Wikidata [Vrandečić and Krötzsch, 2014] uses random URIs to avoid an English bias).

Relational engines (e.g., MonetDB [Sidirourgos *et al.*, 2008]) can optionally use dedicated data structures for the storage of strings (mainly using variants of Tries, e.g., [Leis *et al.*, 2013], or prefix-suppression tables [Carroll *et al.*, 2003]) and use coordinates (i.e., memory addresses) in such data structures as the numerical IDs. These IDs are typically long, and the induced locality reflects the physical storage of the strings rather than the semantics in the KB. These methods and other sophisticated partitioning methods as used in TriAD renders such encoding prohibitively compute expensive [Boncz *et al.*, 2005; Shannon and Benninger, 2014; Harbi *et al.*, 2015] and thus do not support the real-time characteristics of most applications.

Systems like 4Store [Harris *et al.*, 2009], Virtuoso [Erling and Mikhailov, 2009], Clustered TDB [Owens *et al.*, 2008], and SW-Store [Abadi *et al.*, 2009] employ clustering of triples across relations and/or ontological classes to improve data locality. However, term grouping via clustering or bi-simulation [Milo and Suciu, 1999] is typically driven by limited correlational and syntactic criteria only, like neighborhood in the entity-pair graph [Bishop *et al.*, 2011; Patchigolla, 2011], and cannot capture more subtle relations.

Unlike these works, our approach has the advantage that it improves the data locality by exploiting richer implicit and explicit semantic term relations that go beyond the actual strings. Furthermore, we propose and evaluate more robust techniques than sampling to detect the skew.

RDF Compression. The literature hosts a number of approaches for efficient compression techniques for RDF data. One of the most well-known most is HDT [Fernandez *et al.*, 2013], which consists of a sophisticated technique to efficiently compress RDF graphs into a binary data structure. Another approach proposes to enhance the ZLib technique for RDF storage [Fernandez *et al.*, 2014]. Strategies for compression of similar languages, for instance XML compression (e.g., XMill [Liefke and Suciu, 2000]) have also been studied. A systematic comparison of several data structures, such as tries [Leis *et al.*, 2013], for efficient storage of strings in databases and for encoding URI data is presented in [Mavlyutov *et al.*, 2015].

However, our purpose is different than these works, as their primary concern is to minimize space in order to optimize raw storage or transmission, while in our case the primary objective is to improve the database performance.

Semantic Relatedness. Several semantic similarity measures between items have been proposed based on lexicographic features [Zhang *et al.*, 2013], domain-dependent data like Wikipedia [Gabrilovich and Markovitch, 2007] and Wordnet [Budanitsky and Hirst, 2006], or path length between two nodes [Leal, 2013]. However, such functions suffer from high compute efficiency and data sparsity. The use of ontological taxonomies to speed up reasoning via intelligent ID encoding was described in [Curé *et al.*, 2015]. Traditional clustering methods such as k-means consider term similarities to minimize intra-cluster distance while maximizing inter-cluster distance. Clustering based on the underlying graph partitioning (with similarities as edge weights), performed using tools like METIS [Karypis and Kumar, 1998], have been employed by RDF engines like TriAD [Gurajada *et al.*, 2014]; but are unacceptably computationally expensive.

In contrast, $KOGNAC$ utilizes a much more light-weight clustering based on class hierarchies and/or frequent pattern mining for the sake of scalability.

Data Synopsis. Stream algorithms are typically concerned with the extraction of approximate data summaries, like distinct item count, frequent items, frequency moments, etc., from possibly infinite input streams [Charikar *et al.*, 2002; Cormode and Muthukrishnan, 2005; Beyer *et al.*, 2007]. These methods generally involve intelligent sampling-based approaches coupled with efficient hash-based structures for modeling the estimation bounds for the data synopsis obtained with a single-pass. An introduction to such techniques can be found in [Chakrabarti, 2014]. In this work, we evaluate some of the most predominant ones and propose a novel combination yielding the best results in our case.

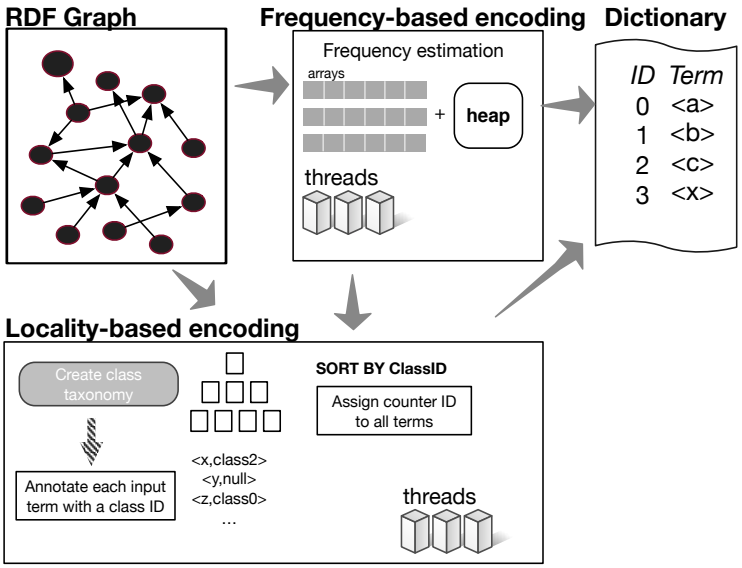

Figure 6.1 – High level overview of the $KOGNAC$ encoding algorithm.

6.3 The $KOGNAC$ Algorithm

The efficiency of the proposed $KOGNAC$ algorithm hinges on the following two important features of modern knowledge bases:

1. *Distribution Skew:* The distribution of the term frequencies in RDF graphs is (typically) highly skewed [Kotoulas *et al.*, 2010] and exhibits the *Pareto principle*. This renders frequency-sensitive encoding techniques potentially quite effective in significantly enhancing index compression for RDF engines.

2. *Data Enculturation:*[1] RDF terms are associated to each other via concepts represented by relations. This semantics can be used to improve the encoding efficiency by clustering related terms together, leading to improved data locality.

A key operation performed by $KOGNAC$ is to distinguish the encoding scheme for (few) highly *frequent* terms from the remaining (many) *infrequent ones*. Intuitively, these two sets of terms semantically fulfill different roles in the graph, and thus potentially should be encoded using different strategies. Frequent terms typically represent broad concepts that connect large parts of the graph (for instance, the *rdf:type* relation, or a hypothetical identifier associated to class *Person*). Hence, from a purely compression point of view, such terms should receive lower IDs (smaller bit length). On the contrary, infrequent terms identify smaller concepts that can potentially be clustered (for semantic relatedness) for more encoding locality efficiency.

1. *Enculturation* is defined as the process of inculcating the surrounding to acquire appropriate values [Grusec and Hastings, 2007].

By segregating the terms, different encoding techniques can be customized to exploit more efficiently the features that characterize them. For frequent terms, $KOGNAC$ implements a frequency-based encoding to maximize downstream index compression. For the infrequent ones, $KOGNAC$ builds an hierarchical clusters over terms (into classes) and accordingly encodes them to better leverage the semantics by assigning "closer" IDs to semantically "closer" terms in the graph, for preserving *data locality*.

Algorithmic Overview. Let V be the set of terms used in the input RDF graph G. $KOGNAC$ receives G and a threshold k (for identifying top-k frequent elements) as input, and returns a dictionary $T \subset V \times \mathbb{N}$ that maps each element in V to a unique ID. Figure 6.1 describes – at a high level – the functioning of $KOGNAC$, comprising sequential execution of two major operational components:

1. *Frequency based encoding* (FBE): It identifies the frequent terms and builds an efficient dictionary for encoding the frequent terms based on their *approximate frequency estimates* computed using parallelized streaming algorithms.
2. *Locality based encoding* (LBE): This provides encoding for the remaining infrequent terms by considering term similarities based on ontological distance and/or using association pattern mining.

The two sets of encodings (frequent and infrequent) form the output mapping, θ, represented as a *dictionary*. Optionally, the graph representation of the input knowledge base might be compressed by loading the mappings in an optimized hash map, the triples range-partitioned, and subsequently replacing each term with the corresponding ID in parallel. However, the RDF store might already have an optimized procedure for such encoding, and therefore only needs the term-ID mappings. We consider this setting in this chapter.

$KOGNAC$ addresses the "twin" challenges of encoding as follows:

1. *Encoded Locality:* Frequency aware encoding enables smaller ID assignment to frequent terms, thus enhancing storage and index compression by the RDF engine. Further, clustering of infrequent terms based on *enculturation* (i.e., semantically associated concepts, relations, etc.) makes $KOGNAC$ agnostic on factors like term appearance order and syntactic structure of URIs, constituting important limitations for existing encoding approaches. This leads to significant benefits on index data locality, positively impacting query response time.
2. *Scalability and Runtime:* Certain computation of $KOGNAC$ can be fully parallelized, allowing better exploitation of modern multi-core architectures. Furthermore, our procedures calculate *approximate* term frequencies and frequent patterns, rather than an exact computations – speeding up the encode/decoding procedures to improve loading runtimes.

We now provide a detailed working description of the primary components of $KOGNAC$.

6.4 Frequency Based Encoding

Frequency-based encoding (FBE) in $KOGNAC$ attempts to encode frequent terms by the classical frequency aware assignment (i.e., highly frequent terms get the smallest IDs) with the goal of facilitating index compression. This strategy entails an accurate identification of such terms (which induces skew [Kotoulas *et al.*, 2010; Urbani *et al.*, 2013]). Unfortunately, an exact computation of term frequencies and their ordering thereafter is a space as well as time-consuming operation, often unacceptable given the large size (order of billions of nodes) of modern knowledge bases. Traditionally, sampling techniques are employed to deal with such issues. However, sampling suffers from three major limitations:

– The samples may be too small and not proper representative of the underlying data distribution, leading to both high false positive and false negative;
– Processing of the sample might still be expensive, especially if a large sample subset is required for tolerable error rates;
– Sampling only provides a synopsis of the data in the sample and cannot determine the relative frequencies of the terms.

These crucial limitations motivated us to explore other approximating techniques; particularly, hash-based methods successfully deployed in other domains, for instance to answer iceberg queries [Fang *et al.*, 1999], or to identify distinct/frequent items in large data streams [Karp *et al.*, 2003; Beyer *et al.*, 2007] as discussed next.

6.4.1 Frequent Term Identification

We initially explore the applicability of state-of-the-art approximate frequent item and frequency estimation algorithms.

Count-Sketch: Count-Sketch [Charikar *et al.*, 2002] is a single pass approach for frequent items identification along with their frequencies in a data stream. It employs a hash-based counter in combination with a *sketch*, another hash function from items to {-1, +1}. The use of a heap structure of size k enables approximate top-k most frequent item extraction. The insertion of an element in the heap involves updating the corresponding hash counter of the element based on the hash family and the sketch. To reduce estimation error variance on expectation, multiple independent hash counters are maintained, and the median value is reported. The frequency estimate (of item i) is known to be bounded by $\hat{f}_i \leq f_i + \epsilon||f||_2$ with high probability, where $||f||_2$ denotes the second moment of item frequencies. However, the favorable theoretical bounds require a quadratic space in the error tolerability ϵ, and is also dependent on the input data distribution. Further, parallelizing the individual operations is non-intuitive and its effect on performance is unknown.

Count-Min: The *Count-Min* algorithm [Cormode and Muthukrishnan, 2005] works

with the same principle as Count-Sketch, and has the advantage that it requires linear space (in error tolerance) and is inherently parallelizable. It works by using $n > 1$ hash array counter tables with n different hash functions, with each observed item hashed to the corresponding hash table positions (via the hash family) and the corresponding counter incremented. The estimated frequency of an item is given by the $minimum$ of hash counter values to which the item maps. The elimination of the heap structure makes the procedure extremely fast. However, it has a worse approximation bound on the frequency estimates than Count-Sketch, with bound $\hat{f}_i \leq f_i + \epsilon||f||_1$ and $||f||_1 \geq ||f||_2$. Further, it requires two passes for counting the term frequencies and extracting the actual frequent terms.

Misra-Gries: The Misra-Gries algorithm [Misra and Gries, 1982] also provides a single-pass method to identify frequent items present in an input stream by maintaining a heap of size k for the currently observed frequent items with a counter for relative frequency estimation. It deterministically reports the items that are at least *k-frequent*, i.e., having a frequency $> \frac{||f||_1}{k}$, where $||f||_1$ denotes the total number of data elements. However, this guarantee no longer holds for items with lower frequencies, and the final relative frequencies heavily depends on the ordering of appearance.

To mitigate the above problems, we propose the *Count-Min+Misra-Gries* (CM+MG) approach, which adopts elements of Count-Min and Misra-Gries and combines them intelligently to obtain the frequent KB terms along with their approximate frequencies.

Count-Min + Misra-Gries Approach

The Count-Min approach provides a good estimate of term frequencies, but cannot identify the top-k elements within a single pass. On the contrary, Misra-Gries detects the top-k elements but does not accurately report the frequencies. The complementary disadvantages make for an ideal combination of the two: *Misra-Gries* for popular item identification and *Count-Min* for approximate frequency estimation. We thus propose a hybrid approach, *Count-Min+Misra-Gries (CM+MG)*, enabling best of both the worlds: precise detection of frequent items along with accurate frequency approximation encapsulated within a single fully parallelizable pass of the input KB.

A graphical overview of the CM+MG approach is depicted in Figure 6.2. The set of triples I (representing the knowledge base), a hash family H comprising n hash functions, m parallel threads, and a threshold k of popular elements are provided as input to the procedure. Given the available memory, m set to the number of physical cores, and the value k requested from the user, we create $m * n$ counter tables for parallel Count-Min and m Misra-Gries heaps, each of size k.

The input I is split into m subsets of equal size, and fed to the m threads, each of which independently calculates n hash codes for each term present in the triples using the

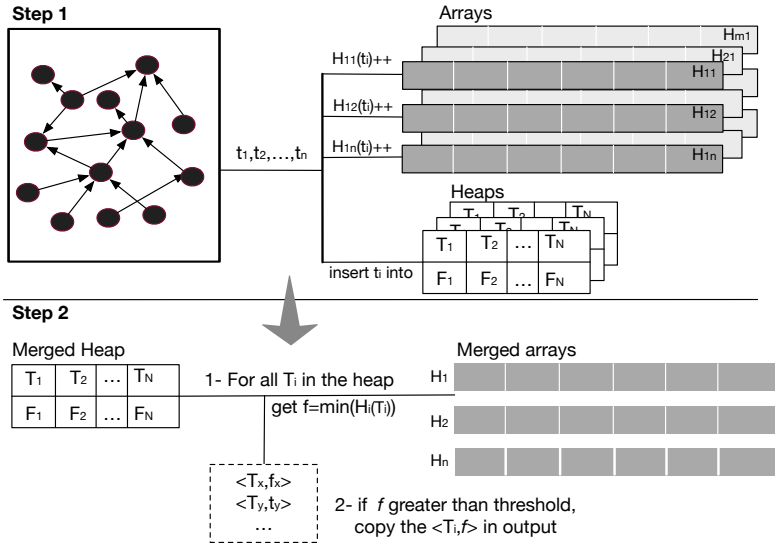

Figure 6.2 – Overview of the working of CM+MG in $KOGNAC$.

hash family H. The corresponding n indices in the counter tables are incremented by 1 and the terms (or hash-codes) are also inserted into the heap.

The m heaps are then merged in CM+MG by using the parallel heap merging technique proposed in [Cafaro and Tempesta, 2011] to obtain a single heap containing the top-k most frequent items in the input. The hash table counters are also merged together into n final tables. As a threshold value for the frequent terms, we select the top-k value in the first array. The algorithm now scans all elements in the merged heap, and instead of using the relative frequencies as estimate, CM+MG queries the count-min arrays using the terms, and uses the minimum of the returned values. If this value is greater than a computed threshold, the term is marked as frequent. Finally, the list of frequent terms along with their estimated frequencies is reported for encoding.

This combined approach addresses the individual limitations of Misra-Gries and Count-Min by providing a hybrid, single pass, fully parallelizable procedure returning the list of top-k frequent terms *and* their estimated frequencies.

6.4.2 Frequent Term Encoding

The estimated term frequencies calculated above by CM+MG are used to sort the identified terms in descending order. Then, an incremental counter value is assigned as the ID to each member of the sorted list. Algorithm 6.1 presents the parallelized pseudo-code for the FBE procedure in $KOGNAC$. In general, state-of-the-art RDF stores adopt a fixed-length bit encoding of the numeric IDs generated.

Assume $RT = \{t_1, t_2, \ldots, t_n\}$ to be n distinct RDF terms, with term t_i having a fre-

Algorithm 6.1: Parallelized Frequency-based Encoding procedure in $KOGNAC$

Require: Collection of RDF triples (I), hash family (H) of n hash functions, hash table size (TS), number of parallel threads (m), and frequency threshold k
Ensure: A partial dictionary mapping $Dict_f$ and ID counter $counter$

Initialize heap and hash tables for each thread as:
for $\forall i \in \{0 \dots n-1\}, \forall j \in \{0 \dots \text{TS}-1\}, l \in \{0 \dots m-1\}$ **do**
  $tmptab_i^l[j] \leftarrow 0$ and $heap^l \leftarrow 0$
end for
split I in $I_{i \in \{0 \dots m-1\}}$ partitions of equal size
for each thread $i \in \{0 \dots m-1\}$ **do**
  for $\forall triple \in I_i$ **do**
    for $\forall term \in triple$ **do**
      for $\forall j \in \{1..n\}$ **do**
        $idx \leftarrow |h_j(term)| \bmod TS$
        $tmptab_j^i[idx] \leftarrow tmptab_j^i[idx] + 1$
      end for
      add $term$ to $heap^i$ using Misra-Gries algorithm
    end for
  end for
end for
wait for all threads i
for $i \in \{0 \dots n-1\}$, $j \in \{0 \dots TS-1\}$ **do**
  $table_i[j] \leftarrow \sum_{x=0}^{nt-1} tmptab_i^x[j]$
end for
$heap \leftarrow$ merge all $heap^i$ for $i \in \{0 \dots m-1\}$
$FreqT \leftarrow \emptyset$
for $\forall term \in heap$ **do**
  $c \leftarrow min(\{table_i[|h_i(term)| \bmod TS] : i \in \{1..n\}\})$
  if $c \geq k$ **then**
    $FreqT \leftarrow FreqT \cup (term \rightarrow c)$
  end if
end for
sort $FreqT$ by the frequency c in descending order
$counter \leftarrow 0$
for $\forall\ (term \rightarrow c) \in FreqT$ **do**
  $block \leftarrow$ ***block-encoding*** value of $counter$
  $Dict_f \leftarrow Dict_f \cup (term, block)$
  $counter \leftarrow counter + 1$
end for
Output $Dict_f$ as the dictionary encoding for frequent terms and $counter$ value for LBE

quency of f_i (estimated) in the input KB. In the worst case, an assignment criterion independent from term frequencies (e.g., order-based) employs an assignment like fixed-length encoding, i.e., all terms will be assigned IDs of length $\lceil \log_2 n \rceil$ bits. In this case, the total space required to store an encoded KB would be,

$$S_{fix} = \sum_{i=1}^{n} f_i \lceil \log_2 n \rceil = F \lceil \log_2 n \rceil \qquad \text{[with } F = \sum_{i=1}^{n} f_i \text{]} \tag{6.1}$$

However, to better characterize storage space and index construction, $KOGNAC$ adopts the *block-based* bit encoding strategy. Specifically, our assignment criteria effectively partitions the k-frequent terms into blocks of different sizes based on the sorted order of the estimated item frequencies.

Thus, block $i \in \{1 \dots b\}$ then contains the top 2^i frequent elements which are not in any previous group, and the terms therein are encoded using i bits. That is, the $i = 1$ block contains the top 2^1 most frequent items encoded using 1 bit (0 and 1), the $i = 2$ block represents the next four (2^2) frequent items to be encoded using 2 bits, and so on. The sequence of bits obtained represents the numerical ID encoding for each term, thus assigning smaller length codes to frequent items. Hence, there exists $b = \lceil \log_2 n \rceil$ non-empty blocks for n distinct items in RT.

However, such ID assignment is not *prefix-free* (i.e., two items with different encoding sizes might share the same prefix), and would lead to ambiguity in the bit length to consider during the decode procedure when items are read from the index for extracting the original RDF term strings. Hence, $\lceil \log_2 b \rceil$ extra bits are prepended to the assigned term ID for identifying the block to which the term belongs and subsequently the number of bits encoding the term. Hence, the total space required for encoding an item in block i is given by $(i + \lceil \log_2 b \rceil)$ bits.

Assuming, f_j^i to denote the frequency of the j^{th} item in block i, the total encoding space required for encoding a KB in this scenario is,

$$S_{kog} = \sum_{i=1}^{b} \sum_{j=1}^{2^i} (i + \lceil \log_2 b \rceil) f_j^i = \lceil \log_2 b \rceil \sum_{i=1}^{b} \sum_{j=1}^{2^i} f_j^i + \sum_{i=1}^{b} i \sum_{j=1}^{2^i} f_j^i \tag{6.2}$$

Since modern KBs have a skewed term distribution [Kotoulas *et al.*, 2010], we assume the item frequencies to be drawn from a Zipfian distribution [2] with parameter $s \geq 2$, such that the frequency of the k^{th} frequent term, $f_k \approx \frac{F}{s^k}$. Substituting this frequency distribution in Equation (6.2), we obtain,

$$S_{kog} = F\lceil \log_2 b \rceil + \sum_{i=1}^{b} i \sum_{j=1}^{2^i} \frac{F}{s^{\sum_{k=1}^{i-1} 2^k + j}} = F\lceil \log_2 b \rceil + \sum_{i=1}^{b} \frac{iF}{s^{\sum_{k=1}^{i-1} 2^k}} \sum_{j=1}^{2^i} \frac{1}{s^j}$$

By algebraic manipulations, for large values of i we have,

$$\begin{aligned} S_{kog} &= F\lceil \log_2 b \rceil + \sum_{i=1}^{b} \frac{iF}{s^{2^i - 2}} \cdot \frac{1}{s-1} \approx F\lceil \log_2 b \rceil + F \sum_{i=1}^{b} \frac{i}{s^{2^i}} \\ &\approx F\left(\lceil \log_2 b \rceil + \frac{1}{s^2} \right) = F\left(\lceil \log_2 \log_2 n \rceil + \frac{1}{s^2} \right) \end{aligned} \tag{6.3}$$

2. This distribution is used for heavy-tailed characteristics observed in natural language sources for KB construction.

Comparing Equation 6.3 with Equation 6.1, we observe that $KOGNAC$ achieves nearly an exponential theoretical decrease (i.e., $log\ log\ n$ vs. $log\ n$) in the total encoding space required to store the KB in contrast to existing fixed-length order based encoding strategies. The mappings of terms to IDs ("encoding") obtained are then used for constructing the *dictionary* for the frequent KB terms.

The ID assignment of $KOGNAC$ bears striking similarity to the optimal *prefix free* Huffman encoding [Huffman, 1952] based on the *Shannon's entropy* criteria. It constructs a *Huffman tree* in bottom-up fashion, by combining two items with the lowest frequencies at each iteration, ensuring that frequent items are combined during later iterations, placing them closer to the root. The tree is then traversed from the root and each edge is labeled as $0/1$, and the encoding assigned to an item is obtained as the sequence of edge labels on the path from root to the leaf containing the item. Interestingly, we observed that in a skewed frequency distribution setting our frequency-based encoding was faster in terms of encode/decode time and was able to attain a close approximation of the optimal encoding provided by Huffman.

6.5 Locality Based Encoding

$KOGNAC$ adopts a different strategy to encode infrequent items in the long tail of the distribution. The reason as to why frequency-based encoding no longer pays off, is:

- Each term appears only a few times and hence the effect of larger sub-optimal ID assignment is stymied; and
- The increased number of bits used for encodings provides an exponential number of disposable IDs, e.g., after the most frequent 2^{24} IDs, the following $2^{32} - 2^{24}$ terms will all require the same amount of bits.

Instead, where different ID assignment can play a role, is in determining the *locality* of relevant associated data for SPARQL query time and the cost of index access. Consider an example SPARQL query, which asks for the list of all students in a particular university. In this case, if the students have IDs that are distant from each other (i.e., distributed over the index), the engine must scan a large part of the index for retrieval.

Lexicographic ordering, adopted by some approaches [Neumann and Weikum, 2008; Urbani *et al.*, 2013], provides a better locality-aware encoding strategy (than appearance order) since it no longer depends on the term ordering, and exploits the fact that connected links tend to occur within the same domain and therefore similar terms exhibit similar URIs [Boldi and Vigna, 2004]. Unfortunately, this approach fails when item similarity does not follow lexicographic ordering. For instance, consider the URIs sites.google.com/site/XYZ-johndoe/ and people.john-doe.uni/ABC/ to refer to two students enrolled in the same university. In this case, the two students are semantically related, even though they are lexicographically distant. If the join succeeds using only sub-portions of the index, the algorithm can save significant computation by ignoring

large chunks of the index.

$KOGNAC$ addresses this problem by seeking an ID assignment criterion to ensure that semantically relevant terms remain close to each other also in the encoded space, so as to increase data locality and reduce index page traversal.

6.5.1 Term Similarity for Data Locality

Observe, that it is not possible to encode the terms so that they are always next to each other, as we can make one assignment only while SPARQL queries could request joins on any subset of the relations. Still, we can leverage a heuristics that is surprisingly effective: *SPARQL joins tend to materialize between semantically related terms.*

$KOGNAC$ addresses this problem by exploiting the fact that RDF data is semantically rich, in the sense that every RDF term carries some meaning that goes beyond its syntactic representation, and that some relations (like *type*) have a clear semantic interpretation which can be used for clustering together similar terms. Although, there exists other similarity measures based on semantic relatedness, typical join queries tend to span terms within closely related ontological hierarchy and relations.

The general idea of locality-based encoding (LBE) in $KOGNAC$ is to partition the terms using the typing relation of terms (i.e., *a type B*) and assign consequent IDs to members of the same class. Similar terms are clustered exploiting various ontological relations between their classes and frequent co-occurrences between their instances. Notice that the typing relation can be either explicitly stated or implicitly inferred by schema information such as the domain/range of properties (e.g., all subjects of *followsCourse* are of type *Student*).

The LBE approach draws on the following benefits:
– the type isA is commonly used in knowledge graphs to denote semantic relations (i.e., instanceOf) which are not domain-dependent;
– new type relations can be inferred from other ontological information when they are not explicitly mentioned (e.g., definitions of the domain/range of properties); and
– classes can be organized in a taxonomy according to the $rdfs : subClassOf$ RDFS relation. This allows efficient distinction of different degrees of similarity between instances of subclasses from instances of siblings of parents (e.g., *Students* should be closer to *Professors* than to *Robots* as the first two are subclasses of *Persons*).

The locality-based encoding framework is composed of $four$ major operations, executed in a sequence, as:

1. *Collecting class information:* Explicit type information and dependencies from the RDF graph pertaining to the constructing of class taxonomy are collected;
2. *Taxonomy construction:* A class taxonomy using the collected class information is

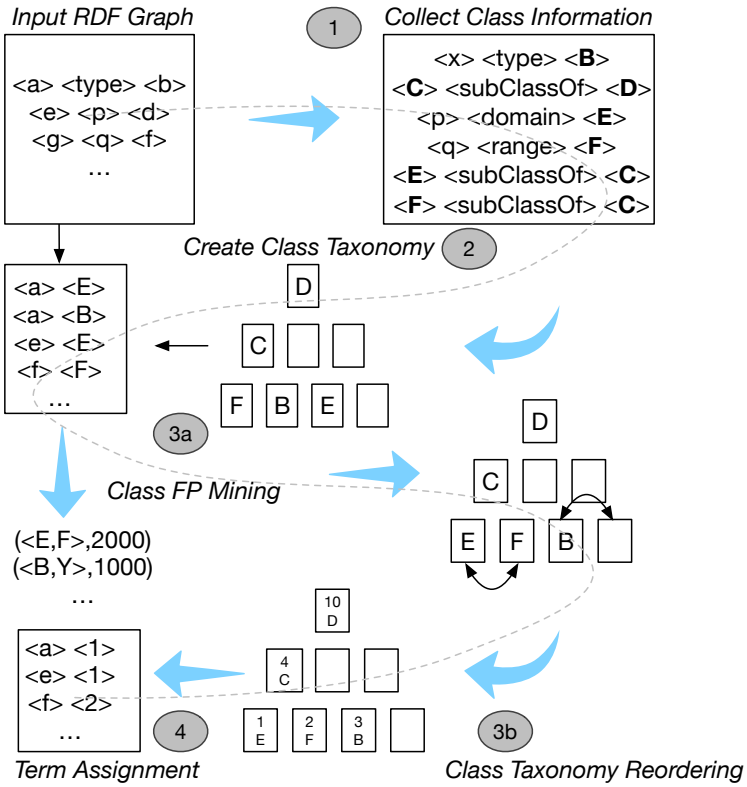

Figure 6.3 – Working of the Locality Based Encoding in $KOGNAC$.

constructed, and the nodes in the graph are annotated with the class they belong;

3. *Class FP mining and taxonomy re-ordering:* In case of sparse taxonomy information, frequent class patterns might be used to re-arrange the nodes in the taxonomy to assign class IDs; and
4. *ID assignment:* The terms are annotated with classes IDs, and then lexicographically sorted within the classes for ID assignment.

Figure 6.3 depicts a graphical representation of the overview of computations performed in these phases. We next describe each of the modules in more detail.

Collecting Information about classes:

$KOGNAC$ initially extracts from the input graph all class names and specific subsets of triples which provide information for constructing the taxonomy of classes. Specifically, the information collected are:
– All objects of *rdf:type* triples that represent the item classes;
– Triples with the *rdfs:subClassOf* or *isA* predicate, since they define a sub-class relation (e.g., ⟨*Student rdfs:subClassOf Person*⟩);

– Triples with the *rdfs:domain* predicate defining the domain of a certain property. These triples help in assessing the typing of subjects for these relations; and
– All triples with the *rdfs:range* predicate, as they define the co-domain of a certain property. These triples help in assessing the typing of objects of these relations.

Taxonomy construction:

The extracted triples with *rdfs:subClassOf* predicate are used to construct a directed weighted graph G_s, where the subjects and objects of triples constitute the vertices, and each triple represents an edge from the object to the subject. Edges from each vertex to the standard class *rdfs:Class* (super-class of all classes [Brickley and Guha, 2004]) are added to ensure that there are no disconnected components in the underlying graph.

Unfortunately, the obtained graph might contain loops and/or multiple paths connecting two classes. Such possible loops are eliminated by finding the *arborescence*[3] of G_s rooted at *rdfs:Class* and having the largest number of edges. For this task, we use the well-known Tarjan's algorithm [Tarjan, 1977], wherein the resulting tree represents the taxonomy of all classes extracted from the dataset.

After the taxonomy is built, the graph is re-scanned, and term-class pairs are generated where each term is associated with the class that they are instance of (if any). For example, for a triple $\langle s, p, o \rangle$ with class C as the domain of s, the pair $\langle s, C \rangle$ is constructed. Notice that a term can be emitted multiple times, and can be associated to multiple classes. If a term cannot be associated with a class, a dummy identifier is used. These pairs can then used in the next stage to extract frequent class co-occurrences.

Class FP mining and Taxonomy reordering:

The taxonomy construction described above depends heavily on ontological information available in the knowledge graph. However, if the input does not contain any of such information (e.g., no *rdfs:subClassOf* triples), then a rich taxonomy cannot be built. In order to improve the taxonomy quality in this scenario, $KOGNAC$ mines possible frequent class co-occurrence associations from the $\langle term, Class \rangle$ pairs emitted before. To this end, first all the obtained pairs are grouped by $term$, and then the list of classes in the same group are extracted. We construct an FP-Tree [Han *et al.*, 2004], an efficient technique for frequent pattern mining, to obtain the popular combination of classes with their relative supports (with default minimum support of 1000 elements).

The original taxonomy class tree is then re-visited and the groups of classes that share the same parent are re-ordered by selecting the one that belongs to the frequent patterns with the highest support, and is thus placed as the first child of the parent. The other classes present in the same group belonging to the same pattern are then

3. The arborescence is a directed graph containing only one path from a designated root vertex to any other vertex in the graph.

placed after the current child. The algorithm continues rearranging all the other classes by looking at the remaining most frequent patterns mined. The built taxonomy tree is used by $KOGNAC$ to encode the long tail infrequent KB terms. In this way, $KOGNAC$ is able to robustly cluster term classes together based on semantic similarity even in the absence of rich ontological information.

6.5.2 Infrequent Term Encoding

After the taxonomy tree is rearranged, $KOGNAC$ assigns unique incremental class IDs to each subclass. The computed taxonomy class tree is assumed to be undirected and sequential IDs are assigned starting from the root and visiting the tree in a post-order manner. The post-order traversal guarantees that a parent node always has an ID that is greater than its children, and smaller than its following siblings.

After each class is associated to a unique class ID, the terms are initially annotated by the IDs of the respective class (using the $\langle term, Class \rangle$ pairs) of which it is an instance of (i.e., from explicit *isA* or implicit domain/range relation). If a term is associated to multiple classes, the class having the smallest class ID is selected (postorder ensures it is a leaf or sibling class). On the other hand, if a term cannot be associated to any class (e.g., a literal), then a dummy class ID is provided.

The terms are then sorted based on their class IDs, and by lexicographic order for terms with the same class IDs. This strategy not only ensures that semantically related terms (belonging to the same class) appear next to each other, but also gains from lexical ordering based on domain similarity. Finally, the incremental term ID counter from the frequency-based encoding stage (Section 6.4) is used to assign IDs to the terms in the order defined by the above sorted list. The union of these term-ID mappings and those previously generated in the FBE phase constitutes the final dictionary output of $KOGNAC$.

Observe, that a large part of the computation of LBE can be inherently parallelized using the parallel FP-growth algorithm [Li *et al.*, 2008] and the task of assigning terms to the smallest class IDs using the read-only taxonomy structure does not require thread synchronization. Only the final ID assignment step is performed sequentially due to the usage of a single incremental counter.

Semantic Distance: Interestingly, the LBE procedure in $KOGNAC$ provides a simple yet fast semantic distance measure between terms in an RDF graph based on their assigned IDs. Assuming x and y to be two terms that are instances of classes in the taxonomy T, a *semantic distance*, $d(x, y)$, between them can then be defined as $d(x, y) = |\theta(x) - \theta(y)|$, where $\theta(i)$ represents the encoding ID assigned to term i.

It is easy to see that this semantic distance measure based on the ID assignment re-

Table 6.2 – Characteristics of uncompressed serialized input datasets for $KOGNAC$.

Dataset	Size (GB)	# Triples	#Terms
LUBM1K	24	133,614,189	32,905,349
LUBM8K	186	1,068,394,982	263,133,316
LDBC	29	168,290,489	177,310,916
DBPedia	228	1,022,545,404	232,902,180

spects the *non-negative*, *symmetric*, *reflexive*, *strictness*, and *triangle inequality* metric properties. However, it reflects a semantic dissimilarity only if the terms are instances of different classes. Otherwise, it returns a less meaningful "lexicographic" distance.

6.6 Experimental Evaluation

This section presents the detailed evaluation of the $KOGNAC$ encoding on several state-of-the-art RDF stores using various real-world and synthetic benchmark datasets.

Implementation & Setup. $KOGNAC$, implemented in C++, takes an RDF data graph (in N3/NT format) as input and returns an (integer) encoding for each vocabulary term, which can thus be used to encode the input graph. To evaluate the effectiveness of our encoding strategy, we adapted several state-of-the-art disk-based centralized SPARQL engines exhibiting varied characteristics and encoding strategies as:

1. native/centralized RDF-3X [Neumann and Weikum, 2008] and TripleBit [Yuan *et al.*, 2013] using syntactic (lexicographic) and order-based encoding respectively;
2. native/distributed in-memory TriAD [Gurajada *et al.*, 2014] employing appearance-based encoding; and
3. RDBMS/centralized column-store MonetDB [Sidirourgos *et al.*, 2008] with memory-based encoding.

The RDF systems were modified to leverage customized dictionaries, and for fair analysis, a single-node setup (in addition to master) for TriAD was considered. Two different types of machines were used to perform the experiments to depict the performance of $KOGNAC$ across varied system architectures. *Machine A*, a dual 8-core 2.4 GHz Intel Haswell CPU, 64 GB memory, with an internal storage of two disks of 4 TB in RAID-0; and *Machine B* with 16 quad-core Intel Xeon CPU of 2.4GHz with 48GB of RAM were used in the empirical setup.

Datasets. The following datasets, both real-world and synthetic, were used to benchmark the performance of the competing approaches in our experiments. Table 6.2 tabulates the characteristics of the datasets.

– *LUBM* [Guo *et al.*, 2005]: a popular benchmark used in the evaluation of almost all major RDF engines. It creates artificial data about universities and proposes a set

of benchmark queries. UBA 1.7 [4] data generator was used to create two datasets of varying scale factors – LUBM1K (scale factor:1000 and 133 million triples) and LUBM8K (scale factor:8000 and 1 billion triples);
– *LDBC* [Angles *et al.*, 2014]: a recent consortium of academic and industrial RDF experts with the goal of providing a reference benchmark for advanced SPARQL 1.1 workloads, in the same spirit of TPC. We created a dataset with around 168 million triples according to the benchmark specifications; and
– *DBPedia* [Bizer *et al.*, 2009]: one of the largest real-world RDF dataset representing a knowledge base containing more than 1 billion triples.

Queries. We considered several benchmark and realistic queries for evaluating the impact of $KOGNAC$ encoding in several RDF stores. The original LUBM benchmark contains 14 example queries, out of which we chose five queries that were selected as representatives in [Yuan *et al.*, 2013] (while the others required reasoning to report results and were thus dropped). For DBPedia, a slight variation of the five example queries reported in the project's main webpage were used.

Unlike the other fixed benchmark queries, LDBC benchmark queries are generated from handcrafted template queries, and are significantly harder as they offer various degrees of complexity which can help in thorough evaluation of RDF stores. For example, a template query involves entity-population SPARQL query with four joins, four OPTIONAL join clauses and one FILTER clause with a parameter for substitution, using the newly defined SPARQL 1.1 operators, which are unsupported by any of the considered systems for evaluation. Thus, the missing operators were implemented in RDF-3X, and correspondingly 7 out of the 12 template queries of the basic benchmark were used to generate the respective queries.

Figure 6.4 reports the LUBM and DBPedia queries in compressed form, while the LDBC queries are freely available at www.ldbcouncil.org (abbreviations like query Q3 refer to the third query of the official benchmark). The class association mining step for the datasets were found to be redundant due to the presence of rich class information for terms, and hence was ignored in the remainder of our experimental setup. The working code for $KOGNAC$ is also available at github.com/jrbn/kognac.

We next discuss the detailed performance effectiveness of the $KOGNAC$ encoding.

6.6.1 Count-Min + Misra-Gries Evaluation

One of the key ingredients of the working of $KOGNAC$ is the RDF term-skew aware FBE procedure to distinguish the frequent items along with their approximate frequencies and their ensuing block-based encoding for improving index and storage

4. obtained from swat.cse.lehigh.edu/projects/lubm/

LUBM Queries.
@prefix r: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix u: <http://www.lehigh.edu/∼zhp2/2004/0401/univ-bench.owl#>
L1. { ?x u:subOrganizationOf < http://www.Department0.University0.edu> . ?x r:type u:ResearchGroup .}
L2. { ?x u:worksFor <http://www.Department0.University0.edu> . ?x r:type u:FullProfessor . ?x u:name ?y1 . ?x u:emailAddress ?y2 . ?x u:telephone ?y3 . }
L3. { ?y r:type ub:University . ?x u:memberOf ?z . ?z u:subOrgOf ?y . ?z r:type u:Department . ?x u:undergradDegreeFrom ?y . ?x r:type u:UndergradStudent.}
L4. { ?y r:type u:University . ?z u:subOrgOf ?y . ?z r:type u:Department . ?x u:memberOf ?z . ?x r:type u:GraduateStudent . ?x u:undergradDegreeFrom ?y .}
L5. { ?y r:type u:FullProfessor . ?y u:teacherOf ?z . ?z r:type u:Course . ?x u:advisor ?y . ?x u:takesCourse ?z . }

DBPedia Queries.
@prefix foaf: <http://xmlns.com/foaf/0.1/>, purl: <http://purl.org/dc/terms/>, db: <http://dbpedia.org/resource/>, dbo: <http://dbpedia.org/ontology/>, rs: <http://www.w3.org/2000/01/rdf-schema#>
D1. { ?car purl:subject db:Category:Luxury_vehicles . ?car foaf:name ?name . ?car dbo:manufacturer ?man . ?man foaf:name ?manufacturer }
D2. { ?film purl:subject db:Category:French_films }
D3. { ?g purl:subject db:Category:First-person_shooters . ?g foaf:name ?t }
D4. { ?p dbo:birthPlace db:Berlin . ?p dbo:birthDate ?b . ?p purl:subject db:Category:German_musicians . ?p foaf:name ?n . ?p rs:comment ?d}
D5. { ?per dbo:birthPlace db:Berlin . ?per dbo:birthDate ?birth . ?per foaf:name ?name . ?per dbo:deathDate ?death .}

Figure 6.4 – Snapshot of LUBM and DBPedia queries for $KOGNAC$.

Table 6.3 – Performance of CM+MG in $KOGNAC$ for frequent item identification.

Dataset	Top 50						Top 500					
	Run-time (sec)			Accuracy			Run-time (sec)			Accuracy		
	S	*CM*	*CM+MG*	*S*	*CM*	*CM+MG*	*S*	*CM*	*CM+MG*	*S*	*CM*	*CM+MG*
LUBM1K	196	109	121	1	1	1	192	104	252	1	0.98	0.96
LUBM8K	3,470	3,585	3,290	1	1	1	3,309	3,509	2,389	0.87	0.99	1
LDBC	373	482	209	1	1	1	270	247	381	0.99	1	1
DBPedia	4,300	4,917	4,227	1	1	1	4,129	4,600	3,043	0.99	1	1

efficiency. To this end, we evaluate the proposed CM+MG procedure and the corresponding setting of the frequency threshold parameter k.

Table 6.3 compares the run-time and accuracy between the naïve sampling method S (with 5% sample rate), Count-Min approach, and Count-Min + Misra-Gries procedure on the benchmark datasets for identifying the top-50 and top-500 frequent terms of the input knowledge base (represented by RDF triples). To this end, the exact frequencies of the elements in the datasets were computed to obtain the "gold standard" top-k elements for comparing with the results retrieved by the algorithms. Note that, the exact sorted ordering of the frequent items on exact frequencies were not considered for accuracy measure, as the order is not relevant and we look at top-k item identification alone, since byte-level compression does not distinguish between elements requiring an equal number of bytes to be stored.

For smaller values of k (set at 50), we observe CM+MG to be the fastest approach, sometimes twice as fast as stand-alone Count-Min (due to its requirement of two passes). In terms of accuracy, all the approaches were able to efficiently extract the top

Table 6.4 – Comparison of $KOGNAC$ with RDF-3X on (a) dictionary encoding time, (b) bulk load time, and (c) compressed disk size.

Dataset	Encoding Time (sec)				Load Time (sec)		Database Size (GB)	
	$KOGNAC$			RDF-3X	$KOGNAC$ RDF-3X		$KOGNAC$ RDF-3X	
	FBE	*LBE*	*Total*					
LUBM1K	429	752	1181	**1136**	**1695**	2681	**7.1**	8.3
LUBM8K	3863	867	12538	**11637**	**17563**	27199	**60**	77
LDBC	612	1127	1739	**1696**	**2923**	3896	14	**12**
DBPedia	5071	10504	15575	**12805**	**21921**	29531	76	76

50 most frequent items. However, with increase in the k to 500, the runtime of CM+MG was seen to degrade slightly (in some cases) due to the expensive heap operations. However, the accuracy of the sampling and Count-Min methods in identifying frequent items in large datasets was seen to suffer, with increase in the value of k, due to overestimation errors in the hash tables. Hence, CM+MG is seen to perform better (on average) than the other approaches in terms of runtime and accuracy.

Since, large knowledge bases typically contain only a few very frequent term, and considering the trade-off between the size compression obtained from FBE and data locality from LBE on the overall performance of RDF engines, the value of the frequency threshold, k in the CM+MG procedure was set to 50 for our experimental setup.

6.6.2 Effectiveness of $KOGNAC$ Encoding

To measure the effectiveness of $KOGNAC$ encoding on RDF systems, its impact was measured on the following two attributes:
(1) Speed-up and space requirement in bulk loading, and
(2) Query processing performance (in terms of runtime, I/O costs, maximum RAM usage, and scan costs).
We consider the aforementioned RDF datasets (LUBM1K, LUBM8K, LDBC, and DBPedia) and the corresponding benchmark queries. As baseline, we initially compare the performance of $KOGNAC$ (with threshold k set to 50) to that of RDF-3X, the best state-of-the-art RDF engine using partial lexicographic ordering for term encoding.

A. Encoding and Bulk loading of RDF data. We first compare the (sequential) RDF term encoding time required to generate the dictionary between the native RDF-3X engine and $KOGNAC$. From Table 6.4, it can be observed that the $KOGNAC$ encoding is slightly slower than the native RDF-3X encoding approach, as it performs a much more complex operation than the simple lexicographic encoding in RDF-3X. In the worst case, $KOGNAC$ is seen to be about 20% slower. Considering that encoding is a one-time operation whose cost gets amortized over time, such performance is generally acceptable, especially in view of the benefits obtained at bulk load, space, and query performances. However, with increase in the number of threads to 8 (in the en-

Table 6.5 – Impact of $KOGNAC$ and RDF-3X encodings on Query runtime, Max. RAM usage, and disk I/O access.

D.	Q.	# Results	Runtime (sec)		Max. RAM (MB)		Disk I/O (MB)	
			KOGNAC	*RDF-3X*	*KOGNAC*	*RDF-3X*	*KOGNAC*	*RDF-3X*
LUBM1K	L1	10	**0.22**	0.31	4	4	**14**	19
	L2	10	**0.04**	0.17	**5**	6	**16**	27
	L3	1	**88.53**	90.83	**708**	854	**53**	69
	L4	2528	**92.21**	98.41	**724**	883	548	**367**
	L5	44190	**7.45**	15.79	**1,261**	1,588	**1,102**	2,132
LUBM8K	L1	10	**0.09**	0.27	4	4	**15**	21
	L2	10	**0.02**	0.64	**5**	7	**18**	32
	L3	1	**700.58**	716.55	**5,335**	6,537	**309**	486
	L4	2528	**717.65**	744.10	**5,320**	6,536	**321**	811
	L5	351919	**75.90**	174.16	**9,739**	12,400	**8,832**	16,928
LDBC	Q2	36	**44.59**	45.58	**1,320**	2,053	**241**	279
	Q3	178	**126.48**	132.55	**524**	574	**588**	624
	Q6	3819127	**60.17**	71.32	**2,157**	5,198	**1,268**	3,953
	Q7	98	**5.85**	6.76	**549**	3,663	**625**	3,675
	Q8	1018	**1,847**	4,915	**2,867**	5,934	**1,804**	3,949
	Q10	14	**420.88**	4,577	**714**	3,662	**729**	3,676
	Q11	114	**24.51**	89.21	**170**	204	**201**	237
DBPedia	D1	449	**0.99**	3.32	**8**	15	55	**52**
	D2	600	**0.23**	3.17	**6**	8	**29**	61
	D3	270	**1.60**	2.96	**6**	11	59	**49**
	D4	68	**0.72**	1.34	6	6	**45**	49
	D5	1643	**5.05**	26.79	**29**	60	330	**263**

coding procedure), the encode time of $KOGNAC$ was seen to decrease exponentially; thereby depicting a highly parallelizable nature.

Interestingly, even with a slower encoding procedure $KOGNAC$ was seen to achieve faster total bulk loading time (i.e., time taken by the RDF engine to read the data and construct the indices and other statistics) compared to the original RDF-3X implementation on all the considered benchmark RDF datasets (Table 6.4). Hence, we observe $KOGNAC$ to be highly scalable for catering to real-time application needs.

Further, the term encodings obtained from $KOGNAC$ generally produced a more compressed database (than RDF-3X) due to the term-skew-aware encoding scheme for assigning smaller IDs to frequent RDF terms, leading to smaller disk storage space requirements for the RDF data as shown in Table 6.4.

B. Query Performance. We next measure the impact of $KOGNAC$ on the overall SPARQL query runtimes for RDF engines. Table 6.5 shows the query performance of RDF-3X system with and without $KOGNAC$ encoding on the various datasets. We report the *cold* query runtimes, maximum RAM usage, and disk I/O access averaged over five executions on both machines, with the OS cache flushed before each execution.

Table 6.6 – Impact of $KOGNAC$ on other RDF stores for different SPARQL queries.

Q.	TriAD		TripleBit		MonetDB	
	$KOGNAC$	*Standard*	$KOGNAC$	*Standard*	$KOGNAC$	*Syntactic*
L1	0.001	0.001	**0.056**	0.149	**0.820**	2.4
L2	0.002	0.002	0.094	n/a	**0.943**	1.2
L3	**0.106**	0.631	1.672	**1.567**	**11.1**	15.2
L4	**2.684**	3.090	**5.626**	6.549	**9.5**	21.1
L5	**2.558**	3.067	**5.082**	6.438	**4.1**	8.2

Simply changing the encoding from a partial lexicographic ordering (that in native RDF-3X) to a combination of frequency/locality-based (that by $KOGNAC$) is observed to have a significant improvement on the runtimes and the other performance metrics. In some cases, the runtime improvements are seen to be around *ten times* (e.g., queries L2 on LUBM8K, Q10, and U4). Specifically, we observed that for queries requiring join on portions of the index, $KOGNAC$ achieves significant improvements by encoding similar terms by assigning consecutive IDs, thereby facilitating join operations.

In order to assess the impact of $KOGNAC$ on *data locality* the maximum resident main memory (peak RAM) and I/O read from disk were compared. From Table 6.5, it can be observed that encoding the input knowledge base graph with $KOGNAC$ leads to less main memory usage, and significant reduction of disk reads. This gives a clear indication that our proposed encoding provides a better *semantically related aware* term encoding capturing possible data locality for enhancing query evaluation performance with smaller I/O scan costs.

KOGNAC encoding in other RDF systems. To explore the performance of $KOGNAC$ on other RDF stores, we conduct similar evaluations with other benchmark systems, namely in-memory distributed TriAD RDF engine (single-node setup with master), centralized TripleBit system, and relational column-store based MonetDB, using the LUBM queries. The obtained runtime results are reported in Table 6.6. We observe similar performance behavior as that previously obtained in RDF-3X, with $KOGNAC$ attaining better query runtime than the competing approaches with memory-based bit-level or partition-based term encoding.

Discussion: Hence, from the experimental setup we observe that the term encoding produced by $KOGNAC$ taking into account term frequencies and semantic similarities enables assignment of smaller IDs for frequently occurring terms (to reduce disk and index sizes) and closer IDs for related terms (to improve data locality). This helps in achieving significant performance improvement in terms of advanced query evaluation runtime, query I/O scan costs, bulk load time, and disk storage space for efficient management of large knowledge bases in RDF stores – addressing all the dimensions of our desiderata.

6.7 Summary

This chapter proposed $KOGNAC$, a scalable algorithm for efficient encoding of RDF terms (with numerical IDs) for storing large Knowledge Bases. $KOGNAC$ adopts a novel combination of estimated frequency-based encoding, FBE, (for frequent terms) and semantic clustering for locality-based encoding, LBE, (for infrequent terms) to encode the underlying entity-relationship graph for improved RDF-style query performance. The FBE procedure enables frequent terms to be represented by smaller IDs thereby improving the storage and index space efficiency for RDF engines. On the other hand, LBE enables related terms to be encoded with close-by IDs to preserve data locality, hence, enhancing advanced SPARQL query (e.g., joins) performance. Large-scale evaluations on benchmark datasets by seamless integration of $KOGNAC$ encoding into multiple state-of-the-art RDF engines and column-stores, exhibited significant improvements in load time, query processing, memory and storage space usage, and I/O and index scan costs. To the best of our knowledge, this work is the first that seeks an improvement of SPARQL query answering via intelligent graph encoding.

7 Conclusion

This dissertation aimed to alleviate certain existing challenges and improve the state-of-the-art pertaining to various stages within the information extraction pipeline, catering to entity-centric knowledge acquisition and management for varied modern day Internet services and data analytics.

7.1 Contributions

The first contribution of this dissertation, $CROCS$, provides a scalable and robust method for high quality named entity co-reference resolution across documents (i.e., CCR). The framework enriches the textual contexts of entity mentions from documents by incorporating co-occurring mentions and extracted features from external knowledge sources to construct *entity summaries*. Words and key-phrases from the entity summaries are used to capture the semantic similarity of mentions across documents for computing equivalence classes of co-referring mentions, by employing a sampling based hierarchical clustering procedure with an information theoretic stopping criterion. Exhaustive experiments on huge news corpora demonstrated the proposed approach to provide enhanced quality for CCR at Web scale, adept at handling different entities with similar surface forms and also *long tail* or newly emerging entities.

The second contribution, $C3EL$, presents a novel framework for joint co-reference resolution and named entity linking (CCR-NEL) of entities across documents to knowledge bases. To this end, $C3EL$ judiciously adopts modules from $CROCS$ and entity disambiguation methods, encapsulated within an iterative approach for interleaved CCR and NEL, enabling the propagation of semantics and contexts across document boundaries. The use of enriched contexts, link confidence validation, along with information feedback across iterations is observed to exhibit improved performance for both CCR and NEL compared to state-of-the-art methodologies, especially in identification of *out-of-knowledge-base* entities.

To assess the quality of extracted entity attributes and relationships from texts, the third contribution proposes the construction of dynamic feature sets based on language models, expressed sentiments labels, and consistency traits for classifying text snippets as credible or deceptive using Support Vector Machines. Additionally, the domain-independent feature models are used to provide a novel concept of possible *evidences* of inconsistency as to why a text was deemed to be non-credible. Empirical evaluations on real-life datasets showcased improved accuracy in detection of misleading facts and robustness of the model across various scenarios.

Efficient management of large knowledge bases, typically represented as labeled graphs, for improving storage and query performance entails the encoding of entity strings to numerical identifiers for RDF engines and column stores. In this regard, the fourth contribution of this dissertation puts forth the $KOGNAC$ algorithm, an intelligent encoding strategy taking into consideration term distribution statistics and semantic relatedness of entities. The combination of *approximate frequent term identification* with *ontological distance based term clustering* enables smaller identifier assignment to highly frequent entities for improved compression (for storage efficiency) and "near-by" identifier assignment to related entities for data locality (for join query performance optimization). Integration of $KOGNAC$ with existing RDF engines for large knowledge repository management exhibited significant improvements in storage requirements, query runtime, and I/O access costs.

In a nutshell, this dissertation presented efficient approaches to entity-centric information harvesting, from text corpora, by co-reference resolution, disambiguation, linking, and computing reliability of facts, along with their precise encoded representation in KBs for enhanced query performance. This provides a robust framework geared towards Web-scale knowledge acquisition and management for named entities from natural language texts.

7.2 Outlook and Future Directions

While this work addresses a number of key problems in the domain of entity-centric knowledge harvesting, the insights obtained and the challenges faced opens up several interesting future research directions for further advancing the state-of-the-art.

Domain Adaptation. The use of factual texts like news articles provides domain skewed entities and information, disregarding other textual depictions such as poetry, works of fiction, and scientific articles. The inculcation of entities and relations extracted from such domains, different from the real-life rendering of entities, requires identification of advanced language constructs (e.g., idioms, metaphors, etc.) and domain-specific terms. Further, the dependence of existing methods on longer texts for capturing contexts and external domain-specific structured knowledge bases

makes these approaches less viable for scenarios with different input characteristics like social media posts and data streams. Extensions of current methods to capture a holistic picture would enable enriched knowledge repositories for better natural language understanding.

Knowledge Inferencing. An interesting area of research involves the reasoning about entities and relationships to derive additional knowledge unavailable in extensional form. Techniques to infer additional attributes and relationships from already known facts about entities would help improve analytics, and alleviate the problems faced due to missing or incomplete information. Reasoning over multiple entity relation hops with richer semantic representation might lead to unprecedented gains for advanced querying and analytics on knowledge repositories.

Richer Semantic Representation. Traditionally knowledge bases have been represented as triples to model entity-entity relationships. However, with the widespread growth of Semantic Web and Linked Open Data, it is increasingly obvious that such simple dyadic relations fall short in capturing multi-entity relationships and information provenance. Hence, there are immense research efforts towards a novel representation scheme or extension of standard RDF format to support richer semantic data. As such, the entire pipeline from index construction to query processing on richer semantic contents needs to be explored and re-engineered.

The Framework of Everything. The life-cycle of information extraction passes through several pipelined stages, such as named entity recognition and mention boundary detection, part-of-speech tagging, co-reference resolution, linking, and knowledge base construction and updation. Existing approaches tend to either efficiently tackle one of the tasks or employ them in sequential phases. The combination of all the different tasks to work in unison within a joint framework stands to benefit an overall quality improvement from information flow across the stages and global cues across documents. Recent algorithms for joint NER-CR-NEL within a document has been shown to exhibit enhanced performance – fueling research for *interleaved approaches* among the related tasks of knowledge acquisition and management.

Bibliography

[Abadi *et al.*, 2009] D. J. Abadi, A. Marcus, S. Madden, and K. Hollenbach. SW-Store: A Vertically Partitioned DBMS for Semantic Web Data Management. *The International Journal on Very Large Data Bases*, 18(2):385–406, 2009.

[Abreu *et al.*, 2013] D. D. Abreu, A. Flores, G. Palma, V. Pestana, J. Pinero, J. Queipo, J. Sanchez, and M. Vidal. Choosing Between Graph Databases and RDF Engines for Consuming and Mining Linked Data. In *International Conference on Consuming Linked Data (COLD)*, pages 37–49, 2013.

[Alexander and Ravada, 2005] N. Alexander and S. Ravada. RDF Object Type and Reification in Oracle. Technical report, Oracle, 2005.

[Alhelbawy and Gaizauskas, 2014] A. Alhelbawy and R. Gaizauskas. Graph Ranking for Collective Named Entity Disambiguation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 75–80, 2014.

[Angles *et al.*, 2014] R. Angles, P. Boncz, J. Larriba-Pey, I. Fundulaki, T. Neumann, O. Erling, P. Neubauer, N. Martinez-Bazan, V. Kotsev, and I. Toma. The Linked Data Benchmark Council: A Graph and RDF Industry Benchmarking Effort. *Special Interest Group on Management of Data (SIGMOD) Record*, 43(1):27–31, 2014.

[Arasu and Garcia-Molina, 2003] A. Arasu and H. Garcia-Molina. Extracting Structured Data from Web Pages. In *International Conference of Special Interest Group on Management of Data (SIGMOD)*, pages 337–348, 2003.

[Arthur and Vassilvitskii, 2007] D. Arthur and S. Vassilvitskii. K-means++: The Advantages of Careful Seeding. In *Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, 2007.

[Artiles *et al.*, 2009] J. Artiles, J. Gonzalo, and S. Sekine. WePS 2 Evaluation Campaign: Overview of the Web People Search Clustering Task. In *International World Wide Web Conference (WWW): Web People Search Evaluation Workshop*, 2009.

[Asahara and Matsumoto, 2003] M. Asahara and Y. Matsumoto. Japanese Named Entity Extraction with Redundant Morphological Analysis. In *Human Language Technology Conference - North American chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 8–15, 2003.

[Bagga and Baldwin, 1998] A. Bagga and B. Baldwin. Entity-based Cross-Document Coreferencing using the Vector Space Model. In *International Conference on Computational Linguistics (COLINGS) - Volume 1*, pages 79–85, 1998.

[Baker, 2012] C. F. Baker. FrameNet, Current Collaborations and Future Goals. *Proceedings of The International Conference on Language Resources and Evaluation (LREC)*, 46(2):269–286, 2012.

[Baron and Freedman, 2008] A. Baron and M. Freedman. Who is Who and What is What: Experiments in Cross-Document Co-Reference. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 274–283, 2008.

[Beheshti *et al.*, 2016] S. Beheshti, B. Benatallah, S. Venugopal, S. H. Ryu, H. R. Motahari-Nezhad, and W. Wang. A Systematic Review and Comparative Analysis of Cross-document Coreference Resolution Methods and Tools. *Computing*, pages 1–37, 2016.

[Bejan and Harabagiu, 2010] C. A. Bejan and S. Harabagiu. Unsupervised Event Coreference Resolution with Rich Linguistic Features. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1412–1422, 2010.

[Beyer *et al.*, 2007] K. Beyer, P. J. Haas, B. Reinwald, Y. Sismanis, and R. Gemulla. On Synopses for Distinct-value Estimation Under Multiset Operations. In *International Conference of Special Interest Group on Management of Data (SIGMOD)*, pages 199–210, 2007.

[Bikel *et al.*, 1997] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: A High-Performance Learning Name-Finder. In *Conference on Applied Natural Language Processing (ANLC)*, pages 194–201, 1997.

[Bishop *et al.*, 2011] B. Bishop, A. Kiryakov, D. Ognyanoff, and I. Peikov. OWLIM: A Family of Scalable Semantic Repositories. *Semantic Web*, 2(1):33–42, 2011.

[Bizer *et al.*, 2009] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - A Crystallization Point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

[Boldi and Vigna, 2004] P. Boldi and S. Vigna. The Webgraph Framework I: Compression Techniques. In *International World Wide Web Conference (WWW)*, pages 595–602, 2004.

[Boncz *et al.*, 2005] P. Boncz, M. Zukowski, and N. Nes. MonetDB/X100: Hyper-Pipelining Query Execution. In *Conference on Innovative Data Systems Research (CIDR)*, pages 225–237, 2005.

[Bornea *et al.*, 2013] M. A. Bornea, J. Dolby, A. Kementsietsidis, K. Srinivas, P. Dantressangle, O. Udrea, and B. Bhattacharjee. Building an Efficient RDF Store Over a Relational Database. In *International Conference of Special Interest Group on Management of Data (SIGMOD)*, pages 121–132, 2013.

[Brickley and Guha, 2004] D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema, 2004. W3C Recommended.

[Brin, 1998] S. Brin. Extracting Patterns and Relations from the World Wide Web. In *International Workshop on The World Wide Web and Databases (WebDB)*, pages 172–183, 1998.

[Broekstra *et al.*, 2002] J. Broekstra, A. Kampman, and F. Van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In *International Semantic Web Conference (ISWC)*, pages 54–68, 2002.

[Budanitsky and Hirst, 2006] A. Budanitsky and G. Hirst. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[Buluc *et al.*, 2013] A. Buluc, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz. Recent Advances in Graph Partitioning. Technical report, Karlsruhe Institute of Technology, 2013.

[Bunescu and Mooney, 2007] R. Bunescu and R. Mooney. Extracting Relations from Text: From Word Sequences to Dependency Paths. In *Natural Language Processing and Text Mining*, pages 29–44. Springer, 2007.

[Bunescu and Paşca, 2006] R. Bunescu and M. Paşca. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 9–16, 2006.

[Cafaro and Tempesta, 2011] M. Cafaro and P. Tempesta. Finding Frequent Items in Parallel. *Concurrency and Computation: Practice and Experience*, 23(15):1774–1788, 2011.

[Carroll *et al.*, 2003] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: Implementing the Semantic Web Recommendations. Technical report, HP Laboratories, 2003.

[Chakrabarti, 2014] A. Chakrabarti. Data Stream Algorithms, 2014. CS49: Dartmouth College.

[Charikar *et al.*, 2002] M. Charikar, K. Chen, and M. Farach-Colton. Finding Frequent Items in Data Streams. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 693–703, 2002.

[Charton *et al.*, 2014] E. Charton, M. J. Meurs, L. Jean-Louis, and M. Gagnon. Mutual Disambiguation for Entity Linking. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 476–481, 2014.

[Chen and Chen, 2015] Y. Chen and H. Chen. Opinion Spam Detection in Web Forum: A Real Case Study. In *International World Wide Web Conference (WWW)*, pages 173–183, 2015.

[Chen and Ji, 2011] Z. Chen and H. Ji. Collaborative Ranking: A Case Study on Entity Linking. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 771–781, 2011.

[Chen and Martin, 2007] Y. Chen and J. Martin. Towards Robust Unsupervised Personal Name Disambiguation. In *Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning: Joint Meeting (EMNLP-CoNLL)*, pages 190–198, 2007.

[Chen *et al.*, 2004] D. Chen, Q. Wu, Y. Ying, and D. Zhou. Support Vector Machine Soft Margin Classifiers: Error Analysis. *Journal of Machine Learning Research*, 5:1143–1175, 2004.

[Cheng and Roth, 2013] X. Cheng and D. Roth. Relational Inference for Wikification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1787–1796, 2013.

[Clark and González-Brenes, 2008] J. H. Clark and J. P. González-Brenes. Coreference Resolution: Current Trends and Future Directions. *Language and Statistics II Literature Review*, 2008.

[Cohen and Sarawagi, 2004] W. W. Cohen and S. Sarawagi. Exploiting Dictionaries in Named Entity Extraction: Combining Semi-Markov Extraction Processes and Data Integration Methods. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 89–98, 2004.

[Cormode and Muthukrishnan, 2005] G. Cormode and S. Muthukrishnan. An Improved Data Stream Summary: The Count-Min Sketch and its Applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[Cornolti *et al.*, 2013] M. Cornolti, P. Ferragina, and M. Ciaramita. A Framework for Benchmarking Entity-Annotation Systems. In *International World Wide Web Conference (WWW)*, pages 249–260, 2013.

[Corro and Gemulla, 2013] L. D. Corro and R. Gemulla. ClausIE: Clause-based Open Information Extraction. In *International World Wide Web Conference (WWW)*, pages 355–366, 2013.

[Cortes and Vapnik, 1995] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

[Crescenzi and Mecca, 2004] V. Crescenzi and G. Mecca. Automatic Information Extraction from Large Websites. *Journal of ACM*, 51(5):731–779, 2004.

[Cucchiarelli and Velardi, 2001] A. Cucchiarelli and P. Velardi. Unsupervised Named Entity Recognition Using Syntactic and Semantic Contextual Evidence. *Computational Linguistics*, 27(1):123–131, 2001.

[Cucerzan, 2007] S. Cucerzan. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning: Joint Meeting (EMNLP-CoNLL)*, pages 708–716, 2007.

[Culotta *et al.*, 2007] A. Culotta, M. L. Wick, and A. McCallum. First-Order Probabilistic Models for Coreference Resolution. In *Human Language Technologies: The Annual*

Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), pages 81–88, 2007.

[Curé *et al.*, 2015] O. Curé, H. Naacke, T. Randriamalala, and B. Amann. LiteMat: A Scalable, Cost-efficient Inference Encoding Scheme for Large RDF Graphs. In *International Conference on Big Data (Big Data)*, pages 1823–1830, 2015.

[Da Silva *et al.*, 2004] J. F. Da Silva, Z. Kozareva, and G. P. Lopes. Cluster Analysis and Classification of Named Entities. In *Conference on Language Resources and Evaluation (LREC)*, 2004.

[Daumé III and Marcu, 2005] H. Daumé III and D. Marcu. A Large-scale Exploration of Effective Global Features for a Joint Entity Detection and Tracking Model. In *Human Language Technology Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP)*, pages 97–104, 2005.

[Dill *et al.*, 2003] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. V. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, and J. Y. Zien. SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation. In *International World Wide Web Conference (WWW)*, pages 178–186, 2003.

[Domingos and Lowd, 2009] P. Domingos and D. Lowd. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan and Claypool, 2009.

[Domingos *et al.*, 2007] P. Domingos, S. Kok, D. Lowd, H. Poon, M. Richardson, and P. Singla. Markov Logic. In *Probabilistic ILP*, pages 92–117. Springer-Verlag, 2007.

[Durrett and Klein, 2013] G. Durrett and D. Klein. Easy Victories and Uphill Battles in Coreference Resolution. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 1971–1982, 2013.

[Durrett and Klein, 2014] G. Durrett and D. Klein. A Joint Model for Entity Analysis: Coreference, Typing, and Linking. *Transactions of the Association for Computational Linguistics*, 2:477–490, 2014.

[Dutta and Weikum, 2015a] S. Dutta and G. Weikum. Cross-Document Co-Reference Resolution using Sample-Based Clustering with Knowledge Enrichment. *Transactions of the Association for Computational Linguistics*, 3:15–28, 2015a.

[Dutta and Weikum, 2015b] S. Dutta and G. Weikum. C3EL: A Joint Model for Cross-Document Co-Reference Resolution and Entity Linking. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 846–856, 2015b.

[Erling and Mikhailov, 2009] O. Erling and I. Mikhailov. Virtuoso: RDF Support in a Native RDBMS. In *Semantic Web Information Management - A Model-Based Perspective*, pages 501–519. Springer, 2009.

[Etzioni *et al.*, 2005] O. Etzioni, M. Cafarella, D. Downey, A. M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Artificial Intelligence*, 165:91–134, 2005.

[Evans, 2003] R. Evans. A Framework for Named Entity Recognition in the Open Domain. In *Recent Advances in Natural Language Processing (RANLP)*, pages 137–144, 2003.

[Fan *et al.*, 2008] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[Fang *et al.*, 1999] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, and J. D. Ullman. Computing Iceberg Queries Efficiently. In *International Conference on Very Large Data Bases (VLDB)*, pages 299–310, 1999.

[Fellbaum, 1998] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[Feng *et al.*, 2012] S. Feng, R. Banerjee, and Y. Choi. Syntactic Stylometry for Deception Detection. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 171–175, 2012.

[Fernandez *et al.*, 2013] J. D. Fernandez, M. A. Martinez-Prieto, C. Gutierrez, A. Polleres, and M. Arias. Binary RDF Representation for Publication and Exchange (HDT). *Journal of Web Semantics*, 19:22–41, 2013.

[Fernandez *et al.*, 2014] N. Fernandez, J. Arias, L. Sanchez, D. Fuentes-Lorenzo, and O. Corcho. RDSZ: An Approach for Lossless RDF Stream Compression. In *European Semantic Web Conference (ESWC)*, pages 52–67, 2014.

[Finkel *et al.*, 2005] J. R. Finkel, T. Grenager, and C. D. Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 363–370, 2005.

[Fleischman and Hovy, 2002] M. Fleischman and E. Hovy. Fine Grained Classification of Named Entities. In *International Conference on Computational Linguistics (COLINGS)*, pages 1–7, 2002.

[Gabrilovich and Markovitch, 2007] E. Gabrilovich and S. Markovitch. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1606–1611, 2007.

[Gabrilovich *et al.*, 2013] E. Gabrilovich, M. Ringgaard, and A. Subramanya. FACC1: Freebase Annotation of ClueWeb Corpora, Version 1 (Format version 1, Correction level 0), 2013.

[Gooi and Allan, 2004] C. H. Gooi and J. Allan. Cross-Document Coreference on a Large Scale Corpus. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 9–16, 2004.

[Grünwald, 2007] P. D. Grünwald. *The Minimum Description Length Principle (Adaptive Computation and Machine Learning)*. MIT Press, 2007.

[Grusec and Hastings, 2007] J. E. Grusec and P. D. Hastings. *Handbook of Socialization: Theory and Research*. Guilford Press, 2007.

[Guo *et al.*, 2005] Y. Guo, Z. Pan, and J. Heflin. LUBM: A Benchmark for OWL Knowledge Base Systems. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2):158–182, 2005.

[Gurajada *et al.*, 2014] S. Gurajada, S. Seufert, I. Miliaraki, and M. Theobald. TriAD: A Distributed Shared-nothing RDF Engine based on Asynchronous Message Passing. In *International Conference of Special Interest Group on Management of Data (SIGMOD)*, pages 289–300, 2014.

[Hachey *et al.*, 2013] B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J. R. Curran. Evaluating Entity Linking with Wikipedia. *Artificial Intelligence Journal*, 194:130–150, 2013.

[Haghighi and Klein, 2007] A. Haghighi and D. Klein. Unsupervised Coreference Resolution in a Nonparametric Bayesian Model. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 640–649, 2007.

[Haghighi and Klein, 2009] A. Haghighi and D. Klein. Simple Coreference Resolution with Rich Syntactic and Semantic Features. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 1152–1161, 2009.

[Haghighi and Klein, 2010] A. Haghighi and D. Klein. Coreference Resolution in a Modular, Entity-Centered Model. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 385–393, 2010.

[Hajishirzi *et al.*, 2013] H. Hajishirzi, L. Zilles, D. S. Weld, and L. S. Zettlemoyer. Joint Coreference Resolution and Named-Entity Linking with Multi-Pass Sieves. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 289–299, 2013.

[Han *et al.*, 2004] J. Han, J. Pei, Y. Yin, and R. Mao. Mining Frequent Patterns Without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004.

[Han *et al.*, 2011] X. Han, L. Sun, and J. Zhao. Collective Entity Linking in Web Text: A Graph-based Method. In *International Special Interest Group Conference on Research and Development in Information Retrieval (SIGIR)*, pages 765–774, 2011.

[Harbi *et al.*, 2015] R. Harbi, I. Abdelaziz, P. Kalnis, and N. Mamoulis. Evaluating SPARQL Queries on Massive RDF Datasets. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 8(12):1848–1851, 2015.

[Harris *et al.*, 2009] S. Harris, N. Lamb, and N. Shadbolt. 4store: The Design and Implementation of a Clustered RDF Store. In *International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS)*, pages 94–109, 2009.

[Harris *et al.*, 2013] S. Harris, A. Seaborne, and E. Prud'hommeaux. SPARQL 1.1 Query Language. *W3C Recommendation*, 21, 2013.

[Hassan *et al.*, 2015] N. Hassan, B. Adair, J. T. Hamilton, C. Li, M. Tremayne, J. Yang, and C. Yu. The Quest to Automate Fact-Checking. In *Computation + Journalism Symposium*, 2015.

[Hassell *et al.*, 2006] J. Hassell, B. Aleman-Meza, and I. B. Arpinar. Ontology-driven Automatic Entity Disambiguation in Unstructured Text. In *International Conference on The Semantic Web (ISWC)*, pages 44–57, 2006.

[Hayes, 2004] P. Hayes, editor. *RDF Semantics*. W3C Recommendation, 2004.

[Hearst, 1992] M. A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. In *International Conference on Computational Linguistics (COLINGS)*, pages 539–545, 1992.

[Heath and Bizer, 2011] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan and Claypool, 2011.

[Hindle, 1990] D. Hindle. Noun Classification from Predicate-argument Structures. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 268–275, 1990.

[Hoffart *et al.*, 2011] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust Disambiguation of Named Entities in Text. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 782–792, 2011.

[Hoffart *et al.*, 2014] J. Hoffart, Y. Altun, and G. Weikum. Discovering Emerging Entities with Ambiguous Names. In *International World Wide Web Conference (WWW)*, pages 385–396, 2014.

[Hoffart, 2015] J. Hoffart. *Discovering and Disambiguating Named Entities in Text*. PhD thesis, Saarland University, 2015.

[Houlsby and Ciaramita, 2014] N. Houlsby and M. Ciaramita. A Scalable Gibbs Sampler for Probabilistic Entity Linking. In *European Conference on Information Retrieval (ECIR)*, pages 335–346, 2014.

[Hourdakis *et al.*, 2010] N. Hourdakis, M. Argyriou, G. M. Petrakis, and E. E. Milios. Hierarchical Clustering in Medical Document Collections: The BIC-Means Method. *Journal of Digital Information Management*, 8(2):71–77, 2010.

[Hu and Liu, 2004] M. Hu and B. Liu. Mining and Summarizing Customer Reviews. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 168–177, 2004.

[Hu *et al.*, 2012] N. Hu, I. Bose, N. S. Koh, and L. Liu. Manipulation of Online Reviews: An Analysis of Ratings, Readability, and Sentiments. *Decision Support Systems*, 52(3):674–684, 2012.

[Huffman, 1952] D. A. Huffman. A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the Institute of Radio Engineers*, 40(9):1098–1101, 1952.

[IBM, 2012] IBM. This is Watson. IBM Journal of Research and Development, 2012. 56(3/4).

[Ji and Grishman, 2011] H. Ji and R. Grishman. Knowledge Base Population: Successful Approaches and Challenges. In *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT) - Volume 1*, pages 1148–1158, 2011.

[Jin *et al.*, 2014] Y. Jin, E. Kıcıman, K. Wang, and R. Loynd. Entity Linking at the Tail: Sparse Signals, Unknown Entities, and Phrase Models. In *International Conference on Web Search and Data Mining (WSDM)*, pages 453–462, 2014.

[Jindal and Liu, 2007] N. Jindal and B. Liu. Analyzing and Detecting Review Spam. In *International Conference on Data Mining (ICDM)*, pages 547–552, 2007.

[Jindal and Liu, 2008] N. Jindal and B. Liu. Opinion Spam and Analysis. In *International Conference on Web Search and Data Mining (WSDM)*, pages 219–230, 2008.

[Jones *et al.*, 2000] K. S. Jones, S. Walker, and S. E. Robertson. A Probabilistic Model of Information Retrieval: Development and Comparative Experiments: Part 1 and 2. *Information Processing and Management*, 36(6):779–840, 2000.

[Karp *et al.*, 2003] R. M. Karp, S. Shenker, and C. H. Papadimitriou. A Simple Algorithm for Finding Frequent Elements in Streams and Bags. *Transactions on Database Systems*, 28(1):51–55, 2003.

[Karypis and Kumar, 1998] G. Karypis and V. Kumar. A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs. *Journal on Scientific Computing*, 20(1):359–392, 1998.

[Kernighan and Lin, 1970] B. W. Kernighan and S. Lin. An Efficient Heuristic Procedure for Partitioning Graphs. Technical report, Bell System Technical Journal, 1970.

[Klyne and Carroll, 2006] G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax, 2006. W3C.

[Koller and Friedman, 2009] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[Kotoulas *et al.*, 2010] S. Kotoulas, E. Oren, and F. Van Harmelen. Mind the Data Skew: Distributed Inferencing by Speeddating in Elastic Regions. In *International World Wide Web Conference (WWW)*, pages 531–540, 2010.

[Koudas *et al.*, 2006] N. Koudas, S. Sarawagi, and D. Srivastava. Record Linkage: Similarity Measures and Algorithms. In *International Conference of Special Interest Group on Management of Data (SIGMOD)*, pages 802–803, 2006.

[Krishnamurty *et al.*, 2012] A. Krishnamurty, S. Balakrishnan, M. Xu, and A. Singh. Efficient Active Algorithms for Hierarchical Clustering. In *International Conference on Machine Learning (ICML)*, pages 887–894, 2012.

[Kulkarni *et al.*, 2009] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective Annotation of Wikipedia Entities in Web Text. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 457–466, 2009.

[Kullback and Leibler, 1951] S. Kullback and R. A. Leibler. On Information and Sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[Kullback, 1987] S. Kullback. Letter to the Editor: The Kullback-Leibler Distance. *The American Statistician*, 41(4):340–341, 1987.

[Kushmerick *et al.*, 1997] N. Kushmerick, D. S. Weld, and R. Doorenbos. Wrapper Induction for Information Extraction. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 729–736, 1997.

[Lample *et al.*, 2016] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural Architectures for Named Entity Recognition. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 260–270, 2016.

[Lao *et al.*, 2011] N. Lao, T. M. Mitchell, and W. W. Cohen. Random Walk Inference and Learning in A Large Scale Knowledge Base. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 529–539, 2011.

[Le and Mikolov, 2014] Q. Le and T. Mikolov. Distributed Representations of Sentences and Documents. In *International Conference on Machine Learning (ICML)*, pages 1188–1196, 2014.

[Leal, 2013] J. P. Leal. Using Proximity to Compute Semantic Relatedness in RDF Graphs. *Computer Science and Information Systems*, 10(4):1727–1746, 2013.

[Leaman and Lu, 2016] R. Leaman and Z. Lu. TaggerOne: Joint Named Entity Recognition and Normalization with Semi-Markov Models. *Bioinformatics*, 32(18):2839–2846, 2016.

[Lee *et al.*, 2011] 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 *Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning: Joint Meeting (EMNLP-CoNLL) – Shared Task*, pages 28–34, 2011.

[Lee *et al.*, 2012] H. Lee, M. Recasens, A. X. Chang, M. Surdeanu, and D. Jurafsky. Joint Entity and Event Coreference Resolution across Documents. In *Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning: Joint Meeting (EMNLP-CoNLL)*, pages 489–500, 2012.

[Lee *et al.*, 2013] H. Lee, A. Chang, Y. Peirsman, Chambers N., M. Surdeanu, and D. Jurafsky. Deterministic Coreference Resolution based on Entity-centric, Precision-ranked Rules. *Computational Linguistics*, 39(4):885–916, 2013.

[Leis *et al.*, 2013] V. Leis, A. Kemper, and T. Neumann. The Adaptive Radix Tree: ARTful Indexing for Main-memory Databases. In *International Conference on Data Engineering (ICDE)*, pages 38–49, 2013.

[Lenat, 1995] D. B. Lenat. CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11):32–38, 1995.

[Li and Sakamoto, 2015] H. Li and Y. Sakamoto. Computing the Veracity of Information through Crowds: A Method for Reducing the Spread of False Messages on Social Media. In *Hawaii International Conference on System Sciences*, pages 2003–2012, 2015.

[Li *et al.*, 2008] H. Li, Y. Wang, D. Zhang, M. Zhang, and E. Y. Chang. PFP: Parallel FP-growth for Query Recommendation. In *Conference on Recommender Systems (RecSys)*, pages 107–114, 2008.

[Li *et al.*, 2013] J. Li, M. Ott, and C. Cardie. Identifying Manipulative Offerings on Review Portals. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 1933–1942, 2013.

[Li *et al.*, 2014a] H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao. Spotting Fake Reviews via Collective Positive-Unlabeled Learning. In *International Conference on Data Mining (ICDM)*, pages 899–904, 2014a.

[Li *et al.*, 2014b] J. Li, M. Ott, C. Cardie, and E. Hovy. Towards a General Rule for Identifying Deceptive Opinion Spam. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1566–1576, 2014b.

[Liefke and Suciu, 2000] H. Liefke and D. Suciu. XMill: An Efficient Compressor for XML Data. In *International Conference of Special Interest Group on Management of Data (SIGMOD)*, pages 153–164, 2000.

[Lim *et al.*, 2010] E. Lim, V. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. Detecting Product Review Spammers using Rating Behaviors. In *International Conference on Information and Knowledge Management (CIKM)*, pages 939–948, 2010.

[Lin and He, 2009] C. Lin and Y. He. Joint Sentiment/Topic Model for Sentiment Analysis. In *International Conference on Information and Knowledge Management (CIKM)*, pages 375–384, 2009.

[Lin and Pantel, 2001] D. Lin and P. Pantel. DIRT - Discovery of Inference Rules from Text. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 323–328, 2001.

[Ling and Weld, 2012] X. Ling and D. S. Weld. Fine-grained Entity Recognition. In *Conference on Artificial Intelligence (AAAI)*, pages 94–100, 2012.

[Liu *et al.*, 2010] B. Liu, L. Chiticariu, V. Chu, H. Jagadish, and F. Reiss. Automatic Rule Refinement for Information Extraction. In *International Conference on Very Large Data Bases (VLDB)*, pages 588–597, 2010.

[Liu *et al.*, 2011] X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing Named Entities in Tweets. In *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 359–367, 2011.

[Liu, 2009] T. Liu. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–233, 2009.

[Loeliger, 2004] H. A. Loeliger. An Introduction to Factor Graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, 2004.

[Luca and Zervas, 2015] M. Luca and G. Zervas. Fake It Till You Make It: Reputation, Competition, and Yelp Review Fraud. Technical report, Harvard Business School, 2015.

[Luo, 2005] X. Luo. On Coreference Resolution Performance Metrics. In *Human Language Technology Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP)*, pages 25–32, 2005.

[Mann and Yarowsky, 2003] G. S. Mann and D. Yarowsky. Unsupervised Personal Name Disambiguation. In *Conference on Natural language learning at HLT-NAACL (CoNLL/HLT-NAACL)*, pages 33–40, 2003.

[Mansell, 2002] R. Mansell. Constructing the Knowledge-base for Knowledge-Driven Development. *Journal of Knowledge Management*, 6(4):317–329, 2002.

[Matthee and Viktor, 2001] M. Matthee and H. Viktor. Data Mining for Organizational Knowledge Management: Aiding Decision, Sense and Policy Making. In *European Conference on Knowledge Management (ECKM)*, pages 353–366, 2001.

[Mausam *et al.*, 2012] Mausam, M. Schmitz, S. Soderland, B. Robert, and O. Etzioni. Open Language Learning for Information Extraction. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 523–534, 2012.

[Mavlyutov *et al.*, 2015] R. Mavlyutov, M. Wylot, and P. Cudré-Mauroux. A Comparison of Data Structures to Manage URIs on the Web of Data. In *European Semantic Web Conference (ESWC)*, pages 137–151, 2015.

[Mayfield *et al.*, 2009] J. Mayfield, D. Alexander, B. Dorr, J. Eisner, T. Elsayed, T. Finin, C. Fink, M. Freedman, N. Garera, P. McNamee, S. Mohammad, D. Oard, C. Piatko, A. Sayeed, Z. Syed, R. Weischedel, T. Xu, and D. 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.

[McCallum and Li, 2003] A. McCallum and W. Li. Early Results for Named Entity Recognition with Conditional Random Fields, Features Induction and Web-Enhanced Lexicons. In *Conference on Computational Natural Language Learning (CoNLL)*, pages 188–191, 2003.

[McNamee *et al.*, 2013] P. McNamee, J. Mayfield, T. Finin, T. Oates, D. Lawrie, T. Xu, and D. W. Oard. KELVIN: A Tool for Automated Knowledge Base Construction. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 32–35, 2013.

[Metzger and Flanagin, 2013] M. J. Metzger and A. J. Flanagin. Credibility and Trust of Information in Online Environments: The Use of Cognitive Heuristics. *Journal of Pragmatics*, 59:210–220, 2013.

[Michelakis *et al.*, 2009] E. Michelakis, R. Krishnamurthy, P. Haas, and S. Vaithyanathan. Uncertainty Management in Rule-based Information Extraction Systems. In *International Conference of Special Interest Group on Management of Data (SIGMOD)*, pages 101–114, 2009.

[Mihalcea and Strapparava, 2009] R. Mihalcea and C. Strapparava. The Lie Detector: Explorations in the Automatic Recognition of Deceptive Language. In *Annual Meeting of the Association for Computational Linguistics and International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP)*, pages 309–312, 2009.

[Mikheev *et al.*, 1999] A. Mikheev, M. Moens, and C. Grover. Named Entity Recognition without Gazetteers. In *Conference of European Chapter of the Association for Computational Linguistics (EACL)*, pages 1–8, 1999.

[Milne and Witten, 2008] D. N. Milne and I. H. Witten. Learning to Link with Wikipedia. In *International Conference on Information and Knowledge Management (CIKM)*, pages 509–518, 2008.

[Milne and Witten, 2013] D. Milne and I. H. Witten. An Open-source Toolkit for Mining Wikipedia. *Artificial Intelligence*, 194:222–239, 2013.

[Milo and Suciu, 1999] T. Milo and D. Suciu. Index Structures for Path Expressions. In *International Conference on Database Theory (ICDT)*, pages 277–295, 1999.

[Mintz *et al.*, 2009] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant Supervision for Relation Extraction Without Labeled Data. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1003–1011, 2009.

[Misra and Gries, 1982] J. Misra and D. Gries. Finding Repeated Elements. *Science of Computer Programming*, 2(2):143–152, 1982.

[Moschitti *et al.*, 2008] A. Moschitti, D. Pighin, and R. Basili. Tree Kernels for Semantic Role Labeling. *Computational Linguistics*, 34(2):193–224, 2008.

[Mukherjee *et al.*, 2012] A. Mukherjee, B. Liu, and N. S. Glance. Spotting Fake Reviewer Groups in Consumer Reviews. In *International World Wide Web Conference (WWW)*, pages 191–200, 2012.

[Mukherjee *et al.*, 2013a] A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance. What Yelp Fake Review Filter Might Be Doing? In *International Conference on Web and Social Media (ICWSM)*, pages 409–418, 2013a.

[Mukherjee *et al.*, 2013b] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh. Spotting Opinion Spammers using Behavioral Footprints. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 632–640, 2013b.

[Mukherjee *et al.*, 2014] S. Mukherjee, G. Basu, and S. Joshi. Joint Author Sentiment Topic Model. In *SIAM International Conference on Data Mining (SDM)*, pages 370–378, 2014.

[Mukherjee *et al.*, 2016] S. Mukherjee, S. Dutta, and G. Weikum. Credible Review Detection with Limited Information using Consistency Features. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML-PKDD)*, pages 195–213, 2016.

[Nadeau and Sekine, 2007] D. Nadeau and S. Sekine. A Survey of Named Entity Recognition and Classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[Nakashole *et al.*, 2011] N. Nakashole, M. Theobald, and G. Weikum. Scalable Knowledge Harvesting with High Precision and High Recall. In *Web Search and Data Mining (WSDM)*, pages 227–236, 2011.

[Nakashole *et al.*, 2013] N. Nakashole, T. Tylenda, and G. Weikum. Fine-grained Semantic Typing of Emerging Entities. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1488–1497, 2013.

[Neumann and Weikum, 2008] T. Neumann and G. Weikum. RDF-3X: A RISC-style Engine for RDF. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 1(1):647–659, 2008.

[Ng, 2007] V. Ng. Shallow Semantics for Coreference Resolution. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1689–1694, 2007.

[Ng, 2008] V. Ng. Unsupervised Models for Coreference Resolution. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 640–649, 2008.

[Ng, 2010] V. Ng. Supervised Noun Phrase Coreference Research: The First Fifteen Years. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1396–1411, 2010.

[Ng, 2016] V. Ng. *Advanced Machine Learning Models for Coreference Resolution*, pages 283–313. Springer Berlin Heidelberg, 2016.

[Ngomo *et al.*, 2014] A. N. Ngomo, M. Röder, and R. Usbeck. Cross-document Coreference Resolution using Latent Features. In *International Conference on Linked Data for Information Extraction (LD4IE)*, pages 33–44, 2014.

[Nguyen *et al.*, 2016] D. Nguyen, M. Theobald, and G. Weikum. J-NERD: Joint Named Entity Recognition and Disambiguation with Rich Linguistic Features. *Transactions of the Association for Computational Linguistics*, 4:215–229, 2016.

[Niepert *et al.*, 2012] M. Niepert, C. Meilicke, and H. Stuckenschmidt. Towards Distributed MCMC Inference In Probabilistic Knowledge Bases. In *Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 1–6, 2012.

[Niu *et al.*, 2004] C. Niu, W. Li, and R. K. Srihari. Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004. article 597.

[Niu *et al.*, 2012] F. Niu, C. Zhang, C. Ré, and J. W. Shavlik. DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. *Very Large Data Search (VLDS)*, 12:25–28, 2012.

[Oracle, 2011] Oracle. *Oracle NoSQL Database*, 2011.

[Ott *et al.*, 2011] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT) - Volume 1*, pages 309–319, 2011.

[Ott *et al.*, 2013] M. Ott, C. Cardie, and J. T. Hancock. Negative Deceptive Opinion Spam. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 497–501, 2013.

[Owens *et al.*, 2008] A. Owens, A. Seaborne, N. Gibbins, and M. C. Schraefel. Clustered TDB: A Clustered Triple Store for Jena. Technical report, University of Southampton, 2008.

[Pacher *et al.*, 2011] D. Pacher, R. Binna, and G. Specht. Data Locality in Graph Databases through N-Body Simulation. In *Grundlagen von Datenbanken*, pages 85–90, 2011.

[Passos *et al.*, 2014] A. Passos, V. Kumar, and A. McCallum. Lexicon Infused Phrase Embeddings for Named Entity Resolution. In *Conference on Computational Natural Language Learning (CoNLL)*, pages 78–86, 2014.

[Patchigolla, 2011] V. Patchigolla. Comparison of Clustered RDF Data Stores. Master's thesis, Purdue University, 2011.

[Pennebaker *et al.*, 2001] J. W. Pennebaker, M. E. Francis, and R. J. Booth. *Linguistic Inquiry and Word Count: A Computerized Text Analysis Program*. Psychology Press, 2001.

[Poesio *et al.*, 2008] M. Poesio, D. Day, R. Artstein, J. Duncan, V. Eidelman, C. Giuliano, and M. Kabadjov. Exploiting Lexical and Encyclopedic Resources For Entity Disambiguation. Technical report, John Hopkins Center for Language and Speech Processing Summer Workshop, 2008.

[Ponzetto and Strube, 2006] S. P. Ponzetto and M. Strube. Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 192–199, 2006.

[Raghunathan *et al.*, 2010] K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning. A Multi-pass Sieve for Coreference Resolution. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 492–501, 2010.

[Rahman and Ng, 2011a] A. Rahman and V. Ng. Coreference Resolution with World Knowledge. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 814–824, 2011a.

[Rahman and Ng, 2011b] A. Rahman and V. Ng. Ensemble-Based Coreference Resolution. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1884–1889, 2011b.

[Rahman *et al.*, 2015] M. Rahman, B. Carbunar, J. Ballesteros, and D. H. Chau. To Catch a Fake: Curbing Deceptive Yelp Ratings and Venues. *Statistical Analysis and Data Mining*, 8(3):147–161, 2015.

[Rao *et al.*, 2010] D. Rao, P. McNamee, and M. Dredze. Streaming Cross Document Entity Coreference Resolution. In *International Conference on Computational Linguistics (COLING): Poster*, pages 1050–1058, 2010.

[Ratinov and Roth, 2009] L. A. Ratinov and D. Roth. Design Challenges and Misconceptions in Named Entity Recognition. In *Conference on Computational Natural Language Learning (CoNLL)*, pages 147–155, 2009.

[Ratinov and Roth, 2012] L. A. Ratinov and D. Roth. Learning-based Multi-Sieve Coreference Resolution with Knowledge. In *Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning: Joint Meeting (EMNLP-CoNLL)*, pages 1234–1244, 2012.

[Ratinov *et al.*, 2011] L. A. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and Global Algorithms for Disambiguation to Wikipedia. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1375–1384, 2011.

[Recasens *et al.*, 2013] M. Recasens, M. C. de Marneffe, and C. Potts. The Life and Death of Discourse Entities: Identifying Singleton Mentions. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 627–633, 2013.

[Reiss *et al.*, 2008] F. Reiss, S. Raghavan, R. Krishnamurthy, H. Zhu, and S. Vaithyanathan. An Algebraic Approach to Rule-based Information Extraction. In *International Conference on Data Engineering (ICDE)*, pages 933–942, 2008.

[Richardson and Domingos, 2006] M. Richardson and P. Domingos. Markov Logic Networks. *Journal of Machine Learning*, 62(1-2):107–136, 2006.

[Robinson *et al.*, 2015] I. Robinson, J. Webber, and E. Eifrem. *Graph Databases: New Opportunities for Connected Data*. O'Reilly, 2015.

[Sahuguet and Azavant, 2001] A. Sahuguet and F. Azavant. Building Intelligent Web Applications using Lightweight Wrappers. *Data and Knowledge Engineering*, 36(3):283–316, 2001.

[Sakr and Al-Naymat, 2010] S. Sakr and G. Al-Naymat. Relational Processing of RDF Queries: A Survey. *Special Interest Group on Management of Data (SIGMOD) Record*, 38(4):23–28, 2010.

[Salton and Buckley, 1988] G. Salton and C. Buckley. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[Salton, 1989] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[Sandhaus, 2008] E. Sandhaus. The New York Times Annotated Corpus Overview. Technical report, Linguistic Data Consortium, 2008.

[Schwarz, 1978] G. E. Schwarz. Estimating the Dimension of a Model. *Annals of Statistics*, 6(2):461–464, 1978.

[Sekine and Nobata, 2004] S. Sekine and C. Nobata. Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy. In *Conference on Language Resources and Evaluation (LREC)*, pages 1977–1980, 2004.

[Sekine and Ranchhod, 2016] S. Sekine and E. Ranchhod. *Named Entities: Recognition, Classification and Use*. John Benjamins Publishing Company, 2016.

[Shamir and Tishby, 2011] O. Shamir and N. Tishby. Spectral Clustering on a Budget. *Journal of Machine Learning Research*, Track 15:661–669, 2011.

[Shannon and Benninger, 2014] M. Shannon and C. Benninger. Telemetry Database Query Performance Review. Technical report, Sophos Lab, 2014.

[Shen *et al.*, 2007] W. Shen, A. Doan, J. Naughton, and R. Ramakrishnan. Declarative Information Extraction using Datalog with Embedded Extraction Predicates. In *International Conference on Very Large Data Bases (VLDB)*, pages 1033–1044, 2007.

[Shen *et al.*, 2012] W. Shen, J. Wang, P. Luo, and M. Wang. Linden: Linking Named Entities with Knowledge Base via Semantic Knowledge. In *International Conference on World Wide Web (WWW)*, pages 449–458, 2012.

[Shen *et al.*, 2015] W. Shen, J. Wang, and J. Han. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2015.

[Sidirourgos *et al.*, 2008] L. Sidirourgos, R. Goncalves, M. L. Kersten, N. Nes, and S. Manegold. Column-Store Support for RDF Data Management: Not All Swans are White. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 1(2):1553–1563, 2008.

[Sil and Yates, 2013] A. Sil and A. Yates. Re-ranking for Joint Named-Entity Recognition and Linking. In *International Conference on Information and Knowledge Management (CIKM)*, pages 2369–2374, 2013.

[Singh *et al.*, 2010] S. Singh, M. L. Wick, and A. McCallum. Distantly Labeling Data for Large Scale Cross-Document Coreference. arxiv.org, 2010. CoRR: abs/1005.4298.

[Singh *et al.*, 2011] S. Singh, A. Subramanya, F. Pereira, and A. McCallum. Large-scale Cross-document Coreference using Distributed Inference and Hierarchical Models. In *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL: HLT) - Volume 1*, pages 793–803, 2011.

[Singh *et al.*, 2013] S. Singh, S. Reidel, B. Martin, J. Zheng, and A. McCallum. Joint Inference of Entities, Relations, and Coreference. In *Workshop on Automated Knowledge Base Construction (AKBC)*, pages 1–6, 2013.

[Singhal, 2012] A. Singhal. Introducing the Knowledge Graph: Things, not Strings. Google Blog, 2012.

[Singla and Domingos, 2006] P. Singla and P. Domingos. Entity Resolution with Markov Logic. In *International Conference on Data Mining (ICDM)*, pages 572–582, 2006.

[Sterner, 2000] H. Sterner. Managing Organizational Knowing: A Conceptual Framework for making Decision, Sense and Policy. In *European Conference on Knowledge Management (ECKM)*, pages 193–200, 2000.

[Strapparava and Valitutti, 2004] C. Strapparava and A. Valitutti. WordNet-Affect: An Affective Extension of WordNet. In *International Conference on Language Resources and Evaluation (LREC)*, pages 1083–1086, 2004.

[Suchanek and Weikum, 2014] F. M. Suchanek and G. Weikum. Knowledge Bases in the Age of Big Data Analytics. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 7(13):1713–1714, 2014.

[Suchanek *et al.*, 2006] F. M. Suchanek, G. Ifrim, and G. Weikum. Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 712–717, 2006.

[Suchanek *et al.*, 2007] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: A Core of Semantic Knowledge. In *International World Wide Web Conference (WWW)*, pages 697–706, 2007.

[Suchanek *et al.*, 2009] F. Suchanek, M. Sozio, and G. Weikum. SOFIE: A Self-organizing Framework for Information Extraction. In *International Conference on World Wide Web (WWW)*, pages 631–640, 2009.

[Suchanek *et al.*, 2012] F. M. Suchanek, J. Fan, R. Hoffmann, S. Riedel, and P. P. Talukdar, editors. *Advances in Automated Knowledge Base Construction*, Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX), 2012.

[Suchanek, 2008] F. M. Suchanek. *Automated Construction and Growth of a Large Ontology*. PhD thesis, Saarland University, 2008.

[Sun *et al.*, 2013] H. Sun, A. Morales, and X. Yan. Synthetic Review Spamming and Defense. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1088–1096, 2013.

[Sutton and McCallum, 2004] C. Sutton and A. McCallum. Collective Segmentation and Labeling of Distant Entities in Information Extraction. Technical report, University of Massachusetts, USA, 2004.

[Taneva *et al.*, 2011] B. Taneva, M. Kacimi, and G. Weikum. Finding Images of Difficult Entities in the Long Tail. In *International Conference on Information and Knowledge Management (CIKM)*, pages 189–194, 2011.

[Tarjan, 1977] R. E. Tarjan. Finding Optimum Branchings. *Networks*, 7(1):25–35, 1977.

[Thielen, 1995] C. Thielen. An Approach to Proper Name Tagging for German. In *Conference of European Chapter of the Association for Computational Linguistics (EACL)*, 1995.

[Udrea *et al.*, 2007] O. Udrea, L. Getoor, and R. J. Miller. Leveraging Data and Structure in Ontology Integration. In *International Conference of Special Interest Group on Management of Data (SIGMOD)*, pages 449–460, 2007.

[Urbani *et al.*, 2013] J. Urbani, J. Maassen, N. Drost, F. Seinstra, and H. Bal. Scalable RDF Data Compression with MapReduce. *Concurrency and Computation: Practice and Experience*, 25(1):24–39, 2013.

[Urbani *et al.*, 2016] J. Urbani, S. Dutta, S. Gurajada, and G. Weikum. KOGNAC: Efficient Encoding of Large Knowledge Graphs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3896–3902, 2016.

[van Ast *et al.*, 2004] J. F. van Ast, J. L. Talmon, W.O. Renier, and A. Hasman. An Approach to Knowledge Base Construction Based on Expert Opinions. *Methods of Information in Medicine*, 43(4):427–432, 2004.

[von Luxburg, 2007] U. von Luxburg. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[Vrandečić and Krötzsch, 2014] D. Vrandečić and M. Krötzsch. Wikidata: A Free Collaborative Knowledge Base. *Communications of the ACM*, 57(10), 2014.

[Wang and Cohen, 2009] R. C. Wang and W. W. Cohen. Character-level Analysis of Semi-structured Documents for Set Expansion. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 1503–1512, 2009.

[Wang *et al.*, 2011] G. Wang, S. Xie, B. Liu, and P. S. Yu. Review Graph Based Online Store Review Spammer Detection. In *International Conference on Data Mining (ICDM)*, pages 1242–1247, 2011.

[Wauthier *et al.*, 2012] F. L. Wauthier, N. Jojic, and M. I. Jordan. Active Spectral Clustering via Iterative Uncertainty Reduction. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1339–1347, 2012.

[Weikum and Theobald, 2010] G. Weikum and M. Theobald. From Information to Knowledge: Harvesting Entities and Relationships from Web Sources. In *Symposium on Principles of Database Systems (PODS)*, pages 65–76, 2010.

[Weld *et al.*, 2008] D. S. Weld, R. Hoffmann, and F. Wu. Using Wikipedia to Bootstrap Open Information Extraction. *Special Interest Group on Management of Data (SIGMOD) Record*, 37(4):62–68, 2008.

[Wu and Weld, 2010] F. Wu and D. Weld. Open Information Extraction using Wikipedia. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 118–127, 2010.

[Wu *et al.*, 2014] Y. Wu, P. K. Agarwal, C. Li, J. Yang, and C. Yu. Toward Computational Fact-Checking. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 7(7):589–600, 2014.

[Yahya *et al.*, 2014] M. Yahya, S. E. Whang, R. Gupta, and A. Halevy. ReNoun: Fact Extraction for Nominal Attributes. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 325–335, 2014.

[Yan *et al.*, 2009] D. Yan, L. Huang, and M. I. Jordan. Fast Approximate Spectral Clustering. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 907–916, 2009.

[Yoo and Gretzel, 2009] K. H. Yoo and U. Gretzel. Comparison of Deceptive and Truthful Travel Reviews. In *International Conference on Information Technology and Travel & Tourism (ENTER)*, pages 37–47, 2009.

[Yosef *et al.*, 2011] M. A. Yosef, J. Hoffart, M. Spaniol, and G. Weikum. AIDA: An Online Tool for Accurate Disambiguation of Named Entities in Text and Tables. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 4(12):1450–1453, 2011.

[Yosef *et al.*, 2012] M. A. Yosef, S. Bauer, J. Hoffart, M. Spaniol, and G. Weikum. HYENA: Hierarchical Type Classification for Entity Names. In *International Conference on Computational Linguistics (COLINGS)*, pages 1361–1370, 2012.

[Yuan *et al.*, 2013] P. Yuan, P. Liu, B. Wu, H. Jin, W. Zhang, and L. Liu. TripleBit: A Fast and Compact System for Large Scale RDF Data. *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 6(7):517–528, 2013.

[Zhang *et al.*, 2013] Z. Zhang, A. Gentile, and F. Ciravegna. Recent Advances in Methods of Lexical Semantic Relatedness – A Survey. *Natural Language Engineering*, 19(04):411–479, 2013.

[Zhang *et al.*, 2015] T. Zhang, H. Li, H. Ji, and S. Chang. Cross-document Event Coreference Resolution based on Cross-media Features. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 201–206, 2015.

[Zhao *et al.*, 2016] G. Zhao, J. Wu, D. Wang, and T. Li. Entity Disambiguation to Wikipedia using Collective Ranking. *Information Processing and Management*, 2016:1–11, 2016.

[Zheng *et al.*, 2013] J. Zheng, L. Vilnis, S. Singh, J. D. Choi, and A. McCallum. Dynamic Knowledge-Base Alignment for Coreference Resolution. In *Conference on Computational Natural Language Learning (CoNLL)*, pages 153–162, 2013.

[Zhu *et al.*, 2009] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J. Wen. StatSnowball: A Statistical Approach to Extracting Entity Relationships. In *International Conference on World Wide Web (WWW)*, pages 101–110, 2009.

[Zuo *et al.*, 2014] Z. Zuo, G. Kasneci, T. Gruetze, and F. Naumann. BEL: Bagging for Entity Linking. In *International Conference on Computational Linguistics (COLINGS)*, pages 2075–2086, 2014.

List of Algorithms

List of Figures

List of Tables