République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A.MIRA-BEJAIA
Faculté des sciences exactes
Département d'informatique

THÈSE

Présentée par

Mlle Ramla BELALTA

Pour l'obtention du grade de

DOCTEUR EN SCIENCES

Filière : Informatique

Option : Cloud Computing

Thème

Developing and Evaluating a Large-Scale Entity Linking System

Soutenue le : **18/12/2024**

Devant le Jury composé de :

Nom et Prénom	Grade		
Mr Abderrahmane Sider	**MCA**	Univ. de Bejaia	Président
Mr Farid Meziane	**Professeur**	Univ. de Derby UK	Rapporteur
Mr Samir Akhrouf	**Professeur**	Univ. de M'sila	Examinateur
Mr Sadik Bessou	**MCA**	Univ. de Sétif1	Examinateur
Mr Zoubeyr Farah	**MCA**	Univ. de Bejaia	Examinateur

Année Universitaire : **2024-2025**

Dedication

This thesis is dedicated to the memory of my father, **Saad Belalta**, who passed away before I had the opportunity to complete this work.

My efforts were motivated by your continuous support and unshakable faith in my abilities Dad. You are no longer with me to witness the completion of this journey. Your spirit and wisdom have guided me the entire time. I am who I am now because of your love and support. My work is a tribute to your lasting legacy.

I sincerely miss your presence and I wish you were here to celebrate this accomplishment with me. In appreciation of everything you have done and still mean to me, this thesis is dedicated to you.

Acknowledgement

Above all, I would like to thank God from the bottom of my heart for giving me the courage, discernment, and determination to finish this work. None of these could have happened without his direction or favor.

My supervisor, **Pr Farid Meziane**, was an essential help and source of patience during my research and thesis writing. I would also like to express my sincere gratitude to him for his support. His advice and thoughts have greatly influenced this work and enabled me to overcome many obstacles. It is a true blessing to have had the opportunity to work and learn from such an inspirational mentor.

I would like to express my deepest gratitude to the members of my Ph.D. jury for their invaluable contributions to the evaluation and improvement of my work. **Mr Abderrahmane Sider**, Maître de Conférences Classe "A", for his insightful feedback, constructive critiques, and guidance that have greatly enhanced the quality of this thesis. **Mr Samir Akhrouf**, Professor, for his expertise, thoughtful questions, and valuable suggestions that helped refine the direction of my research. **Mr Sadik Bessou**, Maître de Conférences Classe "A", for his thorough review, detailed analysis, and supportive comments, which were instrumental in strengthening the rigor of this work. **Mr Zoubeyr Farah**, Maître de Conférences Classe "A", for his encouragement, meticulous evaluation, and valuable recommendations, which provided clarity and precision to my findings. Your collective expertise, dedication, and support have been an incredible source of inspiration throughout this journey. I am deeply honored and grateful for the time and effort you devoted to ensuring the success of this research.

My sincere gratitude is extended to my friend, **Mouhoub Belazzoug**, whose

unwavering support and encouragement have been a continual source of inspiration for me. He provided me with practical and emotional support when I needed it most, and I appreciate incredibly his friendship.

I owe my deepest gratitude to my mother, **Zohra Khababa**, the most important person in my life. Her unending love, tolerance, and sacrifices enabled me to pursue and succeed in my academic journey. Her trust in me motivated me throughout my life, and without her consistent encouragement and support, I could not have completed my thesis. All love goes to my brother **Akrima** for his constant assistance and support during this journey. I am extremely grateful to him.

Finally, I thank all my family, especially **Rachida Belalta**, and all my friends, particularly **Amel Houha**, for their continuous support and motivation. I have found strength and inspiration in your love, understanding, and belief. Thank you for your valuable contribution to this journey.

Thank you all for your invaluable contributions to this journey.

Abstract

Disambiguating name mentions in text is a crucial task in Natural Language Processing, especially in entity linking. The credibility and efficiency of such systems largely depend on this task. For a given name entity mention in the text, there are many potential candidate entities that may refer to this mention in the knowledge base. Therefore, it is very difficult to assign the correct candidate from the whole candidate entities set to this mention. To solve this problem, collective entity disambiguation is a prominent approach. In this thesis we present a new algorithm called CPSR for collective entity disambiguation which is based on the graph approach and semantic relatedness. A clique partitioning algorithm is used to find the best clique that contains a set of candidate entities that provide answers to the corresponding mentions in the disambiguation process. To evaluate our algorithm, we carried out a series of experiments on seven well-known datasets namely, AIDA/CoNLL2003-TestB, IITB ,MSNBC, AQUAINT, ACE2004, Cweb and Wiki. The Kensho Derived Wikimedia Dataset (KDWD) is used as the knowledge base for our system. From the experimental results our CPSR algorithm outperforms both the baselines and other well known state of the art approaches.

Keywords: Named Entity Disambiguation, Entity linking, Clique Partitioning, Semantic Relatedness, Graph Based Approaches.

Résumé

Désambiguïser les mentions de noms dans le texte est une tâche cruciale dans le traitement du langage naturel, en particulier dans la liaison d'entités. La crédibilité et l'efficacité de ces systèmes dépendent largement de cette tâche. Pour une mention d'entité donnée dans le texte, il existe de nombreuses entités candidates potentielles qui peuvent faire référence à cette mention dans la base de connaissances. Par conséquent, il est très difficile d'affecter le bon candidat à partir de l'ensemble des entités candidates définies pour cette mention. Pour résoudre ce problème, la désambiguïsation des entités collectives est une approche importante. Dans cette thèse, nous présentons un nouvel algorithme appelé CPSR pour la désambiguïsation d'entités collectives qui est basé sur l'approche graphique et la relation sémantique. Un algorithme de partitionnement de clique est utilisé pour trouver la meilleure clique qui contient un ensemble d'entités candidates. Ces entités candidates fournissent les réponses aux mentions correspondantes dans le processus de désambiguïsation. Pour évaluer notre algorithme, nous avons effectué une série d'expériences sur sept ensembles de données bien connus, à savoir AIDA/CoNLL2003-TestB, IITB ,MSNBC, AQUAINT, ACE2004, Cweb et Wiki. Le Kensho Derived Wikimedia Dataset (KDWD) est utilisé comme base de connaissances pour notre système. À partir des résultats expérimentaux, notre algorithme CPSR surpasse à la fois les lignes de base et d'autres approches de pointe bien connues.

Mots clé : Disambiguation d'entité nommée, liaison d'entité, partitionnement de clique, relation sémantique, approches basées sur des graphes.

ملخص

يعد توضيح ذكر الأسماء في النص مهمة حاسمة في معالجة اللغات الطبيعية، خاصة في ربط الكيانات . وتعتمد مصداقية وكفاءة هذه الأنظمة إلى حد كبير على هذه المهمة .بالنسبة لكيان اسم معين مذكور في النص، هناك العديد من الكيانات المرشحة المحتملة التي قد تشير إلى هذا الذكر في قاعدة المعرفة .ولذلك فإنه من الصعب جداً تعيين المرشح الصحيح من بين كافة الكيانات المرشحة المحددة لهذا الذكر .ولحل هذه المشكلة، يعد توضيح الكيان الجماعي نهجًا بارزًا .نقدم في هذه الدراسة خوارزمية جديدة تسمى CPSR لتوضيح الكيان الجماعي والتي تعتمد على نهج الرسم البياني والارتباط الدلالي .يتم استخدام خوارزمية تقسيم المجموعة للعثور على أفضل مجموعة تحتوي على مجموعة من الكيانات المرشحة .توفر هذه الكيانات المرشحة الإجابات على الإشارات المقابلة في عملية توضيح الغموض .لتقييم الخوارزمية الخاصة بنا، أجرينا سلسلة من التجارب على سبع مجموعات بيانات معروفة وهي-AIDA/CoNLL2003 TestBو IITBو MSNBCو AQUAINTو ACE2004و Cwebو .Wikiيتم استخدام مجموعة بيانات ويكيميديا المشتقة من Kensho (KDWD) كقاعدة معرفية لنظامنا .من النتائج التجريبية، تتفوق خوارزمية CPSR الخاصة بنا على كل من خطوط الأساس وغيرها من الأساليب الحديثة المعروفة .

الكلمات المفتاحية: توضيح الكيان المسمى، ربط الكيان، تقسيم المجموعة، الارتباط الدلالي، المقاربات القائمة على الرسم البياني.

Preface

This thesis is the product of our exploration of the complex fields of entity linking and entity disambiguation in natural language processing. Our investigation into creating more effective and precise techniques for comprehending text data was motivated by the ever-increasing amount of digital text and the intricacy of human language, prompted further study.

Our fascination with how computers may recognize nuances, comprehend context, and making sense of unclear textual information served as the driving force for our project. Accurately identifying and disambiguating entities inside large text corpora is becoming more and more important as our reliance on digital information grows. Due to this difficulty, we have developed a graph-based method that uses the linkages and interconnectedness found in data to improve efficiency and accuracy when disambiguating mentions in the text.

We hope that the methods and results we have shared in this thesis will spur more research in this fascinating and rapidly developing topic, as well as further the current progress in entity linking and disambiguation.

Contents

List of Tables

List of Figures

List of Acronyms

CNED	Collective Named Entity Disambiguation
CPSR	Clique Partitioning based on Semantic Relatedness
ED	Entity Disambiguation
EL	Entity Linking
IE	Information Extraction
IR	Information Retrieval
KB	Knowledge Base
KDWD	Kensho-derived Wikimedia Dataset
L2R	Learning To Rank
NED	Named Entity Disambiguation
NER	Named Entity Recognition
NE	Named Entity
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
SNED	Single Named Entity Disambiguation
SVM	Support Vector Machine
TAC/KBP	Text Analysis Conference/ Knowledge Base Population
VSM	Vector Space Model
WSD	Word Sens Disambiguation

Chapter 1

Introduction and motivation

1.1 Introduction

Currently, we are experiencing exponential growth in data on the Internet. The increase in this amount of data is driven not only by news, business, or blogging but is also provided by people. Managing a billion web pages seems difficult, if not impossible, for people and businesses. To solve this problem, an obvious solution is to offer automatic tasks like web search, classification, entity linking, etc. Computers are only used to present the content of web pages, while web search engines are used to find a specific set of web pages that match a search query. Unfortunately, the quality of these two services does not fulfill human needs because computers and most search engines operate without regard to the semantic meaning of the contents, [3] thus leading humans to interpret the results themselves.

The concept of the Semantic Web was introduced in 2001 by Berners et al [4] where they extended the current Web to the Semantic Web and gave it a defined meaning that allows humans and computers to work cooperatively to understand textual natural language. Making sense of information is a big task and can be decomposed into many sub-tasks, including co-reference resolution, word sense disambiguation (WSD) [5], named entity recognition (NER) [6], classification [7], and Entity Linking (EL) [8], also known as Named Entity Disambiguation (NED).

Named entities (NE), also called mentions, are one of the principal elements of text on the web. These NE are found in any document, especially news stories, and are semantically richer than most vocabulary words [9] . A textual named entity (mention) refers to a real-world entity, namely a person, location, or organization. The challenge is that one mention in the text may refer to more than one named entity in the knowledge base (KB); this kind of problem is called name ambiguity (polysemy). Another type of problem is name variation (synonymy), where one named entity in the real world may be referred to by more than one mention in the text.

A named entity can appear in a single word, such as "France" or "DELL", or in a collection of words, such as "University of San Francisco" or "United States". Furthermore, the named entity can appear in a dictionary as a word, which means it can be found in the language dictionary; for example "Mark" is both a personal name and an English verb, whereas most named entities, like the name "Cameron Diaz" or the place "France", do not appear in a language dictionary. There is no established dictionary for real named entities; new named entities emerge constantly, some of which are included in KBs. New textual mentions referring to existing named entities are added daily to the content of the website. The only available reference resources for real-world entities are KBs. There are many existing KBs, such as Wikipedia[1], DBpedia[2], Yago[3], and Yahoo[4], among others. Wikipedia is considerably used to disambiguate named entity mentions because of its magnitude and free availability [10–13]. It contains references to relatively well-known real-world entities, and it is useful for researchers to solve entity ambiguity problems.

Named entities are very important in several areas, especially in information mining [14], text mining [15], and knowledge base population [16]. Many researchers have invested in named entity recognition and classification [6]. Besides, many of the recognized named entities from web text content are ambiguous, so it is difficult for software to identify from the KB the right named entities that correspond to these different textual mentions. For instance, the mention "Paris" may refer to the capital of France and must be distinguished from the mention "Paris", which refers to Paris Hilton an American model. Also, an entity

[1]https://www.wikipedia.org/
[2]https://www.dbpedia.org/
[3]https://yago-knowledge.org/
[4]https://uk.yahoo.com/

may be known by several names; for example, "king of pop" and "MJ" refer to the same entity, "Michael Jackson", but they have different surface forms (morphology). These examples demonstrate how difficult the task of linking the NE textual mentions to their real-world entities in the KB is.

Named entity disambiguation is the process of disambiguating and linking named entity mentions that exist in text to their corresponding entities in the KB. By annotating a significant volume of typically noisy data, entity linking contributes to the vision of the Semantic Web. EL, is regarded as one of the two sub-tasks of the Text Analysis Conference-Knowledge Base Population (TAC-KBP) [5]. NED approaches deal only with mentions that have their corresponding entities in a KB, while EL approaches process all mentions, including those that do not have corresponding entities in a KB. EL approaches attempt to predict the unlinkable mentions by leveraging supervised machine learning techniques such as binary classification [17], learning to rank [18] and probabilistic models [19].

From the literature, there are two main classifications of NED [8]. The first classification is based on whether the methods are supervised or unsupervised; the supervised ranking methods rely on annotated training data to "learn" how to rank the candidate entities. These approaches may include binary classification methods [20–26], learning to rank methods [13, 22, 25, 27–36], probabilistic methods [28, 37–39], and graph-based approaches [11, 40, 41].The unsupervised ranking methods are based on unlabeled corpus and do not require any manually annotated corpus to train the model. These approaches may include Vector Space Model (VSM) based methods [10, 42, 43] and information retrieval (IR) based methods [44, 45]. The second classification categorizes the NED approaches into two categories: single named entity disambiguation (SNED) and collective named entity disambiguation (CNED). The first approach, [10, 12, 27, 46–56] aims to disambiguate the mentions in the text independently from each other, while the second one aims to disambiguate all the mentions in the text jointly and mutually. [13, 28, 31, 57–66]

[5] https://www.aclweb.org/portal/content/text-analysis-conference-knowledge-base-population

1.2 Motivation

With the increasing amount of textual information that is available throughout the Web, Information Extraction (IE) techniques [67] have grown in interest. These techniques are being commonly used to process unstructured information from the Web, generally with the objective of building structured KBs from the available information. IE is an area of research that involves Natural Language Processing (NLP) [68] to unstructured or semi-structured documents in order to generate structured information. This information can be used in numerous applications, with examples being the population of KBs or the offering of support to more advanced natural language processing and information retrieval applications, such as question answering [69].

Hence, to retrieve information about specific entities, we first need to identify these entities by assigning their references in a text to a resource usually a KB providing unique identifiers of these entities.

1.2.1 Challenges

The main challenges of entity linking are:

- **Name Ambiguity (Polysemy and Homonymy):** The polysemy problem occurs when a Named Entity (mention) in the text may refer to several entities in the KB; for example, the mention ”jaguar” may refer to ”jaguar” the car or a ”jaguar” the animal. The Homonymy problem is when different entities may share the same name; for example, ”Bank” may refer to several things (the financial institution or the river Bank).

- **Name Variations(Synonymy):** Many different mentions in the text may refer to a single entity in the KB, those mentions represent the synonyms of this entity like aliases, nicknames, abbreviations and acronyms example: all the mentions *”king of pop”*, *”MJ”*,*”Wacko Jacko”* and the *”The Gloved One”* refer to one and only one entity in the KB witch is **Michael Jackson**.

- **Absence of entries(Uncovered entities):** Is seen as some mentions in the text may not have their corresponding entities in the KB.

  While Named Entity Disambiguation is referred to as Named Entity Linking. Entity Linking is a larger task than Named Entity Disambiguation.

NED deals mainly with the name ambiguity problem which means, it tries to find the right named entity in the KB for a given mention in the text and assumes that the KB is complete. EL assumes that the KB is incomplete and deals also with the absence of entries problem in the KB which means the EL system has to predict a link for the mentions who don't have their corresponding entities in the KB.

1.2.2 Application

Entity Linking and disambiguation plays a vital role in many different tasks such as:

- **Information retrieval:** Recently, the semantic entity-based search has taken the place of the conventional keyword-based search. Semantic Entity-Based Search [70–73] takes advantage of entity linking since web text and web documents contain many mentions of entities that need to be disambiguated. Named entities commonly occur in search queries and they are most of the time ambiguous. Disambiguate these ambiguous entity mentions by linking them to a KB by leveraging the query context could potentially enhance the quality of search results [74].

- **Information extraction:** Information extraction systems [75] extract named entities and relations from the text, these named entities are usually ambiguous. A good way to disambiguate the ambiguous named entities is to link them to a KB to facilitate their future exploitation [76, 77].

- **Question answering:** A supported KB is used by question answering systems to response to the user's questions. A promising results were obtained by some question answering systems like Watson [78], these systems harness the entity linking technique to predict the kinds of questions and a potential answers. Thus the entity linking shown to be very beneficial to question answering systems [69].

- **Knowledge base population:** Entity linking is essentially considered as an important sub-task for knowledge base population [79, 80] since new facts are daily generated and digitally expressed on the Web. These new facts need to be linked and added to an existing KB. Therefore, the knowledge base population task could eventually benefit from the entity linking

systems [36, 81–84].

1.3 Entity linking or Named entity disambiguation

1.3.1 Problem definition

Entity Linking is the task of mapping each named entity mentioned in a textual document to the corresponding KB entry (such as Wikipedia, yago), or determining if one such entry does not exist in the KB [79]. The named mentions must be identified previously by a Named entity recognition (NER) system [85–87]. NER is the process of locating a word or a phrase that references to a particular entity within a text. It involves recognition of entity names (people and organizations), place names, temporal expressions and numerical expressions [85]. There are several NER tools that are publicly available, including Stanford NER[6], OpenNLP[7] and LingPipe[8].

Given an input document D containing a set of pre-tagged (using NER system) NE textual mentions M ={ m_1, m_2, m_3 . . . m_k }. Initially, the EL system selects all possible candidate interpretations for each m_i from the KB. I.e. for each NE textual mention $m_i \in$ M the system selects a set of candidates entities $E_i = \{e_{i,1}, e_{i,2}, e_{i,3}...e_{i,j}\}$ from the KB. The NE textual mention m_i is used to search the KB entries titles to find entries with titles that fully or partially contain the NE textual mention. As a subsequent step, the EL system ranks these candidate entities by leveraging a given technique to disambiguate each mention in the text, which means the system will choose for each mention in the text its corresponding entity in the KB. Thereafter, the system assigns each mention to its corresponding entity in the KB by creating a hyperlink between them. An illustrative example is shown in Figure 1.1.

[6] http://nlp.stanford.edu/ner/
[7] http://opennlp.apache.org/
[8] http://alias-i.com/lingpipe/

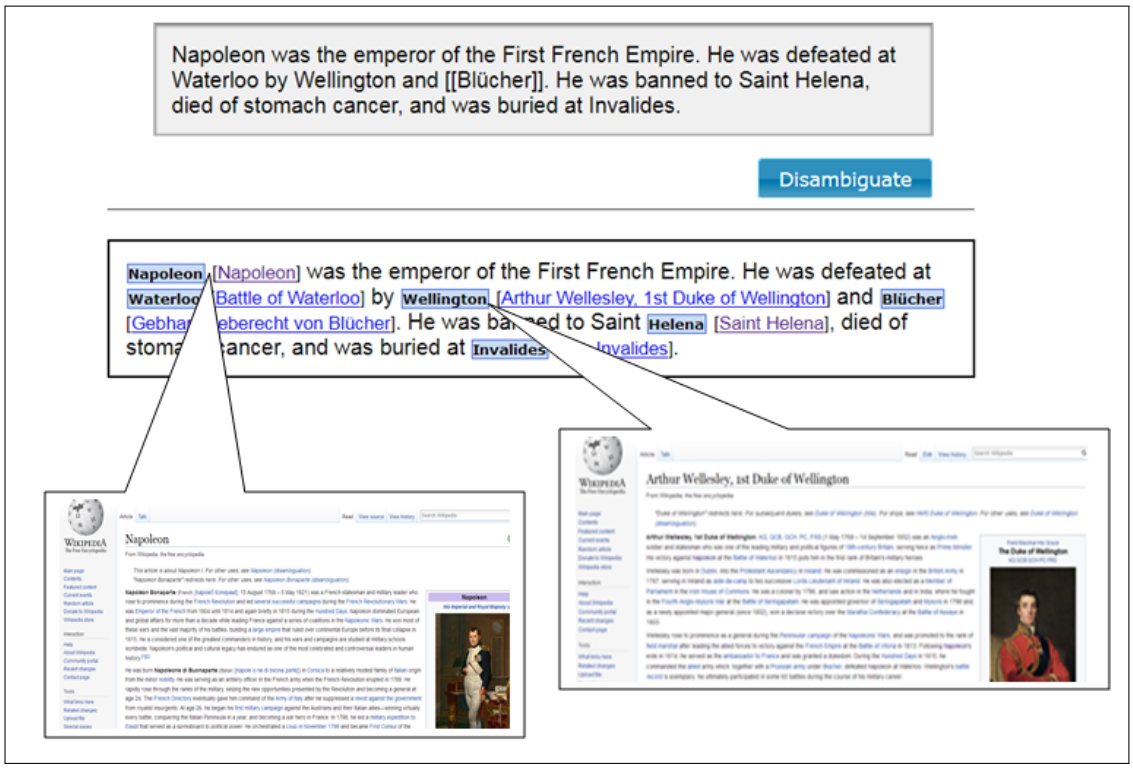

Figure 1.1: *Example of entity disambiguation in text*

1.4 Aim and objectives of the research

In this section, we outline the overarching goal and particular targets that direct our study project. A precise statement of the goals and objectives clarifies the purpose and anticipated results of the study in addition to serving as a roadmap. We create a framework for the methodical investigation of our research topics by identifying these characteristics, which guarantees coherence and focus throughout the investigation. Our entity linking system aims to enhance the accuracy of identifying and associating entities such people, places, or organizations with their corresponding entries in a KB and the main objectives of this research are as follows:

- Leverage the global coherence to disambiguate all the mentions in the text collectively.
- Demonstrate that the context-dependent features are beneficial for the named entity disambiguation.

- Increase the accuracy of the NED systems.
- Propose an unsupervised solution avoiding the training data which is hard labor and time consuming.
- Show that the graph-based methods are more suitable for this kind of research.

1.5 Research contributions

Our research contributions are therefore as follows:

- We propose a new unsupervised algorithm called Clique Partitioning Based on Semantic Relatedness (CPSR). Our algorithm does not require training data or a training phase, but it relies on the KDWD KB that we used to generate the name dictionary. Moreover, it disambiguates iteratively and collectively the mentions inside the same document.
- We also provide a single disambiguation algorithm that disambiguates texts with a single mention as well as the mentions that remain from the collective disambiguation.
- We leverage the semantic relatedness using the **Jaccard measure** to capture the coherence between the mapping entities, therefore using the context-dependent features rather than the context-independent features to create our system.
- We generate a test dataset from the KB we used to build our system, KDWD, to assess our system's performance. We also tested the effectiveness of our system using seven well-known datasets and compared the outcomes against more than 30 other state-of-the-art entity disambiguation systems.
- This study was featured in an article named *«A graph based named entity disambiguation using clique partitioning and semantic relatedness »* published in a top-tier journal called *«Data & Knowledge Engineering »*.[9]

[9]https://doi.org/10.1016/j.datak.2024.102308

1.6 Research questions

In order to provide more accurate natural language comprehension and information retrieval, our system attempts to overcome these following questions and challenges by precisely disambiguating named entities inside the text. The identification of the named entities (NE) or what we call mentions is performed by the NER.

- How does our system handle situations where named entities like "Apple" could refer to multiple things, such as a company or a fruit, by considering the context of the text?
- What criteria should our system consider when distinguishing between different named entities and how can the use of context-dependent attributes and insights from the knowledge base (KB) assist in identifying the most probable reference?
- How can our system assesses the semantic relatedness between named entities and their potential referents to find the perfect match?
- How does our system effectively connect an entity mention to a particular entry in a KB or reference database once it has been disambiguated enhancing overall understanding of the text?
- Which evaluation metrics like precision, recall, F1 score or accuracy would be best suited to assess how well our named entity disambiguation system performs?

1.7 Structure of the thesis

The remaining of the thesis is structured as follows:

Chapter 2 gives a thorough introduction to entity linking, providing background information and laying the groundwork for the following chapters.

A comprehensive overview of the body of literature is presented in chapter 3, where we delve into pertinent studies and research on the topic. This chapter highlights pioneering works in this field and categorize them.

Chapter 4, Research Methodology, describes the specific strategy used in this work also includes a full description of the research process. This provides an explanation of the methodology used to study the NED problem, as well as the research design and data gathering strategies.

We highlight the suggested procedures in chapter 5, Contributions and Methods, in order to deal with the NED issue. This chapter presents our new approach and explains the steps we took in practice to create and improve our method, as well as its theoretical foundations.

The implementation and evaluation is presented in chapter 6, which shows the findings of our suggested study. This chapter offers a thorough explanation an discussion of the obtained results by assessing these findings and offering a thorough breakdown of the efficiency and efficacy of our methodology.

Chapter 7 wraps up the thesis by providing a summary of the research done in the field of NED. This chapter also discusses prospective future viewpoints, offering ideas for future study directions and possible enhancements to the methodologies and approaches discussed in this thesis.

Chapter 2

Background on entity linking

2.1 Introduction

NED tackles the problem of precisely connecting named entities stated in text to their corresponding entries in a KB. NED occupies the space between natural language processing and information retrieval. This chapter explores the background ideas and methods that form the basis of NED, giving readers a basic grasp of the area.

Entity linking, which entails locating and connecting textual occurrences of entities to their canonical representations in a KB, is a key component of EL systems. The foundation of EL or NED systems are entity linking modules, which use a variety of methods to resolve ambiguities and distinguish between entities in text. These modules usually leverage a variety of techniques, including name dictionary based techniques [10, 27, 88, 89], probabilistic models [28, 37–39], graph-based algorithms [11, 40, 41], and machine learning-based techniques like Learning to Rank (L2R) [22, 27–30].

This chapter explains the KB concept, examines entity linking modules, explains the semantic relatedness concept, presents a variety of entity linking techniques that might me used to disambiguated the entities as well as the evaluation criteria that are used to assess the effectiveness of NED systems. These evaluation criteria cover a wide range of measures, including precision, recall, accuracy, and F1 score, which express how well the system can recognize and link entities in text. Comprehending these assessment indicators is crucial for assessing the efficacy and dependability of NED techniques as well as for comparing results to the most advanced methods now in use.

This chapter establishes the foundation for the discussions that follow on advanced NED approaches and methodology by giving an overview of entity linking modules, entity linking techniques, and evaluation criteria. Researchers can create more reliable and accurate NED systems that can handle the complexity of real-world textual data by having a thorough understanding of these fundamental concepts.

2.2 Knowledge base

An indispensable tool for the entity linking process is a knowledge base [90]. It provides extensive details about many global entities (like Marie Curie and Warsaw), including their semantic classifications (Marie Curie is classified as a Scientist, while Warsaw is classified as a City), and the connections between these entities (like the relationship "bornIn" that links Marie Curie and Warsaw).

Knowledge bases can be broadly categorized into two types: domain-specific and global. The purpose of a domain-specific KB is to store concepts, instances, and relationships in a tightly defined field of study. These industry-specific or field-specific specialized KBs offer targeted, in-depth information pertinent to that particular sector. A few notable examples are *echonest* [10], which focuses on music data, *DBLP* [11], which covers publications in computer science, *Google Scholar*[12], which indexes academic articles across disciplines, *DBLife*[13], which keeps track of academic events and activities, and product KBs created by e-commerce businesses to manage product information.

On the other hand, a global KB strives for comprehensiveness in all disciplines by encompassing a wide range of information about the entire world. These KBs combine information from several sources to produce a more comprehensive KB. Notable instances include the vast collaborative structured data base Freebase [91], Google's Knowledge Graph [92], which enriches search results with data from multiple sources; YAGO [93], a semantic KB sourced from Wikipedia and other sources; DBpedia [94], which pulls structured data from Wikipedia; and the assortment of Wikipedia infoboxes [95], which offer succinct factual information about a wide range of subjects. Though global KBs offer breadth over a wide

[10]https://music.us/supporters/echo-nest/
[11]https://dblp.org/
[12]https://scholar.google.com/
[13]https://www.dblife.club/

range of subjects, domain-specific KBs offer depth in specialized areas. Each form of KB serves a distinct purpose.

We will give an overview of six well-known KBs that are frequently used in the entity linking field in the section that follows. These databases provide essential data on entities and their classifications, but they also make the complex web of relationships between them clear, enabling accurate and efficient entity linking.

- **Wikipedia**[14] With millions of entries covering a wide range of subjects, Wikipedia is the largest multilingual online encyclopedia, created by volunteers all over the world. Wikipedia is currently the world's biggest and most well-known online encyclopedia. It is also a resource that is expanding rapidly and is always changing. An article is the fundamental entry on Wikipedia; it defines and discusses an entity or subject and is uniquely referred to by an identifier. Currently, English Wikipedia contains over 6 million articles. In addition, Wikipedia's structure offers a number of helpful features for entity linking, including article categories, entity pages, redirect pages, disambiguation pages, and hyperlinks within Wikipedia articles.

- **Yago** [93] YAGO is a KB that merges Wikipedia's extensive entities with WordNet's organized categories. This provides YAGO with both broad coverage of entities and a well-structured hierarchy. The newest version offers over 10 million entities (people, places, etc.) with 120 million facts defining them, including hierarchical (like kinds) and non-hierarchical (like livesIn) interactions. Notably, YAGO uses a "means" connection to link textual mentions (e.g., "Einstein") to their related entities (Albert Einstein), which is beneficial for tasks like discovering potential entities from text.

- **DBpedia** [94] Wikipedia is the source of structured data extracted by DBpedia, a multilingual KB. This information consists of categories, locations, external links, and infobox summaries. More than 4 million entities are present in the English edition, of which 3 million are regularly categorized. DBpedia changes in tandem with Wikipedia.

[14] http://www.wikipedia.org/

- **Freebase** [91] is a vast online KB that its members have collectively created. By using an accessible interface to alter structured data, non-coders can still make a contribution. Freebase pulls data from a number of sources, including Wikipedia. It currently has 2.4 billion facts describing about 44 million entities (people, places, etc.).

- **Google's Knowledge Graph** [92] Google's Knowledge Graph is a massive web of information about real-world entities (people, places, events) and their connections. Unlike keyword-based search, it understands relationships between these entities. Google builds it by gathering information from various sources, identifying entities, and then using machine learning to analyze connections and constantly improve the network. This rich KB allows Google to understand your searches better and provide informative results with summaries directly on the search page.

- **KDWD** [15] Stands for "Kensho Derived Wikimedia Dataset". The three primary parts of KDWD are the unformatted English Wikipedia articles, annotations indicating which text spans link, and a small sample of the Wikidata KB. It was constructed with the Wikidata snapshot and the English Wikipedia snapshot.

2.3 Entity linking system modules

Typically, the main components of an entity linking system, consists of three major modules, namely (I) candidate generation, (II) candidate ranking, and (III) unlinkable mention prediction. This architecture is presented in Figure 2.1. The following subsections detail each of the modules from the general architecture.

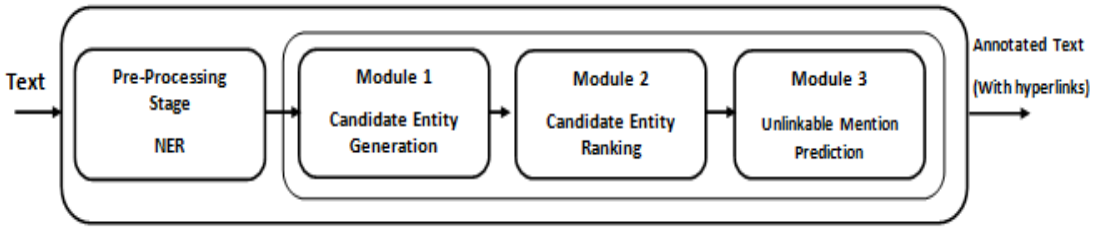

Figure 2.1: *The main modules of an EL system.*

[15] https://datasets.kensho.com/datasets/wikimedia

As illustrated in Figure 2.1 a Pre-Processing stage is required, this step is performed by a NER (Named Entity Recognition System). The NER takes as an input a text and generates a set of mentions (Named Entities). These mentions represent an input for the candidate entity generation module, whereas the output of the NEL represents the link created between the mentions in the text and their corresponding entity in the KB (annotated the text by creating a hyperlink for each mention) .

2.3.1 Candidate Generation module

In this first module, the main goal is to find all possible KB entries that might correspond to each mention (named entity) in the query (text or document). This can be achieved by using a variety of similarity measures (e.g., character-based string similarity metrics [96]), where the module compare the surface form of the named entity from the query with all the KB entries. The top-entries that are most likely to correspond to the entity are returned as candidates. The main objective of this step is to collect a limited set of likely candidate entries that will latter be analyzed in detail during the ranking step. The main approaches that have been applied for generating the candidate entity set E_m for an entity mention m are:

1. **Name Dictionary Based Techniques:** Entity linking systems depend greatly on name dictionary-based methods for the creation of candidate entities [10,11,20,21,27–29,31,82–84,88,89,97–99]. They utilize the Wikipedia structure. Wikipedia is a rich source of many features. These include entity pages redirect pages, disambiguation pages bold phrases and hyperlinks. These features work together to create an offline name dictionary. We will call it dictionary (D). This table maps various names to their related entities. The dictionary holds a wealth of information on named entities. Variations, abbreviations spellings and nicknames are only a few examples. The dictionary's structure is essential to note, it contains key-value mappings. Keys stand for names in this dictionary. The values are sets of entities. These entities are associated with said names mentions. The Name Dictionary Based Techniques are leveraged by many entity linking systems [10, 11, 28, 29, 31]. Table 2.1 represents a portion of a name dictionary and its structure.

Table 2.1 A snapshot of a name dictionary.

Key (Name mentions)	Value (Mapping entities)
Bill Gates	*Bill_Gates*
Michael Jackson	*Michael_Jackson*
King of pop	
MJ	
The gloved one	
...	
Jaguar	*Jaguar*
	Jaguar Cars
	Jaguar (band)
	Jaguar (1979 film)
	...
Usama ibn Mohammed ibn Awad ibn Ladin	*Osama_bin_Laden*
Bin Laden	
John Kennedy	*John_Fitzgerald_Kennedy*

2. **Methods Based on Search Engines:** Certain entity linking systems make use of web search engines to generate candidate entities for each mention in the text. Google for example, is often employed to locate potential entities from the vast expanse of the world wide web where they select the top returned Wikipedia pages results as potential candidate entities for a given mention [82, 100, 101].

2.3.2 Candidate Ranking module

The Candidate Generation module produces a set of candidates. These candidates undergo examination based on a predefined selection of ranking features. The candidates then undergo ranking. This process ensures that the KB entry with the highest likelihood of correct disambiguation appears at the top.

Methods for ranking candidate entities fall under two broad categories. These categories are supervised and unsupervised approaches. Supervised methods employ annotated training data. This data helps to discern ranking strategies. The commonly used methods in this category are: binary classification methods [20, 21, 23, 25, 26, 98, 100], learning to rank (L2R) methods [13, 22, 25, 27–29,

31–33, 35, 36, 99], probabilistic methods [28, 37, 38, 102] and graph-based methods [11, 40, 41]. The Unsupervised methods on the other hand, depend on unlabeled corpora.The methods frequently employed in this category include: Vector Space Model (VSM) based methods [10, 83, 84] and information retrieval based methods [20, 44, 97].

Further categorization comprises: Independent ranking methods [27, 29, 35, 36, 83, 101, 103]. These methods view entity mentions in a document as independent. They then rank candidates according to context similarity. Collective ranking methods [10, 11, 13, 28, 31, 33, 40, 88, 104–107]. These methods presume a coherence in document's topics. Thus they link entity mentions in accordance.

A variety of features are useful in ranking candidate entities. These can be classified as either context-independent or context-dependent. Context-independent features primarily focus on the entity mention's surface form. They also exhibit knowledge concerning the candidate entity. They have an important function. This function resides within a sphere that is separate from the mention's contextual surroundings.

Conversely, context-dependent features function on a different principle. They hinge on the context where the mention emerges. These entail the mention's immediate textual context, but not limited to it. They also account for other mentions found within the same document.

1. **Context-Independent Features**

   (a) ***Name string comparison:*** plays a vital role in ranking candidate entities. Many string similarity measures have been used in the name comparison like edit distance [29, 108], Dice coefficient score [23, 24], character Dice, skip bigram Dice, and left and right Hamming distance scores [32]. Typical comparisons include exact matches, prefix or suffix matches, containment relationships, ordered letter matches or word matches.

   (b) ***Entity Popularity:*** The aspect of entity popularity plays a pivotal role in entity linking. It operates independently of context. Effectively, it is a measure of how likely a potential entity might appear alongside a specific entity mention. Here is a key point: each candidate entity tied to the same mention surface form does not hold

consistent popularity. Some might be more obscure, while others rare. For example, "New York." In this case, "New York City" is a more common entity than "New York (film)". Several state-of-the-art entity linking systems capitalize on Wikipedia count information [13, 28, 31, 33, 40, 88, 100, 104, 108]. This information quantifies entity popularity. Consequently, a popularity feature is defined for each candidate entity, made in relation to the entity mention.

A feature, known as $Pop(e_i)$, is then calculated. It is computed as the portion of links that have the mention form pointing to the specific candidate entity.

$$Pop(e_i) = \frac{Count_m(e_i)}{\sum_{e\,j \in Em} Count_m(e_j)} \tag{2.1}$$

Where $Count_m(e_i)$ is the number of links which point to the entity e_i and have the mention form m as the anchor text.

(c) ***Entity Type:*** The goal of this function is to evaluate the coherence between an entity's type mentioned in a text (for instance people, location organization) and the candidate entity's type in a knowledge database. Many studies use NER systems to identify the entity type for the entity mention in text and some candidate entity whose type is unavailable in the knowledge base [82, 100, 109].

2. **Context-Dependent Features**

(a) ***Textual Context:*** The evaluation of the textual context is done by measuring its similarity. This is primarily between the context that surrounds an entity mention and the document associated to the candidate entity, which means where this mention appears. There are different representations of context used for this objective.

i. **The Bag of Words Method:** The context is often depicted by gathering words (called bag of words) from the entire input document where this mention appears [22, 35, 83, 88, 105, 108]. Alternatively a relevant window around the mention may be employed [11, 13, 27, 28, 76, 104]. For each candidate entity, the context is usually represented as a bag of words from: the entire

Wikipedia entity page [11, 22, 27, 28, 35, 103, 108], the first description paragraph of its Wikipedia page [28], a suitable window around each occurence of that entity in the Wikipedia page corpus [104], or the top-k token TF-IDF summary of the Wikipedia page [13, 88].

ii. **Concept Vector:** Concept vectors are created by systems to capture the semantic heart of a text. This process includes the extraction of different elements from the document or its associated Wikipedia article. These elements encompass keyphrases [40], anchor texts [28], named entities [21, 25, 32], categories [10, 32], descriptive tags [89] and Wikipedia concepts [24, 31, 84, 110]. Moreover, the system constructs the context of a potential entity. This happens through using linked entities, attributes and pertinent facts taken from Wikipedia infoboxes [32, 111, 112]. Ultimately, this comprehensive depiction aids in deciphering the content and context of the document or entity. Therefore, through this detailed understanding, the system's overall comprehension is improved.

(b) ***Coherence Between Mapping Entities:*** Modern entity linking systems typically function under the belief that a document primarily focuses on harmonious entities pertaining to one or few subjects. They exploit this thematic consistency for establishing connections among entity mentions within a same document. The relationship between these mapped entities is leveraged during the linking operation. To calculate the topical coherence between Wikipedia entities under the assumption that two Wikipedia entities are considered to be semantically related if there are many Wikipedia articles that link to both. There is a set of measures that enables to calculate the coherence between mapping entities: Wikipedia Link-based Measure (WLM) [113] which is modeled from the Normalized Google Distance [114]. Given two Wikipedia entities u_1 and u_2, the topical coherence between them is defined as follows:

$$Coh\_G(u_1, u_2) = 1 - \frac{log(max(|U_1|, |U_2|)) - log(|U_1 \cap U_2|)}{log(|WP|) - log(min(|U_1|, |U_2|)} \quad (2.2)$$

Where U_1 and U_2 are the sets of Wikipedia articles that link to u_1 and u_2 respectively, and WP is the set of all articles in Wikipedia.
Another measure is Point-wise Mutual Information (PMI-like) measure [115] used to calculate the topical coherence between Wikipedia entities:

$$Coh\_p(u_1, u_2) = \frac{|U_1 \cap U_2|/|WP|}{|U_1|/|WP|.|U_2|/|WP|} \tag{2.3}$$

The Jaccard distance measure [116] is also used to calculate the topical coherence between Wikipedia entities:

$$Coh\_j(u_1, u_2) = \frac{|\ U_1 \cap U_2\ |}{|\ U_1 \cup U_2\ |} \tag{2.4}$$

The above three measures are based on the link structure of Wikipedia.

2.3.3 Unlinkable Mention Prediction

We previously covered methods for ranking entities in entity linking. They choose the top-ranked entity from a candidate set. This entity then serves as the mapped entity for a mentioned one. However, there is a challenge in practical applications. Some mentioned entities may lack corresponding entities in our base of knowledge. This leads to the problem of predicting unlinkable mentions. The next section offers a comprehensive summary. It covers the main strategies utilized to tackle this issue.

A number of studies in the field of NED simplify their analyses. They do this by assuming that all entity mentions can be connected to entities within a KB. This approach often ignores the challenge of managing mentions that cannot be linked [10, 11, 28, 33, 37, 39, 107]. However some other studies do account for this. They employ basic heuristic strategies to negotiate this obstacle. If a mention cannot be associated with any entities from the candidate set, a problem arises. This set is produced by the Candidate Entity Generation module. In response to this problem some strategies will designate the mention as unlinkable. Then they return a "NIL" value for it [20, 43, 109].

Apart from the techniques specified earlier, numerous entity linking systems employ a NIL threshold approach [26,27,31,44,82,84,104,106,117]. This approach is used in predicting unlinkable entity mentions. In such systems, a score is

ascribed to the highest-ranked entity. The system measures this score against a set threshold value. This is usually obtained automatically from the training data. If the score falls below this threshold, the system anticipates that the mention is unlinkable. Subsequently, it returns a NIL value. On the other hand, if the score does not fall below the threshold, the system takes a different action. It attributes the highest-ranked entity as the correct mapping for the mention.

Various entity linking systems employ supervised machine learning [13, 29, 35, 82, 98–100]. The aim is to predict unlinkable entity mentions via binary classification, where the method involves determining if the top-ranked candidate entity is the appropriate mapping for a mention. Support Vector Machine (SVM) classifiers are frequently used in these systems. They integrate unlinkable mention prediction into entity ranking. This is done by adding a NIL entity. If NIL ranks highest, the mention is identified as unlinkable, and other researchers integrate unlinkable mention prediction. The integration is done into a probabilistic model. This model introduces a NIL entity to the KB [32, 36, 38].

2.4 Semantic relatedness

In NED, the word *"semantic relatedness"* describes the method of determining how closely two concepts or entities are related in meaning within a given context. This idea is crucial to NED, as the objective is to determine the proper entity among several candidates by evaluating each one's significance to the surrounding text.

Semantic relatedness is measured using variety of knowledge-based techniques. This includes utilizing Wikipedia's link structure, taxonomic paths in ontologies like WordNet [118] and connections in structured KBs such as DBpedia [94] are also used. Co-occurrence analysis is another technique. Additionally Pointwise Mutual Information (PMI) [115], Wikipedia Link-based Measure (WLM) [113] and Jaccard disatance measure [116] are used to evaluate the frequency with which entities occur together in huge corpora. Embedding-based methods measure the cosine similarity [119] of the embeddings of entities to capture semantic similarities between them. They use vector space models such as Word2Vec [120], GloVe [121] or BERT [122]. By guaranteeing that the chosen entity is appropriate for the given context. These steps taken together improve the accuracy of NED.

2.5 Entity linking methods

The main entity linking methods harnessed in this field are:

1. **Binary Classification Methods:** binary classification techniques teach classifiers if a candidate entity corresponds to the context of a named entity mention. We treat every candidate-context pair as a binary classification issue, utilizing factors such as contextual words and entity type information to forecast "match" or "no match." Neural networks, support vector machines, and logistic regression are a few of the methods used. These techniques enhance the accuracy and consistency of entity disambiguation by methodically assessing every candidate [20, 21, 23, 25, 26, 98, 100].

2. **Learning to Rank (L2R) Methods:** Learning to rank methods use supervised learning approaches. They assign number to candidate entities. This is based on features extracted from the context of textual mentions. With the help of labeled training data these methods aim to maximize entity ranking. This process makes it possible to effectively disambiguate named entities within the text [13, 22, 25, 27–29, 31–33, 35, 36, 99].

3. **Probabilistic Methods:** Probabilistic models use statistical inference to calculate the probability that a given candidate entity represents the right interpretation of a textual mention. These models use probabilistic factors. They assign entities to mentions where they consider elements like entity popularity context coherence and semantic similarity [28, 37–39].

4. **Graph-Based Methods:** Graph-based methods harness the structural relationships between entities and their contexts. These relationships are represented as nodes and edges in a graph. By modeling the interconnectedness of entities within KB these algorithms can effectively capture semantic relationships. They can also resolve ambiguities through collective inference [2, 11, 40, 41, 123–129].

5. **VSM Based Methods:** The Vector Space Model (VSM) is used by NED systems to represent entities and their environments as high-dimensional vectors inside a continuous vector space. These vectors are produced using methods such as Word2Vec BERT and TF-IDF. To determine which entity is most relevant, the similarity between candidate entity vectors and the

context vector is calculated usually using cosine similarity. Accurate entity disambiguation in text is made possible by efficient capturing of contextual subtleties and semantic relationships by VSM techniques [10, 83, 84].

6. **Information Retrieval Based Methods:** Information retrieval (IR) approaches handle disambiguation as a search problem by referencing database containing context related to the named entity. Term Frequency-Inverse Document Frequency (TF-IDF) and Advanced neural retrieval models are some of the methods. The system uses these methods to retrieve and rank candidate entities according to their relevance to the query. These techniques effectively determine the most contextually relevant entity. They utilize IR principles. This improves the precision of entity disambiguation across a range of texts [44, 45].

2.6 Evaluation criteria

The evaluation of entity linking systems is generally conducted using specific assessment measures. Such measures include precision, recall, F1-measure and accuracy [8] and for each measure we can calculate its Micro and Macro score. The precision of an entity linking system is computed as the fraction of correctly linked entity mentions that are generated by the system, as given by Equation 2.5.

$$Precision = \frac{\{|correctly \quad linked \quad entity \quad mentions|\}}{\{|linked \quad mentions \quad generated \quad by \quad the \quad system|\}} \tag{2.5}$$

The Micro_precision and Macro_precision can be calculated using these following formulas. The Micro_precision is the fraction of correctly linked entity mentions that are generated by the system across all entities and all documents, it is given by Equation 2.6.

$$Micro\_Precision = \frac{\sum_{i=1}^{N}\{|correcly\ entity\ mentions|\}}{\sum_{i=1}^{N}\{|linked\ mentions\ generated\ by\ system_i|\}} \tag{2.6}$$

The Macro_precision is the fraction of correctly linked entity mentions that are generated for each document $D_i \in D$, as shown by Equation 2.7.

$$Macro\_Precision = \frac{1}{D}\sum_{i=1}^{D}\frac{\{|correctly\ linked\ mentions_i|\}}{\{|linked\ mentions\ generated\ by\ system_i|\}} \tag{2.7}$$

Whereas the Recall is the fraction of correctly linked entity mentions that should be linked, as given by Equation 2.8.

$$Recall = \frac{\{|correctly\ \ linked\ \ entity\ \ mentions|\}}{\{|entity\ \ mentions\ \ that\ \ should\ \ be\ \ linked|\}} \tag{2.8}$$

The Micro_recall and Macro_recall can be calculated using these following formulas. The Micro_Recall is the fraction of correctly linked entity mentions that should be linked across all entities and documents as given by the Equation 2.9

$$Micro\_Recall = \frac{\sum_{i=1}^{N}\{|correctly\ \ linked\ \ entity\ \ mentions_i|\}}{\sum_{i=1}^{N}\{|entity\ \ mentions\ \ that\ \ should\ \ be\ \ linked_i|\}} \tag{2.9}$$

The Macro_Recall is the fraction of correctly linked entity mentions that should be linked for each document $D_i \in D$, as shown by Equation 2.10.

$$Macro\_Recall = \frac{1}{D}\sum_{i=1}^{D}\frac{\{|Correctly\ Linked\ Mentions_i|\}}{\{|entity\ mentions\ that\ should\ be\ linked_i|\}} \tag{2.10}$$

However, the F1 measure represents the harmonic mean between precision and recall, as given by Equation 2.11.

$$F1 = \frac{2.precision.recall}{precision + recall} \tag{2.11}$$

The Micro_F1 and Macro_F1 can be calculated using these following formulas. Micro_F1 measure represents the harmonic mean between Micro_precision and Micro_recall across all entities and all documents, as given by Equation 2.12.

$$Micro\_F1 = \frac{2 \cdot Micro\_Precision \cdot Micro\_Recall}{Micro\_Precision + Micro\_Recall} \tag{2.12}$$

Macro_F1 measure represents the harmonic mean between Macro_precision and Macro_recall for each document $D_i \in D$, as given by Equation 2.13.

$$Macro\_F1 = \frac{1}{D}\sum_{i=1}^{D}\frac{2 \cdot Macro\_Precision_i \cdot Macro\_Recall_i}{Macro\_Precision_i + Macro\_Recall_i} \tag{2.13}$$

Accuracy is used when the linked mentions generated by the system from the **Precision formula** equal the entity mentions that should be linked from the **Recall formula**; in this case, Precision = Recall= F1 measure = Accuracy. There are two types of accuracy measures: Micro-averaged accuracy and Macro-averaged accuracy.

The micro-averaged accuracy corresponds to the percentage of correctly disambiguated textual mentions across all entities and all documents. It is calculated as shown by Equation 2.14.

$$Micro\_acc = \frac{correctly \;\; disambiguated \;\; mentions}{total \;\; number \;\; of \;\; mentions} \tag{2.14}$$

Macro-averaged accuracy is the average percentage of correctly disambiguated textual mentions for each document $D_i \in D$, as shown by Equation 2.15.

$$Macro\_acc = \frac{\sum_i^{|D|} \frac{Number \;\; of \;\; correctly \;\; disambiguated \;\; mentions \;\; in \;\; D_i}{Number \;\; of \;\; mentions \;\; in \;\; D_i}}{|D|} \tag{2.15}$$

2.7 Conclusion

To summarize, the background chapter gives a solid knowledge of Named Entity Disambiguation in the context of Natural Language Processing (NLP). This chapter exposes the knowledge needed to understand the arguments and analyses that follow in the research by exploring the fundamental concepts and techniques related to NED. key concepts like knowledge base, entity linking modules, semantic relatedness concept, entity linking techniques and evaluation criteria have all been clarified throughout this chapter.

In conclusion, the background chapter provides a thorough overview of the subject of NED and establishes the framework for additional investigation and analysis in later chapters. This chapter gives a strong grasp of the theoretical foundations and enables better interaction with the topic in an informed and relevant way.

Chapter 3

Related work

3.1 Introduction

Named Entity Disambiguation has been in the spotlight for the past few years. Perhaps the best known related work is the EL shared task challenge first proposed by the National Institute of Standards and Technology (NIST) as part of the Knowledge Base Population (KBP) track within the Text Analysis Conference (TAC) in 2009. [81]

Various methods have been put forth in the quickly developing field of NED to tackle the difficulties of precisely disambiguating and connecting named entities in text to the relevant entries in a KB. In addition to examining the numerous approaches and techniques that have been created over time, this chapter offers a thorough survey of the body of current literature. To illustrate the advantages and disadvantages of each strategy, we will look at how conventional rule-based systems give way to sophisticated machine learning and deep learning models.

The first section of the chapter covers early NED techniques, which mostly depended on manually created rules and heuristics. Although these pioneering methods paved the way for later breakthroughs, they were frequently constrained by their incapacity to grow and adjust to a variety of datasets. After that, we explore the emergence of machine learning techniques, which by utilizing statistical models and feature engineering, brought greater flexibility and enhanced performance.

As the field developed, deep learning emerged and completely changed NED by allowing models to learn intricate patterns and representations from data directly. We evaluate the influence of recent dominance of transformer-based

models, which have set new standards in entity linking tasks, and neural networks, in particular, convolutional and recurrent architectures.

We also investigate graph-based methods that leverage the networked structure of KBs to improve the accuracy of disambiguation. These techniques address the disambiguation of each entity in a document as an interdependent problem, frequently utilizing collective disambiguation procedures.

Furthermore, this chapter outlines the primary issues that still need to be resolved in the field, including how to handle ambiguous mentions, properly integrate context, and handle texts that are multilingual and domain-specific. Our goal is to contextualize our research contributions and draw attention to the gaps that our proposed strategy aims to fill by offering a critical review of the relevant work.

In summary, this chapter provides an overview of the present status of NED research and lays the groundwork for the in-depth examination of our approach and conclusions found in the following chapters.

3.2 State of the art of Entity linking

In the literature, there are two principal classifications for NED approaches: Single Named Entity Disambiguation (SNED) and Collective Named Entity Disambiguation (CNED). The SNED approaches regard the mentions in a text as separate and do not leverage their correlations to predict the disambiguation entity. They rely mainly on the context-independent features to build their systems [22, 46, 103]. The second category, which is the CNED approaches, consider that mentions in the same document are associated with one or many topics. Therefore, the disambiguation of the mentions is interdependent. To harness this interdependence, CNED approaches leverage the topical coherence between the mapping entities [40, 127, 130].

3.2.1 Single named entity disambiguation approaches

There are multiple works developed using SNED. Bunescu and Pasca [27] applied a cosine similarity between the context of the mention and the Wikipedia categories of an entity candidate to solve the disambiguation problem. This similarity was improved by Cucerzan [10] where he added more similarity features

to calculate the topical coherence between a candidate entity and other entities in the context. Moreover, Milne and Witten [12] proposed the Normalized Google Distance measure for mapping the context entities to the unambiguous entities. They introduced the notion of semantic relatedness between a mention's candidate entities and mentions of the text.

Besides, several works demonstrated that learning to rank algorithms as well as deep learning techniques are very effective in handling the EL problem. Zheng et al. [46] proposed a learning to rank algorithm for entity linking. Their algorithm utilizes relationships information among the candidates in the ranking task. Gottipati and Jiang [44] Used the information retrieval with query expansion technique to present an entity linking approach based on a statistical language model where they used both local contexts and global world knowledge to expand query language models.

Furthermore, Mendes et al. [131] participated in the English entity-linking task at TAC KBP 2011 which is an international entity linking competition held every year since 2009. Their DBpedia Spotlight system is not specialize on named entities of certain types, it aims to annotate any of the approximately 3.5M entities and other concepts in DBpedia, a KB extracted from Wikipedia. The unlinkable mentions were mapped to NIL. Besides, Han and Sun [38] introduced their model called *entity-mention model* based on a generative probabilistic method where they harness a set of KB features for a given entity such as: popularity, names and context.

Furthermore, Alhelbawy and Gaizauskas [132] parsed the KB for an entry that present a certain information for a given mention in the text appearing in a certain context. A document similarity function (NEBSim) was formulated in order to calculate the similarity between two documents given a specific NE mention in one of them based on the NE co-occurrence. They also used NEBsim with a cosine similarity measure to learn a model for ranking, a Naive Bayes and SVM classifiers were used to re-rank the extracted documents.

Nebhi [133] presented a process for NED integrated in a rule-based *OBIE* system for French where they used syntactic features and popularity score features extracted from the Freebase KB to build their SVM approach. The evaluation results show that they can improve efficiency considerably. Pink et al. [134] used a simple clustering approaches with a supervised whole-document approach to create their system. The system extends their TAC 2012 system [135] where they

introduced new features for modelling local entity description and type-specific matching as well type-specific supervised models and supervised NIL classification. Moreover, Barrena et al. [136] used a prominent hypotheses in Word Sense Disambiguation (WSD) where they investigate whether these hypotheses hold for entities disambiguation, the idea is to verify if several mentions in a the document tend to refer to the same entity or not. The obtained results from the experiments conducted on different collection and three state-of-the-art NED systems (Spotlight [137], Lexical Knowledge Base (LKB) [138] and Han and Sun system [139]) showed that the NED can take advantage of these hypotheses.

Later after, Chisholm and Hachey [140] replaced Wikipedia with web links, they get entity prior, name, context, and coherence from corpus of web pages that links to Wikipedia. They showed that using 34 million web links approximates Wikipedia performance. Sun et al. [141], in order to encode mentions, contexts, and entities into continuous vector spaces, they suggested to use neural networks. Convolutional neural networks are designed to handle contexts with varying sizes and embed context word placements according to their distance from the mention. A neural tensor network is then used to describe the semantic relationships between the mention and the context.

Furthermore, Lazic et al. [142] presented *Plato*, a probabilistic model for entity resolution which take in consideration noisy or uninformative features where the training and inference can be distributed on several servers.

Yamada et al; [58] suggested an embedding approach that is tailored to NED. Words and entities are both mapped into the same continuous vector space by the suggested technique. They used two models to expand the skip-gram model. The KB graph model uses the KB's link structure to understand how things are connected, whereas the anchor context model uses KB anchors and their context words to align vectors such that comparable words and entities appear adjacent to one another in the vector space.

Besides, Francis-Landau et al. [143] opted for a convolutional neural networks to capture semantic correlation between a mention's context and a candidate entity. These networks were combined to a sparse linear model.

Li et al. [144] proposed a NED system to address the problem of Linkless Knowledge Bases where the cross-document hyperlinks are rarely accessible in many closed domain knowledge bases and it is very expensive to manually add such links unlike the former researchers where they used the context similarity

and the semantic relatedness to solve this problem.

Zhang et al. [145] developed *XLink*, an online bilingual entity linking system based on Wikipeida and Baidu Baike[16]. *XLink* conducts two steps to link the mentions in the text to their corresponding entities in KB: a mention parsing step without using any NER tool and entity disambiguation step. An unsupervised generative probabilistic method was used to model the contextual feature, coherence feature and prior feature jointly.

Additionally, Eshel et al. [146] payed more attention for noisy text such as webpage fragments, social media, or search queries where they are often short, noisy, and less coherent. They proposed a model which used a neural approach that leverages a big amount of training data in Wikilinks NED to learn representations for entity and context.

Ganea and Hofmann [47] combined the benefits of deep learning with more traditional approaches such as graphical models and probabilistic mention entity maps for disambiguation. Furthermore, Sil et al. [48] proposed a neural model that trains fine-grained similarities and dissimilarities between the query and candidate document for entity linking which can be used in zero-short learning for other languages.

Inan and Dikenelli [147] proposed an algorithm that links the unambiguous mentions first then handles the rest of the ambiguous mentions in a specific domain, where they provided a sequence learning model like a translation task in which a sequence of mentions will be translated into a sequence of referent entities in the domain-specific KB. The framework *GERBIL* (General Entity Annotation Benchmark Framework) was used to evaluate their system against several entity Linking approaches available in *GERBIL* [148]. Likewise, Mueller and Durrett [149] noted that neural models relying on attention do not always choose the correct context evidences in short texts even though there is an overlap between these evidences and the correct entity title, thus they augmented their model with sparse features specifically targeting this kind of lexical overlap. Radhakrishnan et al. [150] proposed *ELDEN* a system that densifies Knowledge Graphs with statistical co-occurrence from a large text corpus, this densified KG is later used to train entity embeddings.

Moreover, Shahbazi et al. [151] proposed to start from an initial solution from a local model then expand this solution using Limited Discrepancy Search (*LDS*)

[16]http://baike.baidu.com/

seeking for possible corrections and improvement.

Kundu et al. [152] proposed an entity-centric neural cross-lingual co-reference model that builds on multi-lingual embeddings and language-independent features for English where they build monolingual embeddings for English, Chinese and Spanish using the widely used *CBOW* word2vec model [153] without using any annotated data from Chinese or Spanish.

Furthermore, Shahbazi et al. [154] learned an entity aware extension of Embedding for Language Model (*ELMo*). Each mention is first defined as a function of the entire paragraph then they predict the referent entities. They introduced an approach for learning contextual entity representations by learning an entity-aware extension of *ELMo*, the context-rich entity representations shown to be more suitable for the single disambiguation using only local contexts.
Yao et al. [49] proposed an efficient position embeddings initialization method that initializes larger position embeddings to train a model for the zero-shot entity linking task.

In addition, Chen et al. [50] improved the local model of Ganea and Hofmann [47] by integrating a pre-trained model based on an entity similarity feature to better capture entity type information. Moreover, Mulang et al. [51] contended that the Wikidata graph context provides sufficient signals to guide pre-trained transformer models and enhance their performance for NED on Wikidata.

To achieve end-to-end entity linking over KBs, Ravi et al. [52] suggested *CHOLAN*, a modular strategy. To complete the EL task, *CHOLAN* comprised a pipeline with two transformer-based models that are sequentially combined. In a given text, the first transformer model locates surface forms (entity mentions). A second transformer model is used for each mention to categorize the target entity from a list of predetermined candidates. The latter transformer is supplied with an enriched local context, which comprises a summation of these three following embeddings: Token embedding, segment embedding, and position embedding. Furthermore, De Cao et al. [53] suggested *GENRE*, a system that retrieves entities by creating their distinct names token-by-token in an autoregressive manner and conditioned on the context using a transformer-based architecture that has been pre-trained with a language modeling purpose and fine-tuned to produce entity names. Moreover, De Cao et al. [54] proposed a very effective method that utilizes a shallow and effective decoder and parallelizes autoregressive linking across all possible mentions. Barba et al. [55] presented two transformer-based designs that

perform *EXTEND*, a local formulation for ED, in which they frame this task as a text extraction problem.

Furthermore, Atzeni et al. [56] introduced *DUCK*, a system designed to categorize entities in a knowledge graph based on their relationships, which means the type of entity based on the relations that it has with other entities in a knowledge graph. Relations in a knowledge graph are used to determine entity types. *DUCK* converts box embeddings into spherical polar coordinates, representing relations as boxes on a hypersphere. The model optimizes entity clustering by placing them inside boxes corresponding to their relations.

All of the cited works in the SNED approaches section are compiled in Table 3.1, which also lists the methods used in each work and arranges them chronologically based on publication year.

Table 3.1 Overview of SNED's Works .

Approach	Authors/Reference	Paper's Title	Method	Year
SNED	Bunescu and Pasca. [27]	Using encyclopedic knowledge for named entity disambiguation.	Supervised	2006
	Cucerzan. [10]	Large-scale named entity disambiguation based on wikipedia data.	Unsupervised	2007
	Milne and Witten. [12]	Learning to link with wikipedia.	Supervised.	2008
	Zheng et al. [46]	Learning to link entities with knowledge base.	Supervised	2010
	Gottipati and Jiang [44]	Linking entities to a knowledge base with query expansion.	Unsupervised	2011
	Mendes et al [131]	Evaluating dbpedia spotlight for the tac-kbp entity linking task.	Unsupervised	2011
	Han and Sun [38]	A generative entity-mention model for linking entities with knowledge base.	Supervised	2011
	Alhelbawy and Gaizauskas [132]	Named entity based document similarity with svm-based re-ranking for entity linking	Supervised	2012
	Nebhi [133]	Named entity disambiguation using freebase and syntactic parsing	Supervised	2013
	Pink et al [134]	Sydney cmcrc at tac 2013	Supervised	2013
	Barrena et al [136]	"One entity per discourse" and "one entity per collocation" improve named-entity disambiguation	Unsupervised	2014
	Chisholm and Hachey [140]	Entity disambiguation with web links.	Supervised	2015
	Sun et al [141]	Modeling men- tion, context and entity with neural networks for entity disambiguation.	Supervised	2015
	Lazic et al [142]	Modeling mention, context and entity with neural networks for entity disambiguation.	Supervised	2015
	Yamada et al [58]	Joint learning of the embedding of words and entities for named entity disambiguation.	Supervised	2016
	Francis-Landau et al [143]	Capturing semantic similarity for entity linking with convolutional neural networks.	Supervised	2016
	Li et al [144]	Entity disambiguation with linkless knowledge bases.	Supervised	2016
	Zhang et al [145]	Xlink: An unsupervised bilingual entity linking system.	Unsupervised	2017
	Eshel et al [146]	Named entity disambiguation for noisy text	Supervised	2017
	Ganea and Hofmann. [47]	Deep joint entity disambiguation with local neural attention.	Supervised	2017
	Sil et al. [48]	Neural cross-lingual entity linking.	Supervised	2018

Approach	Authors/Reference	Paper's Title	Method	Year
SNED	Inan and Dikenelli [147]	sequence learning method for domain-specific entity linking.	Supervised	2018
	Mueller and Durrett [149]	Effective use of context in noisy entity linking	Supervised	2018
	Shahbazi et al [151]	Joint neural entity disambiguation with output space search,	Supervised	2018
	Kundu et al [152]	Neural cross-lingual coreference resolution and its application to entity linking	Supervised	2018
	Shahbazi et al [154]	Entity-aware elmo: Learning contextual entity representation for entity disambiguation	Supervised	2019
	Yao et al. [49]	Zero-shot entity linking with efficient long range sequence modeling.	Supervised	2020
	Chen et al. [50]	Improving entity linking by modeling latent entity type information.	Supervised	2020
	Mulang et al. [51]	Evaluating the impact of knowledge graph context on entity disambiguation models.	Supervised	2020
	Ravi et al. [52]	A modular approach for neural entity linking on wikipedia and wikidata.	Supervised	2021
	De Cao et al. [53]	Autoregressive entity retrieval.	Supervised	2021
	De Cao et al. [54]	Highly parallel autoregressive entity linking with discriminative correction.	Supervised	2021
	Barba et al. [55]	Extend: extractive entity disambiguation.	Supervised	2022
	Atzeni et al. [56]	Polar ducks and where to find them: Enhancing entity linking with duck typing and polar box embeddings.	Supervised	2023

3.2.2 Collective named entity disambiguation approaches

In the collective named entity disambiguation approaches, Kulkarni et al. [28] presented a collective approach for entity linking that models the coherence as a probabilistic factor graph for each pair of entity candidates and the different mentions in the text. Besides, Ferragina and Scaiella [106] Introduced *Tagme*, a system that handles the disambiguation of the mentions appearing in short and poorly composed texts like: snippets of search engine results, tweets and news. *Tagme* finds the collective agreement among the candidate entities using scoring functions where they consider the sparseness of the anchors in the short text using Milne and Witten work [12] combining the relatedness function among concepts.

Additionally, Shirakawa et al. [155] used a probabilistic taxonomy then apply

a naive Bayes probabilistic model to disambiguate a mention by identifying its related mentions in the same document.

Ratinov et al. [13] analyzed approaches that utilize Wikipedia link structure information to arrive at coherent sets of disambiguation entities from the input text and compare them to single entity disambiguation approaches.

Besides, Han et sun [107] introduced a probabilistic entity-topic model using the context compatibility, the topic coherence and the correlation between them. They develop a *Gibbs sampling algorithm* to tackle the two inference tasks of their model by identifying the global knowledge from data and then make collective entity disambiguation decisions.

Similarly, Shen et al. [31] proposed a framework called *LINDEN* that aims to link named entities in the text with a KB unifying Wikipedia and WordNet. Sen [57] proposed a latent topic model to learn the context entity association and showed that this improved the disambiguation accuracy. Guo et al. [88] handled Tweet linking where the messages on micro-blogs are short, noisy, informal with restrict context and a lot of text ambiguous meanings. They optimized mention detection and entity disambiguation as a single task using a structural SVM.

Ganea et al. [156] used an effective graphical model to perform collective entity disambiguation where they proposed an unsupervised probabilistic approach Bag-Of-Hyperlinks model (*PBoH*). The mentions in the text are disambiguated jointly across an entire document by combining a document-level prior of entity co-occurrences with local information captured from mentions and their surrounding context.

Yamada et al. [58] modelled textual and global contexts using a continuous vector space to collectively link words and entities. Ganea and Hofmann [47] provided a deep learning model for joint document-level entity disambiguation, which uses learned neural representations, and they showed a highly intriguing neural model for simultaneously learning entity embedding together with mentions and contexts. Entity embeddings, a neural attention mechanism for local context windows, and a differentiable joint inference step for disambiguation are its important components.

Shahbazi et al. [151] provided a model for entity disambiguation that uses Limited Discrepancy Search to integrate local contextual information with global evidence (*LDS*). They started with a full solution generated by a local model and search the space of feasible adjustments to enhance the local solution from a

global point of view, given an input document.

Furthermore, Yang et al. [59] presented an efficient algorithm for approximate bidirectional inference and used a gradient-tree-boosting-based structured learning model for jointly disambiguating named entities in a document. Fang et al. [60] proposed a deep reinforcement learning model to solve the ED problem. They considered both local context and global coherence to disambiguate mentions using the prior designated entity information and made decisions from a global point of view. Besides, Yang et al. [61] introduced a dynamic context augmentation process to integrate the global signal for EL which requires only one pass across all mentions, where they collected information from formerly linked entities to improve further decisions. Moreover, Yamada et al. [62] improved Fang et al. [60] and Yang et al. [61] works by introducing a new model based on contextualized embeddings of words and entities for ED.

Xue et al. [127] presented a unique end-to-end neural network with recurrent random walk layers that incorporates external knowledge to reflect the semantic dependency between distinct EL decisions.

Moreover, El Vaigh et al. [157] designed a Resource Description Framework RDF-based entity relatedness measure for global scores that has the following important properties: it has a clear semantics, it can be computed at a reasonable computational cost, and it accounts for the transitive aspects of entity relatedness by using existing property paths between entities in an RDF KB.

Ayoola et al. [63] offered an ED model that links entities by reasoning over a symbolic KB in a completely differentiable manner, enabling the usage of all KB facts as well as descriptions and types. Furthermore, on the basis of *BERT* (Bidirectional Encoder Representations from Transformers) Devlin et al. [64], Yamada et al. [65] suggested a global ED model. Their approach treats both words and entities as input tokens in order to capture global contextual information for ED. Moreover, Ji et al. [66] Proposed *BI-INTEL*, a model for entity linking, utilizing local compatibility for candidate entity representation through bidirectional interaction and a global interdependence component employing random walk layers to capture dependencies among entity linking choices.

All of the cited works in the CNED approaches section are compiled in Table 3.2, which also lists the methods used in each work and arranges them chronologically based on publication year.

Table 3.2 Overview of CNED's Works .

Approach	Authors/Reference	Paper's Title	Method	Year
CNED	Kulkarni et al. [28]	Collective annotation of wikipedia entities in web text.	Supervised	2009
	Ferragina and Scaiella [106]	Tagme: on-the-fly annotation of short text fragments (by wikipedia entities).	Supervised	2010
	Shirakawa et al [155]	Entity disambiguation based on a probabilistic taxonomy	Supervised	2011
	Ratinov et al. [13]	Local and global algorithms for disambiguation to wikipedia.	Supervised	2011
	Han et sun [107]	An entity-topic model for entity linking	Supervised	2012
	Shen et al. [31]	Linden: linking named entities with knowledge base via semantic knowledge.	Supervised	2012
	Sen [57]	Collective context-aware topic models for entity disambiguation.	Supervised	2012
	Guo et al [88]	To link or not to link? a study on end-to-end tweet entity linking.	Supervised	2013
	Ganea et al [156]	Probabilistic bag-of- hyperlinks model for entity linking.	Unsupervised	2016
	Yamada et al. [58]	Joint learning of the embedding of words and entities for named entity disambiguation.	Supervised	2016
	Ganea and Hofmann [47]	Deep joint entity disambiguation with local neural attention.	Supervised	2017
	Shahbazi et al [151]	Joint neural entity disambiguation with output space search.	Supervised	2018
	Yang et al. [59]	Collective entity disambiguation with structured gradient tree boosting.	Supervised	2018
	Fang et al. [60]	Joint entity linking with deep reinforcement learning.	Supervised	2019
	Yang et al. [61]	Learning dynamic context augmentation for global entity linking.	Supervised	2019
	Yamada et al. [62]	Global entity disambiguation with pretrained contextualized embeddings of words and entities.	Supervised	2019
	Xue et al [127]	Neural collective entity linking based on recurrent random walk network learning.	Supervised	2019
	El Vaigh et al [157]	Novel path-based entity relatedness measure for efficient collective entity linking.	Supervised	2020
	Ayoola et al. [63]	Improving entity disambiguation by reasoning over a knowledge base.	Supervised	2022
	Yamada et al. [65]	Global entity disambiguation with bert.	Supervised	2022
	Ji et al. [66]	A multi-angle bidirectional interaction model for entity linking.	Supervised	2023

On the other hand, graph-based methods were shown to be very effective and are largely used in collective disambiguation approaches. It is worth to mention that the graph-based methods can be used as supervised or unsupervised approaches, also they can be used in single disambiguation as well as in collective disambiguation.

In the remaining of this section, we will highlight works that employ the same CNED approach and graph-based method as our study.

Han et al. [11] proposed a collective EL model based on graph that exploits the global interdependence between different EL decisions; they introduced a referent graph used in conjunction with an inference algorithm. In the meantime, Hoffart et al. [40] presented a framework that combines three measures: (i) the prior probability of the mentioned entity; (ii) the similarity between the contexts of a mention and a candidate entity and (iii) the coherence among candidate entities for all mentions together, where they calculate a dense sub-graph to approximate the best joint mention-entity mapping. Furthermore, Alhelbawy and Gaizauskas [2] presented two collective disambiguation approaches using a graph representation, where the nodes of the graph represent candidate entities and the edges represent the coherence between them. Their first approach uses Page Rank (PR) for the disambiguation, while a clique partitioning technique was used in the second approach to detect the most weighted clique and use its nodes as disambiguation entities. A confidence score is assigned to the nodes in order to be applied in the clique weighting by leveraging three similarity measures and entity popularity. A unified semantic representation for entities and documents was introduced by Guo and Barbosa [158] using the stationary distribution across a random walk with restart on an entity graph. Furthermore, Pershina et al. [159] combined local and global evidence for collective disambiguation where they introduce a graph-based approach that leverage Personalized PageRank (PPR) algorithm. Their algorithm use random walk and does not require supervision.

Huang et al. [160] Measured entity semantic relatedness for topical coherence by introducing a Deep Semantic Relatedness Model ($DSRM$) based on deep neural networks (DNN) and semantic knowledge graphs (KGs).

Moreover, Wang et al. [124] avoided excessive linguistic analysis on the source documents and fully leverages the KB structure where they proposed an unsupervised algorithm named Quantified Collective Validation. Additionally, they

deploy their system in a new language (Chinese) and two new domains (Biomedical and Earth Science).

Zwicklbauer et al. [161] presented a collective graph-based disambiguation algorithm using semantic entity and document embeddings for robust entity disambiguation (*DoSeR*). A k-partite relatedness graph is created between all candidate entities. Semantic embeddings, i.e. real-valued n-dimensional vectors capturing the semantics of entities are used to determine the relatedness between candidate entities where they used *GERBIL* [148] to evaluate their system.

Besides, Ganea et al. [156] leveraged co-occurrence statistics in a fully probabilistic form, where they used a graph-based model that addresses collective entity disambiguation. Gong et al. [162] proposed a graph-based linking algorithm that incorporates the combination of semantic relations and co-reference relations to link at the same time a set of coherent mentions. Chong et al. [123] handled tweets that are close in space and time. They present a graph based model that applies geocoded tweets where they connect tweets close in space and time to form a tweet graph and define a novel objective function over the graph. They also introduce a comparison-based evaluation approach which addresses: noisy mention extraction, incomplete KB and annotation effort.

Furthermore, A bidirectional Long Short-Term Memory (*BiLSTM*) and dynamic convolutional neural network (*DCNN*) were used by Lu et al. [163] to model the mention and the entity candidate respectively. They introduce a graph-based model which represents the semantic relatedness between mentions and their corresponding candidate entities.

Moreover, Guo and Barbosa [164] introduced a new semantic measure based on information theory, balancing lexical and semantic similarity. By calculating mutual information between random walks on the disambiguation graph, their method calculated semantic similarity. They suggested a learning-to-rank extension and an iterative approach. Zeng et al. [165] proposed a graph-based algorithm Gloel to harness co-occurrences in entity lists for mining both explicit and implicit entity relations. They integrated The relations into an entity graph where they incorporate personalized PageRank to calculate entity coherence by combining local mention-entity similarity and global entity coherence.

Cao et al. [125] proposed *NCEL* a neural model for collective entity linking. *NCEL* uses Graph Convolutional Network to combine both local contextual features and global coherence information. They approximated graph convolution

on a subgraph of neighboring entity mentions rather than the complete text to enhance computing efficiency.

Besides, the suggested approach by Le and Titov [126] made use of naturally occurring data in two stages, including Wikipedia and unlabeled documents. For every mention, it first generates a high recall list of potential entities. Next, by treating entities as latent variables, it trains a document-level entity linking model under poor supervision. Using unlabeled texts for estimation, the model chooses entities based on the coherence of each mention in the local context and on how well they fit in with other entities.

Xue et al. [127] proposed a neural network with recurrent random-walk layers for collective EL based on graphs. They incorporated the external KB to collectively infer the referent entities of all mentions of the same document by exploiting local context features. Hu et al. [128] also introduced a graph-neural entity disambiguation model where a heterogeneous entity-word graph is created for each document to model the global semantic relationships among candidate entities in the same document. Furthermore, Xin et al. [129] suggested a Locally-Global (LoG) model for ED, which localizes global properties from a small set of nearby mentions. They used their proposed tree connection method $CoSimTC$ to calculate the cross-tree distance between mentions, and they extract mention neighbors according to the syntactic distance on a dependency parse tree. They also suggested the *Sent2Word* keyword extraction technique to find the keywords in each document. To build a discriminative representation for each candidate item, they also expand their Graph Attention Network (GAT) [166] to incorporate local and global information.

The graph-based studies are emphasized in Table 3.3, which is organized by publication year and precise the method applied in each work.

Table 3.3 Overview of graph-based works using CNED approach.

Approach	Authors/Reference	Paper's Title	Year	Method
CNED	Han et al. [11]	Collective entity linking in web text: a graph-based method.	2011	Unsupervised
	Hoffart et al. [40]	Robust disambiguation of named entities in text.	2011	Supervised
	Alhelbawy and Gaizauskas. [2]	Collective named entity disambiguation using graph ranking and clique partitioning approaches.	2014	Unsupervised
	Guo and Barbosa. [158]	Robust entity linking via random walks.	2014	Unsupervised
	Pershina et al [159]	Personalized page rank for named entity disambiguation.	2015	Unsupervised
	Huang et al [160]	everaging deep neural networks and knowledge graphs for entity disambiguation.	2015	Supervised
	Wang et al [124]	Language and domain independent entity linking with quantified collective validation.	2015	Unsupervised
	Zwicklbauer et al [161]	Robust and collective entity disambiguation through semantic embeddings.	2016	Supervised
	Ganea et al. [156]	Probabilistic bag-of-hyperlinks model for entity linking.	2016	Supervised
	Chong et al [123]	Collective entity linking in tweets over space and time.	2017	Unsupervised
	Gong et al. [162]	Collective entity linking on relational graph model with mentions.	2017	Supervised
	Lu et al [163]	Boosting collective entity linking via type-guided semantic embedding.	2017	Supervised
	Guo and Barbosa. [164]	Robust named entity disambiguation with random walks.	2018	Supervised
	Zeng et al [165]	Collective list-only entity linking: A graph-based approach.	2018	Supervised
	Cao et al [125]	Neural collective entity linking.	2018	Supervised
	Le and Titov [126]	Boosting entity linking performance by leveraging unlabeled documents.	2019	Supervised
	Xue et al. [127]	Neural collective entity linking based on recurrent random walk network learning.	2019	Supervised
	Hu et al. [128]	Graph neural entity disambiguation.	2020	Supervised
	Xin et al. [129]	Log: a locally-global model for entity disambiguation.	2021	Supervised

To tackle the above-described issues, we exploit in this work the cliques derived from a graph representation of the candidate entities referring to the mentions of the same document to disambiguate them collectively and iteratively. Our model leverages the semantic relatedness between the mapping entities rather

than the context-independent features to enhance accuracy.

3.3 Position of the current work

Our work focuses on the idea that several named entities (NEs or mentions) in a document might work together to disambiguate each other. This method takes into account the possibility that some textual mentions in a document could be confusing on their own. To jointly address the ambiguities of all NE textual mentions in a document, a collective disambiguation method is therefore required. Utilizing collective disambiguation technique, our work makes use of an unsupervised graph-based method that have proven effective in solving the Named Entity Disambiguation problem.

Our goal is to increase the accuracy and coherence of entity disambiguation which means guarantying that the linked entities are related and compatible with one another in the text's larger context, in addition to being accurate on their own. For instance, in a sentence on technology, connecting "Apple" to the business and "Jobs" to Steve Jobs enhances coherence, whereas connecting "Apple" to the fruit would weaken it. This increase can be achieved through collective techniques since the connectivity of entities in the graph facilitates more reliable and context-aware ambiguity resolution.

3.4 Conclusion

The relevant works in the field of entity linking were examined in this chapter, with an emphasis on notable developments and ongoing challenges.

In summary, the examination of related works in the entity linking field shows the obstacles that researchers continue to encounter as well as the progress that has been made. Notable advancements have resulted from the transition from early rule-based and machine learning techniques to modern deep learning and transformer-based models. In particular, these developments have improved entity linking systems' precision and effectiveness. It has proven possible to obtain more successful entity disambiguation and contextual understanding across a variety of intricate datasets. Nonetheless, the field is still struggling with a number of important problems. Two significant obstacles still remain: the ambiguity of

natural language by nature and the requirement for large, high-quality KBs. Furthermore, there are many difficulties in processing texts in specific domains and multiple languages. Combining entity linking with more general tasks related to natural language processing, like information extraction and semantic search, presents further areas for development.

Chapter 4

Research methodology

4.1 Introduction

This Chapter provides a detailed account of our research process. It describes the techniques utilized in conducting and developing this research. Furthermore, it provides a rational and justifications for the methods selected and adopted in various aspects and phases of the research.

Denzin and Lincoln [167] state that the choice of a research methodology or strategy depends on the specific research question and the topic under investigation. Consequently, the research design employed in the study should be regarded as a means to address the research questions.

A quantitative research approach was chosen as the methodology of our work. This kind of approaches focuses on collecting and analyzing numerical data.

A very succinct definition of quantitative research, provided by Creswell [168], is a sort of research that gathers numerical data and uses mathematically based methodologies (namely, statistics) to evaluate and interpret the data in order to explain phenomena. The principles of quantitative research are:

- **Objectivity**: Reducing bias and subjective interpretations is a top priority in quantitative research. In order to ensure objectivity in data gathering and analysis, researchers create experiments and procedures that others can reproduce to reach similar results.

- **Measurement:** Measurable data is the center of everything. The ability to quantify variables and convert them into terms that can be measured makes it possible to gather numerical data appropriate for statistical analysis.

- **Statistical Analysis:** For quantitative research, statistical techniques are essential. To investigate correlations between variables, test hypotheses, and reach data-driven conclusions, researchers use a variety of methodologies, including regression analysis, correlation analysis, and hypothesis testing.
- **Testing Hypotheses:** A hypothesis, or precise prediction regarding the relationship between variables, is frequently the starting point of quantitative research. The gathered data is then analyzed using statistical tests to see whether the hypothesis is supported or refuted.
- **Generalizability:** Extrapolating findings beyond the particular sample under study is the aim. To ensure that findings may be generalized, researchers take into account how well their sample represents a larger population and employ suitable sampling strategies.
- **Replication:** The capacity to repeat results is a significant strength. To increase the research's credibility, it would be ideal for additional researchers to use the same methodology and examine comparable data in order to validate the links or effects that have been noted.
- **Control:** In some situations, especially experiments, scientists try to keep an eye on unrelated factors that could affect the result. In doing so, the precise impact of the independent variable on the dependent variable is more easily isolated.

Quantitative research is based on the collecting of numerical data, where numbers serve as the fundamental unit. These can include surveys, experiments, measured or tallied observations, or pre-existing databases which contains numerical information. In our work, the data was collected using pre-existed datasets.

The reminder of this chapter provides descriptions of different stages of the research as shown in figure 4.1. Such stages encompass: Identification of needs, problem definition, data collection and preparation, design of the proposed system and finally system implementation and evaluation. This careful depiction ensures a comprehensive understanding.

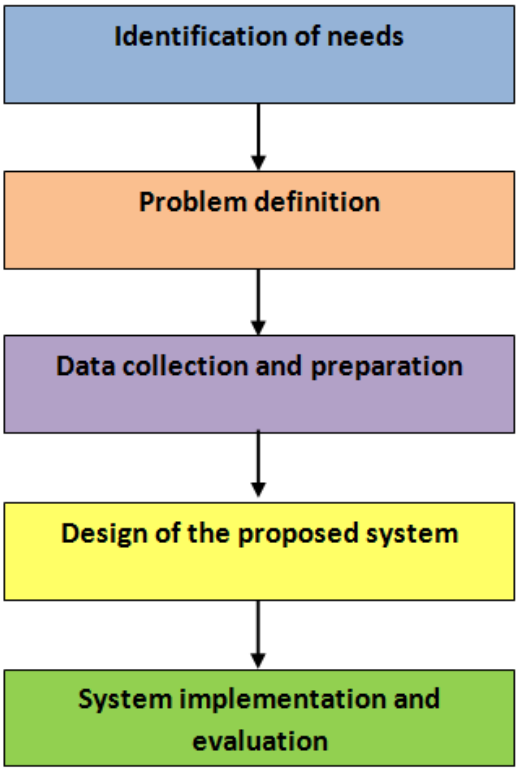

Figure 4.1: *Research methodology adopted*

4.2 Identification of needs

The process of creating a new NED system requires a thorough comprehension of both the existing constraints and areas for development. Current systems might be limited to performing effectively within their domains, have trouble with big and diverse datasets, or struggle with particular entity kinds. Furthermore, the processes by which certain systems come at disambiguation conclusions may not be transparent.

In order to solve some of these issues, this study suggests a revolutionary NED system with higher accuracy is its ultimate goal. With careful consideration of the unique requirements of the planned application, the properties of the data it will analyze, and any practical limitations, this new method could provide a great deal of improvement over current NED systems.

4.3 Problem definition

In natural language processing (NLP), named entity disambiguation is a crucial problem. Its goal is to establish a connection between textual references to real-world entities (such as individuals, groups, and places) and their relevant entries in a KB. High precision and robustness are hindered by issues with current NED systems, despite notable developments.

One of the main issues is that unclear entities have limited precision. Large datasets provide scalability challenges. A common requirement of domain specificity is retraining for new domains. Furthermore, the process of disambiguation lacks explainability which means the capacity to comprehend and interpret the system's methods and reasoning for resolving ambiguities between entities. It offers insights into the decision-making process, ensuring transparency, trust, and simpler debugging. Developing more accurate NED systems is the goal of this research in an effort to address this problem which means the ability to correctly identify and link ambiguous entities to their intended references. Accurate entity disambiguation is necessary for the advancement of NLP.

In order to increase disambiguation accuracy, our research suggests an innovative and unsupervised approach to tackle the disambiguation problem.

4.4 Data collection and preparation

In this study we use the Kensho-derived Wikimedia Dataset (KDWD) [1] as a KB, where we leverage it in the development of our system and the creation of the samples (benchmarks) used as an initial test and evaluation of our system.

There are many strong reasons of using the KDWD KB in our NED system. An extensive and current source of structured data is offered by the KDWD. This dataset covers a wide variety of subjects. Ensuring a comprehensive and up-to-date coverage is crucial for precise entity disambiguation. It also helps with clarification.

The reliability is increased by its community-validated data and open-access design. Additionally, the information's reliability is strengthened. The interrelated facts and extensive metadata of the dataset make it easier to create intricate knowledge graphs. This enhances the entities' semantic comprehension. Also improved the contextual relevance of the entities.

Moreover, the fact that Wikimedia projects are inherently multilingual guarantees our NED system is capable of handling multiple languages. This capability may prove useful in the future where we might need to accommodate additional languages.

The KDWD provides improved data quality and is enhanced by Kensho Technologies' experience. One further enhanced feature is structure. For strong performance, these qualities are essential.

The KDWD is a structured, multi-layered rendition of the Wikidata knowledge graph. It has three connected layers of data. The base layer is *Wikipedia text* which is a plain text English Wikipedia corpus. The middle layer is *Wikipedia links* which annotates the corpus by including which text spans are links. The top layer, *Wikidata graph* connects the link text spans to items in Wikidata. Figure 4.2 shows these three layers. In total, KDWD contains over 5 million pages, 51 million entities, and over 140 million relations (see Table 4.1) .

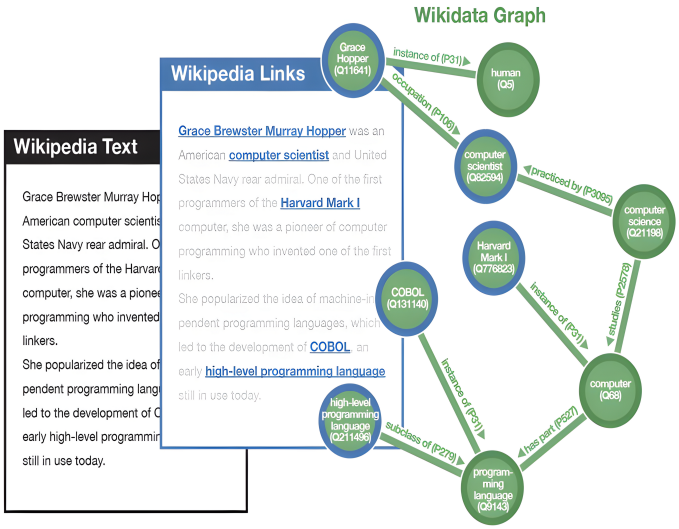

Figure 4.2: *Snapshot of KDWD Knowledge Base. [1]*

Table 4.1 KDWD Knowledge Base. [1]

Pages	Tokens	Entities	Relations
5.3M	2.3B	51M	140M

The size of this KB is 25.32 GB zipped, we download it from **Kaggle**[17]. The KDWD contains seven files, two of them are extracted from Wikipedia and the remaining five files are extracted and constructed from the Wikidata which is a KB in a graph form.
The Wikipedia files are:

- page.csv (page metadata and Wikipedia-to-Wikidata mapping)
- link_annotated_text.jsonl (plaintext of Wikipedia pages with link offsets)

The Wikidata files are:

- item.csv (item labels and descriptions in English)
- item_aliases.csv (item aliases in English)
- property.csv (property labels and descriptions in English)
- property_aliases.csv (property aliases in English)
- statements.csv (truthy qpq statements)

From the files described above we have prepared the following files as shown in Figure 4.3 :

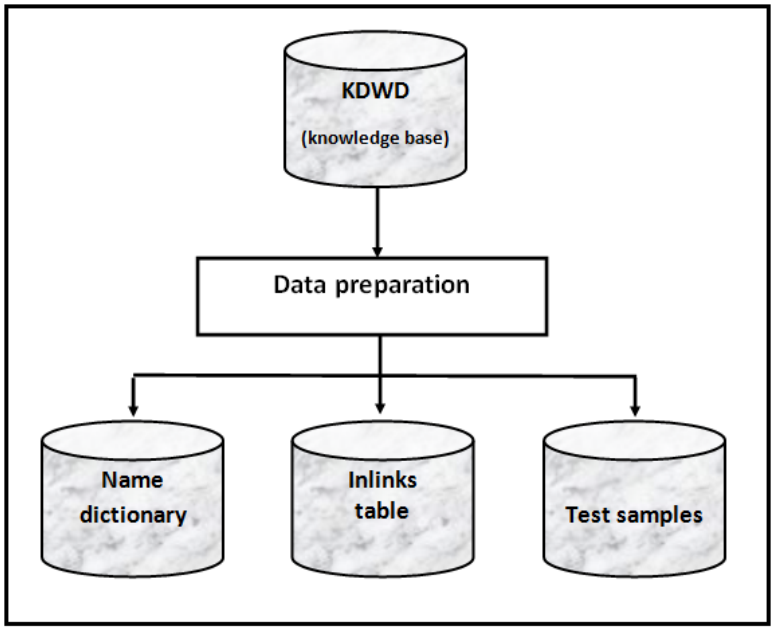

Figure 4.3: *Data preparation*

[17]https://www.kaggle.com/kenshoresearch/kensho-derived-wikimedia-data

- **A name dictionary** In our work we use a name dictionary technique to generate the candidate entities. The candidate entity generation module in NED provides clear benefits over search engine-based techniques when it makes use of name dictionary. Name dictionaries guarantee excellent precision. They offer a carefully selected list of domain-specific items. Name dictionaries expedite the process. They directly access a precompiled list leading to speedier identification ideal for real-time applications. In contrast, search engine methods may produce irrelevant results. Name dictionaries customized for certain domains improve disambiguation. They closely align with encountered entities. This increases relevance and accuracy.

  To create the name dicionary we concatenated the tables *item.csv*, *item_aliases.csv* and another file *entity_df.csv* derived from *link_annotated_text.jsonl*. Before the concatenation, we filtered each table by dropping the NIL rows, then we gave the same names for the two rows of each table. Our name dictionary is a key-value table, containing over 48 million distinct name mentions and the referent candidate entities of each mention. The dictionary was prepared in the same manner as Chen et al. [169]. Thereafter, the candidate entities of each mention in the document are generated using this table.

- **An In-links table** The main goal of creating this table is to calculate the semantic relatedness between two any given entities. This table is constructed from the *page.csv* file. It contains two columns. For each page, we assign a row where the first column contains its unique page id and the second contains the set of page ids that represent the incoming links to this page. This table is then used to calculate the Jaccard distance measure between two entities which represents the semantic relatedness between these two entities.

- **Test samples:** Our benchmark is created using this file *link annotated text.jsonl*, where we create eight samples. A systematic variable sampling method is employed in our work. Systematic variable sampling is straightforward and simple to use involves selecting samples at regular intervals is a simple and productive research strategy. By ensuring uniform distribution throughout the population, this method improves representativeness and

lessens clustering. When the population list is sorted randomly, systematic variable sampling reduces selection bias, producing more objective and dependable results. Furthermore, it enhances data quality by identifying trends and patterns that are frequently overlooked in random sampling and is reasonably priced, particularly when combined with a full population list.

In our work, items from a list are chosen at random intervals. For instance, each $\mathbf{n}^{th}$ item in the dataset is selected. **n** can vary depending on the sample and indicate how many documents were chosen for that particular sample. Using this sampling technique, many samples with varying numbers of documents and mentions can be produced. The eight samples contain, in total, **237** documents and **2856** mentions, with an average of **12.05** mentions per document. The following Table 4.2 represents a detailed description of each sample .

Table 4.2 The eight KDWD test samples.

Samples	**# Doc**	**# Ment**
Sample 1	30	423
Sample 2	29	235
Sample 3	30	282
Sample 4	28	303
Sample 5	29	311
Sample 6	30	632
Sample 7	44	432
Sample 8	17	238
All Samples	237	2856

4.5 Design of the proposed system

In our Named Entity Disambiguation generation, we employ a graph-based method since it provides better contextual understanding and makes use of rich semantic linkages between entities. Graph structures are very good at collecting and displaying intricate dependencies and linkages, which are important for correctly disambiguating entities in a variety of contexts.

Moreover, the scalability of graph databases enables us to effectively manage extensive and complex datasets, rendering the graph-based method a resilient and adaptable option for our NED system.

4.6 System implementation and evaluation

Our system is implemented using *Python* programming language. There are many benefits in building a NED system in Python. Python is widely known for its ease of use. Its comprehensibility makes it a perfect option for rapidly creating NED models. Text analysis, machine learning and natural language processing are all crucial components of NED systems. These are made possible by its vast ecosystem of libraries and frameworks. These include *scikit-learn*. *NLTK* and *spaCy* are also part of the ecosystem. Python's versatility makes it easy to integrate with other technologies. Integration with external KBs or semantic resources is thus easier in the NED workflow (for more implementation details, see Chapter 6).

For the evaluation of our system and besides the test dataset we have created from the KDWD KB, we have used AIDA/ CoNLL 2003 -TestB, IITB, MSNBC, AQUAINT, ACE2004, Cweb, and Wiki datasets to evaluate the performance of our system (Figure 4.4). Assessing our NED system with datasets like AIDA/CoNLL2003-TestB IITB MSNBC, AQUAINT ACE2004, Cweb and Wiki has number of benefits. These datasets offer extensive and varied sets. They contain text documents from multiple genres. The documents span various areas. This guarantees reliable analysis. It works in a variety of settings. The annotated entities and ground truth annotations associated with the AIDA/CoNLL2003-TestB IITB MSNBC AQUAINT, ACE2004 and Cweb datasets provide for a quantitative assessment. They can measure the precision, recall, F1 score and accuracy of the NED system. These datasets also include a broad spectrum of entity types, including named entities, enabling extensive testing of the system's capacity to disambiguate various entity mentions. By granting access to a large body of knowledge, the Wiki dataset's inclusion enhances the evaluation process even more and makes it possible to gauge how well the system links entities to the KB. In general, using these datasets guarantees thorough assessment of NED systems, making it easier to pinpoint areas for development as well as their advantages and disadvantages.

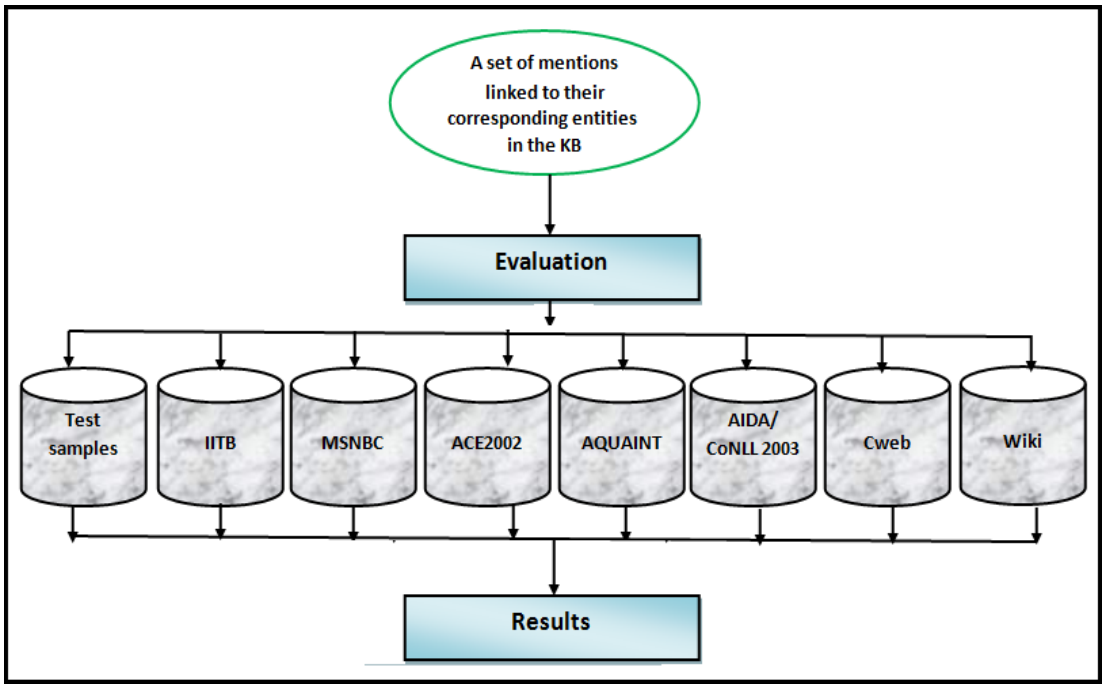

Figure 4.4: *Our system evaluation process*

4.7 Conclusion

In this Research methodology chapter, we carefully described the methodical process used for the creation and assessment of our Named Entity Disambiguation system. Every stage, from identification of needs, problem definition, gathering data and preprocessing to choosing techniques , is thoroughly justified in light of how well it will help to accomplish the goals of the study. The choice of tools and techniques is based on their demonstrated efficacy and alignment with the particular demands of NED assignments. Through the quantitative assessment method, the study guarantees a thorough evaluation of the correctness, robustness, and general efficacy of the system. This comprehensive methodological framework not only supports the validity of the results but also offers a replicable model for further NED research.

A comprehensive research methodology seeks to assure the transparency, repeatability, and credibility of the study by painstakingly recording and justifying the research design and procedures. This chapter gives a thorough justification of the selected techniques and resources, supports their suitability, and describes the procedures followed in the gathering, processing, and analysis of data. It seeks

to improve the study's scientific rigor and make the research easier to understand in its whole by addressing the validity, reliability, and ethical aspects of the research. In the end, it provides a reproducible foundation for further research and validates the research findings.

Chapter 5

Contributions and methods

5.1 Introduction

In this work, we propose a new system based on a graph approach to solve the named entity disambiguation problem. A clique algorithm is used to create our system. The clique algorithm was shown to be efficient in several domains, such as social network analysis [170], biological network analysis [171], combinatorial optimization problems [172] and heuristic development and approximation algorithms [173]. Clique partitioning algorithms are an effective tool for graph analysis, particularly with huge graphs. Although the application of clique partitioning is ultimately determined by the particular requirements of the analysis, its advantages make it a worthwhile choice for a variety of graph-related activities where they excel in three main areas: flexibility, scalability, and efficiency. These algorithms can process even the most complicated and huge datasets quickly by dividing big graphs into smaller clique-based communities. They also adapt well to different kinds of graphs and applications; some even optimizing certain attributes to enhance the identification of cohesive subgroups in the graph [174–176].

The clique approach, based on the concepts of graph theory [177], detects subsets of vertices within an undirected graph where every pair of vertices is directly connected by an edge. These subsets, known as cliques, serve as vital elements for understanding network structure and dynamics in diverse fields. These algorithms usually investigate the graph by taking into account various vertices combinations and determining if they constitute cliques. The program begins with a blank set of vertices and iteratively adds vertices to construct possible cliques, retracing its steps as needed to investigate different routes. This

process continues until all maximal cliques in the graph are identified.

5.2 Proposition of an iterative clique partitioning algorithm CPSR

In this work we introduce the Clique Partitioning based on Semantic Relatedness Algorithm (CPSR), which exploits iteratively a clique partitioning algorithm in conjunction with a semantic relatedness measure. This measure represents the consistency between mapping entities retrieved from a KB as a context-dependent feature. The concept of consistency between mapping entities retrieved implies that when entities are retrieved and mapped, the results are coherent and correct, ensuring that the same mention is accurately identified and linked correctly to its corresponding entity in the KB each time it occurs in a document.

Our algorithm aims to find an optimal sub-graph in which its nodes represent the disambiguation entities. As mentioned before, we harness the KDWD as a KB in order to build and test our NED system. Figure 5.1 depicts our entity disambiguation approach, where we recommend both collective and single disambiguation methods. Our proposed NED system is composed of the two modules as described in the following sections.

5.2.1 Clique concept

A clique in graph theory is a subset of vertices (nodes) in a graph where every vertex (node) is directly connected with an edge to every other vertex (node) in the subset. In other words, it is a complete sub-graph within a larger graph. As more formal definition, given a graph G = (V, E), where V is the set of vertices and E is the set of edges, a clique C is a subset of V such that for every pair of distinct vertices u, $v \in C$, there exists an edge $(u, v) \in E$. [178]

5.2.2 Candidate Entity Generation module

Given a set of input documents, $D = \{d_1, d_2, d_3 \ . \ . \ . \ d_n\}$, each document $d_i \in D$ contains a set of pre-tagged mentions $M = \{m_1, m_2, m_3 \ . \ . \ . \ m_k\}$, for each mention $m_i \in M$ in d_i we select its different candidate entities E_i=$\{e_{i,1}$,$e_{i,2}$,$e_{i,3}$,$e_{i,4}$,...,$e_{i,j}\}$ from the KDWD KB using the name dictionary. The

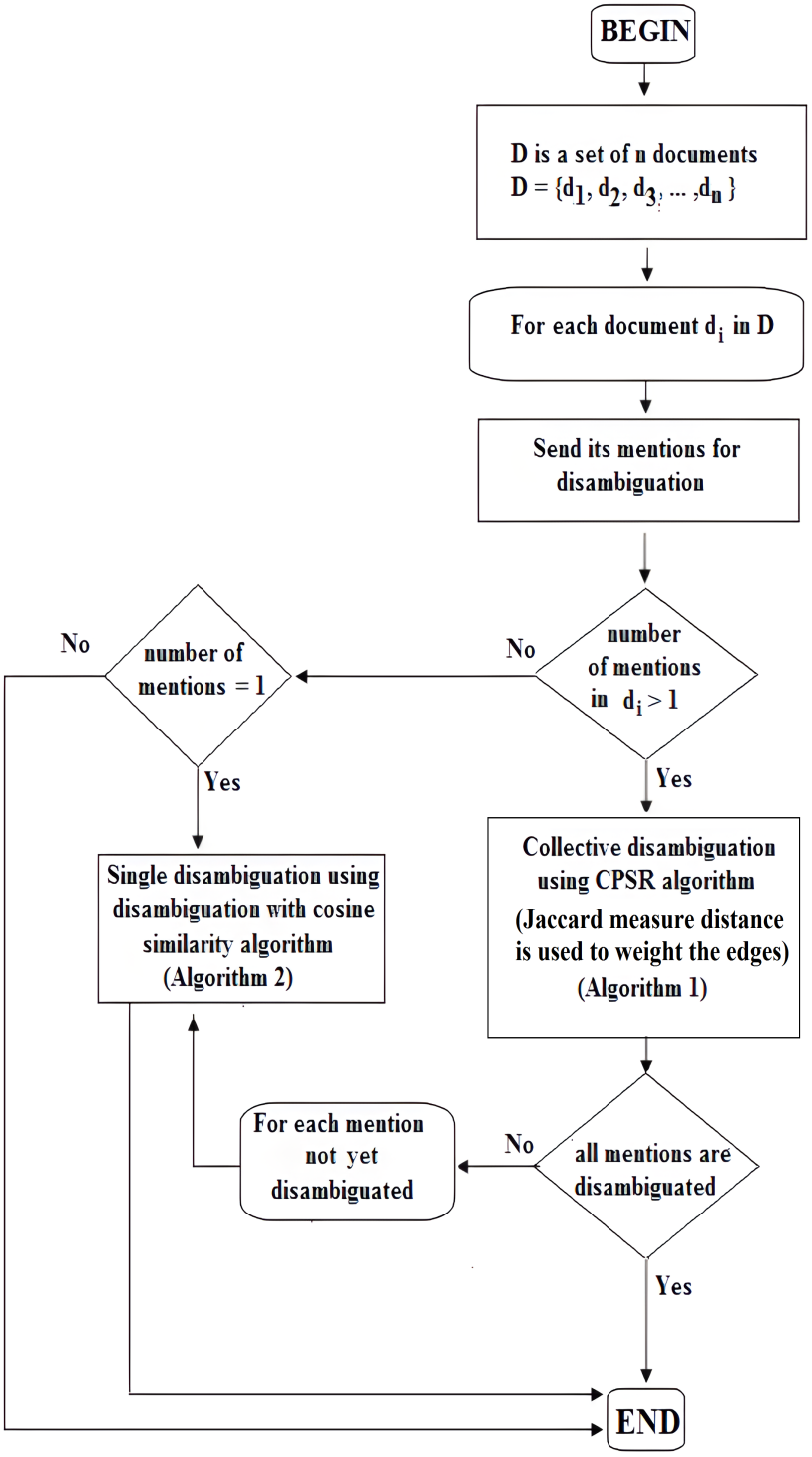

Figure 5.1: *The proposed entity disambiguation process.*

name dictionary is a key-value table that we build from the KDWD KB, where the first column of this table (en-label) contains all the mentions with their name variations that are available in the KDWD KB. The second column includes one or more pages that represent the referent entities for this mention. We note here that mentions without candidate entities are not taken into consideration in our entity disambiguation process. Besides, to generate the candidate entities, we compare the surface form of each mention in the document with the surface form of the first column of the name dictionary. Subsequently, we retrieve the second column's content as candidate entities that correspond to that mention.

5.2.3 Candidate Entity Ranking module

This second module represents our solution, which is based on a graph approach for named entity disambiguation. Our system relies on an iterative clique partitioning algorithm and a semantic relatedness measure. Our entity ranking strategy includes two main phases: the graph creation and the disambiguation algorithm (CPSR), which are presented in detail in the following subsections:

Graph creation

The candidate entities of different mentions in a document are represented as an undirected and weighted graph G = (V,E,W), where V is a set of nodes representing the candidate entities, E is a set of edges between these nodes, and W is a set of weights where each weight is calculated and assigned to its corresponding edge. To build our graph, we applied the following rules to create edges and calculate their weights. An edge is created between two entities when there is semantic relatedness between them. Semantic relatedness is a measure that represents the coherence or functional association between two concepts or words. In our study it represents the semantic coherence between two Wikipedia pages. Two Wikipedia entities are deemed to be semantically related if several Wikipedia articles link to both of them, which implies that a relation holds between them in the real world. Meanwhile, edges are not drawn between the nodes referring to the same mention. Regarding the edges weighting rule, we use **Jaccard distance measure** [8]. The Jaccard distance measure was introduced in the work titled "Etude de quelques méthodes de classification automatique" by Paul Jaccard, published in 1901. The choice of the Jaccard measure is justified as it has been

proven to be effective in many works and used particularly in Guo et al. [88] in which they achieved **89,60%** of micro precision surpassing Hoffart et al. [40] and Han et al. [11] which employed Wikipedia Link-based Measure (WLM), modeled from the Normalized Google Distance [114], and reached **81,82%** and **87%** of micro precision respectively. The Jaccard distance measure is used to calculate the topical coherence between candidate entities which is based on the link structure of Wikipedia, namely weights here express the semantic relatedness between the different candidate entities. In addition, since the same entity candidate can be found multiple times as a candidate for different textual mentions, each occurrence must be evaluated independently. Equation 5.1 calculates the Jaccard distance measure between two entities.

$$Cohj(u_1, u_2) = \frac{| U_1 \cap U_2 |}{| U_1 \cup U_2 |} \tag{5.1}$$

Where U_1 and U_2 are the sets of Wikipedia articles that are linked to u_1 and u_2 respectively.

Disambiguation Algorithm

Our CPSR algorithm, as mentioned earlier, is based on a graph approach dedicated to handle the NED problem. In this section, we describe the behavior and structure of our proposed algorithm in details. We introduce a pseudocode (script) which explains the iterations steps of the algorithm; in addition, we depict in Figure 5.2 a scenario of our CPSR algorithm for more explanations.

Given an undirected and weighted graph G(V,E,W), a clique $G_s = (V_s, E_s, W_s)$ is a sub-graph of G where $V_s \subseteq V$, $E_s \subseteq E$ and $W_s \subseteq W$. G_s is a complete sub-graph where all the nodes of the clique are linked to each other by an edge.

Our main idea is to iteratively find all the cliques in the graph and choose the clique with the highest weight, then use its nodes as the disambiguation entities. The clique's nodes represent the disambiguation entities, where each node of this clique represents the candidate entity predicted for a given mention in the text. Next, we delete all the wrong candidate entities for this textual mentions that have been previously disambiguated by the chosen clique. Afterwards, we merge all the nodes of this chosen clique into one single node in the new graph, which will be used in the next iteration as explained in Algorithm 1.

Algorithm 1 : Collective disambiguation - Clique Partitioning algorithm based on Semantic Relatedness (CPSR)

Input: Undirected and weighted graph G(V,E,W) / V is a set of candidate entities, for each edge $e \in E$, we assign an associated weight $w \in W$ which represents the Semantic Relatedness (SR) calculated using the Jaccard distance measure mentioned above (equation 1).

Output: Sub-graph, each node of this sub-graph represents the disambiguation entity of a given mention

Clique-List = find all the cliques in the graph (G).

while Clique-List is not empty **do**

1- Weight each clique by summing the SR scores of its edges.

2- Choose the highest scoring clique and use its nodes as disambiguation entities.

3- Remove all wrong candidates for any mention disambiguated in step 2.

4- Merge all nodes of the chosen clique into one single node and update G.

5- Clique-List= find all the cliques in the new graph (G).

end while

We will illustrate our algorithm using an example. Let us consider a document with six textual mentions (A, B, C, D, E, and F) and each mention has three candidate entities. The candidate entities corresponding to a given mention take the name of this mention in lowercase, concatenated with an index to distinguish between them. For example, the candidate entities of the mention A are a_1, a_2, a_3. The same process is used for the remaining candidate entities corresponding to the other mentions in the text. These candidate entities represent the nodes in our graph. An edge is created between two candidate entities of two different mentions if there is a semantic relatedness between them. The weight of the edge is calculated using the Jaccard distance measure. The candidate entities referring to the same mention are not related to each other with edges. Figure 5.2 shows the different iterations of this example using our CPSR algorithm. Cliques are shown in different line styles. The clique with the highest weight is represented in bold. In this example, in the first iteration and after the detection of all the cliques in the graph, the clique with the highest weight is the clique that encompasses

the nodes a_2, b_3, and e_2 with a weight of 80. This means that these nodes are the disambiguation entities of the mentions A, B, and E, respectively. After the disambiguation of these mentions, the system merges the nodes a_2, b_3, and e_2 into one single node and removes the rest of the candidate entities of the mentions A, B, and E from the graph, which means the system deletes a_1, a_3, b_1, b_2, e_1 and e_3. During the second iteration, the clique that encompasses the nodes c_1, d_3, and the merged nodes from the first iteration has the highest weight of 55. In this iteration, the mentions C and D are disambiguated, and c_1 and d_3 are their disambiguation entities, respectively. c_1 and d_3 will be merged with the previous disambiguation entities of the first iteration. The rest of the candidate entities of the mentions C and D, which means c_2 ,c_3, d_1, and d_2, will be deleted from the graph. While in the third iteration, there is only one clique with 18 as a weight; this clique englobs the merged nodes of the second iteration and f_3. f_3 is the disambiguation mention of F; f_3 will be merged with the previous disambiguation entities of the second iteration. The rest of the candidate entities of the mention F which means f_1, f_2 will be deleted from the graph. The algorithm comes to an end in the fourth and final iteration when all mentions in the text have been disambiguated.

As a summary of the example, in each iteration, the algorithm disambiguates collectively a set of mentions in the text until there are no more cliques in the graph and no more mentions to be disambiguated.

Moreover, our system handles documents containing only one mention. The CPSR algorithm is not dedicated to dealing with this case since it disambiguates collectively and jointly all the mentions in a document and requires at least two mentions to create the clique, which is the core of the CPSR algorithm. Therefore, we implement an alternative solution explained in Algorithm 2 that supports the single mention disambiguation problem by exploiting the cosine similarity measure. The cosine similarity measure is used to compute the similarity score between the surface form of the text mention and the surface form of the titles of its candidate entities. Here, the surface form refers to the exact spelling or written representation of a word or a phrase, which means its textual morphology. Hence, the candidate entity with the highest cosine similarity measure score is chosen as the disambiguation entity for this mention. We highlight that N-gram representation is used to represent the surface form of the mention and

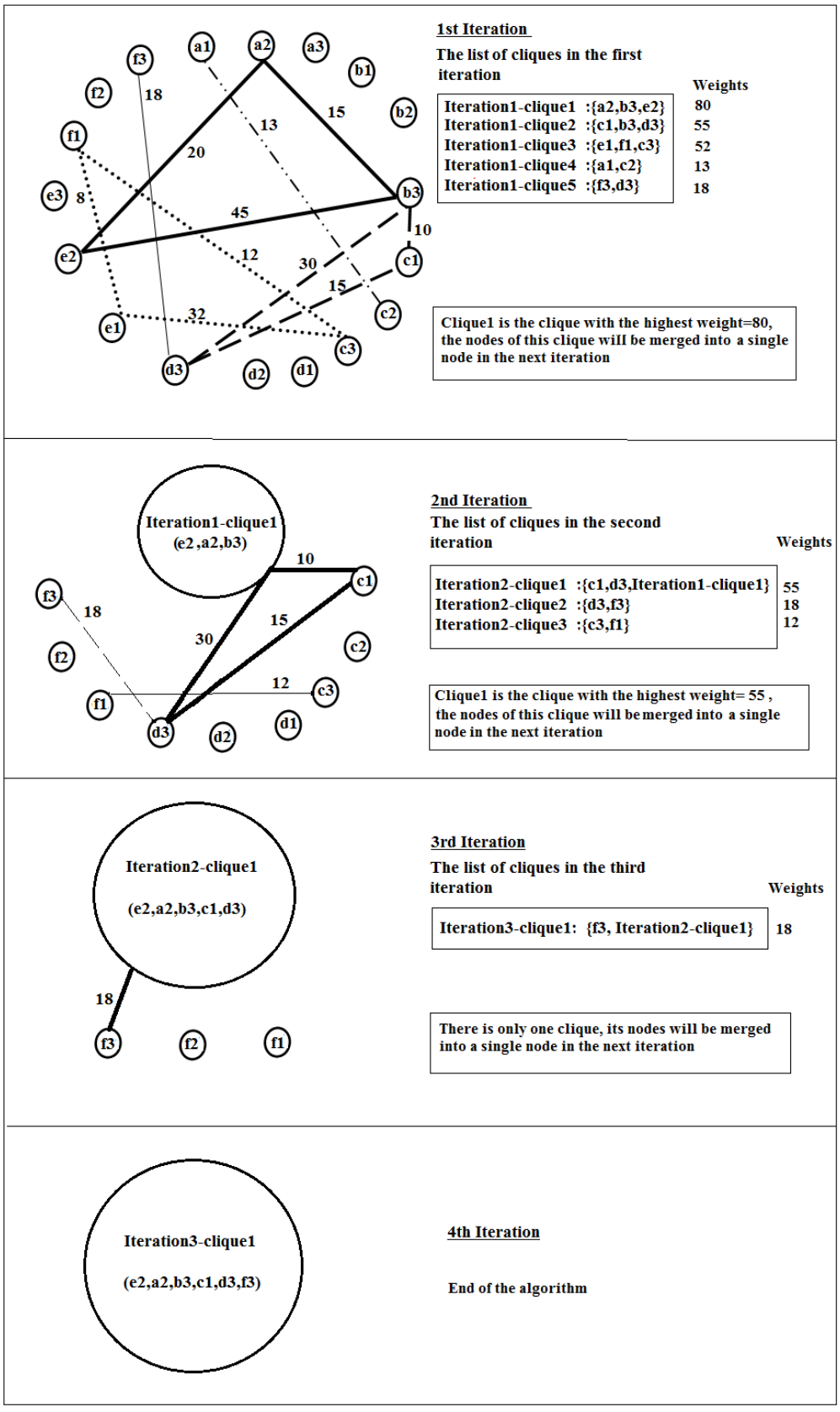

Figure 5.2: *Illustration of an execution of our CPSR algorithm on an example.*

its candidate entities titles. The N-gram representation is a method of text representation where a sequence of n consecutive words (or characters) is treated as a unit. For example, in a 2-gram (bigram) representation, the sentence "machine learning is fun to learn", will be represented by the following phrases: "machine learning", "learning is", "is fun", "fun to", "to learn". Whereas, when using a 3-gram representation , the same sentence will be presented by the following phrases: "machine learning is","learning is fun", "is fun to", "fun to learn". The same thing in a 4-gram representation, the sentence will be represented by the following phrases:"machine learning is fun","learning is fun to", "is fun to learn". The same principle is applied in 5-gram, 6-gram and others [179]. In our work, we have used 4-gram representation, additional information are provided in Chapter 6, Experimental Results.

After representing the surface form of the mention and its candidate entities titles in 4-gram representation, the next steps are: creating a vocabulary, vectorizing the sentences (creating vectors) and calculating the similarity score between the mention and each one of its candidate entities. The candidate entity with the highest cosine similarity score will be selected as the disambiguation entity for this mention [180]. The cosine similarity measure is shown in this following formula.

$$\cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\|\|\vec{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5.2}$$

This formula calculates the cosine similarity between two vectors $\vec{A}$ and $\vec{B}$, where:

- $\cos(\theta)$ represents the cosine of the angle between the two vectors.
- $\vec{A} \cdot \vec{B}$ is the dot product of the vectors.
- $\|\vec{A}\|$ and $\|\vec{B}\|$ are the magnitudes of the vectors.

The main goal of the single disambiguation is to disambiguate the mentions left by the CPSR algorithm, as well as the isolated nodes (mentions) in the graph, which means that these nodes are not connected to each other in the graph.

Algorithm 2 : Single disambiguation - Disambiguation with cosine similarity

Input: List of mentions / one mention.

Output: List of disambiguation entities / one disambiguation entity.

for each mention **do**

1- Represent the surface form of the mention and the surface form of its candidate entities titles with 4 gram representation.

2- Calculate the cosine similarity between the surface form of the mention and its candidate entities titles.

3- Choose the candidate entity with the highest similarity score as a disambiguation entity.

end for

Algorithm 3 encapsulates the entire disambiguation process as illustrated in Figure 5.1, which combines the collective and single disambiguation methods to disambiguate all the mentions in the document without omitting any.

Algorithm 3 : Global algorithm

Input: List of documents.

Output: a set of disambiguated mentions which belongs to the input documents.

L= a list of documents

for (each document d in L) **do**
  if (number of mentions in d) > 1 **then**
    step1- Disambiguate collectively these mentions using CPSR algorithm, Algorithm 1 (Collective disambiguation).
    step2-
    if not(all mentions are disambiguated) **then**
      Disambiguate the remaining mentions from step1 one by one using Algorithm 2 (Single disambiguation).
    end if
  else
    if (number of mentions = 1) **then**
      Disambiguate this mention using Algorithm 2 (Single disambiguation).
    end if
  end if
end for

Figure 5.3 provides a general representation of our system and disambiguation design; for a more in-depth analysis of each system module, see Figure 5.4.

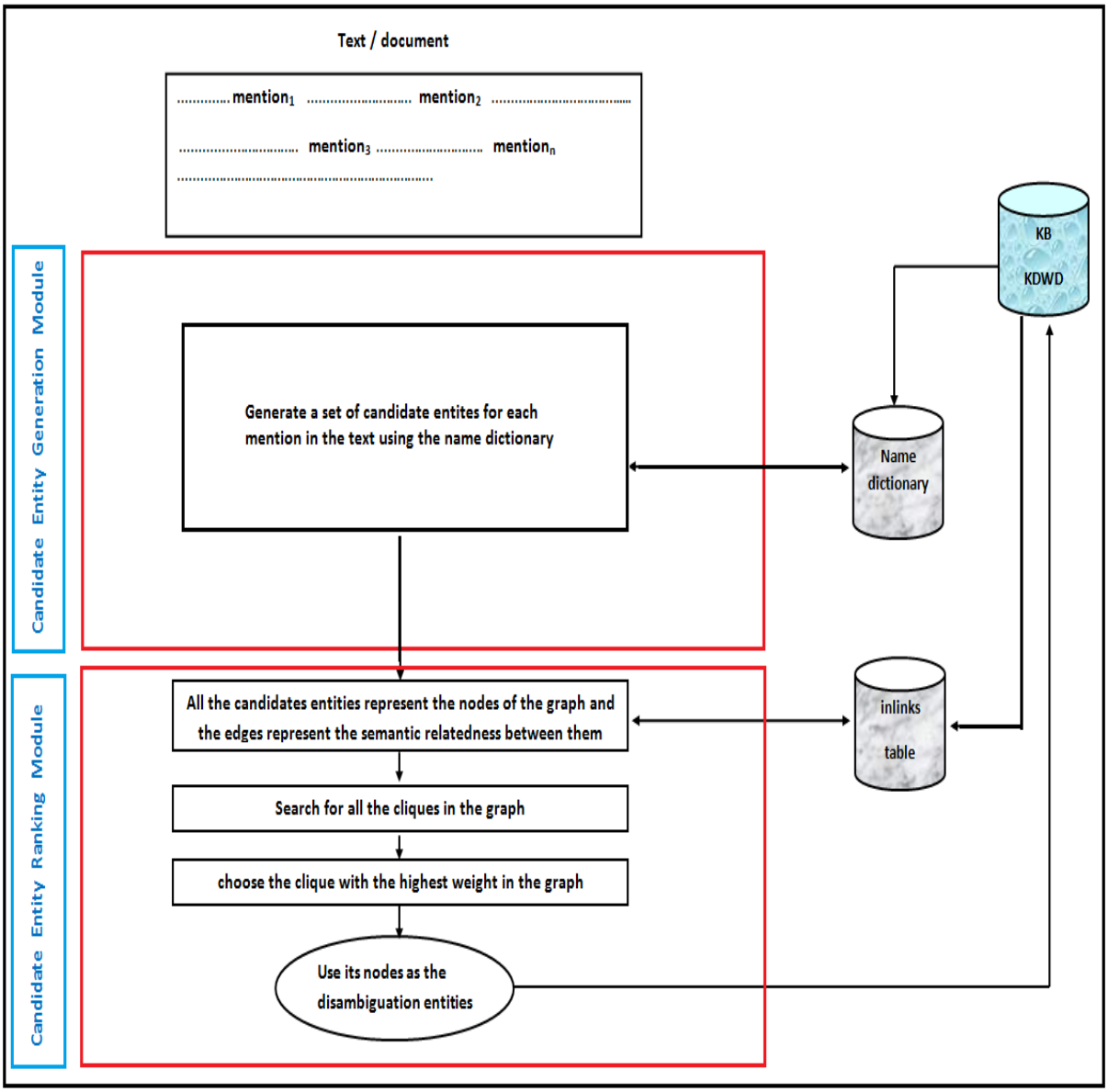

Figure 5.3: *Our system and disambiguation process design*

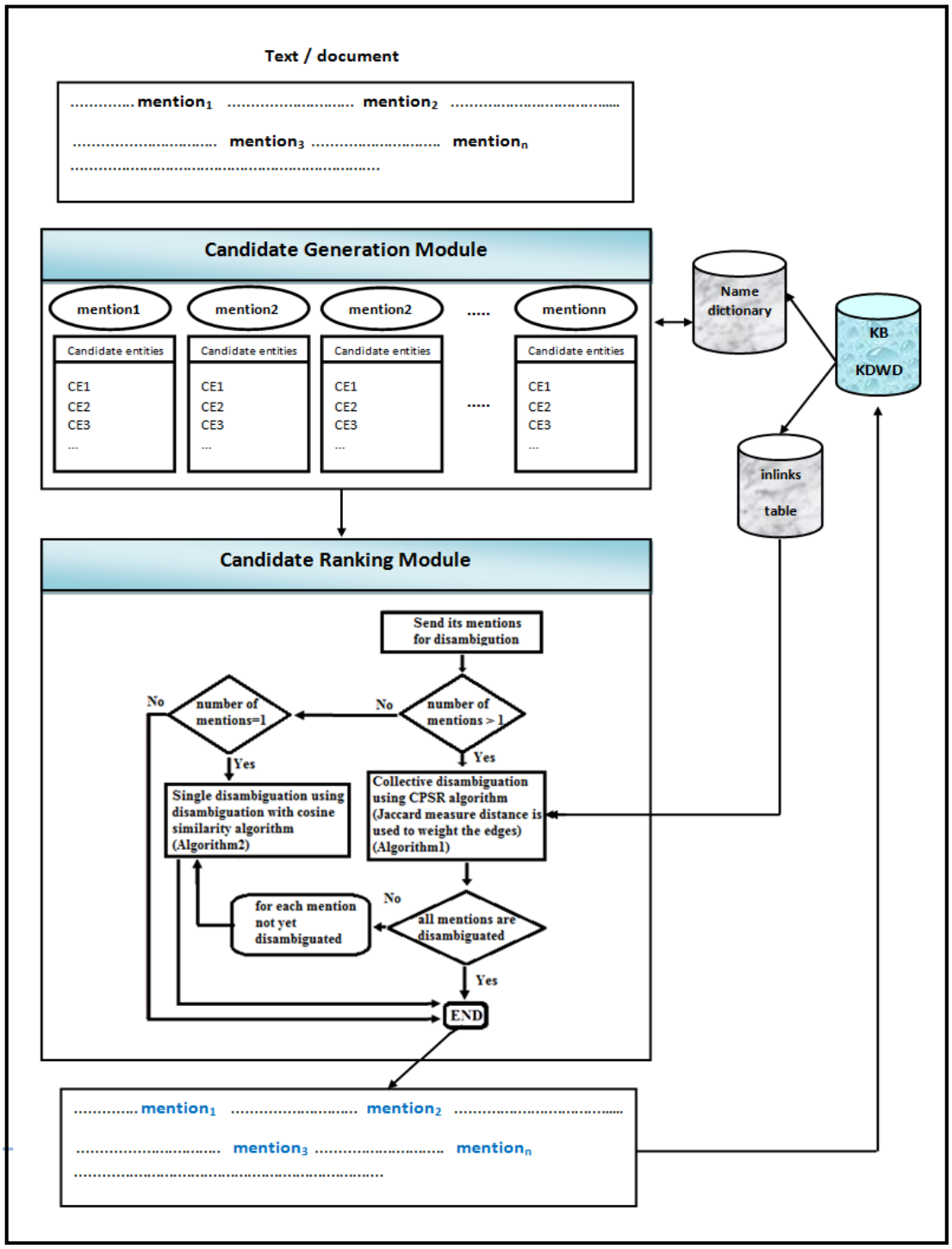

Figure 5.4: *Our system and disambiguation process design (more detailed)*

5.3 Conclusion

In summary, this chapter has introduced a new method for Named Entity Disambiguation, known as Clique Partitioning based on Semantic Relatedness (CPSR), which addresses the drawbacks of the previous approaches and introduces fresh ways to improve accuracy.

To disambiguate all of the mentions in the text collectively, our suggested method makes use of graph-based techniques namely clique partitioning. Our approach also manages the single disambiguation in the unlikely event that some mentions remain isolated and do not form part of a clique or in case a document contains only one mention. Using the **Jaccad distance measure**, we employ semantic relatedness in this work to determine the topical coherence between the various mapping entities. It is important to note that our method is unsupervised and doesn't call for any special training.

Chapter 6

Implementation and Evaluation

6.1 Introduction

This chapter describes how the experiment was carried out, where we delve into the practical parts of our study by presenting the findings from this implementation and giving a thorough rundown of the experiment's setup are the objectives.

This chapter outlines the setting of the experiment, where we have already described the environment in which the experiment was conducted, including the software configurations and the KB KDWD used to create our system including the data preparation in chapter 4 (Research Methodology).

This chapter also encompasses the description of the datasets (benchmarks) used in the evaluation of our system, the evaluation criteria and the baselines. This detailed description aims to provide a clear understanding of the context and conditions under which the experiment was carried out.

The results of the experiment are exposed in the second subsection, where we offer a thorough analysis of the data from the performance evaluation. The study's objectives are taken into consideration when analyzing the results, and significant conclusions and observations are emphasized. Overall, we provide an analysis of the consequences and efficacy of the applied solution.

6.2 Implementation

All the experimental tests were conducted on a workstation with 128 GB of RAM, 24 processors, and 1 terabyte of hard disk.

We use *Python* for coding and we leverage the **Networkx**[18] library for clique generation. This library use the function **find_cliques(G, nodes=None)**, this function returns all maximal cliques in an undirected graph. For each node n, a maximal clique for n is a largest complete subgraph containing n. The largest maximal clique is sometimes called the maximum clique. This function returns an iterator over cliques, each of which is a list of nodes. It is an iterative implementation, so should not suffer from recursion depth issues. This function accepts a list of nodes and only the maximal cliques containing all of these nodes are returned. It can considerably speed up the running time if some specific cliques are desired.

1. **Parameters:**
   - **G:NetworkX graph:** is an undirected graph.
   - **nodes:list, optional (default=None):** If provided, only yield maximal cliques containing all nodes in nodes. If nodes is not a clique itself, a *ValueError* is raised.
2. **Returns:** An iterator over maximal cliques, each of which is a list of nodes in G. If nodes is provided, only the maximal cliques containing all the nodes in nodes are returned. The order of cliques is arbitrary.

This implementation is based on the algorithm published by Bron and Kerbosch in 1973 [168], as adapted by Tomita, Tanaka and Takahashi in 2006 [181] and discussed in Cazals and Karande in 2008 [182]. This algorithm ignores self-loops and parallel edges, since cliques are not conventionally defined with such edges.

6.3 Experiments

The experimental section is divided into two subsections. The first subsection describes the setting of the experiment, and the second subsection presents the results. In particular, we describe the datasets, and the performance evaluation results.

[18] https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.clique.find _cliques.html

6.3.1 Experimental settings

In this subsection, we describe the datasets (benchmarks) commonly used in the NED domain for the evaluation of our system. This is followed by the presentation of the evaluation metrics used to evaluate the performance of our system. In addition, we present two baselines that are used in the comparison with our CPSR algorithm.

Datasets

In this section, we introduce seven well-known and publicly available datasets that are used in our evaluation. These datasets provide different characteristics in terms of surface form frequency and length of surrounding context (cf. Table 6.1). These datasets are used as benchmarks for entity disambiguation and are suitable for collective named entity disambiguation. We evaluate our system over these datasets to demonstrate its strength across different documents/datasets.

1. **AIDA/CoNLL-TestB:** This corpus is manually annotated by Hoffart et al. [40] and comprises 231 documents with an average of 19.4 mentions per document.

2. **IITB:** This corpus gathers 103 manually annotated documents. Each document contains, on average, 109.1 mentions collected from the web sites belonging to different domains. This dataset is characterized by its density (highest entity/document of all corpora). This corpus was developed by Kulkarni et al. [28].

3. **AQUAINT:** This corpus consists of 50 documents, and each document contains 14.5 mentions on average. The documents originate from different news services, from Xin-hua News Service, the New York Times, and Associated Press news corpus, and was created by Milne and Witten [12].

4. **MSNBC:** This corpus was presented in 2007 by Cucerzan [10] and contains 20 news documents with an average of 32.9 mentions per document.

5. **ACE2004:** This corpus is a subset of ACE2004 co-reference documents of Ratinov et al. [13] annotated by Amazon Mechanical Turk. It has 35 documents, with an average of 7.3 mentions per document.

6. **Wiki:** This corpus is extracted from Wikipedia in Gabrilovich et al. [183] and contains 320 documents with an average of 21.3 mentions per document.

7. **Cweb:** This corpus is extracted from ClueWeb by Guo and Barbosa [164] and contains 320 documents with an average of 34.8 mentions per document.

Table 6.1 Test datasets (benchmarks) description.

Datasets	# Doc	# Mentions	Men/Doc
AIDA/CoNLL-TestB	231	4485	19.4
IITB	103	11245	190.1
AQUAINT	50	726	14.5
MSNBC	20	658	32.9
ACE2004	35	257	7.3
Wiki	320	6821	21.3
Cweb	320	11154	34.8

Evaluation Criteria

In our work, we use **Accuracy** because our system ensures the disambiguation all the mentions detected in the document since the mentions that remain from the collective disambiguation CPSR (Algorithm 1) are disambiguated using the single disambiguation (Algorithm 2). In our case, the entity mentions that should be linked are given as an input to our system; therefore, the number of linked mentions generated by the system in Equation 2.5 equals the number of entity mentions that should be linked in Equation 2.8. Which means in our work, Accuracy = Precision = Recall= F1 measure. Precisely, we used Micro-averaged accuracy represented in Equation 2.14 and Macro-averaged accuracy represented in Equation 2.15 explained in Chapter 2, Evaluation Criteria section.

Baselines

As baselines, we compared our work with two simple baselines proposed by Chen et al. [169] since they also worked on the KDWD KB. These baselines do not require any treatment; they just select the most popular candidate entity as a disambiguation entity for the mention. In the first baseline, the popularity feature

represents the number of views of the predicted entity, which is a statistical value retrieved from the KDWD KB, while in the second baseline, the popularity feature represents the number of directed in-links. The disambiguation decision is made in the two baselines directly by choosing the candidate entity with the highest popularity feature score.

6.3.2 Experimental results

This section presents the experimental results, where we first evaluate the performance of our CPSR system on eight samples that we have generated from the KDWD KB. Then, we tested our system on seven well-known datasets that are widely used in the ED domain. We then compared our system with the two baselines mentioned in the previous section as well as with several annotators from the state of the art, including graph-based systems. Regarding the single mention disambiguation problem and after multiple quick tests, we tuned the N-gram representation to 4-gram because it returns a significant difference in cosine similarity measures between the candidate entities of a mention. Therefore, 4-gram representation leads to more accurate prediction.

Experimental results on our KDWD test dataset

Our first experiment focused on eight samples that we prepared from the KDWD KB as explained in Chapter 4 Data collection and preparation section. To test the performance of our system, we carried out a series of experiments on these eight samples. Table 6.2 shows the details of the different samples in terms of the number of documents, the number of mentions, and the performance results expressed by the Micro and Macro-accuracy of our CPSR system.

Table 6.2 Micro and Macro accuracy of our CPSR system over the eight KDWD test samples.

Samples	# Doc	# Ment	Micro_acc	Macro_acc
Sample 1	30	423	**90.56%**	**87.70%**
Sample 2	29	235	**88.51%**	**82.82%**
Sample 3	30	282	**89%**	**86.15%**
Sample 4	28	303	**92.73%**	**85.54%**
Sample 5	29	311	**91.96%**	**88.02%**
Sample 6	30	632	**96.51%**	**84.18%**
Sample 7	44	432	**90.27%**	**89.79%**
Sample 8	17	238	**92.01%**	**79.40%**
All Samples	237	2856	**92.01%**	**91.44%**

The experimental results show the good performance of our CPSR system. We observe that we achieved an average of **92.01%** and **91.44%** in terms of micro and macro accuracy, respectively, on the whole dataset samples. From these two average measures, we observe that micro and macro accuracy results are close which indicates a good balance between the documents of the sample and also it indicates that the model is performing consistently. These results are a good performance indicator for our CPSR system. Despite the high micro-accuracy reported on sample 6 by our CPSR system, where we register **96.51%**, the macro-accuracy does not exceed **85%**, which expresses a poor balance between micro and macro accuracy, in which we report a difference of more than **12%**. The same result was also obtained in sample 8. Besides, samples 2 and 4 acquired a moderate balance, where we report a difference of **7%** between micro and macro accuracy. On samples 1, 3, 5, and 7, the balance is reasonable, where we recorded a difference of less than **4%** in the worst case and inferior to **1%** in sample 7 as the best case.

Experimental results on the seven benchmarks datasets

Furthermore, we tested our system on seven widely used datasets in the named entity disambiguation domain, namely: AIDA/CoNNL2003-TestB, IITB, MSNBC, AQUAINT, ACE2004, Cweb, and Wiki. From Table 6.3, we observe that our algorithm gave overall good performance on the seven datasets in terms of micro and macro accuracy. The algorithm achieved an average of **92,05%** and

91.39% of micro and macro-accuracy, respectively. In particular, we recorded our best score on the AIDA/CoNLL2003-TestB dataset with **97.07%** for micro-accuracy. The second best score of **96.58%** of micro-accuracy was achieved with the MSNBC dataset. We also obtained great results on the Wiki, ACE2004, and Cweb datasets with **95.23%**, **92.94%** and **91.26%** of micro-accuracy, respectively. Whereas, compared to the previous datasets, we report slightly better results over the IITB and AQUAINT datasets with **83.96%** and **87.37%** of micro-accuracy, respectively.

Table 6.3 Results of Micro and Macro-accuracy obtained by our CPSR system on the seven datasets

DataSet	**Mico_acc**	**Macro_acc**
AIDA/CoNLL2003-TestB	97.07	96.91
IITB	83.96	83.97
ACE2004	92.94	90.11
AQUAINT	87.37	86.46
MSNBC	96.58	95.43
Wiki	95.23	95.23
Cweb	91.26	91.68
Avg	**92.05**	**91.39**

Comparison of our CPSR system against other annotators

This section is dedicated to the comparison of our CPSR system against several annotators, such as the baselines explained previously, state-of-the art systems, and also graph-based approaches.

a. Comparison with the baselines

We compared our method with the baselines that have been described in the previous section:

Baseline 1: The model with highest page views as popularity.

Baseline 2: The model with most links directed popularity.

The experimental results of the CPSR system on our test dataset show a significant improvement in performance compared to the results of the two baselines.

We note that we recorded **92.01%** for micro-accuracy while baseline1 and baseline2 reached **61.83%** and **59.13%** respectively, a difference of over **30%** was reported. Table 6.4 shows the results in terms of micro-accuracy.

Table 6.4 Comparison results with the baselines.

Systems	Micro-Accuracy	Macro-Accuracy
Baseline1	61.83	/
Baseline2	59.13	/
CPSR (ours)	**92.01**	**91.44**

b. Comparison with state-of-the art approaches

In this part, we introduce a second comparison between our CPSR system and 38 former state-of-the-art systems from SNED and CNED approaches since 2007 until 2023, including some well known and prominent works like Hoffart et al. [40], Han et al. [11], Yang et al. [61], Fang et al. [60] and Yamada et al. [62]. In this section, we offer three comparison tables: one using AIDA/CoNLL2003-TestB dataset, one using the IITB dataset, and one using five other datasets, namely MSNBC, AQUAINT, ACE2004, Cweb, and Wiki.

Table 6.5 shows the comparison results that were conducted on the AIDA/ CoNLL 2003-TestB dataset against 32 annotators, and the results are expressed in terms of a micro-F1 score. Micro-F1 score, also denoted as F1 @MI, is defined by the Equation 2.12 and is discussed in more detail in Chapter 2, in the Evaluation Criteria section. When compared to other annotators, our CPSR system's performance results show a significant improvement. It is important to note that we achieved a micro-F1 score of **97.07%**, whereas the second-ranked system of Yang et al. [59] only scored **95.90%**. Also, we found that our CPRS system differed by **35%** from the system that came in last.

Table 6.5 Comparison results against state-of-the-art annotators on AIDA/ CoNLL 2003-TestB. The best value in bold and second is underlined.

System	F1 @MI
Cucerzan(2007) [10]	74,00
Milne and Witten (2008) [12]	68,00
Han et al.(2011) [11]	62,00
Hoffart et al.(2011) [40]	81,82
Usbeck et al.(2014) [184]	55,00
Alhelbawy and Gaizauskas (2014) (clique) [2]	86,11
Alhelbawy and Gaizauskas (2014) (PR) [2]	87,59
Yamada et al.(2016) [58]	91,50
Ganea and Hofmann (2017) [47]	92,22
Guo and Barbosa (2018) [164]	89,00
Cao et al. (2018) [130]	80,00
Le and Titov (2018) [185]	93,07
Raiman and Raiman (2018) [186]	94,88
Yang et al. (2018) [59]	<u>95,90</u>
Fang et al. (2019) [60]	94,30
Shahbazi et al. (2019) [154]	93.46
Le and Titov (2019) [126]	89,66
Yamada et al. (2019) [62]	95,04
Yang et al. (2019) (DCA-SL) [61]	94,64
Yang et al. (2019) (DCA-RL) [61]	93.73
Hu et al. (2020) [128]	92,40
Chen et al. (2020) [50]	93,54
Mulang et al. (2020) [51]	94,94
De Cao et al. (2020) [53]	93,30
Xin et al. (2021) [129]	92,02
Ravi et al. (2021) [52]	83,10
De Cao et al. (2021) [54]	85,5
Barba et al. (2022) [55]	92,60
Ayoola et al. (2022) [63]	90,40
Yamada et al. (2022) [65]	95,00
Ji et al. (2023) [66]	92,09
Atzeni et al. (2023) [56]	93,70
CPSR (ours)	**97,07**

Moreover, we performed another comparison of our system against other annotators on the IITB dataset. We have noticed from the literature that there is a limited number of systems that have used the IITB dataset to evaluate their performances. This is the main reason why we compared our system with only a restricted number of annotators (four annotators). It is important to note that our algorithm performed better when the prior and later datasets were used rather than the IITB dataset. Nevertheless, our CPSR system performed better than the state-of-the-art competitors. Indeed, we report **83.96%** of micro_F1 score, while the second ranked system achieved only **74.10%** of micro_F1 score. The difference in performance between our CPSR system and the second-ranked one is more than **9%** which is a substantial improvement. The results of the comparison using the IITB dataset are given in Table 6.6.

Table 6.6 Comparison results against annotators using IITB dataset

Systems	F1 @MI
Han et al. (2011) [11]	73.0
Hulpuş et al. (2015) [187]	71.7
Ganea et al. (2016) [156]	62,47
Zwicklbauer et al. (2016) [161]	74.10
Our system (CPSR)	**83.96**

To enrich our work, we have added other experiments on several datasets, namely MSNBC, AQUAINT, ACE2004, Cweb, and Wiki. We compared our CPSR system with 21 former systems (most of them are mentioned in the AIDA/ CoNLL2003- TestB comparison and some are new). The performance results are communicated in Table 6.7 in terms of micro and macro F1 scores.

Table 6.7 Comparison results against state-of-the-art annotators on MSNBC, AQUAINT, ACE2004, Cweb and Wiki datasets. The best value in bold and second is underlined.

	Datasets									
	MSNBC		AQUAINT		ACE2004		Cweb		Wiki	
Systems	**F1 @MI**	**F1 @MA**	**F1 @MI**	**F1 @MA**	**F1 @MI**	**F1 @MA**	**F1 @MI**	**F1 @MA**	**F1 @MI**	**F1 @MA**
Cucerzan (2007) [10]	88.34	87.76	78.67	78.22	79.30	78.22	_	_	_	_
Milne and Witten (2008) [12]	78.43	80.37	85.13	84.84	81.29	84.25	64.10	_	81.70	_
Ratinov et al. (2011) [13]	75.37	75.37	83.14	82.97	81.91	83.18	56.20	_	67.20	_
Cheng and Roth (2013) [188]	90.22	90.87	87.72	**87.74**	86.60	87.13	_	_	_	_
Ganea and Hofmann (2017) [47]	93.70	_	88.50	_	88.50	_	77.90	_	77.50	_
Phan et al. (2017) [189]	91.80	_	_	_	92.90	_	_	_	_	_
Yang et al. (2018) [59]	92.60	_	89.90	_	88.50	_	81.80	_	79.20	_
Le and Titov (2018) [185]	93.90	_	88.30	_	89.90	_	77.50	_	78.00	_
Fang et al. (2019) [60]	92.80	_	87.50	_	91.20	_	78.50	_	82.80	_
Shahbazi et al. (2019) [154]	92.30	_	90.10	_	88.70	_	78.40	_	79.80	_
Yang et al. (2019)(DCA-SL) [61]	94.57	_	87.38	_	89.44	_	73.47	_	78.16	_
Yang et al. (2019)(DCA-RL) [61]	93.80	_	88.30	_	90.10	_	75.60	_	78.80	_
Yamada et al. (2019) [62]	96.30	_	**93.50**	_	91.90	_	78.90	_	89.10	_
Chen et al. (2020) [50]	93.40	_	89.80	_	88.90	_	77.90	_	80.01	_
De Cao et al. (2020) [53]	94.30	_	89.90	_	90.10	_	77.30	_	87.40	_
Ravi et al. (2021) [52]	83.40	_	76.80	_	86.80	_	_	_	_	_
Barba et al. (2022) [55]	94.70	_	91.60	_	91.80	_	77.70	_	88.80	_
Ayoola et al. (2022) [63]	94.80	_	92.60	_	93.60	_	78.20	_	90.40	_
Yamada et al. (2022) [65]	96.30	_	93.50	_	91.90	_	78.90	_	89.10	_
Ji et al. (2023) [66]	93.30	_	**94.23**	_	93.48	_	80.81	_	87.48	_
Atzeni et al. (2023) [56]	94.60	_	91.30	_	**95.00**	_	78.20	_	85.90	_
Our system (CPSR)	**96.58**	**95.43**	87.37	86.46	92.94	**90.11**	**91.26**	**91.68**	**95.23**	**95.23**

From Table 6.7, we note that our CPRS system outperforms all the other systems on the MSNBC, Cweb, and Wiki datasets in terms of micro and macro F1 score, where we reached **96.58%** of micro_F1 measure on the MSNBC dataset, thereby slightly exceeding the second-best system of Yamada et al. [62] and Yamada et al. [65] in which they report **96.30%** of micro_F1. On the Cweb dataset, **91.26%** of micro_F1 measure was registered with a margin of more than **9%** from the second-best system of Yang et al. [59] where they obtained **81.80%** of micro_F1 measure. We recorded a difference of around **5%** against the second-best system of Ayoola et al. [63] on the Wiki dataset, where we reached **95.23%** of the micro_F1 measure, whereas Ayoola et al. [63] achieved **90.40%** of the micro_F1 measure.
Furthermore, using the ACE2004 dataset, Atzeni et al. [56] came in first with **95.00%** of the micro_F1 measure, followed by Ayoola et al. [63], where they achieved **93.60%** of the micro_F1 measure, and our system was ranked in the fourth position with **92.94%** of micro_F1 measure.

In contrast, on the AQUAINT dataset, our system does not perform as well as on the previous datasets, and it came in at the sixteenth position with **87.37%** of micro_F1 measure, while Ji et al. [66] is ranked first with **94.23%** of micro_F1 measure and Yamada et al. [62] and Yamada et al. [65] came in at the second position with **93.50%** of micro_F1 measure.

c. Comparison with graph-based approaches

In this sub-section, we report the results of our system against 15 annotators of graph-based approaches. Our aim here is to emphasize the contributions of our system with respect to graph-based approaches and show the achieved performance improvement.

At first, we start by comparing our system against Alhelbawy and Gaizauskas [2] work, where they introduce two approaches for NED. The first approach is based on the Page Rank algorithm (we call it PR), and the second, on which our work is based, uses clique partitioning (we call it Clique). In their second approach, they suggested collectively disambiguating all the mentions in the document using an undirected and unweighted graph, in which they assigned a confidence score to the nodes (vertices) that represent candidate entities. This score is calculated for each candidate entity separately from the other candidate entities in the document. Only the textual context of the mention was utilized

to generate this score (along with cosine similarity and Jaro-Winkler similarity, as well as entity popularity). They employed a clique partitioning technique to discover the most weighted clique, where they weighted each clique by adding the confidence scores of all its nodes. The nodes of the most weighted clique were then used to disambiguate the mentions. In our study, we employed an undirected but weighted graph, where we weighted the edges with a Jaccard similarity score rather than affecting scores to the nodes. The Jaccard similarity measure indicates the semantic relatedness between two candidate entities and the strength of their relationship, thus using a context-dependent feature that represents the coherence between the mapping entities. Unlike Alhelbawy and Gaizauskas [2] where they used only context-independent features, these features just rely on the surface form of the entity mention and the candidate entity's knowledge and are not related to the context in which the entity mention appears. Table 6.8 presents the key differences between our approach and that of Alhelbawy and Gaizauskas, where they adopt a clique-based method.

Table 6.8 Comparison table between our system and Alhelbawy and Gaizauskas's system (based on clique)

Alhelbawy and Gaizauskas's system (based on clique) [2]	**Our system**
Unweighted graph	Weighted graph
Affect a score to the nodes	Affect a score to the edges
Use a combination of measures (cosine similarity, Jaro Winkler and entity popularity) to score the nodes	Use Jaccard distance measure to score the edges
Leverage context-independent features	Leverage context-dependent features

From the experimental results shown in Table 6.9, we can notice that we achieved **97.07%** of micro-F1 measure against **86.11%** for the second approach (Clique) of Alhelbawy and Gaizauskas [2] on the AIDA/CoNLL2003-TestB dataset. Therefore, we increased the micro-F1 score by **10.96%**. We also exceeded their first approach (PR) in which they recorded **87.59%** of the micro-F1 score. We consider these results a huge enhancement achieved by our CPSR system within graph-based approaches.

Table 6.9 Comparison results against Alhelbawy and Gaizauskas (2014) [2] work on AIADA/CoNLL2003-TestB.

Systems	**F1 @MI**
Alhelbawy and Gaizauskas (2014)(PR) [2]	87.59
Alhelbawy and Gaizauskas (2014)(Clique) [2]	86.11
Our system (CPSR)	**97.07**

Second, we carried out a comparison of our CPSR algorithm against several graph-based approaches. The comparison was conducted on the five datasets used in the previous comparison with the state-of-the-art systems, namely MSNBC, AQUAINT, ACE2004, Cweb, and Wiki datasets. Besides, we selected fourteen graph-based annotators such as Hoffart et al. [40], Cao et al. [130], and Le and Titov [185]. Table 6.10 captures the significant performance of our system in almost all the datasets in terms of micro and macro-F1 measures.

Table 6.10 Comparison results against graph-based annotators on MSNBC, AQUAINT, ACE2004, Cweb, and Wiki datasets. The best value in bold and second is underlined.

	Datasets									
	MSNBC		AQUAINT		ACE2004		Cweb		Wiki	
Systems	**F1 @MI**	**F1 @MA**	**F1 @MI**	**F1 @MA**	**F1 @MI**	**F1 @MA**	**F1 @MI**	**F1 @MA**	**F1 @MI**	**F1 @MA**
Han et al. (2011) [11]	88.46	87.93	79.46	78.80	73.48	66.80	61.00	_	78.00	_
Hoffart et al. (2011) [40]	78.81	76.26	56.47	56.46	80.49	84.13	58.6	_	63	_
Usbeck et al. (2014) [184]	_	_	73	59.90	66	78	_	_	_	_
Guo and Barbosa (2014) [158]	91.37	91.73	90.74	**90.58**	87.68	89.23	_	_	_	_
Hulpuş et al. (2015) [187]	_	_	73.3	_	_	_	_	_	_	_
Ganea et al. (2016) [156]	91.06	91.19	89.27	88.94	88.71	88.46	_	_	_	_
Gong et al. (2017) [162]	85.44	_	86.78	_	86.73	_	_	_	90.14	_
Guo and Barbosa (2018) [164]	92	91	90	90	88	89	78	_	85	_
Cao et al. (2018) [130]	_	_	87	88	88	89	_	_	86	_
Le and Titov (2019) [126]	92.20	_	90.70	_	88.10	_	78.20	_	81.70	_
Xue et al. (2019) [127]	94.43	_	**91.94**	_	90.64	_	79.65	_	85.47	_
Parravicini et al. (2019) [190]	92	_	86	_	83	_	_	_	_	_
Hu et al. (2020) [128]	95.50	_	91.60	_	90.14	_	77.50	_	78.50	_
Xin et al. (2021) [129]	94.32	_	90.75	_	92.92	_	77.91	_	76.24	_
Our system (CPSR)	**96.58**	**95.43**	87.37	86.46	**92.94**	**90.11**	**91.26**	**91.68**	**95.23**	**95.23**

It is worth noting here that we achieved our best score of **96.58%** of micro_F1 on the MSNBC dataset, thus slightly surpassing Hu et al.'s system [128] with **1%**. We reached **92.94%** of micro_F1 whereas Xin et al. [129], as a second-ranked system, obtained **92.92%** on ACE2004. However, on the Cweb dataset, we achieved much better results with an improvement of more than **11.5%** where we reached **91.26%** of micro_F1 against Xue et al. [127] with **79.65%** which represents a considerable improvement. We also acquired a good enhancement on the Wiki dataset where we reached **95.23%** of micro_F1 compared to the second ranked system of Gong et al. [162] who reached **90.14%**. Therefore, we obtained more than **5%** improvement on the F1 score. However, we noticed that our CPSR method performed slightly less on the ACQUAINT dataset. Where our system was ranked in the eighth position with **87.37%** of micro_F1 score, while Xue et al.'s system [127] came in the first position with **91.94%** of micro_F1 score, and Hu et al.'s system [128] came in the second position with a score of **91.60%**.

6.3.3 Discussion

In this study, we tested the performance of our system on seven datasets as well as on our own test dataset, which is derived from the KDWD KB. We also supported our study with a comparison of our CPSR system against 41 annotators from the literature. At first, we tested the performance of our system on the dataset that we prepared from the KDWD KB. Our dataset is characterized by an average of 12.05 mentions per document, with 2856 mentions over all the test samples. The experimental results were very significant, where we achieved **92.01%** and **91.44%** of micro and macro accuracy, respectively. We also compared our system with two baselines that leverage the KDWD. Similarly, our system outperformed largely these two baselines with an improvement of more than 30% in micro-accuracy.

Furthermore, we carried out a second experiment on seven datasets commonly used in the NED domain, namely: AIDA /CoNLL2003-TestB, IITB, MSNBC,AQUAINT, ACE2004, Cweb, and Wiki. We compared our proposed CPSR solution against different ED systems taken from the state-of-the-art review and then with some other approaches based on graphs. From the experimental results, the improvement of our CPSR system is considerable in almost all the datasets used in this study.

Some particularly satisfying results were observed on the AIDA/CoNLL2003-TestB dataset, where we achieved our best scores of **97.07%** and **96.91%** of the micro and macro F1 scores, respectively. Moreover, our great enhancement was reached notably on Cweb and Wiki datasets against state-of-the-art systems, where we reported a gain of more than **9%** around **5%** over the two datasets, respectively, while we obtained more than **11%** and **5%** of improvement against the graph-based systems.

Whereas in the AQUAINT dataset, our CPSR system offers modest results compared to other systems where we recorded a difference of more than **6%** and **4%** of micro-F1 score with the first ranked system of the state-of the art proposed by Ji et al. [66] and the first one in graph-based approaches proposed by Xue et al. [127] respectively. Regarding the IITB dataset, our system scored **83.96 %** of the micro-F1 score. Despite the slight decrease of performance results reported by our system on the IITB dataset compared to its results with the other datasets, our system still outperforms its competitors on the IITB dataset. This decline may be attributed to the higher density of mentions in the IITB dataset, with an average of 190.1 mentions per document. It is also possible that many documents in both AQUAINT and IITB contain only a single mention, implying that the single disambiguation component is solely responsible for disambiguation. This suggests that collective disambiguation is not occurring, and in such cases, the single disambiguation component might not perform as effectively.

It is worth noting that our system coped really well with the following datasets: AIDA/CoNLL2003-TestB, MSNBC, ACE2004, Cweb, and Wiki, where we obtained excellent results (more than **90%**) and we were ranked first against the state-of-the-art systems and fourth on ACE2004. Our system doesn't cope with AQUAINT dataset; even though we obtained **87.37%** of micro-F1 measure, we were ranked in the sixteenth position against the state-of-the-art systems and in the eighth position against the graph-based systems.

The creation of the name dictionary was the most challenging and time-consuming task; some documents are not suitable for the graph approach disambiguation. In such a case, the system detects a clique that contains a small number of mentions, and thus it disambiguates only a few mentions collectively instead of disambiguating the maximum number of mentions in the document collectively. Therefore, a large number of remaining mentions were disambiguated singly using Algorithm 2 (disambiguation using cosine similarity).

6.4 Conclusion

Our Named Entity Disambiguation system is fully implemented in this chapter. The hardware, software, datasets, and environment used in the experiment are the first things we lay out. We can clearly see the testing ground for our NED system thanks to this methodical approach. Once this thorough configuration was complete, we showed the outcomes that our system has produced.

We proved our NED system's efficacy in a range of situations by means of an extensive performance assessment. Results showed that our technique is highly accurate and robust, with notable advantages over the state-of-the-art systems in terms of accuracy.

Our NED approach has the ability to tackle the difficulties of entity disambiguation in a variety of datasets, as demonstrated by its effective installation and encouraging outcomes. The knowledge gathered from this chapter provides a strong basis for the system's future development and practical implementation.

In the next and last chapter, we will talk about the consequences of these findings, and look at possible directions for further research and development.

Chapter 7

Conclusion and perspectives

In this work, we propose a new system for collective entity disambiguation. We introduced the CPSR algorithm based on clique partitioning and semantic relatedness. Our model does not require any training, therefore it is registered under the unsupervised category. It disambiguates mentions in the texts by leveraging the KDWD KB, from which we constructed our own test dataset. Our system disambiguates all the mentions in the document since the remaining mentions from the collective disambiguation are disambiguated one by one using the single disambiguation.

The experimental results demonstrated the effectiveness and strength of our algorithm across a wide range of entity disambiguation datasets, comparing it against 41 annotators from the literature. It was found that semantic relatedness plays a very important role in the disambiguation process and offers better results compared to the methods that use only the context-independent features. The cliques helped to find the maximum number of mentions that have the highest level of semantic relatedness between them. Although graph-based approaches are computationally expensive since the graph may contain hundreds of nodes for documents with multiple mentions, nevertheless, they produced the best results.

Based on these experiments, semantic relatedness used in conjunction with the graph-based method has proved that it has a great impact on the NED systems' performance and dramatically increases their accuracy. However, the only drawback is the size of the graph, which requires high memory to avoid longer execution times.

In future works, we would like to refine and annotate our test dataset that we prepared from the KDWD KB and make it publicly available so it can be used as

a benchmark for NED. We also plan to use different measures to weight the edges of our graph, like the Wikipedia Link-based Measure (WLM), which is modeled from the Normalized Google Distance [114] and Point-wise Mutual Information measure (PMI-like) [191] to calculate the topical coherence between Wikipedia entities and evaluate their effectiveness within our CPSR algorithm. Also, we aim to use Word2vec to capture the semantic relatedness between the different candidate entities.

And for a more thorough analysis, we intend to test our system to determine the number of iterations it needs, the average duration of each iteration, the maximum and minimum execution times of an iteration, the total execution time, and the frequency of calls to each branch of our algorithm regarding to the KDWD KB. We also plan to carry out additional tests, particularly focusing on the Single Disambiguation component, to investigate the performance decline on AQUAINT and IITB. Although our system achieved over 80% of accuracy on these datasets, this is considered a decrease compared to the results on the other datasets where we achieved more than 90% of accuracy.

Moreover, we plan to accommodate additional languages to our system and give special attention to execution time and make sure to decrease it as much as possible. Furthermore, we plan to employ energy-efficient techniques as an alternative to enhance the performance of our model [192–194]. With all these perspectives, we intend to employ Large Language Models (LLMs) [195], which are revolutionizing the entity linking field with their ability to comprehend context, handle massive volumes of data, and learn from extensive datasets.

Bibliography

[1] Gabriel Altay, "Introducing the kensho derived wikimedia dataset," 2020, https://blog.kensho.com/announcing-the-kensho-derived-wikimedia-dataset-5d1197d72bcf/, Last accessed on 2020-02-03.

[2] A. Alhelbawy and R. Gaizauskas, "Collective named entity disambiguation using graph ranking and clique partitioning approaches," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 1544–1555.

[3] K. Breitman, M. A. Casanova, and W. Truszkowski, *Semantic web: concepts, technologies and applications*. Springer Science & Business Media, 2007.

[4] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 34–43, 2001.

[5] A. R. Pal and D. Saha, "Word sense disambiguation: A survey," *arXiv preprint arXiv:1508.01346*, 2015.

[6] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[7] S. B. Kotsiantis, I. Zaharakis, P. Pintelas *et al.*, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[8] W. Shen, J. Wang, and J. Han, "Entity linking with a knowledge base: Issues, techniques, and solutions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 2, pp. 443–460, 2015.

[9] D. Petkova and W. B. Croft, "Proximity-based document representation for named entity retrieval," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 731–740.

[10] S. Cucerzan, "Large-scale named entity disambiguation based on wikipedia data." in *EMNLP-CoNLL*, vol. 7, 2007, pp. 708–716.

[11] X. Han, L. Sun, and J. Zhao, "Collective entity linking in web text: a graph-based method," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 765–774.

[12] D. Milne and I. H. Witten, "Learning to link with wikipedia," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 509–518.

[13] L. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to wikipedia," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 1375–1384.

[14] M. Kobayashi and K. Takeda, "Information retrieval on the web," *ACM computing surveys (CSUR)*, vol. 32, no. 2, pp. 144–173, 2000.

[15] Y. Zhang, M. Chen, and L. Liu, "A review on text mining," in *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2015, pp. 681–685.

[16] H. Ji and R. Grishman, "Knowledge base population: Successful approaches and challenges," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics, 2011, pp. 1148–1158.

[17] R. Kumari and S. K. Srivastava, "Machine learning: A review on binary classification," *International Journal of Computer Applications*, vol. 160, no. 7, 2017.

[18] H. Li, "Learning to rank for information retrieval and natural language processing," *Synthesis lectures on human language technologies*, vol. 4, no. 1, pp. 1–113, 2011.

[19] N. Chater and C. D. Manning, "Probabilistic models of language processing and acquisition," *Trends in cognitive sciences*, vol. 10, no. 7, pp. 335–344, 2006.

[20] V. B. Vasudeva Varma, S. Kovelamudi, P. Bysani, K. K. N. Santosh GSK, K. Reddy, K. Kumar, and N. Maganti, "Iiit hyderabad at tac 2009," in *Proceedings of the Second Text Analysis Conference (TAC 2009), Gaithersburg, Maryland, USA, November*, 2009.

[21] W. Zhang, J. Su, C. L. Tan, and W. T. Wang, "Entity linking leveraging: automatically generated annotation," in *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 1290–1298.

[22] W. Zhang, C. L. Tan, Y. C. Sim, and J. Su, "Nus-i2r: Learning a combined system for entity linking." in *TAC*, 2010.

[23] J. Lehmann, S. Monahan, L. Nezda, A. Jung, and Y. Shi, "Lcc approaches to knowledge base population at tac 2010." in *TAC*, 2010.

[24] S. Monahan, J. Lehmann, T. Nyberg, J. Plymale, and A. Jung, "Cross-lingual cross-document coreference with entity linking." in *TAC*, 2011.

[25] Z. Chen and H. Ji, "Collaborative ranking: A case study on entity linking," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 771–781.

[26] A. Pilz and G. Paaß, "From names to entities using thematic context distance," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 857–866.

[27] R. C. Bunescu and M. Pasca, "Using encyclopedic knowledge for named entity disambiguation." in *Eacl*, vol. 6, 2006, pp. 9–16.

[28] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, "Collective annotation of wikipedia entities in web text," in *Proceedings of the 15th*

ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009, pp. 457–466.

[29] Z. Zheng, F. Li, M. Huang, and X. Zhu, "Learning to link entities with knowledge base," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 483–491.

[30] W. Zhang, Y. C. Sim, J. Su, and C. L. Tan, "Entity linking with effective acronym expansion, instance selection, and topic modeling," in *IJCAI*, vol. 2011, 2011, pp. 1909–1914.

[31] W. Shen, J. Wang, P. Luo, and M. Wang, "Linden: linking named entities with knowledge base via semantic knowledge," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 449–458.

[32] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin, "Entity disambiguation for knowledge base population," in *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 277–285.

[33] W. Shen, J. Wang, P. Luo, and M. Wang, "Liege: link entities in web lists with knowledge base," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1424–1432.

[34] T.-Y. Liu, *Learning to rank for information retrieval*. Springer Science & Business Media, 2011.

[35] W. Zhang, C. L. Tan, J. Su, B. Chen, W. Wang, Z. Toh, Y. Sim, Y. Cao, and C.-Y. Lin, "I2r-nus-msra at tac 2011: Entity linking." in *TAC*. Citeseer, 2011.

[36] P. McNamee, "Hltcoe efforts in entity linking at tac kbp 2010." in *TAC*, 2010.

[37] G. Limaye, S. Sarawagi, and S. Chakrabarti, "Annotating and searching web tables using entities, types and relationships," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1338–1347, 2010.

[38] X. Han and L. Sun, "A generative entity-mention model for linking entities with knowledge base," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 945–954.

[39] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 469–478.

[40] 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 *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 782–792.

[41] W. Shen, J. Wang, P. Luo, and M. Wang, "Linking named entities in tweets with knowledge base via user interest modeling," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 68–76.

[42] G. Tang, Y. Guo, D. Yu, and E. Xun, "A hybrid re-ranking method for entity recognition and linking in search queries," in *Natural Language Processing and Chinese Computing*. Springer, 2015, pp. 598–605.

[43] Z. Chen, S. Tamang, A. Lee, X. Li, W.-P. Lin, M. Snover, J. Artiles, M. Passantino, and H. Ji, "Cuny-blender tac-kbp2010," 2010.

[44] S. Gottipati and J. Jiang, "Linking entities to a knowledge base with query expansion," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 804–813.

[45] D. M. Nemeskey, G. A. Recski, A. Zséder, and A. Kornai, "Budapestacad at tac 2010," 2010.

[46] Z. Zheng, F. Li, M. Huang, and X. Zhu, "Learning to link entities with knowledge base," in *Human Language Technologies: The 2010 Annual*

Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010, pp. 483–491.

[47] O.-E. Ganea and T. Hofmann, "Deep joint entity disambiguation with local neural attention," *arXiv preprint arXiv:1704.04920*, 2017.

[48] A. Sil, G. Kundu, R. Florian, and W. Hamza, "Neural cross-lingual entity linking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[49] Z. Yao, L. Cao, and H. Pan, "Zero-shot entity linking with efficient long range sequence modeling," *arXiv preprint arXiv:2010.06065*, 2020.

[50] S. Chen, J. Wang, F. Jiang, and C.-Y. Lin, "Improving entity linking by modeling latent entity type information," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 7529–7537.

[51] I. O. Mulang, K. Singh, C. Prabhu, A. Nadgeri, J. Hoffart, and J. Lehmann, "Evaluating the impact of knowledge graph context on entity disambiguation models," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2157–2160.

[52] M. P. K. Ravi, K. Singh, I. O. Mulang, S. Shekarpour, J. Hoffart, and J. Lehmann, "Cholan: A modular approach for neural entity linking on wikipedia and wikidata," *arXiv preprint arXiv:2101.09969*, 2021.

[53] N. De Cao, G. Izacard, S. Riedel, and F. Petroni, "Autoregressive entity retrieval," *arXiv preprint arXiv:2010.00904*, 2020.

[54] N. De Cao, W. Aziz, and I. Titov, "Highly parallel autoregressive entity linking with discriminative correction," *arXiv preprint arXiv:2109.03792*, 2021.

[55] E. Barba, L. Procopio, and R. Navigli, "Extend: extractive entity disambiguation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 2478–2488.

[56] M. Atzeni, M. Plekhanov, F. A. Dreyer, N. Kassner, S. Merello, L. Martin, and N. Cancedda, "Polar ducks and where to find them: Enhancing

entity linking with duck typing and polar box embeddings," *arXiv preprint arXiv:2305.12027*, 2023.

[57] P. Sen, "Collective context-aware topic models for entity disambiguation," in *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 729–738.

[58] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji, "Joint learning of the embedding of words and entities for named entity disambiguation," *arXiv preprint arXiv:1601.01343*, 2016.

[59] Y. Yang, O. Irsoy, and K. S. Rahman, "Collective entity disambiguation with structured gradient tree boosting," *arXiv preprint arXiv:1802.10229*, 2018.

[60] Z. Fang, Y. Cao, Q. Li, D. Zhang, Z. Zhang, and Y. Liu, "Joint entity linking with deep reinforcement learning," in *The World Wide Web Conference*, 2019, pp. 438–447.

[61] X. Yang, X. Gu, S. Lin, S. Tang, Y. Zhuang, F. Wu, Z. Chen, G. Hu, and X. Ren, "Learning dynamic context augmentation for global entity linking," *arXiv preprint arXiv:1909.02117*, 2019.

[62] I. Yamada, K. Washio, H. Shindo, and Y. Matsumoto, "Global entity disambiguation with pretrained contextualized embeddings of words and entities," *arXiv preprint arXiv:1909.00426*, 2019.

[63] T. Ayoola, J. Fisher, and A. Pierleoni, "Improving entity disambiguation by reasoning over a knowledge base," *arXiv preprint arXiv:2207.04106*, 2022.

[64] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[65] I. Yamada, K. Washio, H. Shindo, and Y. Matsumoto, "Global entity disambiguation with bert," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 3264–3271.

[66] B. Hui, L. Zhang, Y. Nian, K. Yan, and J. Chen, "A multi-angle bidirectional interaction model for entity linking," *Available at SSRN 4200929*, 2023.

[67] R. Grishman, "Information extraction: Techniques and challenges," in *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology: International Summer School, SCIE-97 Frascati, Italy, July 14–18, 1997*. Springer, 1997, pp. 10–27.

[68] J. O'Connor and I. McDermott, *NLP*. Thorsons, 2001.

[69] A. M. N. Allam and M. H. Haggag, "The question answering systems: A survey," *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, vol. 2, no. 3, 2012.

[70] T. Cheng, X. Yan, and K. C.-C. Chang, "Entityrank: searching entities directly and holistically," in *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007, pp. 387–398.

[71] G. Demartini, T. Iofciu, and A. P. De Vries, "Overview of the inex 2009 entity ranking track." in *INEX*, vol. 9. Springer, 2009, pp. 254–264.

[72] K. Balog, P. Serdyukov, and A. P. d. Vries, "Overview of the trec 2010 entity track," NORWEGIAN UNIV OF SCIENCE AND TECHNOLOGY TRONDHEIM, Tech. Rep., 2010.

[73] I. Bordino, Y. Mejova, and M. Lalmas, "Penguins in sweaters, or serendipitous entity search on user-generated content," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2013, pp. 109–118.

[74] M. Mitra and B. Chaudhuri, "Information retrieval from documents: A survey," *Information retrieval*, vol. 2, pp. 141–163, 2000.

[75] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan, "A survey of web information extraction systems," *IEEE transactions on knowledge and data engineering*, vol. 18, no. 10, pp. 1411–1428, 2006.

[76] T. Lin, O. Etzioni *et al.*, "Entity linking at web scale," in *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale*

Knowledge Extraction. Association for Computational Linguistics, 2012, pp. 84–88.

[77] N. Nakashole, G. Weikum, and F. Suchanek, "Patty: A taxonomy of relational patterns with semantic types," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 1135–1145.

[78] C. Welty, J. W. Murdock, A. Kalyanpur, and J. Fan, "A comparison of hard filters and soft evidence for answer typing in watson," in *International Semantic Web Conference*. Springer, 2012, pp. 243–256.

[79] H. Ji and R. Grishman, "Knowledge base population: Successful approaches and challenges," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 1148–1158.

[80] P. McNamee, H. T. Dang, H. Simpson, P. Schone, and S. M. Strassel, "An evaluation of technologies for knowledge base population." in *LREC*. Citeseer, 2010.

[81] P. McNamee and H. T. Dang, "Overview of the tac 2009 knowledge base population track," in *Text Analysis Conference (TAC)*, vol. 17, 2009, pp. 111–113.

[82] J. Lehmann, S. Monahan, L. Nezda, A. Jung, and Y. Shi, "Lcc approaches to knowledge base population at tac 2010." in *TAC*, 2010.

[83] Z. Chen, S. Tamang, A. Lee, X. Li, W.-P. Lin, M. Snover, J. Artiles, M. Passantino, and H. Ji, "Cuny-blender tac-kbp2010," 2010.

[84] X. Han and J. Zhao, "Nlpr_kbp in tac 2009 kbp track: A two-stage method to entity linking." in *TAC*, 2009.

[85] R. Sharnagat, "Named entity recognition: A literature survey," 2014.

[86] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE transactions on knowledge and data engineering*, vol. 34, no. 1, pp. 50–70, 2020.

[87] P. Sun, X. Yang, X. Zhao, and Z. Wang, "An overview of named entity recognition," in *2018 International Conference on Asian Language Processing (IALP)*. IEEE, 2018, pp. 273–278.

[88] S. Guo, M.-W. Chang, and E. Kiciman, "To link or not to link? a study on end-to-end tweet entity linking." in *HLT-NAACL*, 2013, pp. 1020–1030.

[89] A. Gattani, D. S. Lamba, N. Garera, M. Tiwari, X. Chai, S. Das, S. Subramaniam, A. Rajaraman, V. Harinarayan, and A. Doan, "Entity extraction, linking, classification, and tagging for social media: a wikipedia-based approach," *Proceedings of the VLDB Endowment*, vol. 6, no. 11, pp. 1126–1137, 2013.

[90] O. Deshpande, D. S. Lamba, M. Tourn, S. Das, S. Subramaniam, A. Rajaraman, V. Harinarayan, and A. Doan, "Building, maintaining, and using knowledge bases: a report from the trenches," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. ACM, 2013, pp. 1209–1220.

[91] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008, pp. 1247–1250.

[92] X. Zou, "A survey on application of knowledge graph," in *Journal of Physics: Conference Series*, vol. 1487, no. 1. IOP Publishing, 2020, p. 012016.

[93] M. Fabian, K. Gjergji, and W. Gerhard, "Yago: A core of semantic knowledge unifying wordnet and wikipedia," in *16th International World Wide Web Conference, WWW*, 2007, pp. 697–706.

[94] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*. Springer, 2007, pp. 722–735.

[95] F. Wu and D. S. Weld, "Automatically refining the wikipedia infobox ontology," in *Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 635–644.

[96] W. Cohen, P. Ravikumar, and S. Fienberg, "A comparison of string metrics for matching names and records," in *Kdd workshop on data cleaning and object consolidation*, vol. 3, 2003, pp. 73–78.

[97] V. Varma, P. Bysani, K. Reddy, V. B. Reddy, S. Kovelamudi, S. R. Vaddepally, R. Nanduri, N. K. Kumar, S. Gsk, and P. Pingali, "Iiit hyderabad in guided summarization and knowledge base population," *International Institute of Information Technology*, 2010.

[98] W. Zhang, Y. C. Sim, J. Su, and C. L. Tan, "Nus-i2r: Learning a combined system for entity linking," in *Proc. TAC 2010 Workshop*, 2010.

[99] W. Zhang, Y.-C. Sim, J. Su, and C.-L. Tan, "Entity linking with effective acronym expansion, instance selection and topic modeling," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[100] S. Monahan, J. Lehmann, T. Nyberg, J. Plymale, and A. Jung, "Cross-lingual cross-document coreference with entity linking." in *TAC*, 2011.

[101] M. Dredze, P. McNamee, D. Rao, A. Gerber, T. Finin *et al.*, "Entity disambiguation for knowledge base population," in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010.

[102] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 469–478.

[103] T. Lin, O. Etzioni *et al.*, "Entity linking at web scale," in *Proceedings of the joint workshop on automatic knowledge base construction and web-scale knowledge extraction (akbc-wekex)*, 2012, pp. 84–88.

[104] W. Shen, J. Wang, P. Luo, and M. Wang, "Linking named entities in tweets with knowledge base via user interest modeling," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 68–76.

[105] T. Štajner and D. Mladenić, "Entity resolution in texts using statistical learning and ontologies," in *The Semantic Web: Fourth Asian Conference, ASWC 2009, Shanghai, China, December 6-9, 2009. Proceedings 4*. Springer, 2009, pp. 91–104.

[106] P. Ferragina and U. Scaiella, "Tagme: on-the-fly annotation of short text fragments (by wikipedia entities)," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 1625–1628.

[107] X. Han and L. Sun, "An entity-topic model for entity linking," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 105–115.

[108] X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu, "Entity linking for tweets." in *ACL (1)*, 2013, pp. 1304–1311.

[109] D. M. Nemeskey, G. A. Recski, A. Zséder, and A. Kornai, "Budapestacad at tac 2010," 2010.

[110] S. Cucerzan, "Tac entity linking by performing full-document entity extraction and disambiguation." in *TAC*, 2011.

[111] Z. Chen, S. Tamang, A. Lee, X. Li, W.-P. Lin, M. Snover, J. Artiles, M. Passantino, and H. Ji, "Cunyblender tac-kbp2010 entity linking and slot filling system description," in *Proc. TAC 2010 Workshop*, 2010.

[112] J. Lehmann, S. Monahan, L. Nezda, A. Jung, and Y. Shi, "Lcc approaches to knowledge base population at tac 2010." in *TAC*, 2010.

[113] I. H. Witten and D. N. Milne, "An effective, low-cost measure of semantic relatedness obtained from wikipedia links," 2008.

[114] R. L. Cilibrasi and P. M. Vitanyi, "The google similarity distance," *IEEE Transactions on knowledge and data engineering*, vol. 19, no. 3, pp. 370–383, 2007.

[115] O. P. Damani, "Improving pointwise mutual information (pmi) by incorporating significant co-occurrence," *arXiv preprint arXiv:1307.0596*, 2013.

[116] P. Jaccard, "Étude comparative de la distribution florale dans une portion des alpes et des jura," *Bull Soc Vaudoise Sci Nat*, vol. 37, pp. 547–579, 1901.

[117] Y. Li, C. Wang, F. Han, J. Han, D. Roth, and X. Yan, "Mining evidences for named entity disambiguation," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1070–1078.

[118] R. Al-Halimi, R. C. Berwick, J. Burg, M. Chodorow, C. Fellbaum, J. Grabowski, S. Harabagiu, M. A. Hearst, G. Hirst, D. A. Jones *et al.*, "Wordnet an electronic lexical database," *MA: MIT Press, Cambridge*, p. 422, 1998.

[119] P. Xia, L. Zhang, and F. Li, "Learning similarity with cosine similarity ensemble," *Information sciences*, vol. 307, pp. 39–52, 2015.

[120] K. W. Church, "Word2vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.

[121] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[122] T. Mickus, D. Paperno, M. Constant, and K. Van Deemter, "What do you mean, bert? assessing bert as a distributional semantics model," *arXiv preprint arXiv:1911.05758*, 2019.

[123] W.-H. Chong, E.-P. Lim, and W. Cohen, "Collective entity linking in tweets over space and time," in *European Conference on Information Retrieval*. Springer, 2017, pp. 82–94.

[124] H. Wang, J. G. Zheng, X. Ma, P. Fox, and H. Ji, "Language and domain independent entity linking with quantified collective validation," in *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP2015)*, 2015.

[125] Y. Cao, L. Hou, J. Li, and Z. Liu, "Neural collective entity linking," *arXiv preprint arXiv:1811.08603*, 2018.

[126] P. Le and I. Titov, "Boosting entity linking performance by leveraging unlabeled documents," *arXiv preprint arXiv:1906.01250*, 2019.

[127] M. Xue, W. Cai, J. Su, L. Song, Y. Ge, Y. Liu, and B. Wang, "Neural collective entity linking based on recurrent random walk network learning," *arXiv preprint arXiv:1906.09320*, 2019.

[128] L. Hu, J. Ding, C. Shi, C. Shao, and S. Li, "Graph neural entity disambiguation," *Knowledge-Based Systems*, vol. 195, p. 105620, 2020.

[129] K. Xin, W. Hua, Y. Liu, and X. Zhou, "Log: a locally-global model for entity disambiguation," *World Wide Web*, vol. 24, pp. 351–373, 2021.

[130] Y. Cao, L. Hou, J. Li, and Z. Liu, "Neural collective entity linking," *arXiv preprint arXiv:1811.08603*, 2018.

[131] P. N. Mendes, J. Daiber, M. Jakob, and C. Bizer, "Evaluating dbpedia spotlight for the tac-kbp entity linking task," in *Proceedings of the TAC-KBP 2011 Workshop*, vol. 116, 2011, pp. 118–120.

[132] A. Alhelbawy and R. Gaizauskas, "Named entity based document similarity with svm-based re-ranking for entity linking," in *International Conference on Advanced Machine Learning Technologies and Applications*. Springer, 2012, pp. 379–388.

[133] K. Nebhi, "Named entity disambiguation using freebase and syntactic parsing," in *Proceedings of the First International Workshop on Linked Data for Information Extraction (LD4IE 2013) co-located with the 12th International Semantic Web Conference (ISWC 2013)*. Gentile, AL; Zhang, Z.; d'Amato, C. & Paulheim, H., 2013.

[134] G. Pink, A. Naoum, W. Radford, W. Cannings, J. Nothman, D. Tse, and J. R. Curran, "Sydney cmcrc at tac 2013." in *TAC*, 2013.

[135] W. Radford, W. Cannings, J. Nothman, D. Tse, J. R. Curran, A. Naoum, and G. Pink, "(almost) total recall-sydney cmcrc at tac 2012." in *TAC*, 2012.

[136] A. Barrena, E. Agirre, B. Cabaleiro, A. Penas, and A. Soroa, ""one entity per discourse" and "one entity per collocation" improve named-entity disambiguation," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 2260–2269.

[137] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes, "Improving efficiency and accuracy in multilingual entity extraction," in *Proceedings of the 9th International Conference on Semantic Systems*, 2013, pp. 121–124.

[138] E. Agirre and A. Soroa, "Personalizing pagerank for word sense disambiguation," in *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 2009, pp. 33–41.

[139] X. Han and L. Sun, "A generative entity-mention model for linking entities with knowledge base," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 945–954.

[140] A. Chisholm and B. Hachey, "Entity disambiguation with web links," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 145–156, 2015.

[141] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, and X. Wang, "Modeling mention, context and entity with neural networks for entity disambiguation." in *IJCAI*, vol. 15, 2015, pp. 1333–1339.

[142] N. Lazic, A. Subramanya, M. Ringgaard, and F. Pereira, "Plato: A selective context model for entity resolution," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 503–515, 2015.

[143] M. Francis-Landau, G. Durrett, and D. Klein, "Capturing semantic similarity for entity linking with convolutional neural networks," *arXiv preprint arXiv:1604.00734*, 2016.

[144] Y. Li, S. Tan, H. Sun, J. Han, D. Roth, and X. Yan, "Entity disambiguation with linkless knowledge bases," in *Proceedings of the 25th international conference on world wide web*, 2016, pp. 1261–1270.

[145] J. Zhang, Y. Cao, L. Hou, J. Li, and H.-T. Zheng, "Xlink: An unsupervised bilingual entity linking system," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Springer, 2017, pp. 172–183.

[146] Y. Eshel, N. Cohen, K. Radinsky, S. Markovitch, I. Yamada, and O. Levy, "Named entity disambiguation for noisy text," *arXiv preprint arXiv:1706.09147*, 2017.

[147] E. Inan and O. Dikenelli, "A sequence learning method for domain-specific entity linking," in *Proceedings of the Seventh Named Entities Workshop*, 2018, pp. 14–21.

[148] R. Usbeck, M. Röder, A.-C. Ngonga Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann *et al.*, "Gerbil: general entity annotator benchmarking framework," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 1133–1143.

[149] D. Mueller and G. Durrett, "Effective use of context in noisy entity linking," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1024–1029.

[150] P. Radhakrishnan, P. Talukdar, and V. Varma, "Elden: Improved entity linking using densified knowledge graphs," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1844–1853.

[151] H. Shahbazi, X. Z. Fern, R. Ghaeini, C. Ma, R. Obeidat, and P. Tadepalli, "Joint neural entity disambiguation with output space search," *arXiv preprint arXiv:1806.07495*, 2018.

[152] G. Kundu, A. Sil, R. Florian, and W. Hamza, "Neural cross-lingual coreference resolution and its application to entity linking," *arXiv preprint arXiv:1806.10201*, 2018.

[153] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[154] H. Shahbazi, X. Z. Fern, R. Ghaeini, R. Obeidat, and P. Tadepalli, "Entity-aware elmo: Learning contextual entity representation for entity disambiguation," *arXiv preprint arXiv:1908.05762*, 2019.

[155] M. Shirakawa, H. Wang, Y. Song, Z. Wang, K. Nakayama, T. Hara, and S. Nishio, "Entity disambiguation based on a probabilistic taxonomy," *Microsoft Research, Seattle, WA, USA, Tech. Rep. MSR-TR-2011-125*, 2011.

[156] O.-E. Ganea, M. Ganea, A. Lucchi, C. Eickhoff, and T. Hofmann, "Probabilistic bag-of-hyperlinks model for entity linking," in *Proceedings of the 25th International Conference on World Wide Web*, 2016, pp. 927–938.

[157] C. B. El Vaigh, F. Goasdoué, G. Gravier, and P. Sébillot, "A novel path-based entity relatedness measure for efficient collective entity linking," in *International Semantic Web Conference*. Springer, 2020, pp. 164–182.

[158] Z. Guo and D. Barbosa, "Robust entity linking via random walks," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 499–508.

[159] M. Pershina, Y. He, and R. Grishman, "Personalized page rank for named entity disambiguation," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 238–243.

[160] H. Huang, L. Heck, and H. Ji, "Leveraging deep neural networks and knowledge graphs for entity disambiguation," *arXiv preprint arXiv:1504.07678*, 2015.

[161] S. Zwicklbauer, C. Seifert, and M. Granitzer, "Robust and collective entity disambiguation through semantic embeddings," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 425–434.

[162] J. Gong, C. Feng, Y. Liu, G. Shi, and H. Huang, "Collective entity linking on relational graph model with mentions," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Springer, 2017, pp. 159–171.

[163] W. Lu, Y. Zhou, H. Lu, P. Ma, Z. Zhang, and B. Wei, "Boosting collective entity linking via type-guided semantic embedding," in *National CCF Conference on Natural Language Processing and Chinese Computing*. Springer, 2017, pp. 541–553.

[164] Z. Guo and D. Barbosa, "Robust named entity disambiguation with random walks," *Semantic Web*, vol. 9, no. 4, pp. 459–479, 2018.

[165] W. Zeng, X. Zhao, J. Tang, and H. Shang, "Collective list-only entity linking: A graph-based approach," *IEEE Access*, vol. 6, pp. 16 035–16 045, 2018.

[166] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.

[167] N. K. Denzin and Y. S. Lincoln, *The Sage handbook of qualitative research*. sage, 2011.

[168] J. W. Creswell and J. D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.

[169] W. Chen, D. Hathout, D. Zheng, and T. Yoo, "Context-based entity linking using kdwd, Fact Sheet N°282," 2020, https://towardsdatascience.com/context-based-entity-linking-using-kdwd-69a633f9e4e7/, Last accessed on 2020-05-09.

[170] T. Gschwind, S. Irnich, F. Furini, R. W. Calvo *et al.*, "Social network analysis and community detection by decomposing a graph into relaxed cliques," Tech. Rep., 2015.

[171] B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, and T. Vicsek, "Cfinder: locating cliques and overlapping modules in biological networks," *Bioinformatics*, vol. 22, no. 8, pp. 1021–1023, 2006.

[172] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, "The maximum clique problem," *Handbook of Combinatorial Optimization: Supplement Volume A*, pp. 1–74, 1999.

[173] E. Marchiori, "A simple heuristic based genetic algorithm for the maximum clique problem," in *Symposium on Applied Computing: Proceedings of the 1998 ACM symposium on Applied Computing*, vol. 27. Citeseer, 1998, pp. 366–373.

[174] M. Ovelgönne, "Scalable algorithms for community detection in very large graphs," Ph.D. dissertation, Karlsruhe, Karlsruher Institut für Technologie (KIT), Diss., 2011, 2011.

[175] R. Gao, S. Li, X. Shi, Y. Liang, and D. Xu, "Overlapping community detection based on membership degree propagation," *Entropy*, vol. 23, no. 1, p. 15, 2020.

[176] M. Bloznelis and V. Kurauskas, "Large cliques in sparse random intersection graphs," *The Electronic Journal of Combinatorics*, vol. 24, no. 2, pp. P2–5, 2017.

[177] J. A. Bondy and U. S. R. Murty, *Graph theory*. Springer Publishing Company, Incorporated, 2008.

[178] ——, *Graph theory*. Springer Publishing Company, Incorporated, 2008.

[179] G. Kondrak, "N-gram similarity and distance," in *International symposium on string processing and information retrieval*. Springer, 2005, pp. 115–126.

[180] F. Rahutomo, T. Kitasuka, M. Aritsugi *et al.*, "Semantic cosine similarity," in *The 7th international student conference on advanced science and technology ICAST*, vol. 4, no. 1. University of Seoul South Korea, 2012, p. 1.

[181] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," *Theoretical computer science*, vol. 363, no. 1, pp. 28–42, 2006.

[182] F. Cazals and C. Karande, "A note on the problem of reporting maximal cliques," *Theoretical computer science*, vol. 407, no. 1-3, pp. 564–568, 2008.

[183] E. Gabrilovich, M. Ringgaard, and A. Subramanya, "Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0)," 06 2013.

[184] R. Usbeck, A.-C. Ngonga Ngomo, M. Roder, D. Gerber, S. A. Coelho, S. Auer, and A. Both, "Agdistis-graph-based disambiguation of named entities using linked data," in *International Semantic Web Conference*. Springer, 2014, pp. 457–471.

[185] P. Le and I. Titov, "Improving entity linking by modeling latent relations between mentions," *arXiv preprint arXiv:1804.10637*, 2018.

[186] J. Raiman and O. Raiman, "Deeptype: multilingual entity linking by neural type system evolution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[187] I. Hulpuş, N. Prangnawarat, and C. Hayes, "Path-based semantic relatedness on linked data and its use to word and entity disambiguation," in *International Semantic Web Conference*. Springer, 2015, pp. 442–457.

[188] X. Cheng and D. Roth, "Relational inference for wikification," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1787–1796.

[189] M. C. Phan, A. Sun, Y. Tay, J. Han, and C. Li, "Neupl: Attention-based semantic matching and pair-linking for entity disambiguation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1667–1676.

[190] A. Parravicini, R. Patra, D. B. Bartolini, and M. D. Santambrogio, "Fast and accurate entity linking via graph embedding," in *Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, 2019, pp. 1–9.

[191] D. Lin, "Extracting collocations from text corpora," in *First workshop on computational terminology*. Citeseer, 1998, pp. 57–63.

[192] R.-J. Zhu, Q. Zhao, G. Li, and J. K. Eshraghian, "Spikegpt: Generative pre-trained language model with spiking neural networks," *arXiv preprint arXiv:2302.13939*, 2023.

[193] R. K. Chunduri and D. G. Perera, "Neuromorphic sentiment analysis using spiking neural networks," *Sensors*, vol. 23, no. 18, p. 7701, 2023.

[194] R. A. Knipper, K. Mishty, M. Sadi, and S. K. K. Santu, "Snnlp: Energy-efficient natural language processing using spiking neural networks," *arXiv preprint arXiv:2401.17911*, 2024.

[195] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 3, pp. 1–45, 2024.

Abstract

Disambiguating name mentions in text is a crucial task in Natural Language Processing, especially in entity linking. The credibility and efficiency of such systems largely depend on this task. For a given name entity mention in the text, there are many potential candidate entities that may refer to this mention in the knowledge base. Therefore, it is very difficult to assign the correct candidate from the whole candidate entities set to this mention. To solve this problem, collective entity disambiguation is a prominent approach. In this thesis we present a new algorithm called CPSR for collective entity disambiguation which is based on the graph approach and semantic relatedness. A clique partitioning algorithm is used to find the best clique that contains a set of candidate entities. These candidate entities provide the answers to the corresponding mentions in the disambiguation process. To evaluate our algorithm, we carried out a series of experiments on seven well-known datasets namely, AIDA/CoNLL2003-TestB, IITB ,MSNBC, AQUAINT, ACE2004, Cweb and Wiki. The Kensho Derived Wikimedia Dataset (KDWD) is used as the knowledge base for our system. From the experimental results our CPSR algorithm outperforms both the baselines and other well known state of the art approaches.

Keywords: Named Entity Disambiguation, Entity linking, Clique Partitioning, Semantic Relatedness, Graph-Based Approaches.

Résumé

Désambiguïser les mentions de noms dans le texte est une tache cruciale dans le traitement du langage naturel, en particulier dans la liaison d'entités. La crédibilité et l'efficacité de ces systèmes dépendent largement de cette tâche. Pour une mention d'entité donnée dans le texte, il existe de nombreuses entités candidates potentielles qui peuvent faire référence à cette mention dans la base de connaissances. Par conséquent, il est très difficile d'affecter le bon candidat à partir de l'ensemble des entités candidates définies à cette mention. Pour résoudre ce problème, la désambiguïsation des entités collectives est une approche importante. Dans cette thèse, nous présentons un nouvel algorithme appelé CPSR pour la désambiguïsation d'entités collectives qui est basé sur l'approche graphique et la relation sémantique. Un algorithme de partitionnement de clique est utilisé pour trouver la meilleure clique qui contient un ensemble d'entités candidates. Ces entités candidates fournissent les réponses aux mentions correspondantes dans le processus de désambiguïsation. Pour évaluer notre algorithme, nous avons effectué une série d'expériences sur sept ensembles de données bien connus, à savoir AIDA/CoNLL2003-TestB, IITB ,MSNBC, AQUAINT, ACE2004, Cweb et Wiki. Le Kensho Derived Wikimedia Dataset (KDWD) est utilisé comme base de connaissances pour notre système. A partir des résultats expérimentaux, notre algorithme CPSR surpasse à la fois les lignes de base et d'autres approches de pointe bien connues.

Mots clé: Désambiguïsation d'entité nommée, liaison d'entité, partitionnement de clique, relation sémantique, approches basées sur des graphes.

ملخص

يعد توضيح ذكر الأسماء في النص مهمة حاسمة في معالجة اللغات الطبيعية، خاصة في ربط الكيانات .وتعتمد مصداقية وكفاءة هذه الأنظمة إلى حد كبير على هذه المهمة .بالنسبة لكيان اسم معين مذكور في النص، هناك العديد من الكيانات المرشحة المحتملة التي قد تشير إلى هذا الذكر في قاعدة المعرفة .ولذلك فإنه من الصعب جداً تعيين المرشح الصحيح من بين كافة الكيانات المرشحة المحددة لهذا الذكر .ولحل هذه المشكلة، يعد توضيح الكيان الجماعي نهجًا بارزًا .نقدم في هذه الدراسة خوارزمية جديدة تسمى CPSRلتوضيح الكيان الجماعي والتي تعتمد على نهج الرسم البياني والارتباط الدلالي .يتم استخدام خوارزمية تقسيم المجموعة للعثور على أفضل مجموعة تحتوي على مجموعة من الكيانات المرشحة .توفر هذه الكيانات المرشحة الإجابات على الإشارات المقابلة في عملية توضيح الغموض .لتقييم الخوارزمية الخاصة بنا، أجرينا سلسلة من التجارب على سبع مجموعات بيانات معروفة وهي AIDA/CoNLL2003-TestB و IITBو MSNBCو AQUAINTو ACE2004و Cwebو .Wikiيتم استخدام مجموعة بيانات ويكيميديا المشتقة من Kensho (KDWD) كقاعدة معرفية لنظامنا .من النتائج التجريبية، تتفوق خوارزمية CPSRالخاصة بنا على كل من خطوط الأساس وغيرها من الأساليب الحديثة المعروفة .

الكلمات المفتاحية توضيح الكيان المسمى، ربط الكيان، تقسيم المجموعة، الارتباط الدلالي، المقاربات القائمة على الرسم البياني.