Towards an Accurate, Robust, and Scalable Named Entity Disambiguation System

by

Zhaochen Guo

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

Knowledge bases (KBs), repositories consisting of entities, facts about entities, and relations between entities, are a vital component for many tasks in artificial intelligence and natural language processing such as semantic search and question answering. *Named Entity Disambiguation* (NED), the task of disambiguating mentions of named entities in a textual document by linking them to the actual entities in a KB, enables expanding or correcting the KB with facts extracted from documents – a task called Knowledge Base Population. This thesis focuses on the NED task with the goal of building an accurate, robust, and scalable NED system.

We first propose a graph-based approach that collectively disambiguates mentions of entities in a given document, with the assumption that entities mentioned in a document are semantically related under a single topic. Our approach uses a carefully-curated disambiguation graph built from a KB, and applies personalized random walks on the graph to compute semantic representations of entities, which are used to measure semantic relatedness and disambiguate named entities.

We then improve the robustness of our NED approach with a supervised learning to rank algorithm using publicly available datasets. We find that the public benchmarks, mainly from news articles, are biased towards well-known entities and not representative to evaluate the robustness of an NED approach. Thus we develop a framework for deriving new benchmarks and construct two benchmarks with varying disambiguation difficulties from two large corpora (Wikipedia and ClueWeb) for the evaluation of robustness.

Finally, to address the scalability issue of our NED approach, we explore various features from entity graphs, contextual texts, and document corpora that can be efficiently pre-computed offline. Instead of performing random walks on online constructed graphs, we use a set of selected landmark nodes from entity graphs to compute the semantic representations of entities. We also explore features derived from the describing documents and associated categories of entities. By pre-computing all these features offline, our approach can reduce the computing and memory resources to improve the running efficiency and scale out the NED system. The evaluation shows that our approach is very competitive and efficient compared to previous NED approaches.

Preface

Most of the work in this thesis is a collaborative effort with my supervisor, Dr. Denilson Barbosa. Dr. Denilson Barbosa has provided guidance and feedback on many of the approaches, experiments, and writing.

Except Chapter 6, all chapters are my original work. The preliminary result of Chapter 4 was published in the Data Extraction and Object Search (DEOS) workshop in the 23rd International World Wide Web Conference [31] in 2014, and the complete work of Chapter 4 was published in the 23rd ACM International Conference on Information and Knowledge Management [32] in 2014. Chapter 5 and Appendix A, which are two extended works with a new NED approach and two new benchmarks, were published together with Chapter 4 in the Semantic Web Journal [33] in 2017. I am the sole contributor for the problem formalization, method design, experimental evaluation, and result analysis. Dr. Denilson Barbosa also provided guidance for evaluation and result analysis and assisted with the manuscript by providing editorial feedback.

The work in Chapter 6 is the result of a collaboration with Michael Strobl and Victor Olivares. Victor helped build the initial system, and Michael contributed the idea of using landmarks for the NED and the implementation. I built features from the two disambiguation graphs, Wikipedia text and categories, and the connection strength. I am also mainly responsible for the system implementation, experiments evaluation, result analysis, and the manuscript writing. Dr. Denilson Barbosa provided guidance for the work and editorial feedback for the manuscript.

To the memory of my father

who was always there to encourage me but could not see this thesis completed

Acknowledgements

It is my honor to meet so many great people during this journey and I sincerely thank them and appreciate their help and accompany.

First and foremost, I would like to thank my supervisor, Dr. Denilson Barbosa, for his guidance and support throughout my research. His high standards for research and writing always push me to a higher level I never think of. Thank you, Denilson! Many thanks to the members of the committee, Greg Kondrak, Davood Rafiei, Marek Reformat, and Bruno Martins for contributing your precious time to read the thesis and providing excellent feedback. Thanks to Daniel Chui for his time on proofreading this thesis. I would also like to thank my colleagues and collaborators: Michael Strobl, Victor Olivares, Filipe Mesquita, Ying Xu, and Mirko Bronzi. This work was only possible because of you.

Thanks to the wonderful staffs at the department. Special thanks to Daneel Blair and Karen Berg for dealing with the graduate issues and providing help whenever needed. I would also like to thank Evan Chrapko and Shane Chrapko and everyone in Trust Science for their support and trust. It was great working with you guys in the past two years and I wish Trust Science all the best in the future. Thanks to Diffbot for believing me and offering me the opportunity to continue the research on this challenging problem.

Lastly but most importantly, my deepest thanks and appreciation go to my family. I would like to thank my wife for her support, love, and sacrifice, and my beloved daughter for bringing joy to my life and being such a good girl always cheering me up. Nothing is a big deal with your accompany. Thanks to my parents for making who I am today with their unconditional love, and my sisters for your inspiration and support. I am so proud of your achievements.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

A knowledge base (KB) is a repository of structured information consisting of entities, facts about entities, and relations between entities. The recent advent of large KBs has renewed the interest in algorithmic understanding of natural language text, especially in the context of the Web where facts about named entities are described in many documents.

Two crucial tasks in natural language understanding have to do with *named entities*, which are the persons, organizations, locations, *etc*. that are explicitly mentioned in text using proper nouns: (1) *Named Entity Recognition* (NER), which corresponds to finding *mentions* to named entities in the text; and (2) *Named Entity Disambiguation* (NED), which is the task of disambiguating the named entities by linking them to the actual entities in the KB (when possible). This thesis is concerned with the NED task, assuming that mentions to named entities have been identified by NER.

A typical task that requires NED to help resolve the ambiguity of named entities is *question answering* (QA), the task of generating answers in response to questions in natural language. There are two main paradigms of QA systems: text-based QA which relies on text from large corpora, such as the Web, and knowledge-based QA which relies on structured KBs. To answer questions like "*which NBA teams has Karl Malone played for?*", a text-based QA system [40] would parse the question into query terms, search the corpora for documents matching these query terms, and rank relevant answers or passages extracted from the matching documents, such as the ones shown in Example 1 and 2.

Example 1.

Malone, a retired professional basketball player, is mostly known for his time with the Washington Bullets, where he was an NBA All-Star twice. He also played for Utah, Philadelphia, and Miami.

Example 2.

Malone, nicknamed "The MailMan" spent his first 18 seasons in NBA with the Utah Jazz and final season with Los Angeles. He was a two-time NBA MVP, a 14-time NBA All-Star.

As can be seen, finding answers to the above question is challenging for text-based QA systems because both documents contain the query terms *Malone* and *NBA*. Moreover, Example 1, which describes *Jeff Malone* (a different basketball player), might rank higher than Example 2 (the correct answer) since it contains the phrase *played for* in the question while Example 2 does not. Even if Example 2 is ranked higher, deciding the team that mention *Los Angeles* refers to, *Los Angeles Clippers* or *Los Angeles Lakers*, remains a challenge.

A knowledge-based QA system, on the other hand, would formalize the question into a query in logical form: **playsFor(Jeff Malone, x?)** $\wedge$ **isA(x, NBA Team)**, and retrieve the answers by executing the query against a structured KB like the one shown in Figure 1.1. In such a knowledge-based QA system, the primary role of NED is in understanding the question by disambiguating named entities mentioned in the question to their counterparts in the KB. It is worth mentioning that NED is also useful for understanding text that can be used to augment the KB with new facts, provided one can also map phrases in the text to predicates in the KB. For instance, Example 1 states several facts, including that *Karl Malone* played for *Utah Jazz* and *Washington Wizards* (at the time the team was known as *Washington Bullets*), which can only be added to the KB after NED. Besides its importance in question understanding and knowledge base population, NED is also important in many other tasks that require resolving ambiguity of named entities, such as topic classification based on named entities [47] and sentiment analysis on named entities

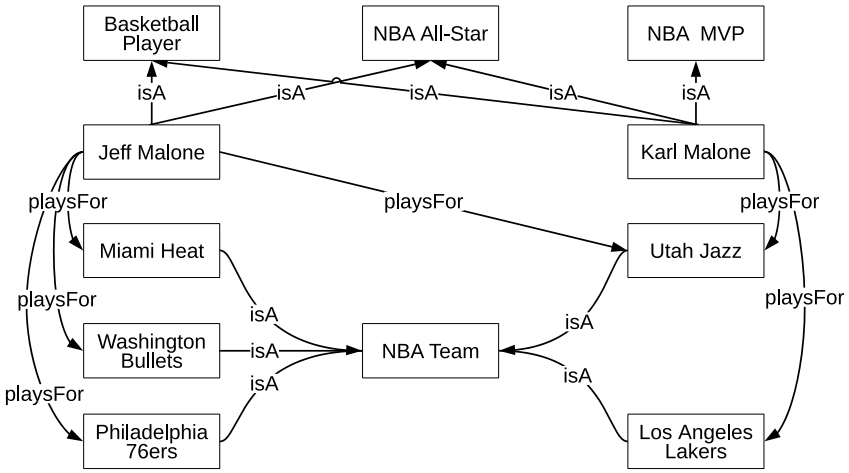


Figure 1.1: An example knowledge base.

like restaurants or products [78].

NED is a difficult problem, even for humans, due to the inherent ambiguity of natural language. As shown in the examples, one named entity can be mentioned in different forms (*e.g.*, the NBA team *Utah Jazz* is mentioned by *Utah* in Example 1 and *Utah Jazz* in Example 2), while each mention can refer to multiple named entities (*e.g.*, the mention *Malone* refers to two different NBA basketball players: *Jeff Malone* in Example 1 and *Karl Malone* in Example 2).

Most approaches perform NED in two stages: *candidate selection* and *mention disambiguation*, with the first stage to select a suitable set of candidate entities for each mention, and the second stage to perform the actual disambiguation of mentions. Selecting candidate entities is done, primarily, by consulting an alias dictionary, a mapping from aliases to their referent named entities. As for mention disambiguation, most approaches can be categorized into two main groups, as discussed next.

Local NED The local NED approaches focus mainly on lexical features, such as words or entities surrounding each mention in the document [3], [9], [19], [66]. They disambiguate each mention independently, typically by ranking the candidate entities according to their similarity with the mention, and picking the most similar one. These approaches work best when the context is rich enough to uniquely

identify a mention, which is not always the case. For example, the documents in the examples lack sufficient context to disambiguate *Malone* and *Los Angeles*.

Global NED Unlike local approaches which handle each mention independently, global approaches perform the disambiguation *collectively* on all mentions in a document [15], [17], [38], [45], [51], [85], motivated by the premise that disambiguation of one mention contributes to the disambiguation of the remaining mentions in the same document. For example, disambiguating *Washington Bullets* to named entity *Washington Wizards* will make it easier to disambiguate *Utah, Philadelphia*, and *Miami* in Example 1 to their corresponding NBA teams as opposed to cities or states since they are more semantically related to *sports* and *NBA* instead of the *locations*.

Following the *topic coherence assumption* that an input document belongs to a single topic (*e.g.*, sports) under which all entities mentioned in the document are tightly related, most global approaches aim at taking into account the semantics of the mentions and candidate entities, represented as a graph consisting of entities and links in the KB, as shown in Figure 1.2. In general, they start with a graph that has all mentions in the document (*e.g.*, the *mention* column in Figure 1.2), linked to every one of their candidate entities in the KB. In turn, the candidate entities are also linked to a small subset of their full neighborhood in the KB. Disambiguation in this approach then seeks to find a forest embedded in the constructed graph in which each mention remains linked to a single candidate entity and entities in the forest are more connected with each other than other candidates by a measure of *semantic relatedness* – a notion captured as some property of the forest. The measure of semantic relatedness used by each method is key to its accuracy and cost and has been the focus of many NED approaches [44], [69].

Another observation about global approaches is that, to produce meaningful results, the disambiguation must be constrained so that it produces a small mention-to-entity assignment (*i.e.*, forest in the original graph) with high global coherence (*e.g.*, based on semantic relatedness). However, finding such an assignment is NP-hard [51], and all approaches turn to approximate algorithms or heuristics.

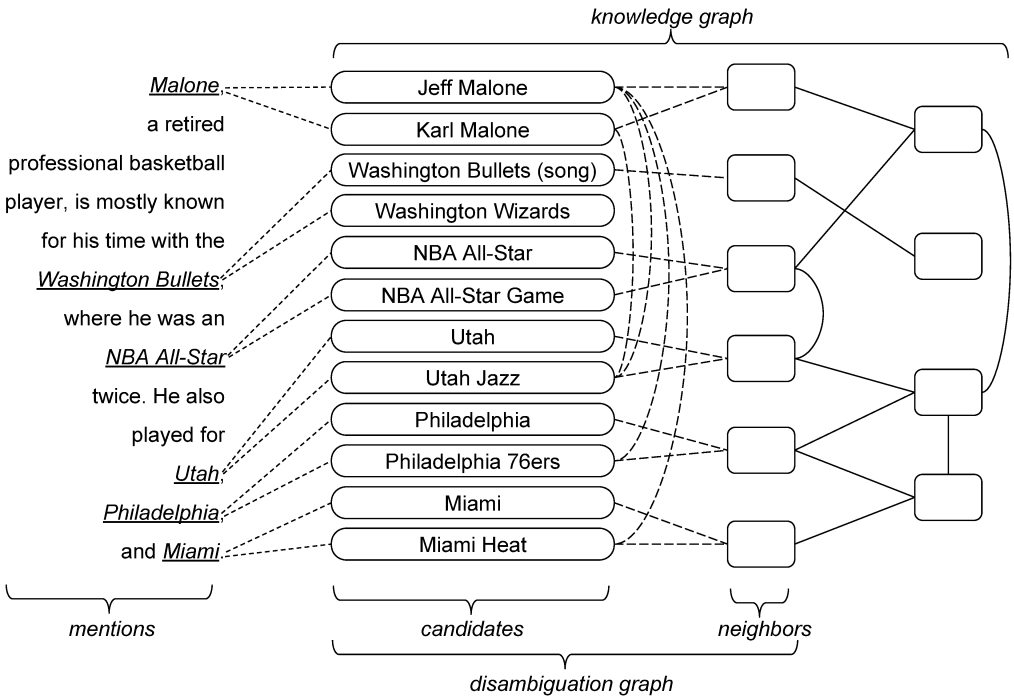


Figure 1.2: Example named entity disambiguation scenario.

In this thesis, we address the NED problem following the paradigm of global approaches with the goal of building an *accurate*, *robust*, and *scalable* NED system. Being *accurate* requires our system to be competitive to the state-of-the-art NED approaches in terms of accuracy; being *robust* requires the model in our system to be flexible to incorporate new features and also portable to be applied on different datasets; and being *scalable* requires our system to be efficient to process each document and also have the ability to scale out for large-scale data processing.

1.1 Background

Our approaches use an entity graph, a graph consisting of entities and their relations, to derive effective measures of semantic relatedness and features for our models. We build our entity graphs using Wikipedia as the knowledge repository for its popularity and high coverage, although our NED approaches also work with other KBs. In this section, we first define the terminology and notation used in the thesis, as listed in Table 1.1, and then give a brief introduction of Wikipedia, its alias dictionary, and two entity graphs constructed from it.

Term	Notation	Definition
Token	t	A **token** is a "sequence of characters grouped together as a semantic unit for processing" [60], and is usually an individual word or a symbol.
Lemma	l	A **lemma** is a canonical form of a set of words. *E.g.*, *bake* is the lemma of *baking* and *baked*.
Document	d	A **document** is an ordered sequence of tokens, denoted as $d = \langle t_1, t_2, \ldots, t_n \rangle$.
Mention	m	A **mention** is an ordered sequence of tokens $m = \langle t_i, \ldots, t_j \rangle$, mainly as a surface form referring to a specific object.
Context	$context$	**Context** of a mention is a subsequence of a document d that is related to m, denoted by $context(m)$. The context could be the surrounding tokens of a mention or the entire document.
Entity	e	An **entity** is a real-word object in a repository, such as a person or an organization. It is usually referenced by a unique identifier (*e.g.*, a URI).
NIL	NIL	**NIL**, also known as *Out-of-KB Entity*, refers to any entities that are not in a KB.
Alias	$alias$	An **alias** is an ordered sequence of tokens that are used to mention an entity. We denote the alias set of an entity e as $aliases(e)$.
Alias Dictionary	AD	An **alias dictionary** is a mapping from each alias to a list of entities the alias could refer to.
Candidate Entity	e	A **candidate entity** is an entity potentially referred to by a mention m. A candidate set of mention m is denoted as $cand(m) = \{e_i, \ldots, e_j\}$.
Knowledge Base	KB	A **knowledge base** is a repository of structured information consisting of entities, facts about entities, and relations between entities.
Entity Graph	EG	An **entity graph** is a graph representation of entities and connections between them, usually derived from a knowledge base [6]. Formally, an entity graph is defined as $\mathsf{EG} = (E, L)$, where nodes in E correspond to entities and links in L are derived from connections between entities.
Landmarks	LM	**Landmarks** are a set of distinctively selected entities that meet some pre-defined criteria.
Context Similarity	$ctxSim$	**Context similarity** is a metric to measure the similarity between objects (could be mentions and entities) by their context.
Semantic Signature	$\mathsf{SS}(e)$	**Semantic signature** is a vector representation of the semantics of an entity.
Semantic Relatedness	$\psi(e_i, e_j)$	**Semantic relatedness** is a metric to measure the strength of the relatedness between entities through semantic signatures.
Assignment	Γ	An **assignment**, denoted as $\Gamma : M \rightarrow E \cup \{\mathsf{NIL}\}$, is a mapping between mentions in a document and entities in an EG, with each mention assigned with at most one entity.
Global Coherence	Ψ	**Global coherence** is a metric to measure the semantic coherence of entities in a document, usually by computing the overall semantic relatedness among entities in an assignment Γ, defined as $\Psi(\Gamma)$.

Table 1.1: Terminology and notation.

1.1.1 Wikipedia

Wikipedia is the largest free online encyclopedia, covering a wide range of topics. At the time of writing, it had about 90 thousand active editors, over 5 million articles in English alone, and covered 293 other languages. The Wikipedia reposi-

tory consists of Wikipedia pages, each of which is uniquely identified by a title and an internal id and provides a definitional description about a person, a location, or an event, *etc*. In addition to the textual description, most pages also have a semi-structured field, called *Infobox*, with a summary of facts about the corresponding entity, such as birthday or education. Furthermore, pages are grouped into *categories* based on their topics [1]. For example, the page of *Karl Malone* is classified into categories of *National Basketball Association All-Stars* and *Power forwards (basketball)*. Wikipedia uses hyperlinks and special-purpose pages to build a better navigation experience, as discussed below.

WikiLinks A WikiLink is a hyperlink between Wikipedia pages, connecting a mention to their true entity page. As markups manually annotated by editors, WikiLinks provide high-quality annotations for entity disambiguation and semantic relatedness measure. Also as entity annotations, WikiLinks can be used to construct large volume of training and testing datasets for NED and collect aliases of entities to build alias dictionaries.

Redirect Pages A redirect page is used to redirect readers to the page of the actual entity from its aliases. Usually, redirect pages account for alternative names, including abbreviations, plurals, alternative spellings, and misspellings, among other aliases. For example, the page about the *United States* can be redirected from several redirect pages: *United States of America*, *The States*, *U.S.*, and *USA*. As we see, these pages can serve as reliable sources of aliases for the entities.

Disambiguation Pages Contrary to a redirect page, a disambiguation page lists all known entities of a single name or lemma. For example, the disambiguation page for lemma *tree* contains a link to the page describing the woody plant [2] and another to the page about the data structure [3].

[1] `https://en.wikipedia.org/wiki/Category:Main_topic_classifications`

[2] `https://www.wikipedia.org/wiki/Tree`

[3] `https://www.wikipedia.org/wiki/Tree_(data_structure)`

1.1.2 Alias Dictionary

An alias dictionary is a mapping of known aliases of entities to their identifiers in the KB. From Wikipedia, we build an alias dictionary using 4 different sources: *Wikipedia page title*, which gives the canonical name of an entity, *Redirect page*, which provides alternative names of an entity, *Disambiguation page*, which gives commonly used names of an entity, and *WikiLink*, which provides aliases of an entity through anchor text. In addition to the mapping from aliases to their entities, we also record the frequency of each mapping – the number of times an entity is mentioned by an alias.

An example entry in an alias dictionary for alias *Utah* would be:
$\{$*utah :* $\{$*Utah:11267, Utah Jazz:710, University of Utah:376, Utah Railway:2*$\}\}$

1.1.3 Entity Graphs

We build two entity graphs from Wikipedia, as described below.

PageLink Graph Using WikiLinks, we construct a *PageLink Graph* $EG_{PageLink}$ – an entity graph with *entities* (what Wikipedia articles describe) as nodes and their connections via WikiLinks as edges. Each WikiLink, though directional from one entity e_i to another e_j, also indicates a semantic relation from e_j to e_i. Therefore, we construct the *PageLink Graph* as an undirected entity graph.

Co-occurrence Graph Taking the distance between entities into account, we say two entities e_i and e_j co-occur when: (1) e_i is mentioned in the document describing e_j or the other way around (the same relation as that in the PageLink Graph); or (2) e_i and e_j are both mentioned within a window of 500 words [10]. Using WikiLinks, we can build a *Co-occurrence graph* $EG_{Cooccur}$, in which nodes are entities and edges are co-occurrence relations between entities.

One difference of the two EGs is the weight of edges, which is set to 1 in the PageLink graph and the total number of co-occurrences of two entities in the Co-occurrence graph.

1.2 Problem Statement

This thesis focuses on the two sub-tasks of NED: *candidate selection* and *mention disambiguation*, as defined below.

Definition 1 (Candidate Selection)**.** *Given a mention m in a document d, and an Entity Graph $EG = (E, L)$, the candidate selection task is to find a set of candidates that m could potentially refer to: $cand(m) = \{e_i, \ldots, e_j\} \subseteq E$.*

Definition 2 (Mention Disambiguation)**.** *Given a document d, a set of mentions $M = \{m_1, \ldots, m_N\}$ in d and their candidates $cand(m_i)$ from an Entity Graph $EG = (E, L)$, the mention disambiguation task is to find an assignment $\Gamma : M \rightarrow E \cup \{\text{NIL}\}$, such that:*

- *$\Gamma(m_i) \in cand(m_i)$ and $\Gamma(m_i)$ is the referent entity of m_i; or*
- *$\Gamma(m_i) =$ NIL, otherwise.*

The *candidate selection* task is mainly to reduce the actual search space of entities that can be referred to by a mention. It is done by consulting an alias dictionary with an approximate similarity estimation method that identifies a list of plausible candidates for each mention. The *mention disambiguation* task is more rigorous and time-consuming: it determines the best match for each mention among the candidates identified in candidate selection.

1.3 Overview of Proposed Approaches

1.3.1 Candidate Selection

Our NED system employs a candidate selection approach that is experimentally tuned to balance the recall and cardinality of candidate sets. We review the current approaches for candidate selection, including name expansion methods to find the full names of mentions and aliases of named entities (using an alias dictionary), and effective pruning criteria to prune irrelevant candidates. We then experimentally evaluate the impact of these name expansion methods and pruning criteria on candidate selection, and find a balanced setting for our NED system.

1.3.2 Mention Disambiguation

An Accurate NED Approach

Our first approach for *mention disambiguation*, named WNED (Walking Named Entity Disambiguation), is a global NED approach. It advances the state-of-the-art in accuracy using a semantic relatedness measure based on a novel *semantic signature*. To compute the semantic signature, we build a disambiguation graph like the one shown in Figure 1.2, and represent each candidate entity by the stationary probability distribution resulting from a random walk with restart [96] on that graph. We call such distributions the semantic signatures of the entities.

As demonstrated in the personalized PageRank algorithm [41], a random walk with restart on a graph can propagate information along the edges and provide a relatedness measure between indirectly connected nodes. The probability in the stationary distribution can be viewed as the relatedness between these target entities and each entity in the graph, with higher value indicates higher relatedness. Thus, our semantic signatures capture the semantics of the entities in terms of their relevance with respect to all other entities in the graph, which represent an entity in a more fine-grained manner than the 0-1 coarse weighting in local approaches. Furthermore, we capture the semantic signature of a document by performing a random walk with re-starting from the *set of entities* in that document. As shown later, these signatures allow computing the relatedness of entities and partially disambiguated documents using Information Theory.

We propose an iterative algorithm for WNED from the observation that the disambiguation of one mention can benefit the disambiguation of others. In each round, the algorithm performs a random walk with restart using entities that have been disambiguated up to that point to compute the semantic signature of the document, which is used to compute the global coherence between the document and each candidate. In a greedy fashion, the algorithm picks the candidate with the highest total score above a threshold, or NIL if no such candidate exists, and proceeds to the next mention. Our approach using this iterative algorithm can achieve state-of-the-art accuracy. One potential issue of WNED, however, is its robustness: we employ a hand-tuned algorithm to combine different similarity scores between

a mention and its candidate, which is sensitive to the datasets and inflexible to incorporate new similarity scores.

A Robust NED Approach

Our second approach, named L2R (Learning to Rank), addresses the robustness issue of WNED with a supervised machine learning algorithm. With labeled data for NED task available, especially large-scale datasets that can be generated automatically (*e.g.*, via WikiLinks in Wikipedia), we employ a learning to rank algorithm to build a ranking model for NED using these datasets. In addition to the lexical and statistical features, our ranking model also incorporates the semantic relatedness (we use the one in WNED), which is commonly ignored in other supervised NED approaches. The evaluation results show that our L2R can outperform all state-of-the-art NED systems we evaluated, and the ranking model trained on one dataset is robust and can achieve high accuracy on other datasets. Our WNED and L2R, while very competitive to other approaches, are two memory intensive and computationally expensive approaches because of the online graph construction and the random walks, thus are not scalable to handle large datasets.

A Scalable NED Approach

Our last approach addresses the scalability issue in several ways. We first propose to approximate the disambiguation graph used in WNED and L2R with a set of landmarks, which are representative entities carefully selected from the entity graph, so that we can avoid the online graph construction and perform the random walk offline to pre-compute the semantic signatures of entities using only landmarks. We then explore features from the PageLink and Co-occurrence graphs, including the set of neighboring entities and connection strength between entities, to further improve the accuracy. We also employ MinHash [54] to pre-process textual features to improve the efficiency of the similarity computation. The experiments show that our new NED approach can achieve comparable accuracy as our previous systems with a large gain in efficiency.

1.3.3 Evaluation Methodology

We perform extensive experimental evaluation of these approaches using both well-known public benchmarks and new benchmarks with more challenging cases. We assess their accuracy and efficiency using standard metrics including precision, recall, F1 score, and running time. In addition, we also leverage a general benchmarking framework for NED to compare our approaches with more than 11 state-of-the-arts and demonstrate the superiority of our approaches.

1.4 Summary of Contributions

- First, we propose a global NED approach that can achieve the state-of-the-art accuracy using a novel semantic signature obtained through a random walk with restart on a disambiguation graph.
- Second, we introduce a supervised approach to improve the robustness of the first approach using a learning to ranking algorithm, which is robust to changes of datasets and features. Our ranking model, trained on one dataset, can achieve high accuracy on other datasets.
- Third, we present an efficient approach to address the scalability issue of the first two approaches. This approach employs efficient methods to compute semantic signatures of entities and semantic relatedness between entities, which can help scale out the approach and also achieve competitive accuracy to the state-of-the-arts.
- Last, we develop a framework for deriving new benchmarks and construct two balanced benchmarks from large corpora with documents of varying difficulty, which complement the public benchmarks with the potential to advance the research of NED.

In addition to the main contributions, our system also participated in two evaluation challenges: TAC KBP 2015 [4] and NEEL Challenge [5].

[4] https://tac.nist.gov/2015/KBP/

[5] http://scc-research.lancaster.ac.uk/workshops/microposts2015/

1.5 Organization

The rest of this thesis is organized as follows: Chapter 2 first gives a comprehensive review of the literature concerning NED. Chapter 3 describes the candidate selection. Chapter 4 introduces our unsupervised approach WNED. Chapter 5 presents the supervised learning to rank approach L2R. Chapter 6 describes our NED approach to address the scalability issue. Finally, in Chapter 7, we conclude the thesis with some directions for future work.

Chapter 2

Background and Related Work

In this chapter, we first give some background of NED in the domain of NLP and KBP, then review the state-of-the-art NED approaches, and briefly discuss a few closely-related tasks at the end.

2.1 Information Extraction

Information Extraction (IE) is the task of extracting structured information such as entities, facts about entities, and relations between entities from unstructured sources. Two sub-tasks of IE are *named entity recognition* and *relation extraction*.

Named Entity Recognition (NER) NER is the task of identifying named entities from a given document and classifying them into predefined categories, such as person, organization, and location. NER is an important research problem on its own merit. Standard approaches model NER as a sequence labeling problem and employ supervised machine learning algorithms for label prediction. Given annotated sentences with *IOB* tags (which correspond to Inside, Outside, and Beginning of an entity), a sequence classifier can be trained to tag words in a sentence and use those tags to identify named entities. Stanford NER [23], one of the well-known NER systems, provides a Conditional Random Fields (CRF) based classifier to label sequences of terms with part-of-speech tags and other features. They also provide models trained on a mixture of CoNLL, MUC-6, MUC-7, and ACE named entity corpora. More reviews about work related to NER can be found in [73], [75].

Relation Extraction (RE) RE is the task of detecting and extracting relations between named entities, such as *playsFor* between *Malone* and *Utah Jazz* in the examples. These relations encode the semantic connections between entities which can be used to support reasoning in applications [2]. There are different types of approaches proposed for the RE task. The earliest approaches are based on hand-built lexico-syntactic patterns [43], which can achieve high precision, but low recall. Supervised machine learning approaches [8] are also common solutions for RE, which train classifiers to predict if a relation between entities exists and then label the relation. While achieving high accuracy, these supervised solutions have to face the high cost of data annotations and the fact that most models cannot be generalized to different domains [87]. To solve this problem, semi-supervised approaches [14], [94] are proposed to exploit a limited number of training examples to bootstrap classifiers and use them to discover new patterns, which can then help find more entity pairs. Furthermore, distance supervision methods [70] are used to acquire a large number of seed examples from KBs like DBpedia and Freebase and combine these seed examples with supervised approaches to improve relation classification.

With these advances in IE, many open IE systems (*e.g.*, ReVerb [21] and NELL [71]) are built to harvest knowledge from the Web – a task known as *knowledge base population*.

2.2 Knowledge Base Population

Knowledge Base Population(KBP) is the task of populating knowledge bases with information extracted from unstructured sources. There are two main sub-tasks in KBP systems: *Slot Filling* which fills in values and relations of given entities with facts extracted using IE techniques, and *Entity Linking* which resolves the ambiguity of mentions in a given document by linking them to entries in a KB through NED approaches. As illustrated in Figure 2.1, starting with unstructured sources (*e.g.*, Web documents), a KBP system first identifies named entities and extracts their relations through an IE component, and then performs NED to populate the extracted knowledge into a KB.

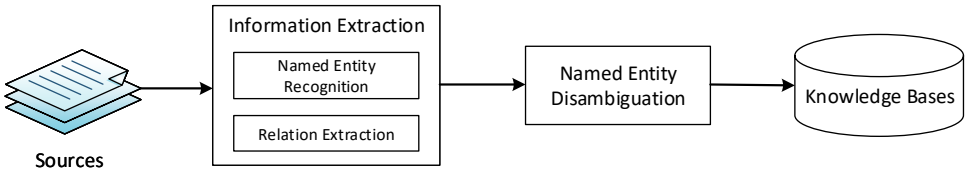


Figure 2.1: An overview of a general KBP system.

Realizing the importance of KBs, many applications start to harvest general or domain-specific knowledge from various sources and build their own KBP systems. Existing high-quality KBs, such as Cyc [53] and DBpedia [4], are mainly constructed manually or through crowd-sourcing by online communities. The main issues of constructing these KBs are the high cost of population and maintenance and the moderate coverage of facts. Recent advances in IE have enabled automatic KBP (*e.g*. NELL [12], TrueKnowledge [97], and Probase [106]). The NIST Text Analysis Conference (TAC) [1] has also been organizing a KBP track in the past few years with the goal of promoting research in knowledge discovery and KBP. The track provides benchmarks and tools for the evaluation of various tasks. Besides slot filling and entity linking, they also have an *Event* track to extract information about events from unstructured text and *Belief and Sentiment* track to detect beliefs and sentiments about entities, both of which need to be combined with NED to populate the extracted information into KBs.

2.3 NED: State-of-the-Art

The literature about NED is vast, with an extensive list of approaches. Shen *et al*. [89] gave a qualitative survey, framing most of the approaches covered here. Dai *et al*. [18] provided a brief survey of NED approaches for bioinformatics applications. There are also several tutorials covering the topic of NED [62], [63]. In the following sections, we will review the state-of-the-art NED methods within a unifying framework and standard notation. Unlike previous surveys, this chapter offers a quantitative comparison of these methods, consisting of a summary of the

[1] https://tac.nist.gov/

experimental evidence provided by each method.

There are mainly four categories of approaches proposed in the literature: to disambiguate mentions individually using ranking (Section 2.3.1); to disambiguate mentions collectively by solving an optimization problem (Section 2.3.2); to disambiguate mentions using topic models (Section 2.3.3); and the recent methods using deep neural networks (Section 2.3.4).

2.3.1 Local NED

As formalized in Chapter 1, NED is to map a set of mentions $M = \{m_1, \ldots, m_N\}$ to entities in an entity graph $EG = (E, L)$. Early works on NED focus on disambiguating each mention in isolation. They propose to use a compatibility function $\phi : M \times E \rightarrow [0, 1]$ to measure the local compatibility between a mention m and its candidate entities $e_i \in cand(m)$ with the goal of finding:

$$e^* = \underset{e_i \in cand(m)}{\arg\max}\ \phi(m, e_i) \tag{2.1}$$

To measure the compatibility, a variety of features pertaining to the context in which the mention appears are used, in both unsupervised [3], [9], [39] and supervised [19], [66], [68], [113] ways. Here we review a few approaches using the idea of compatibility functions ϕ for NED, as summarized in Table 2.1.

Unsupervised approaches

The canonical approach to compute $\phi(m, e)$ is to model each mention and entity using feature vectors and employ vector-based similarity measures to compute the compatibility. One common way is to extract features from the context of mentions and entities. Recall that the context of a mention can be defined by a window of words surrounding it, while for entities in the KB, their context could be a describing document (usually the text in a Wikipedia page).

Bagga and Baldwin [3] were among the first few to measure the compatibility using bag-of-words as features; they processed contextual texts by removing stopwords and weighing words with their $tf\,idf$. Bunescu and Pașca [9] followed a

Method	Features	Similarity Measure	Training Corpus	External Resources
Unsupervised local NED (Section 2.3.1)				
[9]	- Bag-of-words - Weighted word-category correlation	- Cosine Similarity	–	–
Supervised local NED (Section 2.3.1)				
[66]	- Bag-of-words - Part-of-speech - Entity-specific keywords	- Cosine similarity - Naive Bayes classifier	Wikipedia	–
[68]	- Commonness - Semantic relatedness - Unambiguous named entities	- Naive Bayes - SVM - C4.5 - Feature selected C4.5 - Bagged C4.5	Wikipedia	–
[111]	- Bag-of-words - Named entities - Word-category correlation - Entity types	- Cosine similarity - SVM classifier	Wikipedia	–
[113]	- Bag-of-words - Named entities - Text surrounding links - Relatedness - Category overlap - Commonness - Mention ambiguity - Name string similarity	- Cosine similarity - GBDT - GBRank	Wikipedia	Web search click logs
[19]	- 200 Atomic features - 26569 combined features	- SVM Ranker	TAC-KBP Dataset	Search engine
[110]	- Topic vector	- Hellinger distance - Classifier	Wikipedia	–

Table 2.1: Summary of local NED approaches.

similar strategy, using a surrounding window (of 55 words) as the context of mentions. Besides words, named entities extracted from documents were also used as the context by Cucerzan [17]. For an entity e in a KB, they used all entities in the first paragraph of the describing document of e as well as entities that link back to e. For mentions without surrounding entities, they proposed to use the candidate entities of all mentions in the document to approximate their entity features.

When textual features are used, as in the methods described above, the disambiguation accuracy could be affected by various issues, such as word variations (marriage *vs*. married) or neglect of semantics (*e.g*., marriage *vs*. spouse).

To solve this problem, Bunescu and Pașca [9] defined a word-category correlation which measures the semantic similarity of mention m and entity e by the correlation between words in the context of m and the categories of e . For m and e, the feature vector has $|V| \times |C|$ as dimensions, and the weights are binary:

$$V_{w,c}(m,e) = \begin{cases} 1, & \text{if } w \in context(m) \text{ and} \\ & c \in category(e) \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

in which w is a word, c is a category, $|V|$ is the size of the word vocabulary, and $|C|$ is the number of the categories. Given the word-category correlation vector, Bunescu and Pașca [9] then learned the weights of the correlation between words and categories using Wikipedia as the training dataset. The weights, in turn, are used to compute the compatibility between a mention and its candidate entities.

Supervised approaches

Critics of unsupervised approaches argue that tuning parameters is not only challenging but also brittle, advocating the use of supervised approaches, given the abundance of training data that can be obtained from Wikipedia. As shown in Table 2.1, a significant number of supervised NED approaches were proposed aiming to build a classification or regression model to replace the compatibility function $\phi(m,e)$ in unsupervised approaches. Instead of treating the NED problem as a binary classification (*i.e.*, determine whether an entity e is the correct entity of a mention m or not), most approaches use regression models to predict a confidence score (or probability) and use it as the compatibility to rank candidate entities, which the disambiguation procedure outlined by Equations 2.1 is applied.

Training Data Wikipedia has been a favorite source to build training data for supervised approaches. With entities annotated via WikiLinks, positive training examples can be extracted by sampling mentions (from anchor text), and the contextual words surrounding them, together with the target entity. For negative samples, a traditional strategy is to consult an alias dictionary and use candidate entities that are *not* the true entity of a given mention. In this way, we can automatically generate a large number of training examples.

Features In practice, features are as important as learning algorithms [104]. The methods surveyed here show a great diversity when it comes to feature selection.

Mihalcea and Csomai [66] employed a Naive Bayes classifier to combine a list of features, including mention names and their parts-of-speech, three words to the left and right side of the mentions and their parts-of-speech, as well as a list of frequent keywords (occurring at least 3 times in a corpus) in the context. Their wikification experiments on a Wikipedia dataset gave high quality of annotations that were hardly distinguishable from the human-generated annotations in Wikipedia.

Milne and Witten [68] proposed to use entities referred to by unambiguous mentions (*i.e.*, mentions with only one candidate entity) in the surrounding context of a mention. Three features were used in their approach: *commonness* which is the probability a mention refers to a candidate entity, *relatedness* which defines the semantic relatedness between entities, and *context quality* which measures the weight of contexts (computed from the commonness and relatedness). Several different classifiers were trained using a Wikipedia dataset, of which *Bagged C4.5* achieved the best results. Also, evaluation on a newswire dataset showed that classifiers trained on a Wikipedia dataset can achieve fairly high accuracy on non-Wikipedia datasets.

Dredze *et al*. [19] developed 55 classes of features, including almost 200 atomic features for each mention-entity pair, and 26,569 combined features from these atomic features in mentions, entities, source documents, and a KB. These rich and extensible sets of features cover spelling variations, misspelling, acronyms and abbreviations, named entity types, as well as a wide variety of statistics from their KB. Their SVM ranking based learning algorithm can achieve 94% accuracy on a Newswire dataset and 80% on a more challenging TAC-KBP dataset.

In addition to features extracted from documents and KBs, Zhou *et al*. [113] also explored features from external sources including 20 types of semantic relatedness scores between entities, such as overlapping of categories, links, and co-occurrence count of entities obtained from the browsing data of Web users. They employed two learning algorithms: Gradient Boosted Decision Trees (GBDT) and Gradient Boosted Ranking (GBRank) to build a disambiguation model using Wikipedia

datasets, which can achieve 84% accuracy on the MSNBC dataset and 81% accuracy on the Yahoo! News dataset.

2.3.2 Global NED

While local approaches handle mentions in isolation, ignoring the semantic relations between mentions, *global NED* follows another direction to exploit the global coherence of mentions in a given document to collectively disambiguate mentions.

Table 2.2: Summary of global NED approaches.

Method	Features	Similarity Measure	Training Corpus	External Resources
Global approaches: optimization problem (Section 2.3.2)				
[17]	- Named entities - Category	- Cosine similarity	–	- The Web - CoNLL 2003
[51]	- Bag-of-words of the first paragraph - Bag-of-words of the document - Anchor text and surrounding text	- Cosine similarity - Jaccard similarity	–	–
[85]	- Commonness of entity - Bag-of-words - Prediction results of a classifier - Link-based relatedness	- SVM Ranker	Wikipedia	–
[15]	- Commonness of entity - Bag-of-words - Syntactico-semantic relations - Co-reference relations	- Cosine similarity - Relation confidence	–	DBpedia
[10]	- Co-occurrence between entities	- Co-occurrence	–	–
Global approaches: graph problem (Section 2.3.2)				
[45]	- Commonness - Keyphrases - Entity graph	- Semantic relatedness	–	–
[38]	- Bag-of-words - Hyperlink structure of Wikipedia	- Cosine similarity - Semantic relatedness	–	–
Topic-model approaches (Section 2.3.3)				
[36]	- Entity-mention model	–	Wikipedia	–
[37]	- Entity-topic model	–	Wikipedia	–

Most global NED approaches are based on the *topic coherence assumption* and aim to find an assignment Γ with maximum global coherence among all entities in Γ. These approaches are mostly formalized as one of the following two problems.

NED as an Optimization Problem

Suppose $\Psi : \Gamma \to \mathbb{R}$ is a measure for the global coherence of an assignment Γ, we can cast the NED problem to an optimization problem aiming to find an assignment Γ such that:

$$\Gamma^* = \arg\max_{\Gamma} \left(\sum_{i=1}^{N} \phi(m_i, e_j) + \Psi(\Gamma) \right) \tag{2.3}$$

Here N is the number of mentions in a document. The first component $\phi(m_i, e_j)$ measures the compatibility between mention m_i and entity e_j, and the second component $\Psi(\Gamma)$ measures the global coherence of assignment Γ. Note that the problem becomes a local ranking problem when the second component is eliminated.

NED as a Dense Subgraph Problem

With an entity graph where semantic relatedness is encoded in the edges between entities, NED can be formalized as a problem of finding a subgraph of entities with the maximum coherence score. Formally, given a graph $G = (N_G, R_G)$ with $N_G = M \cup E$ and $R_G = \{m \to e, e_i \to e_j\}$, in which $m \to e$ is a mention-entity relation weighted by their compatibility $\phi(m, e)$ and $e_i \to e_j$ is an entity-entity relation weighted by their semantic relatedness $\psi(e_i, e_j)$, the NED problem is to identify a subgraph that contains exactly one mention-entity edge for each mention. As we can see, the dense subgraph problem can be cast to the optimization problem defined above by defining the coherence score as follow:

$$\sum_{i=1}^{N} \phi(m_i, e_j) + \Psi(\Gamma)$$

Global Coherence A common way of measuring global coherence $\Psi(\Gamma)$ is to add up the semantic relatedness $\psi(e_i, e_j)$ of all entity pairs in Γ, defined as follows:

$$\Psi(\Gamma) = \sum_{i=1}^{N} \sum_{j \geq i}^{N} \psi(e_i, e_j)$$

Here we assume that the semantic relatedness ψ is symmetric: $\psi(e_i, e_j) = \psi(e_j, e_i)$. For asymmetric measure, we can use an averaged value:

$$\psi(e_i, e_j) = \frac{1}{2} \times (\psi(e_i \to e_j) + \psi(e_j \to e_i))$$

With the global coherence measure, the goal of a global NED approach is then to find an assignment Γ^* that maximizes the following objective function.

$$\Gamma^* = \arg\max_{\Gamma} \left(\sum_{i=1}^{N} \phi(m_i, e_i) + \sum_{i=1}^{N} \sum_{j \geq i}^{N} \psi(e_i, e_j) \right) \tag{2.4}$$

Approaches

Cucerzan [17] presented the first global NED approach using categories of entities, in which the semantic relatedness $\psi(e_i, e_j)$ was measured using the overlapping of categories of entities collected from Wikipedia. Their approach approximated the global coherence by the coherence between e_i and all other entities e_j in Γ. Since the assignment is not available during the disambiguation process, it cannot be used for the global coherence measure. Instead, they proposed to use the candidate entities of all mentions M in the document as a representation of the assignment. Their evaluation on the MSNBC news dataset showed a significant improvement over local NED approaches.

Kulkarni *et al*. [51] formalized NED as a collective optimization problem taking into account both local compatibility and global coherence. They used a supervised model trained on Wikipedia datasets to measure the local compatibility, and employed a semantic relatedness measure [69] based on in-links of entities in an EG to measure the global coherence. With these measures, NED is formalized as an optimization problem as described in Equation 2.3. Realizing that the inference solution is NP-hard, they cast the optimization into a 0/1 integer linear program and further relaxed it to a linear program which can be solved with rounding policy and hill-climbing techniques. Their collective inference solution outperformed the approaches of Cucerzan [17] and Milne and Witten [68] on both MSNBC dataset and IITB dataset.

Ratinov *et al*. [85] also treated NED as an optimization problem. They measured the compatibility by combining a few text-based features using a weighting scheme, including the similarity between the context of entities (top-200 tokens from their Wikipedia page weighted by $tf\,idf$) and the context of mentions (all tokens from their document), and semantic relatedness using both Normalized Google

Distance (NGD) and Point-wise Mutual Information (PMI) on the in-links and out-links of entities in an EG. Their algorithm used a two-stage approach in which the first stage used a linear SVM ranker to select the best candidate entity and the second stage used an SVM linker to predict if the selected candidate is the right entity. Similar to the approach of Kulkarni *et al*. [51], this approach showed the advantage of using global coherence.

AIDA [45] is a graph-based approach which casts NED as a subgraph problem. It first constructs a graph consisting of mentions and their candidate entities as nodes, and mention-entity $\langle m, e\rangle$ and the entity-entity $\langle e_i, e_j\rangle$ relations as edges. The $\langle m, e\rangle$ relation is weighted by local compatibility $\phi(m, e)$ and $\langle e_i, e_j\rangle$ relation is weighted by semantic relatedness $\psi(e_i, e_j)$. The goal of AIDA is to find a subgraph containing all mentions and entities with the one-entity-per-mention constraint and has the highest minimum weighted degree which is measured by the total weight of a node's incident edges. An approximate algorithm is proposed to solve the graph problem and is shown to outperform the approach of Kulkarni *et al*. [51].

Han *et al*. [38] proposed a graph-based approach that can make use of indirect connections between entities in an EG which are ignored in the global NED approaches discussed above. Similar to AIDA, their approach relies on a *referent graph* which is built from mentions, entities, and the mention-entity and entity-entity relations. Given the referent graph, a random walk is performed on the graph so that the interdependence between NED decisions is enforced by the iterative evidence propagation and the result of NED decisions can be propagated to other nodes in the graph. Once the random walk converges or a condition is met, the entity that can maximize the similarity $\phi(m, e) \times r(e)$ (in which $r(e)$ is the importance score of entity e accumulated during the random walk) is chosen as the true entity.

Cheng and Roth [15] further improved the accuracy of NED by exploiting relations between entities in the candidate selection and entity disambiguation through relational inference. They formalized NED as an integer linear programming (ILP) problem with the goal of finding an assignment that can maximize the global coherence and satisfy relational constraints between mentions. They derived these relational constraints through syntactico-semantic relations which were further re-

fined by matching against relational triples collected from Wikipedia and DBpedia, and co-reference relations which were extracted through text clustering techniques. By integrating the relational constraints into the ILP formalization, their system can find an assignment with better coherence to the document and outperform the above systems.

2.3.3 Topic Model-Based NED

While various types of features are explored in these NED approaches described above, many of them are heterogeneous, making it difficult to incorporate them all in one model, such as the prior probability of entities and the contextual compatibility. To resolve this problem, probabilistic models that are commonly used in various text mining tasks are employed for the NED task.

Han and Sun [36] proposed a generative mention-entity model to leverage three types of heterogeneous features. In their model, mentions are modeled as samples generated from a three-step process:

- Pick an entity from the KB according to distribution $P(e)$ (the normalized popularity of entities).
- Generate mention names according to distribution $P(m|e)$.
- Generate contexts of mentions (document) according to distribution $P(c|e)$.

The three distributions are estimated in the following way.

- $P(e) = \frac{count(e)+1}{|M|+N}$, in which $count(e)$ is the number of times m mentions e. Add-one smoothing is used here to handle the zero probability problem.
- $P(m|e) = \frac{\epsilon}{(l_e+1)^{l_m}} \prod_{j=1}^{l_m} \sum_{i=1}^{l_e} t(m_i|e_j)$, in which l_m is the length of the mention and l_e is the length of the entity name, m_i is the i-th word of m, and e_j is the j-th word of e. This estimation is based on the assumption that the name of mention m is a translation of the name of entity e using the IBM model 1 [7]

- $P(c|e) = P(t_1 t_2 \dots t_n | M_e) = P_e(t_1) P_e(t_2) \dots P_e(t_n)$, in which $M_e = P_e(t)$, $P_e(t)$ is the probability of term t in the context of entity e

The main task is then to find the entity e that maximizes the probability $P(m, c, e)$ as follows:

$$e = \arg\max_e P(m, c, e) = \arg\max_e P(e) P(m|e) P(c|e)$$

Han and Sun [37] also proposed another generative entity-topic model based on two assumptions: *topic coherence assumption* which assumes that entities in the assignment should be centered around a main topic of the document, and *context compatibility assumption* which assumes that the context of mentions should be consistent with that of their true entities. The generative process in their model requires the following knowledge to be estimated from a document corpus: *topic knowledge*, in which each topic is modeled as a multinomial distribution of entities, *entity name knowledge* which is a multinomial distribution with the probability indicating how likely an entity is mentioned by a name (*i.e.*, alias); and *entity context knowledge* which models the context of entities as a multinomial distribution of words. Then a document is generated as follows: first a topic distribution of the document is generated, then an entity assignment is generated from the document's underlying topics and the knowledge which ensures the topic coherence, at last words in the document are generated from the entity context knowledge and the entity assignment ensuring the context compatibility. To estimate the global knowledge from the document corpus, a Gibbs sampling algorithm is proposed by extending the Gibbs sampling algorithm using a Latent Dirichlet Allocation (LDA) model. Once the global knowledge is estimated, the entity assignment generated from the inference will be the final assignment Γ to mentions in the document.

2.3.4 Neural Network Based NED

More recently, as neural networks (NNs) show their potential in various NLP tasks [16], researchers start to explore NN-based models for NED. He *et al.* [42] was one of the first few to apply deep neural network (DNN) to the NED task. Their

method first automatically learned semantic representations of entities using a denoising auto-encoder on the Wikipedia corpus and then stacked another layer on top of the learned representation to learn a similarity measure for local compatibility. Experiments showed that their DNN-based approach can outperform a few global approaches using only local compatibility.

Sun *et al*. [95] built positional information of words in the context of mentions into their model, and used convolutional neural network (CNN) to learn the semantic representations of mentions, their context, and entities, which were then used to measure the compatibility between mentions and its candidate entities to rank and disambiguate entities.

Francis-Landau *et al*. [24] also used CNN to map semantic of mentions and entities into a continuous vector space. Moreover, their approach measured three different granularities of semantics for each mention: the mention itself, the surrounding texts of the mention, and the document containing the mention. For entities, semantics were measured on the entity name and the describing document of the entity. Their disambiguation algorithm then used the continuous vector representations to measure the compatibility between mentions and their candidate entities. Nguyen *et al*. [77] extended the work of Francis-Landau *et al*. [24] by using a recurrent neural network (RNN) to incorporate disambiguation results from previous iterations into each disambiguation decision to enforce the topic coherence assumption.

Yamada *et al*. [107] proposed a new embedding method extended from the Skip-Gram model [67], which can incorporate the structure of entity graphs into an entity model. Besides, they also built a word-entity model to unify the word and entity models. By combining the word, entity, and word-entity models, their model can map words and entities into the same continuous vector space, which can be used to measure both local compatibility and global coherence. Their NED approach using this model can achieve state-of-the-art accuracy.

Besides the describing text of entities, Gupta *et al*. [34] also considered surrounding context of entities (obtained from the context of mentions linking to those entities) and types of entities in their model to learn embeddings of entities. Phan *et al*. [80] further improved the approach using two long short-term memory (LSTM)

networks to model the positional information of mentions and the ordering of words. They also applied attention mechanism in their model to handle noises in context. Experiments showed that the positional information and word ordering can improve the accuracy by 5% to 10% on their evaluation datasets.

2.3.5 Online NED

While datasets from Newswire or Encyclopedia are the focus of most state-of-the-art NED systems, datasets from social sites or online forums are gaining more attention and need to be disambiguated in many applications, such as mining tweets for political preference prediction [59] and analyzing online reviews for business intelligence [49]. Moreover, documents from these datasets are more ambiguous and could pose more challenges for NED because of the informal expression, noisy text, and short context.

TAGME [22] was the first system to perform NED on short texts. Instead of using expensive collective inference, a simple scoring function was proposed to improve the efficiency. For each candidate entity e of mention m, TAGME measured its local compatibility to m and its global coherence to other entities in Γ through a voting scheme based on the semantic relatedness between entities.

Guo *et al*. [29] treated the mention extraction and mention disambiguation tasks on tweets as an end-to-end task, and employed a structured SVM algorithm to handle the two tasks jointly. They explored local features, such as *capitalization rate, popularity, entity type*, and $tf\,idf$, and semantic features from the neighboring entities in an EG. They also explored the voting strategy in TAGME [22] and showed its effectiveness in NED.

Meij *et al*. [64] performed NED on tweets in two steps: a high-recall ranking step to rank candidate entities of a mention, and a machine learning based re-ranking step to re-rank the candidates using various features. Among the machine learning algorithms they evaluated, Random Forest achieved the best accuracy compared with Naive Bayes, C4.5, SVM, and GBRT.

2.3.6 NED to Generic KBs

While specific features can be derived from different KBs and used to further improve the accuracy of NED, models learned using these features are often not portable from one KB to another. Thus, approaches that exploit only common features from different KBs are preferred in many cases.

Motivated by this requirement, Sil *et al*. [90] proposed approaches to disambiguate mentions of named entities to generic KBs. Instead of using domain-specific features, they selected only two common features: the number of occurrences the attributes of an entity e appear in the surrounding context of mention m, and the number of occurrences *similar* entities of entity e appear in the surrounding context of mention m. Here similar entities of e are entities sharing some common attributes with e. Using the two features, a distance supervision based approach was proposed for the NED task. Results on a movie KB and a sports KB showed the superiority of this generic approach over some state-of-the-art NED algorithms [113]. While adding domain-specific features (*e.g*., Wikipedia-derived features) can significantly improve the accuracy of their system (from 61% to 69%), their work demonstrated the benefits of leveraging other KBs other than Wikipedia and showed the portability of these features and models on different KBs.

2.4 Evaluation of NED Systems

In this section, we will briefly introduce public benchmarks and commonly used metrics for NED evaluation.

2.4.1 Evaluation Datasets

Wikipedia is one of the best resources with annotation data available, thus is commonly used to generate test datasets for the NED task. Besides Wikipedia, many other benchmarks are also constructed and released for public use. Cucerzan [17] manually annotated 20 news articles from 10 MSNBC news categories, such as *business*, *politics*, and *sports*, and created the *MSNBC* dataset, in which mentions are explicitly provided and annotated with entities in Wikipedia (all mentions have

their corresponding entities). Kulkarni *et al*. [51] also built the IITB dataset from web pages of popular sites using a browser-based annotation system. The IITB dataset contains more ambiguous samples and also NIL annotations. Milne and Witten [68] and Ratinov *et al*. [85] used a subset of 50 documents from the AQUAINT corpus of newswire to build the AQUAINT benchmark. Ratinov *et al*. [85] also used Amazon's Mechanical Turk to construct the ACE2004 dataset from a subset of the ACE co-reference dataset which contains mention types and has the co-reference resolved. Hoffart *et al*. [45] created a large dataset AIDA-CoNLL[2] based on the CoNLL 2003 data, consisting of annotations for 1393 Reuters newswire articles and $34,956$ mentions. In addition to Wikipedia, the CoNLL dataset is also annotated with entities from YAGO2 and Freebase when available, thus can be used for NED against YAGO2 and Freebase.

Another set of public datasets for NED are provided by the TAC-KBP shared task [3], which are collected mainly from sources like newswire and blogosphere and focus on persons, organizations, and locations.

2.4.2 Evaluation Metrics

Results of an NED system can be divided into four sets:

- *True Positive (TP)*: Mentions correctly linked to their entities in the KB.
- *True Negative (TN)*: Mentions correctly linked to NIL
- *MisMatch (MM)*: Mentions whose true entities exist in the KB, but incorrectly linked to other entities in the KB.
- *False Positive (FP)*: Mentions that should be linked to NIL, but linked to entities in the KB.
- *False Negative (FN)*: Mentions that should be linked to entities in the KB, but linked to NIL.

[2] `http://www.mpi-inf.mpg.de/yago-naga/aida/download/aida-yago2-dataset.zip`

[3] `https://tac.nist.gov/2017/KBP/`

The metrics used to evaluate NED systems are defined as follow:

$$\begin{aligned} accuracy &= \frac{TP+TN}{TP+FP+TN+FN+MM} \\ precision &= \frac{TP}{TP+FP+MM} \\ recall &= \frac{TP}{TP+FN+MM} \\ F1 &= 2 \times \frac{precision \times recall}{precision + recall} \end{aligned}$$

Other than these standard metrics which measure on each mention, *Bag of Title* (BOT) is another metric that evaluates NED results at the document level instead of the mention level so that duplicates in one document will be ignored.

Since NED is commonly formalized as a ranking problem, measures from the information retrieval community can also be used to measure NED approaches. *Mean reciprocal rank (MRR)*, which calculates the average of the reciprocal rank of the true entity among all candidates of the mentions, is an appropriate measure for NED, given that NED concerns only the true entity instead of all candidates. Assuming that the rank position of the true entity for each mention m_i is $rank_i$, then the *MRR* of a document will be computed as follows:

$$MRR = \frac{1}{|M|} \sum_{i=1}^{|M|} \frac{1}{rank_i}$$

Here M is the set of mentions in a document and $|M|$ is the number of mentions.

2.4.3 Summarized Results of NED Systems

Table 2.3, Table 2.4, and Table 2.5 summarize the evaluation results of various NED approaches on different datasets. For methods that report multiple results, the most representative one is chosen here. For example, if a method reports precision, recall, and F1, only the F1 will be listed.

Since it is easy to generate test datasets from Wikipedia, various NED systems are evaluated using Wikipedia datasets. Table 2.3 lists the reported results of NED systems on Wikipedia-generated datasets. Note that the test datasets for most systems are different from each other, although they are collected from Wikipedia.

Measure	Method	Baselines
Accuracy	[9] - 84.8	82.3 - Bag-of-Words
	[17] - 88.3	86.2 - String matching+Commonness
	[36] - 80.0	60.0 - [17] 66.0 - [61] 70.0 - [68]
F-1	[66] - 87.7	60.2 - Random selection 82.0 - Commonness 76.0 - Bag-of-Words
	[68] - 97.1	53.1 - Random selection 90.7 - Commonness 92.9 - [61]
BOT	[85] - 90.5	81.8 - Commonness 80.3 - [68]
	[15] - 93.1	80.3 - [68] 90.5 - [85]

Table 2.3: Reported experimental results on datasets derived from Wikipedia.

Corpus	Measure	Method	Baselines
MSNBC	Accuracy	[17] - 91.4	51.7 - String matching+Commonness
		[19] - 94.7	91.4 - [17]
	F-1	[51] - 69.0	63.0 - [68]
		[113] - 87.6	62.3 - Random selection 84.2 - Commonness 81.7 - [17]
		[10] - 77.7	62.9 - [17] 55.5 - [51] 66.0 - [85]
	BOT	[85] - 74.9	72.8 - Commonness 68.5 - [68]
		[15] - 81.2	68.5 - [68] 74.9 - [85]
AQUAINT	BOT	[85] - 83.9	82.7 - Commonness 83.6 - [68]
		[15] - 88.9	83.6 - [68] 83.9 - [85]
ACE2004	BOT	[85] - 77.3	69.5 - Commonness 72.8 - [68]
		[15] - 85.3	72.8 - [68] 77.3 - [85]

Table 2.4: Reported experimental results on popular benchmarks.

Thus it is not fair to directly comparing the methods based on the reported results. Also, the results of baselines reported in each NED system are either based on re-running of the baseline systems if publicly available (e.g. [68]) or based on the

Corpus	Measure	Method	Baselines
IITB	F-1	[51] - 69.7	51.8 - [17]
		[38] - 73.0	37.0 - [66] 45.0 - [17] 52.0 - [68] 69.0 - [51]
		[37] - 80.0	37.0 - [66] 52.0 - [68] 69.0 - [51] 73.0 - [36]
		[10] - 79.1	49.9 - [17] 63.1 - [51] 48.9 - [85]
TAC-KBP-09	Accuracy	[111] - 83.8	61.9 - Bag-of-words+NE
		[19] - 79.4	77.0 - [17]
		[36] - 86.0	72.0 - [17] 80.0 - [61] 83.0 - [68]
CoNLL	Accuracy	[45] - 81.8	65.8 - Commonness 51.0 - [17] 72.9 - [51]

Table 2.5: Reported experimental results on custom benchmarks.

re-implemented systems. Some systems have results reported using other metrics, and in the table we report only the results with the most commonly used metrics.

Two observations are in order here: first, a multitude of performance measures have been used in the literature; second, most systems are not comparable, even on the same datasets. As shown in the tables, most results of baselines are not comparable with the results reported in their original papers. Besides the parameter setting, the Wikipedia corpus used in each NED system may be different and affect the results. For example, the model in [68] was trained using a 2007 Wikipedia dump, and may not fit for other Wikipedia dumps. Another main reason is the candidate selection process. To the best of our knowledge, no two NED systems are using the same candidate selection approach. As shown in [35], candidate selection can significantly affect the accuracy of an NED system since the recall of candidate selection defines the upper bound of the accuracy of an NED system. These issues make it hard to conduct fair comparisons of NED approaches, thus call for a general evaluation framework for NED.

2.5 Closely-Related Problems

In many ways, NED resembles other *de-duplication* problems. Entity Resolution (*a.k.a*, Record Linkage) [5] is one of them, although its focus is on matching entire (semi-)structured data records from disparate sources, typically in the context of data integration. Word Sense Disambiguation (WSD) [74] and co-reference resolution [75] are two related problems from NLP, and many ideas proposed for WSD can also be applied to the NED task [72]. Another closely related problem is annotating structured tables on the Web [56], [100], [102] for which surrounding context of mentions is not available for disambiguation.

Chapter 3

Candidate Selection

In this chapter, we study the problems in the candidate selection task. More specifically, we give a review of the current candidate selection approaches, which include name expansion methods to find the full names of mentions and aliases of named entities, and effective pruning criteria to filter out irrelevant candidates. We then perform an experimental evaluation on the impacts of these approaches, from which we find a solution for our NED system that can balance the recall of candidate selection and cardinality of candidate sets.

3.1 Overview

Intuitively, NED is to match all mentions from a document against all entities in a KB and find the best-matched pairs using a set of matching criteria. This all-against-all strategy, however, is very inefficient since most entities in a KB are irrelevant to the given mentions either lexically or semantically, resulting in a large number of unnecessary comparisons, especially with the growing size of most KBs (*e.g.*, DBpedia has over 5 million entities). Candidate selection aims to address this efficiency issue by restricting the comparisons between the mention and those selected candidates.

One evaluation metric for candidate selection is *recall*, which is the ratio of mentions whose true entities are in their candidate set. High recall of candidate selection is crucial for NED since it determines the upper bound accuracy of the system. However, it could potentially bring in more noisy candidates (the candi-

dates that are irrelevant to the mention), resulting in high *cardinality* of candidate sets, which will affect the efficiency of an NED system: the higher the cardinality of a candidate set is, the more computations the system will perform. Therefore, a good candidate selection system should strike a balance between two factors: (1) maximizing the recall; and (2) minimizing the cardinality of candidate sets. Building a system like this has to face a few challenges, as discussed below.

Name Variation The first challenge is the name variation. Many entities are commonly mentioned by nicknames instead of their canonical names (*e.g.*, *Edmonton* is known as *The City of Champions*). Acronym is another example. For instance, *ABC* refers to over 120 entities in Wikipedia[1]. Besides the *All Basotho Convention* political group and the *Artificial Bee Colony* algorithm, *ABC* is also an abbreviation of two broadcasting corporations: *American Broadcasting Company* and *Australian Broadcasting Corporation*. With these variations and constant invention of alternative names, building an exhaustive alias dictionary to include all aliases of each entity is unrealistic.

Noisy Candidates As described above, methods that improve the recall of candidate selection might bring in irrelevant noisy candidates and affect the efficiency or the accuracy of an NED system. A pruning step, in this case, is needed to clean the candidate sets. With the requirements of being *lightweight* and *effective*, it is challenging to find pruning criteria that can balance the effectiveness and the complexity of candidate selection.

Most approaches perform candidate selection in two steps, with the first step to select all potential candidate entities referred to by a mention, and the second step to prune noisy entities from those selected candidates. Various methods are proposed to address challenges in each step, including *name expansion* methods and *candidate pruning* methods. Name expansion methods address the name variation issue by expanding shortened names or acronyms to their full names in the same document using techniques like co-reference resolution, or building an alias dictio-

[1] `https://en.wikipedia.org/wiki/ABC`

nary from external corpora for entities in the KB. Candidate pruning methods solve the noisy candidate issue using effective pruning criteria. In the following sections, we will review the current methods for each step.

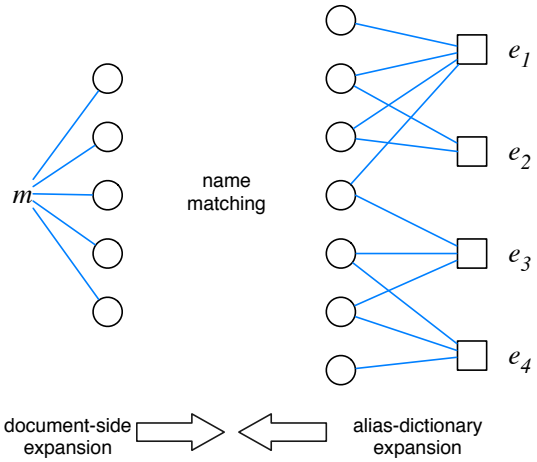


Figure 3.1: Name expansion for candidate selection. Mention m is expanded with alternative names from its document, while entity names are expanded with entries in an alias dictionary.

3.2 Name Expansion

Name expansion is a step to *expand* the set of lemmas for both mentions and entities. As illustrated in Figure 3.1, with expanded name sets from both the document and KB sides, we can improve the recall of candidate selection by matching the name sets using exact or fuzzy string matching. A number of methods are proposed for the name expansion, and several of them can be applied on both mentions and entities (*e.g.*, using external dictionaries of acronyms and nicknames). When disambiguating mentions against Wikipedia entities, an alias dictionary, which associates different names with entity ids, turns out to be straightforward and useful. We discuss each method next.

3.2.1 Name Expansion with Alias Dictionary

Alias dictionaries are common resources for name expansion. At the time of writing, the largest alias dictionary, built from the WikiLinks Corpus [91] by mining the anchor text in links from Web pages to Wikipedia articles, consists of 40M aliases

to 2.9M entities. Most NED approaches use Wikipedia to build alias dictionaries, as described in Chapter 1. Besides Wikipedia, there are also other high-quality alias resources for specific types of entities. For example, the Intelius Nickname Collection [13], regarded as the largest database of nicknames, is particularly useful to collect aliases for *person*.

In Wikipedia, among the 4 alias sources described in Section 1.1.1, page titles and redirect pages are the most reliable ones, since they refer to internal article ids that uniquely identify entities in the KB. According to Cucerzan [17], anchor texts from WikiLinks, such as "[[Texas (TV Series)|Texas]]", can add quite a lot of noise to the alias dictionary. However, these WikiLinks also provide extra value to estimate the popularity of entities by their number of times being mentioned in Wikipedia.

Besides the pages and WikiLinks, there are also other ways to collect aliases from Wikipedia. Guo *et al*. [30] extract nicknames of entities from Infoboxes and also through patterns based on a guideline in Wikipedia: "the names of an article's subject [to be] written in **bold** when they are first mentioned in the article". For example, *Marie Curie*'s aliases are in the first paragraph: "**Marie Skłodowska-Curie**, often referred to as **Marie Curie** or **Madame Curie**"[2]. While apparently not well explored in the literature, one could envision using NLP tools to identify nicknames from other sections of Wikipedia articles. For instance, the second paragraph in the article about *Michael Jordan* states, at the time of writing, that his performance "earned him the nicknames *Air Jordan* and *His Airness*". Zhang *et al*. [111] and Zhang *et al*. [110] use the **Did You Mean** and the **Wikipedia Search Engine** services to find additional aliases from Wikipedia. With **Wikipedia Search Engine**, we can complement the candidates from the search results by matching query strings against the top-K entities using string similarity measures, such as the longest common subsequence [110], [111].

Table 3.1 lists the sources used by different systems to build alias dictionaries.

[2] `https://en.wikipedia.org/wiki/Marie_Curie`

	Wikipedia				
	Title Pages	Redirect Pages	Disamb. lists	Anchor text	External resources
Bunescu and Paşca [9]	✓	✓	✓		
Cucerzan [17]	✓	✓	✓	✓	
Mihalcea and Csomai [66]				✓	
Milne and Witten [68]	✓	✓			
Kulkarni *et al*. [51]	✓	✓	✓		
Zhang *et al*. [111]	✓	✓	✓	✓	Wikipedia Service
Zhou *et al*. [113]	✓	✓		✓	Web Search Engine
Dredze *et al*. [19]	✓	✓			Freebase
Zheng *et al*. [112]	✓	✓	✓	✓	
Hoffart *et al*. [45]	✓	✓	✓	✓	DBPedia/YAGO
Ratinov *et al*. [85]				✓	
Han *et al*. [38]				✓	
Han and Sun [36]	✓	✓	✓	✓	
Zhang *et al*. [110]	✓	✓	✓	✓	
Cheng and Roth [15]				✓	
Cai *et al*. [10]	✓	✓	✓		

Table 3.1: Sources used in the candidate selection component of different NED systems.

3.2.2 Name Expansion for Mentions

A mention is ambiguous when it could potentially refer to multiple entities, and becomes more ambiguous when it is a shortened name or acronym of entities. *Michael* or *MJ*, for example, could refer to *Michael Jordan* – the basketball player or the machine learning researcher, or *Michael Jackson* – the deceased pop singer. Fortunately, the full names of most shortened names or acronyms is commonly mentioned in the same document. Correctly identifying the full name of those mentions could greatly reduce the ambiguity and potentially improve the accuracy of NED. For instance, if *Michael* or *MJ* can be linked to *Michael Jackson*, we can then narrow down their candidates to entities only referred to by *Michael Jackson*. Below we first discuss the name expansion for acronyms and then the shortened names.

Acronym Expansion

Acronym, an abbreviated form of entity names, is another type of ambiguous mentions, for which acronym expansion is the main approach to resolve the ambiguity.

Many approaches have been proposed for acronym expansion, using syntactic patterns, such as "Acronym (full name)" and "full name (Acronym)", to extract their full names from the surrounding text [30]. For the case that the full name is not adjacent to the acronym, Zhang *et al*. [109] employ a heuristic approach. It first retrieves the matching candidates of each acronym from an alias dictionary, and then goes through the document searching for occurrences of either the candidate names or their known aliases (which are available from the alias dictionary). The first match found in the document is selected as the final expanded name.

Zhang *et al*. [110], [111] handle both the simple and complicated cases in a uniform way using a machine learning approach. Their approach first builds a list of candidates using the simple patterns mentioned above, and then searches for tokens in the document whose first letter matches the first letter of the acronym. Once a match is found, they add the longest continuous sequences of tokens with no punctuations and no more than two stop words to a candidate list. Next, an SVM classifier is employed to select the candidate with the highest confidence. The experiment shows a recall of 92.9% on the KBP-2010 dataset, which, in their tests, improves the accuracy of NED from 76.1% to 82.8%. Anastácio *et al*. [1] also handle both cases with results showing that name expansion could reduce missed candidates by more than 50%, and improve the overall accuracy of NED by 4%.

Shortened Name Expansion

Shortened name, similar to acronym, is another source of ambiguity for entities. The main approach for shortened name expansion is intra-document co-reference resolution [17], [101]. Radford *et al*. [83] employ a naive co-reference resolution which matches a mention to its previous mentions and chooses the one with extract string match or right-aligned name match. Ratinov and Roth [84] use some simple string matching rules to discard professional titles and honorifics. Besides, they also employ case insensitive and punctuation insensitive matching rules for acronym expansion. Lehmann *et al*. [52] use string similarity to recursively expand mentions, which, in combination with full-text search over Wikipedia, can achieve a 97% recall on the TAC-KBP 2009 dataset. Another strategy, proposed by Zheng *et*

al. [112], uses automatic spelling correction to improve the results of name expansion. The *entity linking* task in TAC [48] has also motivated many name expansion methods aiming to improve the recall of candidate selection.

3.3 Candidates Generation

With expanded mention names and an enriched alias dictionary, the next step is to generate a set of candidates for each mention through name matching, as shown in Figure 3.1. Most approaches apply exact string matching against a look-up table to retrieve all entities associated with any expanded name of a mention [17], [36], [38], [39], [45], [51], [66], [68], [85], [110], [111], [113]. Radford *et al*. [83] further refine the candidate generation by taking into account the reliability of alias sources (from Wikipedia), defined as follows: *title* $>$ *redirect* $>$ *bold words* $>$ *partial title match* $>$ *disambiguation* $>$ *link anchor text*. Ploch *et al*. [82] also use a similar idea with weighted alias sources. Dredze *et al*. [19] take a step further when no candidates can be found for a mention in the alias dictionary: they select entities (corresponding to Wikipedia pages) among the top 20 search results from querying the Web with the mention name.

One issue with the look-up table method is that true candidates could be filtered out because of the name variations or typos not captured in the name expansion and alias dictionary. To solve this issue, Ploch *et al*. [82] propose to use fuzzy matching over the alias dictionary, on which an inverted index is built to support queries, and choose the top-k entities from a query as candidates. As is often the case, a balance must be struck when tuning such approaches to prevent noise from negatively affecting the results.

3.4 Candidate Pruning

While the methods described above can significantly improve the recall of candidate selection, each of them, however, could increase the cardinality of candidate sets with additional noisy candidates, which will affect the efficiency of the NED system. Candidate pruning aims to solve this issue with effective pruning criteria.

Several criteria are proposed for this issue. Prior probability $prior(m, e)$, which measures the probability of entity e being the true entity of mention m using the frequency of the mention to entity mapping in the alias dictionary, is a commonly used one to rank and select candidates. Context similarity is another one using the similarity between the lexical context of a mention and its candidates, such as the surrounding words, named entities, or keyphrases [44]. With these criteria, a system can rank candidate entities by their similarity to the given mention, and select the top K candidates with the rest pruned.

3.5 Experimental Evaluation

In this section, we experimentally evaluate the impacts of alias sources, name expansion methods, and candidate pruning criteria on the performance of candidate selection. We measure the context similarity using indexes built with Apache Lucene[3]. For co-reference resolution, we employ the system ANNIE in Gate 8.1[4].

Datasets For the evaluation, we choose 4 widely used benchmarks: (1) MSNBC [17], with 20 news articles from 10 different topics (two articles per topic) and 656 linkable mentions in total; (2) AQUAINT, compiled by Milne and Witten [68], with 50 documents and 727 linkable mentions from a news corpus from the Xinhua News Service, the New York Times, and the Associated Press; (3) ACE2004 [85], a subset of the ACE2004 Coreference documents with 35 articles and 257 linkable mentions, annotated through *crowdsourcing*; and (4) AIDA-CoNLL [45], a hand-annotated dataset based on the CoNLL 2003 data, with 1388[5] Reuters news articles and 27817 linkable mentions.

Evaluation Metric We measure the accuracy of candidate selection using *recall* and the effectiveness of candidate pruning criteria using $Recall@K$, which is defined as the percentage of mentions with their referent entities in the top K candi-

[3] `https://lucene.apache.org/`

[4] `https://gate.ac.uk/ie/annie.html`

[5] The original dataset includes 5 other documents where all mentions are linked to NIL, and are therefore removed from our analysis.

dates ranked by a criterion. In our experiments, we report the results with K set to 1, 5, 10, 20, 50, 100, and 200.

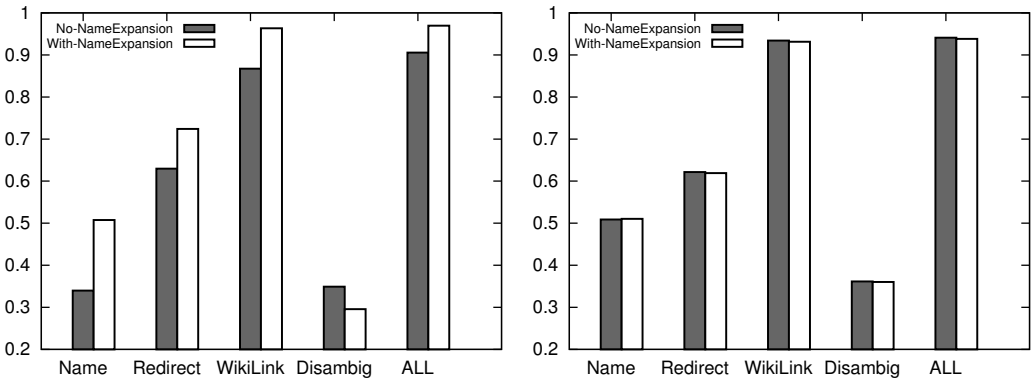
3.5.1 Impact of Alias Sources

We first evaluate the contribution of each alias source to the recall of candidate selection. In the experiment setting, we use exact name matching for candidates selection *without* candidate pruning.

Figure 3.2 shows the recall of candidate selection using aliases from 4 sources with (white bar) and without (gray bar) name expansion. As we see, aliases from Wikipedia titles can only help achieve around 33% to 54% recall, indicating that at least half of the entities in the datasets are not mentioned by their canonical names. Among the 4 sources, *WikiLink* gives the highest coverage of aliases, corresponding with the fact that WikiLink is the major source of aliases. With all 4 sources used, we can achieve over 90% recall on all datasets and over 97% on AIDA-CoNLL.

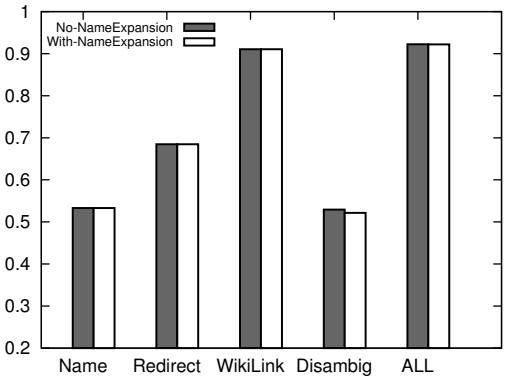
3.5.2 Impact of Mention Name Expansion

We employ co-reference resolution to help expand shortened names and acronyms. Figure 3.2 (white bar) gives the recall of candidate selection using name expansion (through co-reference resolution). Compared to results without using name expansion (gray bar), we can see a slight decrease of recall with name expansion on all datasets except MSNBC. One reason is the errors from name expansion. For example, *NYSE*, an acronym of *New York Stock Exchange*, is mistakenly expanded to *New York Mercantile Exchange*. We also find that correct name expansion could potentially harm the recall due to the limited coverage of alias dictionaries. For instance, *Nardelli*, although correctly expanded to *Bob Nardelli*, still fails to find its true entity *Robert Nardelli* since *Bob Nardelli* is not in the alias list of *Robert Nardelli*. For the MSNBC datasets on which the recall is improved from 90.55% to 96.95%, a further analysis reveals that the improvement mainly comes from persons mentioned by their first names but expanded to their full names via co-reference resolution. Thus name expansion could be considered as an effective method in candidate selection when handling datasets with a lot of person names.

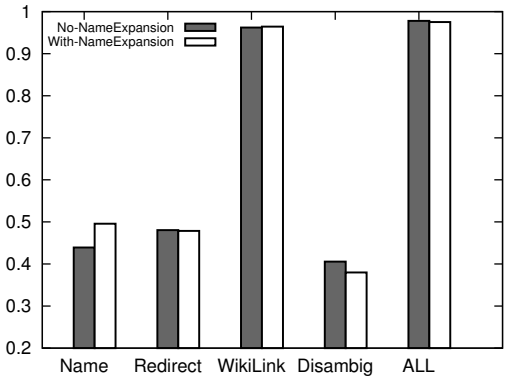


(a) Recall on MSNBC dataset

(b) Recall on AQUAINT dataset



(c) Recall on ACE2004 dataset



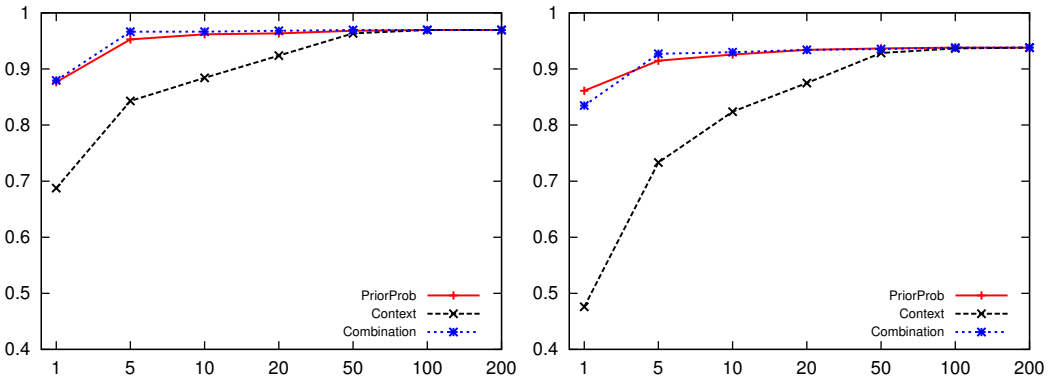
(d) Recall on AIDA-CoNLL dataset

Figure 3.2: Recall of candidate selection using different alias sources with (white) and without (gray) name expansion.

3.5.3 Impact of Candidate Pruning

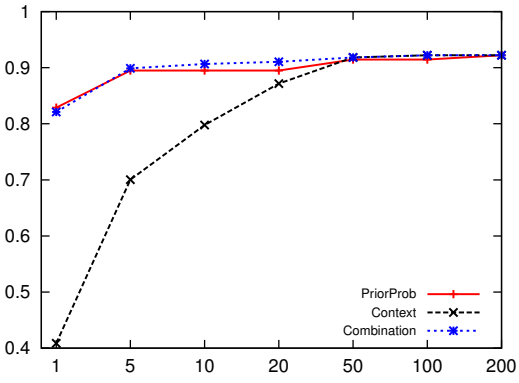
In this experiment, we evaluate the impact of different pruning criteria on the performance of candidate selection. We report the *Recall@K* using 3 criteria: *prior probability*, and *local context similarity*, and their *combination* which is defined as $w\ prior(m, e_i) + (1-w)ctxSim(m, e_i)$ with the weight w set to 0.3 experimentally.

Each figure in Figure 3.3 shows *Recall@K* on one of the 4 datasets. $Recall@1$ gives the recall of choosing the top 1 candidate, which is equivalent to the accuracy of NED based on prior probability. As we can see, a prior probability based approach can serve as a strong baseline for NED. Another observation from using prior probability for pruning is that we can achieve over 90% recall on most datasets when increasing the number of candidates of each mention to 5. We also find that the recall can be improved with more candidates selected, but not significantly.

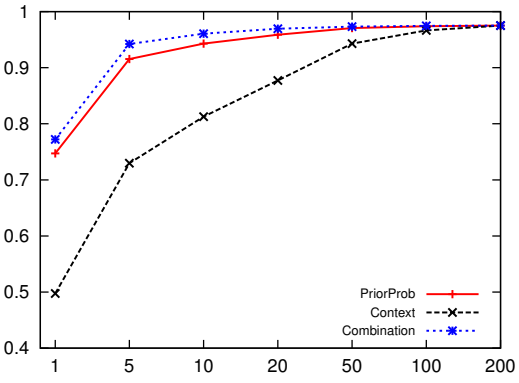


(a) MSNBC

(b) AQUAINT



(c) ACE2004



(d) AIDA-CoNLL

Figure 3.3: Recall of candidate selection using different pruning criteria.

The continuous increase of the recall, when increasing K, only happens on *AIDA-CoNLL*, among the 4 datasets. One reason is that *AIDA-CoNLL* is a much larger dataset with more variations. Compared to prior probability, context similarity is only a moderate criterion when limiting K to any value less than 10.

We can see that both criteria contribute to the recall from the results of *combination*. For example, $Recall@5$ using *combination* is higher than that using *prior probability* or *context similarity* alone. We also notice that the results of $Recall@1$ increase on two datasets and decrease on the other two, indicating that *context similarity* does not always help with candidate selection. Thus, *prior probability* could be a good trade-off when efficiency is a concern for NED.

3.6 Summary

In this chapter, we reviewed and experimentally evaluated the state-of-the-art methods for the *candidate selection* task. We found that most entities are not mentioned by their canonical names, but other alternative names. We also found that name expansion is more helpful for persons than other types of entities, and it could potentially decrease the recall due to expansion errors and incomplete coverage of alias dictionaries. At last, for candidate pruning, we evaluated 3 pruning criteria. The results showed that *prior probability* is a much better criterion than *context similarity*, and combining them can further improve the recall. In the following work, we will use the alias dictionary constructed using all 4 sources, the co-reference resolution for name expansion, and the combined criteria for candidate pruning with K set to 10.

Chapter 4

NED with Random Walks

In this chapter, we consider exploiting the connectivity in entity graphs for the *mention disambiguation* task. We propose a novel method to compute *semantic signatures* of entities and documents using the random walk with restart on a disambiguation graph. We use the semantic signatures to measure semantic relatedness between entities and global coherence between entities and documents. We also introduce an iterative disambiguation algorithm, which as demonstrated in our experimental evaluation, can achieve state-of-the-art accuracy.

4.1 Overview

The mention disambiguation problem, as formalized in Chapter 1, is to find an assignment $\Gamma : M \rightarrow E \cup \{\mathsf{NIL}\}$, which maps the mentions in M to their true entities in E of an entity graph EG or NIL if the entities are not in E.

A good assignment Γ balances two factors: the *local compatibility* between mention m_i and the entity $e_j = \Gamma(m_i)$ assigned to m_i, and the *global coherence* $\Psi(\Gamma)$ among the entities in the assignment.

As usual, we define the local compatibility $\phi(m_i, e_j)$ as:

$$\phi(m_i, e_j) = \alpha\, prior(m_i, e_j) + (1 - \alpha) ctxSim(m_i, e_j) \tag{4.1}$$

where $prior(m_i, e_j)$ is a corpus prior probability of e_j being the referent entity of m_i, and $ctxSim(m_i, e_j)$ is the similarity between local features (*e.g.*, keywords) extracted from the surrounding text of m_i and the describing document of e_j.

The global coherence $\Psi(\Gamma)$ of the assignment measures how each entity in the assignment relates to the others:

$$\Psi(\Gamma) = \sum_{e \in \Gamma[M]} \psi(e, \Gamma) \tag{4.2}$$

in which $\psi(e, \Gamma)$ measures the semantic relatedness between an entity e and other entities in the assignment Γ. Maximizing the sum in Eq. 4.2 is consistent with the *topic coherence assumption*.

Under the reasonable assumption that the local compatibility is normalized, we can formulate the NED problem as a min-max optimization where the goal is to *maximize* the global coherence while minimizing the *loss* in pairwise local compatibility within the assignment, which can be estimated as $|M| - \sum_{m_i \in M} \phi(m_i, \Gamma(m_i))$. Here $|M|$ is the number of mentions in the document. An equivalent and simpler formulation of the problem is to find an assignment Γ^* that maximizes:

$$\Gamma^* = \arg\max_{\Gamma} \left(\Psi(\Gamma) \cdot \sum_{m_i, e_j \in \Gamma} \phi(m_i, e_j) \right) \tag{4.3}$$

The primary role of $\Psi(\Gamma)$ in the optimization above is to leverage connections between entities in the EG to prevent disambiguation mistakes caused by disproportionately high priors of some candidate entities. For example, *Karl Malone* has a higher prior than *Jeff Malone* and thus would be incorrectly assigned to mention *Malone* in Example 1; however, once *Washington Bullets* is disambiguated, $\Psi(\Gamma)$ will *counter* the effect of the high prior because *Jeff Malone* is directly connected to the team in the EG.

Since solving Eq. 4.3 is NP-hard in general [45], [51], our approach uses a greedy and iterative disambiguation algorithm: in each iteration, we re-compute the semantic signature of the document d, and disambiguate a mention m according to:

$$\Gamma(m) = \arg\max_{e_i \in cand(m)} \left(\phi(m, e_i) \cdot \psi(e_i, d) \right) \tag{4.4}$$

where $cand(m)$ are the candidates for m, and $\psi(e_i, d)$ is the semantic relatedness measure. In the next iteration, the entity e_i linked to the mention m will be taken into account when re-computing the semantic signature of the document. By doing so, we guide the search and increase the coherence of the resulting assignment.

4.2 Semantic Signatures

As defined, a disambiguation graph consists of entities and relations between entities, in which the graph connectivity can be used to measure relatedness between entities as follows. Let $G = (V, E)$ be a graph and let $\bar{V} \subseteq V$ be a set of vertices such that $|\bar{V}| = n$. The semantic signature of a vertex $v \in \bar{V}$ is an n-dimensional vector where the weight of index i is the *relatedness* between v and the vertex i in $\bar{V}$. In this work, relatedness is defined as the probability that node i is visited in a random walk process restarting at vertex v. We call each n-dimensional vector the *semantic signature* of the vertex v.

The notion of signature extends naturally to a *set* of k vertices of $\bar{V}$: perform a random walk with restart from the k vertices and consider the resulting probability distribution over the n vertices. Thus, we can obtain the semantic signature of a *document* by performing random walks from the entities that are mentioned in it.

Figure 4.1 illustrates the idea of the semantic signature of entities and documents.

	e_1	...	e_i	...	e_j	...	e_n
e_1	w_{11}	...	w_{1i}	...	w_{1j}	...	w_{1n}
$\vdots$							
e_i	w_{i1}	...	w_{ii}	...	w_{ij}	...	w_{in}
$\vdots$							
e_j	w_{j1}	...	w_{ji}	...	w_{jj}	...	w_{jn}
$\vdots$							
e_k	w_{k1}	...	w_{ki}	...	w_{kj}	...	w_{kn}
d	w_{d1}	...	w_{di}	...	w_{dj}	...	w_{dn}

Figure 4.1: Semantic signatures of entities and documents.

4.2.1 Random Walks with Restart

A random walk with restart is a stochastic process to traverse a graph, resulting in a probability distribution over the vertices corresponding to the likelihood those vertices are visited. This probability can be interpreted as the relatedness between nodes in the graph. A random walk starts with an initial distribution over the nodes

in the graph, propagating the distribution to adjacent vertices proportionally, until convergence.

Let A be the transition matrix of the disambiguation graph, with A_{ij} being the probability of reaching entity e_j from entity e_i, which can be computed as follows:

$$A_{ij} = \frac{w_{ij}}{\sum_{e_k \in OUT(e_i)} w_{ik}}$$

in which, $OUT(e_i)$ is the set of entities directly reachable from e_i, and w_{ij} is the weight of the edge between e_i and e_j.

Let r^t be the probability distribution at iteration t, and r_i^t be the value of entity e_i, then r_i^{t+1} is computed as follows:

$$r_i^{t+1} = \sum_{e_j \in IN(e_i)} r_j^t * A_{ji} \tag{4.5}$$

in which $IN(e_i)$ is the set of entities linking to e_i.

As is customary, we incorporate a random restart probability in the *preference vector* to avoid the issues caused by sinks and guarantee convergence. Formally, the random walk model can be modeled as:

$$r^{t+1} = \beta \times r^t \times A + (1 - \beta) \times \mathbf{v} \tag{4.6}$$

where $\mathbf{v}$ is the preference vector, and $\sum v_i = 1$. We also follow the standard convention and set $\beta = 0.85$.

When a random walk process converges to a stationary state, we obtain a *stationary distribution*, which is what we use as our semantic signature.

4.2.2 Disambiguation Graph

A disambiguation graph that can capture the semantic relations between candidate entities is important for our algorithm. In this work, we use the *Co-occurrence graph* to construct our disambiguation graph (as described in Section 1.1.3). An example disambiguation graph is shown in Figure 1.2, in which the leftmost column lists the mentions, the *candidates* column lists the candidate entities of these mentions, and the *neighbors* column lists the neighboring entities of the candidates in the Co-occurrence graph.

We post-process the disambiguation graph to remove noisy entities and reduce the size of the graph by pruning non-candidate entities that are connected to just one candidate entity as well as entities with a low degree (200 was experimentally chosen as the minimum value). Our EG is so dense that, without pruning, the disambiguation graph of most mentions quickly becomes prohibitively large. Notice that candidate entities are never pruned, to ensure unpopular entities are included.

Discussion In addition to reducing the graph size and improving the efficiency of the random walk process, another advantage of the smaller disambiguation graph is that those candidates and their semantically-connected entities are more densely connected (as per co-occurrence and direct linkage in Wikipedia), which preserves the topic coherence assumption in most global approaches. As illustrated in Figure 1.2, the basketball related entities such as *Jeff Malone*, *Utah Jazz*, and others form a much denser subgraph than other entities.

4.2.3 Semantic Signature Computation

With the disambiguation graph, we can then compute the semantic signatures using the random walk models.

Semantic Signature of an Entity

To compute the semantic signature of a target entity e_i, we need to ensure that the random walk always restarts from e_i. This can be done by setting the preference vector $\mathbf{v}$ with $v_i = 1$, and $v_{j(j \neq i)} = 0$.

Semantic Signature of a Document

In principle, computing the semantic signature of a document is no different from doing so for a single entity. Given a set of entities E_d representing a document, we set their preference probability in vector $\mathbf{v}$, and then compute the semantic signature of the document through the random walk with restart from entities in E_d.

There are, however, two problems here. First, we do not know the true entity set E_d representing the document (finding this set is the task of NED after all). Second,

it is not clear how to set the weights in the preference vector. Different entities may have different importance to the document, thus a uniform weight may not reflect their importance. To solve these two problems, we explore the following strategies.

Finding E_d We say a mention is *unambiguous* if it is associated with only one entity in the alias dictionary. Unambiguous mentions have been shown to help with NED [68]. In Example 2, *NBA MVP* is one such unambiguous mention, which is useful for the disambiguation of other mentions. We initialize the set E_d with the referent entities of all unambiguous mentions in M and expand it as more mentions are disambiguated.

In case all mentions in the document are ambiguous, we approximate E_d using candidates of the mentions in M. While this approximation could bring in a lot of noise which would potentially decrease the accuracy of the disambiguation, the effectiveness may not be affected much by the noisy entities if the true entities are well connected in the graph.

Determining Weights The preference probability of an entity e_i could be affected mainly by two factors: $P(m, e_i)$—the probability mention m refers to e_i, and $I(m)$—the importance of the mention in its document.

For the case where the referent entities of unambiguous mentions are used, $P(m, e_i)$ is 1 since e_i is considered as the true entity of m. When using the candidate entities, the probability can be measured in several ways. Prior probability is one measure that has been shown to be a strong baseline [31]. Other alternatives include the context similarity between e_i and m and uniform weights. We experimented with several options, and, fortunately, as shown in our experimental evaluation, WNED is very robust to the choice of weights, consistently yielding good results.

While there could be other better measures, we use the standard *tf-idf* scheme to compute the importance of a mention $I(m)$ for simplicity.

Combining $P(m, e_i)$ and $I(m)$ together, we compute the preference probability as follows:

$$\mathbf{v}_i = I(m) * P(m, e_i) \tag{4.7}$$

With the preference vector $\mathbf{v}$, the semantic signature of a document can be computed using a random walk with restart on the disambiguation graph. As shown in Figure 4.1, the row for document d gives its semantic signature.

4.3 Mention Disambiguation

4.3.1 Semantic Relatedness

Let $SS(e_i)$ be the semantic signature of a candidate entity $e_i \in cand(m)$, and $SS(d)$ be the semantic signature of the document d. There are several ways one can use to compare these semantic signatures to estimate the semantic relatedness between m and d. One standard way of doing so is to use the Kullback-Leibler (KL) divergence: given two probability distributions P and Q, their KL divergence measures their distance, as follows:

$$D_{KL}(P \parallel Q) = \sum_i P_i \log \frac{P_i}{Q_i} \tag{4.8}$$

In this work, we use Zero-KL Divergence [46], a better approximation of the KL divergence that handles the case when Q_i is zero.

$$ZKL_\gamma(P, Q) = \sum_i P_i \begin{cases} \log \frac{P_i}{Q_i} & Q_i \neq 0 \\ \gamma & Q_i = 0 \end{cases} \tag{4.9}$$

in which γ is a real number coefficient. Following the recommendation in [46], we set $\gamma = 20$, arriving at the semantic relatedness used in Eq. 4.4:

$$\psi(e_i, d) = \frac{1}{ZKL_\gamma(SS(e_i), SS(d))} \tag{4.10}$$

4.3.2 Disambiguation Algorithm

As previously observed (see, *e.g.*, [45], [51]), the NED problem is intimately connected with a number of NP-hard optimizations on graphs, including the maximum m-clique problem [28], from which a polynomial time reduction is not hard

Algorithm 1 Iterative WNED

Input: $M = \{m_1, m_2, \ldots, m_n\}$, $\mathsf{EG} = (E, L)$
Output: Assignment $\hat{\Gamma} : M \to E \cup \{\mathsf{NIL}\}$

1: $\hat{\Gamma} = \langle \Gamma_i, 1 \leq i \leq |M| \,|\, \Gamma_i = \mathsf{NIL} \rangle$
2: $L = \langle m_i \in M$ sorted by $|aliases(m_i)| \rangle$
3: **for** $i = 1$ to $|L|$ **do**
4: **if** $|cand(m_i)| = 1$ **then**
5: $\hat{\Gamma}_i(m_i) = cand(m_i)$
6: **end if**
7: **if** $|cand(m_i)| > 1$ **then**
8: $\mathbf{d}$ = vecInit$(M, \mathsf{EG}, \hat{\Gamma})$; $Q = SS(\mathbf{d})$
9: $max = 0$
10: **for** $e_j \in cand(m_i)$ **do**
11: $P = SS(e_j)$
12: $\psi(e_j, d) = \dfrac{1}{ZKL_\gamma(P, Q)}$
13: $score(e_j) = \psi(e_j, d) \cdot \phi(m_i, e_j)$
14: **if** $score(e_j) > max$ **then**
15: $e^* = e_j$; $max = score(e_j)$
16: **end if**
17: **end for**
18: **if** $score(e^*) < \theta$ **then**
19: $\hat{\Gamma}_i(m_i) = \mathsf{NIL}$
20: **else**
21: $\hat{\Gamma}_i(m_i) = e^*$
22: **end if**
23: **end if**
24: **end for**
25: **return** $\hat{\Gamma}$

to construct. Thus we resort to an iterative greedy algorithm, called Walking NED (WNED), and described in Alg. 1.

WNED starts with the mentions sorted by their degree of ambiguity, measured by the number of entities with that mention as an alias (line 2). Note that the ambiguity of a mention is typically much higher than the number of candidates that are considered (after pruning). If a mention has a single candidate, WNED assigns that candidate to the mention (line 5). Otherwise, the main loop of the algorithm goes through that mention (lines 7–23): updating the semantic signature of the partial entity assignment (line 8), computing the signature of each candidate (line 11), and selecting the best candidate based on the greedy approximation of the original opti-

Algorithm 2 vecInit

Input: $M = \{m_1, m_2, \ldots, m_n\}, \mathsf{EG} = (E, L), \Gamma : M \to E$
Output: Document disambiguation vector $\mathbf{d}$

1: let n be the size of the disambiguation graph
2: $\mathbf{d} = \mathbf{0}_{(n)}$
3: **if** $\Gamma \neq \emptyset$ **then**
4: **for** $m, e \in \Gamma$ **do**
5: $\mathbf{d}_e = 1$
6: **end for**
7: **else**
8: **for** $m \in M$ **do**
9: **for** $e \in cand(m)$ **do**
10: $\mathbf{d}_e = prior(m, e) \cdot tf\,idf(m)$
11: **end for**
12: **end for**
13: **end if**
14: normalize $\mathbf{d}$
15: **return** $\mathbf{d}$

mization in Eq. 4.4. A final step of the algorithm is to assign NIL to those mentions whose even the best candidate entity has a low score (lines 18–22).

Parameters The experimental evaluation reported here was obtained with the following parameter setting: in Eq. 4.1 $\alpha = 0.8$; in Eq. 4.6, $\beta = 0.85$; and in Eq. 4.9, $\gamma = 20$. These settings were obtained experimentally.

4.3.3 Computational Cost

There are two factors contributing to the cost of WNED: computing the signatures, and greedily selecting the best candidate for each mention.

Let $n = |M|$ be the number of mentions, then the total number of candidates considered by WNED is at most Kn, assuming a constant number K of promising candidates are selected [32]. Thus the total number of semantic signature computations is $Kn + |M| = O(n)$. Given that the size of the disambiguation graph is $O(n)$ vertices and $O(n^2)$ edges (unless some non-trivial pruning is performed) and the number of iterations in the random walks is fixed, computing all signatures can be done in $O(n^2)$ time (and space).

As for the time required for the actual scoring, for fixed EG and input document, computing the prior probability and the context similarity is done through database lookups at $O(1)$ time. In the standard WNED, we also need to compute the Zero-KL divergence on vectors of length n, which can be done in $O(n)$ time.

In our experience, the highest actual costs lie in building the disambiguation graphs, which must be done for each input document, and performing the random walks. Our current implementation keeps the entire disambiguation graph in main memory to speed up the random walks.

4.4 Experimental Evaluation

The EG used in our experiment is built from the Wikipedia 20130606 dump. The source code for our approach is available publicly[1]. For the metrics, we use the standard *accuracy, precision, recall*, and *F1*. Note that we only focus on mentions whose referent entities are in the EG. For the system implementation, we manage our entity graphs on disks using the WebGraph[2], and compute the random walk with restart using the WeightedGraph library[3].

4.4.1 Evaluation using Public NED Framework

Given the host of applications where NED is useful and the inherent difficulty of the problem, a lot of effort has been devoted recently to establishing fair and comprehensive benchmarks for this task. In particular, Web-based evaluation platforms, such as GERBIL [99], are a clear step in the right direction, as they go a long way in automating the collection and reporting of results of different algorithms under the same benchmarks and evaluation conditions.

GERBIL [99] is a general framework for entity annotation, which has more than 11 NED approaches and 16 public datasets for the NED task. We compare our approach with all available approach including graph-based approach: AGDISTIS [98], AIDA [45], Babelfy [72], FOX [92], WAT [81], xLisa [108],

[1] https://github.com/U-Alberta/wned
[2] http://webgraph.di.unimi.it/
[3] http://law.di.unimi.it/satellite-software/weighted/

corpus	topic	#docs	#mentions/doc
ACE2004	news	57	4.44
AQUAINT	news	50	14.54
MSNBC	news	20	32.50
AIDA/CoNLL	news	1393	19.97
DBpediaSpotlight	news	58	5.69
KORE50	mixed	50	2.86
Microposts2014	tweets	3505	0.65
N3-RSS-500	RSS-feeds	500	0.99
N3-Reuters-128	news	128	4.85
OKE 2015 Task 1	encyclopedia	199	5.41

Table 4.1: Statistics of datasets in GERBIL [99].

and PBoH [27], and context-based approaches: DBpedia Spotlight [65], FREME NER [88], Kea [93], and NERD-ML [86]. The datasets in GERBIL, with detailed statistics shown in Table 4.1, mainly contain news articles, RSS feeds, tweets, and encyclopedia, most of which are from news articles, indicating that mentions in them likely refer to popular entities. Performance on datasets with very few mentions per document, such as *Microposts2014* and *N3-RSS-500*, will depend more on the local compatibility and less on the global coherence.

Table 4.2 gives the results of different approaches using GERBIL[4], including PBoH [27][5] and our approach[6]. Note that the results of PBoH in Table 4.2 are from their updated report on GERBIL 1.2.4, which is different from the results reported in their original work [27] [7].

We can see that comparing to other NED approaches, our WNED approach is very competitive on most of the datasets. Although no special processing is applied on the micropost2014 datasets, our approach still performs better than all other approaches except the PBoH. One main type of errors on microposts is from the candidate selection because of the casual writing style in microposts, which introduces many unseen name variations.

[4]http://gerbil.aksw.org/gerbil/experiment?id=201611040001
[5]http://gerbil.aksw.org/gerbil/experiment?id=201610270004
[6]Results are available in http://dx.doi.org/10.7939/DVN/10968
[7]See details for the accuracy drop: https://github.com/AKSW/gerbil/issues/98

Datasets	AGDISTIS [98]	AIDA [45]	Babelfy [72]	DBpedia Spotlight [65]	FOX [92]	FREME NER [88]	Kea [93]	NERD-ML [86]	WAT [81]	xLisa [108]	PBoH [27]	WNED
ACE2004	0.65	0.69	0.53	0.48	0.00	0.49	0.66	0.58	0.66	0.70	**0.72**	**0.77**
	0.77	0.82	0.70	0.68	0.37	0.65	0.77	0.73	0.77	0.80	**0.83**	**0.88**
	0.66	0.80	0.61	0.58	0.00	0.58	0.76	0.67	0.76	**0.81**	0.79	**0.83**
	0.78	0.89	0.76	0.75	0.39	0.71	0.84	0.79	0.85	**0.88**	0.86	**0.91**
AQUAINT	0.52	0.55	0.68	0.53	0.00	0.56	0.78	0.60	0.73	0.76	**0.81**	**0.79**
	0.51	0.55	0.68	0.52	0.00	0.43	0.78	0.58	0.74	0.75	**0.81**	**0.79**
	0.73	0.57	0.70	0.55	0.00	0.58	0.81	0.62	0.75	0.79	**0.84**	**0.83**
	0.59	0.56	0.70	0.54	0.00	0.44	**0.80**	0.60	0.76	0.77	**0.83**	**0.83**
MSNBC	0.73	0.69	0.71	0.42	0.02	0.22	0.78	0.62	0.73	0.50	**0.82**	**0.88**
	0.73	0.65	0.68	0.44	0.02	0.16	0.77	0.64	0.73	0.50	**0.82**	**0.90**
	0.74	0.74	0.76	0.46	0.02	0.24	0.84	0.67	0.79	0.55	**0.86**	**0.89**
	0.73	0.70	0.73	0.48	0.02	0.18	0.84	0.70	0.80	0.57	**0.85**	**0.91**
AIDA/CoNLL-Complete	0.55	0.68	0.66	0.50	0.51	0.38	0.61	0.20	0.71	0.47	**0.75**	**0.76**
	0.53	0.66	0.60	0.50	0.48	0.29	0.57	0.12	0.68	0.45	**0.75**	**0.76**
	0.57	0.77	0.74	0.58	0.54	0.44	0.68	0.24	**0.80**	0.54	**0.80**	**0.79**
	0.52	**0.76**	0.68	0.59	0.50	0.33	0.65	0.14	**0.78**	0.52	**0.78**	**0.78**
AIDA/CoNLL-Test A	0.54	0.67	0.65	0.48	0.49	0.28	0.61	0.00	0.70	0.45	**0.75**	**0.76**
	0.50	0.62	0.59	0.47	0.45	0.23	0.56	0.00	0.66	0.41	**0.73**	**0.75**
	0.56	0.74	0.74	0.55	0.53	0.33	0.67	0.00	**0.78**	0.52	**0.80**	**0.78**
	0.49	0.71	0.68	0.55	0.47	0.25	0.64	0.00	**0.76**	0.48	**0.77**	0.75
AIDA/CoNLL-Test B	0.54	0.69	0.68	0.52	0.49	0.35	0.61	0.01	**0.72**	0.47	**0.75**	**0.75**
	0.54	0.68	0.62	0.51	0.48	0.22	0.61	0.00	0.70	0.46	**0.75**	**0.76**
	0.55	**0.77**	0.76	0.60	0.52	0.40	0.69	0.00	**0.80**	0.54	**0.80**	**0.77**
	0.54	0.78	0.70	0.60	0.51	0.26	0.70	0.01	**0.80**	0.53	**0.79**	0.78
AIDA/CoNLL-Training	0.55	0.69	0.65	0.50	0.52	0.39	0.61	0.28	0.71	0.48	**0.75**	**0.76**
	0.53	0.66	0.60	0.50	0.50	0.30	0.56	0.17	0.68	0.45	**0.73**	**0.77**
	0.57	0.77	0.74	0.58	0.55	0.45	0.69	0.33	**0.81**	0.56	**0.80**	0.79
	0.52	0.76	0.68	0.59	0.51	0.35	0.64	0.21	**0.79**	0.53	**0.78**	**0.78**
DBpediaSpotlight	0.27	0.25	0.52	0.71	0.15	0.45	**0.74**	0.56	0.67	0.71	**0.79**	**0.79**
	0.28	0.21	0.51	0.69	0.12	0.31	0.73	0.53	0.69	0.71	**0.80**	**0.81**
	0.40	0.25	0.52	0.71	0.15	0.45	**0.74**	0.56	0.67	0.71	**0.80**	**0.80**
	0.36	0.21	0.51	0.69	0.12	0.31	0.73	0.53	0.69	0.71	**0.80**	**0.82**
KORE50	0.33	**0.69**	**0.74**	0.46	0.27	0.17	0.60	0.31	0.62	0.51	0.63	0.56
	0.30	**0.64**	**0.70**	0.42	0.22	0.14	0.53	0.25	0.52	0.45	0.58	0.52
	0.33	**0.69**	**0.74**	0.46	0.27	0.17	0.60	0.31	0.62	0.51	0.63	0.56
	0.30	**0.64**	**0.70**	0.42	0.22	0.14	0.53	0.25	0.52	0.45	0.59	0.52
Microposts2014-Test	0.33	0.42	0.48	0.50	0.22	0.42	**0.64**	0.52	0.60	0.55	**0.73**	0.63
	0.60	0.59	0.63	0.66	0.49	0.60	**0.76**	0.67	0.74	0.68	**0.85**	0.75
	0.42	0.42	0.48	0.50	0.22	0.42	0.64	0.52	0.60	0.55	**0.74**	**0.65**
	0.61	0.59	0.63	0.66	0.49	0.60	**0.76**	0.67	0.74	0.68	**0.85**	**0.76**
Microposts2014-Train	0.42	0.51	0.51	0.48	0.31	0.46	**0.65**	0.52	0.63	0.59	**0.71**	0.64
	0.61	0.61	0.61	0.61	0.48	0.56	**0.74**	0.63	0.73	0.67	**0.81**	**0.74**
	0.51	0.51	0.51	0.48	0.31	0.46	0.65	0.52	0.63	0.59	**0.73**	**0.67**
	0.63	0.61	0.61	0.61	0.48	0.56	0.74	0.63	0.73	0.67	**0.82**	**0.75**
N3-RSS-500	**0.61**	0.45	0.44	0.20	0.56	0.28	0.44	0.38	0.44	0.45	0.53	**0.69**
	0.61	0.39	0.38	0.16	0.54	0.20	0.39	0.30	0.37	0.38	0.53	**0.69**
	0.52	**0.66**	0.64	0.32	0.50	0.44	0.62	0.57	0.64	**0.65**	0.55	**0.65**
	0.52	**0.64**	0.63	0.41	0.49	0.45	0.61	0.58	0.63	**0.66**	0.48	0.62
N3-Reuters-128	**0.66**	0.47	0.45	0.33	0.54	0.24	0.51	0.41	0.52	0.39	**0.65**	0.63
	0.72	0.38	0.39	0.27	0.57	0.16	0.46	0.35	0.44	0.34	**0.72**	**0.63**
	0.64	0.57	0.55	0.41	0.52	0.31	0.61	0.51	0.63	0.49	**0.69**	0.62
	0.68	0.51	0.55	0.41	0.54	0.29	0.60	0.51	0.59	0.52	**0.72**	0.60
OKE 2015 Task 1 evaluation dataset	0.59	0.56	0.59	0.31	0.56	0.32	**0.63**	0.61	0.57	**0.62**	**0.63**	**0.62**
	0.60	0.55	0.58	0.27	0.53	0.26	**0.63**	0.60	0.56	0.61	**0.63**	**0.62**
	0.62	0.63	0.66	0.36	0.60	0.38	**0.71**	**0.70**	0.65	**0.71**	0.68	0.65
	0.61	0.62	0.65	0.30	0.56	0.28	**0.71**	0.68	0.62	**0.70**	0.67	0.64
OKE 2015 Task 1 example set	**1.00**	0.60	0.4	0.22	**0.78**	0.25	0.55	0.00	0.60	0.5	0.50	0.67
	1.00	0.72	0.65	0.44	0.67	0.44	0.69	0.33	0.72	0.69	0.67	**0.75**
	1.00	**0.86**	0.57	0.50	0.80	0.40	0.75	0.00	**0.86**	0.80	0.67	0.75
	1.00	**0.89**	0.80	0.33	**0.89**	0.50	0.82	0.33	**0.89**	**0.89**	0.78	0.82
OKE 2015 Task 1 gold standard sample	0.62	0.67	0.71	0.25	0.54	0.41	**0.78**	**0.77**	0.72	0.75	0.76	**0.78**
	0.64	0.65	0.68	0.20	0.49	0.32	0.76	0.74	0.69	0.73	**0.76**	**0.78**
	0.64	0.71	0.75	0.27	0.56	0.44	**0.81**	**0.81**	0.77	0.79	0.80	**0.82**
	0.64	0.67	0.72	0.22	0.53	0.35	**0.79**	0.77	0.73	0.76	0.78	**0.80**

Table 4.2: Results reported by GERBIL. The rows in each cell report the F1@Micro, F1@Macro, InKB F1@Micro, and InKB F1@Macro, in which **red** marks the highest F1 and **blue** marks the second highest F1.

4.4.2 Evaluation on Established Benchmarks

Despite their effectiveness and convenience, the information reported by platforms such as GERBIL (particularly, aggregate accuracy measurements) is not enough for a deeper analysis that can lead to algorithmic improvements. Thus, we re-implement and experiment with several state-of-the-art NED approaches and compare our WNED against them. In the remaining section, we report our experimental evaluation on well-known publicly available benchmarks.

Method	MSNBC			AQUAINT			ACE2004			AIDA-CoNLL		
	Acc.	F1@MI	F1@MA	Acc.	F1@MI	F1@MA	Acc.	F1@MI	F1@MA	Acc.	F1@MI	F1@MA
PRIOR	0.86	0.86	0.87	0.84	0.87	0.87	**0.85**	0.85	0.87	0.75	0.75	0.76
CONTEXT	0.77	0.78	0.72	0.66	0.68	0.68	0.61	0.62	0.57	0.40	0.40	0.35
Cucerzan	0.88	0.88	0.88	0.77	0.79	0.78	0.79	0.79	0.78	0.73	0.74	0.72
M&W	0.68	0.78	0.80	0.80	0.85	0.85	0.75	0.81	0.84	0.60	0.68	0.68
Han11	0.88	0.88	0.88	0.77	0.79	0.79	0.72	0.73	0.67	0.62	0.62	0.58
AIDA	0.77	0.79	0.76	0.53	0.56	0.56	0.77	0.80	0.84	0.78	0.79	0.79
GLOW	0.66	0.75	0.77	0.76	0.83	0.83	0.75	0.82	0.83	0.68	0.76	0.71
RI	0.89	0.90	0.90	0.85	0.88	0.88	0.82	0.87	0.87	0.79	0.81	0.80
WNED	0.89	0.90	0.90	**0.88**	**0.90**	**0.90**	0.83	0.86	0.89	0.84	0.84	0.83

Table 4.3: Accuracy results of all methods on the 4 public benchmarks.

We compare WNED to the following approaches: Cucerzan [17], M&W [68], Han11 [38], AIDA [45], GLOW [85], and RI [15]. Detailed descriptions of these approaches are given in Chapter 2 (Related Work).

We also evaluate two useful baselines: PRIOR which picks the entity with the highest prior probability for each mention using $prior(m, e)$, and CONTEXT which chooses the candidate entity with the highest textual similarity to the mention using $ctxSim(m, e)$. These baselines are informative as virtually all methods rely on these measures in one way or another, including ours (recall Eq. 4.4). Somewhat surprisingly, as shown next, not every method improves on both of them.

Also, as mentioned in [27], GERBIL uses an old version of the public datasets. Thus we update the 4 widely used public benchmarks as described in Section 3.5.

To avoid any discrepancy in the results caused by different Wikipedia versions used in different approaches, we update all datasets and results of the compared NED approaches to their redirected entities in our Wikipedia dump. All datasets used in this evaluation and the results obtained with each method on each document can be downloaded from `http://dx.doi.org/10.7939/DVN/10968`.

Table 4.3 shows the results of the two baselines, 6 competing NED approaches, and our WNED on the 4 public benchmarks. As customary, we report F1 aggregated across mentions (micro-averaged, indicated as **F1@MI**) and across documents (macro-averaged, **F1@MA**).

Discussion A few observations are worth making here. Among previous work, RI has the best performance across benchmarks. The disambiguation via textual similarity alone, as done by the CONTEXT baseline, leads to poor accuracy in general, especially on the more challenging AIDA-CoNLL benchmark. The PRIOR baseline, on the other hand, performs well across the board, outperforming several approaches. This points to limitations in the benchmarks themselves: they use high-quality news articles, where the entities are likely to be mentioned at least once by their full name (which is easy to disambiguate with the prior probability alone).

The reader will notice that virtually every method in the literature is evaluated against the baseline PRIOR, and if one looks back to earlier works, the reported accuracy of PRIOR is not nearly as high as what we report. This can be explained by the continuous cleaning process on Wikipedia—from which the statistics are derived. As we use a more recent and cleaner corpus, where the support for good and appropriate entity aliases is markedly higher than for inappropriate mentions.

With respect to WNED, it outperforms all competitors on all benchmarks. Another observation is that there is quite a lot of variability in the relative ordering of the previous methods across benchmarks, except for RI and our methods. This somewhat surprising lack of robustness in some approaches may have been caused by over-tuning for the development benchmark, resulting in poor generalization when tested on different benchmarks.

4.4.3 Qualitative Error Analysis

We now look at the types of errors made in our approach. We manually inspected every error for the smaller datasets: MSNBC, AQUAINT, and ACE2004, and analyzed 20 errors randomly picked in each bracket of AIDA-CoNLL (a collection of documents with accuracy of PRIOR within a range).

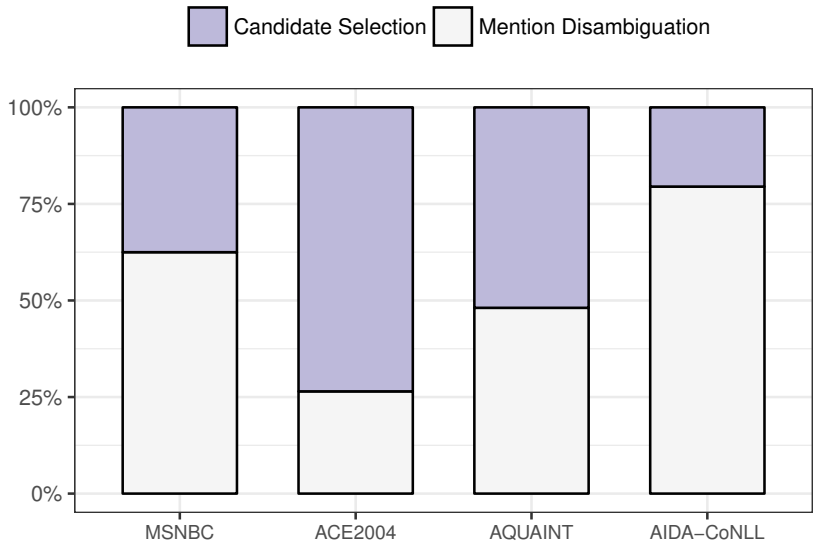


Figure 4.2: Breakdown of errors by WNED across benchmarks; for AIDA-CoNLL, the errors are estimated using a sample.

The first observation is that, in the benchmarks, a large fraction of errors in our method happen in the candidate selection phase, as illustrated in Fig. 4.2. On average, 54% of the errors in the smaller benchmarks are due to issues in candidate selection (compared with 18% in AIDA-CoNLL). This reinforces the hypothesis that the entities mentioned in these public benchmarks are easy to disambiguate. Below we discuss prototypical errors in each of the phases.

Errors during Candidate Selection

Incorrect Co-reference Resolution We employ a co-reference resolution algorithm in our text processing pipeline to increase recall. It is possible that distinct named entities are incorrectly deemed to be the same because of the heuristic nature of the algorithm. For example, in the sentence

> "Time Warner stagnated for five years after it was created in 1990 by the merger of Time and Warner."

the entity "Time" at the end is incorrectly resolved to "Time Warner", leading to an error. About 1% of the errors are due to the incorrect resolution of named entities.

Incomplete Alias Dictionary Currently, we disambiguate only those mentions corresponding to an alias from Wikipedia, leading to problems in sentences like

> "Thirteen miners were trapped inside the Sago Mine near Buckhannon, W. Va."

In this case, we miss the abbreviation "W. Va." for West Virginia. This kind of error was noticeably more common in the easier benchmarks (accounting for 30% of the errors in the ACE2004 dataset). In the AIDA-CoNLL benchmark, only 2% of the errors are due to this problem.

Aggressive Pruning Another source of error by our method is pruning the correct entity from the candidate set. For example, in sentence

> "A state coordinator for the Florida Green Party said she had been ..."

the correct entity (the *Green Party of Florida*) is pruned due its low prior, but it could be correctly resolved given the mention of *Florida* in the same sentence. Instead, WNED links the mention to the *US Green Party*. Of course, pruning is done to reduce the cost of the random walks, and future algorithmic improvements can alter this trade-off.

Errors during Mention Disambiguation

These are errors where the correct entities according to the ground truth were selected as the candidates but not chosen during the mention disambiguation phase by our algorithm.

Lack of Application Domain We observe that most of the errors associated with locations happen because the documents in most benchmarks are news articles that start with the location of the news source reporting the news (*e.g.*, *New York Times* documents always start with a mention to New York). More often than not, such locations are unrelated to the topic of the document and other mentions in the document, breaking the topic coherence assumption. These errors, which can be easily avoided via pre-processing, accounts for 5% of the mistakes of our algorithm in the MSNBC and AIDA-CoNLL benchmarks and 2% across all benchmarks.

Need for Deeper Text Analysis There are of course very hard disambiguation cases where a more in-depth understanding of the text would be needed for a successful algorithmic approach. One example is the sentence:

> "Maj. Gen. William Caldwell, a U.S. military spokesman, told reporters that ..."

In this case, there are two candidates with the same name and high military rank, thus being semantically related to the document and confusing the algorithm. However, extraneous facts about the candidates, though unrelated to the text itself, could be used for disambiguating the mention. For instance, the candidate incorrectly chosen by our algorithm died in the 1820s while the correct candidate was still alive at the time the benchmark article was written. Given that the document states the facts as current news, the incorrect candidate could have been pruned out.

Questionable Errors

We argue that in many cases, our approach (as well as other approaches) chose an entity that is considered erroneous by the ground truth, but that would be acceptable to a human judge. For example, in the sentence:

> "Coach Saban said the things Crimson Tide fans most wanted to hear."

our approach links "Crimson Tide" in the sentence to the *Alabama Crimson Tide football*, which is the *men's* varsity football team of the university while the ground truth refers to *Alabama Crimson Tide* which corresponds to both the men's and women's teams. We found that about 17% of the errors are in this category, with a higher prevalence in the harder benchmarks (21%). Table 4.4 lists many other similar errors, where a case can be made that the ground-truth itself is probably too strict.

Impact of the Greedy Approach

Given that the iterative WNED is a greedy algorithm, it is interesting to see how an erroneous disambiguation decision influences future ones, especially in the very first round. In all benchmarks, we found 1 error in the first round among all the

Mention	WNED Suggestion	Ground Truth
Iraqi	Iraqi people	Iraq
Hungarian	Hungary	Hungarian people
executives	Corporate title	Chief executive officer
Russian	Russian language	Russians
Iranian	Iranian people	Iran
Greek	Greek language	Ancient Greece
civil war	American Civil War	Civil war
broadcaster	Presenter	Broadcasting

Table 4.4: Questionable disambiguation errors

errors in MSNBC, AQUAINT, and ACE2004 datasets[8], and less than 8 errors from the random samples in the AIDA-CoNLL benchmark. In all cases, the first error did not prevent the algorithm from correctly disambiguating other mentions.

As for the initialization step, we found that most documents in our benchmarks do have unambiguous mentions available, and most of them are correctly linked to the true entity[9]. In MSNBC, we have 2 errors from the unambiguous mentions, *New York Stock Exchange* and *NYSE*; both are linked to *New York Mercantile Exchange*. This error barely affects the semantic signature since they are still stock related entities. There are 5 such errors in AQUAINT, and 1 error in ACE2004, all of which have little effect on the linking results of other mentions in the same document.

Finally, we found that most other errors happened after 5 iterations when the document disambiguation vector already captures the topic fairly well. These errors are on mentions that are not semantically related to other mentions in the document, or simply due to the disproportionately high priors favoring (incorrectly) head entities.

[8]A mention to the *USS Cole* which should have been linked to *USS Cole (DDG-67)*, was linked to *USS Cole bombing*.

[9]Recall (Sec. 4.3) we initialize the document disambiguation vector with unambiguous mentions when available.

4.5 Summary

In this chapter, we described a mention disambiguation approach that combines lexical and statistical features with *semantic relatedness* computed using *semantic signatures*, a semantic representation of entities derived from random walks over suitably designed disambiguation graphs. Our semantic signature uses more relevant entities from an entity graph, thus reducing the effect of feature sparsity, and results in a substantial accuracy gain. We proposed a hand-tuned greedy algorithm which outperformed the previous state-of-the-arts by a wide margin. We also evaluated our WNED approach using the GERBIL framework on 16 public datasets and showed the superiority of our approach.

Moreover, we demonstrated several shortcomings of the existing NED benchmarks: they use high-quality news articles, where most entities are popular entities, thus easy to disambiguate with the prior probability alone. This finding motivates us to build more *balanced* benchmarks with documents of varying disambiguation difficulty, as described in Chapter 5. We also performed a comprehensive qualitative analysis on the disambiguation results, which gives directions for future improvement. Our analysis found that around 54% of the disambiguation errors were due to issues in candidate selection such as incorrect name expansion, incomplete alias dictionary, and aggressive pruning. The errors during mention disambiguation pointed out a need for deeper text understanding using resources like the common knowledge and structural relations.

Chapter 5

NED via Learning to Rank

In this chapter, we consider using a supervised algorithm to address the robustness issue of the hand-tuned algorithm in WNED. We employ a learning to rank algorithm to combine local features and the semantic relatedness to learn a robust ranking model for the mention disambiguation task. Our experimental evaluation demonstrates the superiority of this supervised approach in both accuracy and robustness.

5.1 Overview

While our unsupervised approach WNED can achieve the state-of-the-art accuracy, its hand-tuned algorithm is sensitive to datasets and inflexible to incorporate new features as needed. With abundant datasets available for NED, supervised machine learning methods, in this case, would be a suitable solution for NED to build robust models with better parameter tuning and flexibility to new data and features.

The idea of applying supervised algorithms on NED is not new. Milne and Witten [68] employ a few classifiers, including Naive Bayes, C4.5, and SVM, on features like local context and prior probability. Their approach, however, takes all mentions as independent instances and ignores the global coherence. Zheng *et al*. [112] employ a learning to rank algorithm for the NED task and show that the ListNET algorithm outperforms both SVM and Perceptron. Similar to Milne and Witten, their approach mainly focuses on local features, overlooking the global coherence and semantic relatedness.

We consider NED as an entity ranking problem, which is similar to the ranking problem in the document retrieval task in Information Retrieval (IR). While document retrieval ranks web documents by their relevance to a user query, NED ranks candidate entities by their similarity to a given mention. One difference is that document retrieval returns a list of ranked documents while NED considers only the top 1 candidate. Our approach employs a pairwise learning to rank algorithm LambdaMART [105] to combine both local features and the global coherence into a ranking model which can benefit from the robustness of the supervised algorithm and the effectiveness of the global coherence.

5.2 Supervised Walking NED

5.2.1 Learning to Rank

A number of supervised learning algorithms can be employed to learn ranking models for the measure of similarity scores between a mention and its candidates. Most approaches, however, treat mention-candidate pairs as independent instances without considering the ranking structure of candidate entities, resulting in mediocre results. To handle this issue, learning to rank algorithms take the ranking structure into evaluation metrics and objective loss functions, and exploit it to learn models that are specific to ranking tasks.

Learning to rank approaches originate from IR and have broad applications in many other tasks, including entity search, question answering, and machine translation [55]. Based on how the ranking structure of instances is used, learning to rank approaches can be divided into three categories [55].

Pointwise Approach In pointwise learning to rank approaches, the ranking structure of candidates is ignored, and each mention-candidate pair is treated as an independent instance, for which a relevance score is computed using a ranking model. Depending on the output of the ranking model (*e.g.*, a category label or a real number), we can employ traditional classification or regression supervised algorithms to train the ranking model.

Pairwise Approach The pairwise approaches transform a ranking problem into a pairwise classification problem. Given a mention $m_i \in M$, its candidate entities $e_{ij} \in cand(m_i)$, and the ranking of each candidate r_j, pairwise approaches consider each preference pair of candidates $\langle e_j, e_k \rangle$ as an instance and their relative ranking as the label. For example, the preference pair $\langle e_j, e_k \rangle$ is a positive instance when $r_j > r_k$, or a negative instance otherwise. In this way, we can employ existing supervised classification algorithms to learn ranking models. Compared to the pointwise approaches, which handle each candidate separately, the pairwise approaches consider partial ranking structure in their learning algorithms.

Listwise Approach Listwise approaches take a step further to exploit the full ranking structure of candidates. They consider each ranking list as one instance, and train a ranking model by minimizing a loss function on the list, using metrics like the KL-divergence between the learned ranking list and the true ranking list [11].

In this work, we use LambdaMART [105], a pairwise learning to rank algorithm, to build our ranking model. LambdaMART combines the strength of MART and LambdaRank. MART (Multiple Additive Regression Trees) is a gradient boosted decision tree algorithm that learns prediction models in the form of an ensemble of weak decision trees, and has been shown to work remarkably well on classification tasks [25]. The algorithm, however, conducts the Gradient Descent in the functional space which still faces the discontinuous issue in most IR measures. LambdaRank, on the other hand, leverages the fact that neural net training only needs the gradients of the cost function instead of the function itself, and proposes to use a *gradient* function (Lambda Function) as a surrogate loss function to the IR measure, which can bypass the discontinuous issue in typical IR measures.

Below are details about the features and training data we use in our approach.

5.2.2 Features

While many features [112] can be used to improve the ranking accuracy, our main goal, however, is to exploit the semantic relatedness computed from the random walks and the annotation datasets to learn a robust ranking model for the NED

task, rather than performing exhaustive feature engineering at the risk of overfitting. Therefore we use only 4 features, all of which are familiar in this research area. More precisely, given a mention-entity pair $\langle m, e \rangle$, we extract the following features: (1) prior probability $prior(m, e)$, (2) context similarity $ctxSim(m, e)$, (3) semantic relatedness $\psi(e, \mathbf{d})$ in which $\mathbf{d}$ is obtained by Alg. 2 using the initial $\hat{\Gamma}$, computed as in lines 1–4 in Alg. 1 in Chapter 4, and (4) name similarity $nameSim(m, e)$, which is measured by the N-Gram distance [50] between the name of m and the canonical name of e.

5.2.3 Training data

High-quality training data is critical to develop ranking models. For learning to rank algorithms, an ideal training dataset would consist of a complete ranking of candidates for each mention. In practice, however, obtaining such ranking would be infeasible because it is subjective to users' preference and also costly to collect.

Without a full ranking of candidates, partial ranking from annotation datasets is also valuable for learning to rank algorithms. For any pairwise ranking methods, the training instances for each mention are *ordered* pairs of entities $\langle e_i, e_j \rangle$ such that e_i is ranked *higher* than e_j. In annotation datasets, the true entity of a mention always ranks higher than other candidates, thus can be used to create such training instances. Given a mention m and its true entity e in the candidate set, the training instances will be created as $\langle e, e_i \rangle$ in which $e_i \neq e$ and is ranked lower than e.

5.3 Experimental Evaluation

We evaluate our approach in two ways. The first one uses 5-fold cross-validation on the 4 datasets (MSNBC, AQUAINT, ACE2004, and AIDA-CoNLL), and reports the average results of each validation. The second one uses the whole AIDA-CoNLL as the training dataset and evaluates on the other 3 datasets. For the two models, L2R refers to the model trained on a fraction of the respective datasets, and L2R-CoNLL is the model trained on the AIDA-CoNLL dataset, regardless of the test corpora. For the LambdaMART algorithm, we use the implementation provided in

the RankLib[1] with default parameters.

5.3.1 Evaluation on Established Benchmarks

Method	MSNBC			AQUAINT			ACE2004			AIDA-CoNLL		
	Acc.	F1@MI	F1@MA	Acc.	F1@MI	F1@MA	Acc.	F1@MI	F1@MA	Acc.	F1@MI	F1@MA
PRIOR	0.86	0.86	0.87	0.84	0.87	0.87	**0.85**	0.85	0.87	0.75	0.75	0.76
CONTEXT	0.77	0.78	0.72	0.66	0.68	0.68	0.61	0.62	0.57	0.40	0.40	0.35
Cucerzan	0.88	0.88	0.88	0.77	0.79	0.78	0.79	0.79	0.78	0.73	0.74	0.72
M&W	0.68	0.78	0.80	0.80	0.85	0.85	0.75	0.81	0.84	0.60	0.68	0.68
Han11	0.88	0.88	0.88	0.77	0.79	0.79	0.72	0.73	0.67	0.62	0.62	0.58
AIDA	0.77	0.79	0.76	0.53	0.56	0.56	0.77	0.80	0.84	0.78	0.79	0.79
GLOW	0.66	0.75	0.77	0.76	0.83	0.83	0.75	0.82	0.83	0.68	0.76	0.71
RI	0.89	0.90	0.90	0.85	0.88	0.88	0.82	0.87	0.87	0.79	0.81	0.80
WNED	0.89	0.90	0.90	**0.88**	**0.90**	**0.90**	0.83	0.86	0.89	0.84	0.84	0.83
L2R-CoNLL	**0.91**	**0.92**	**0.92**	0.85	0.87	0.87	**0.85**	**0.88**	**0.90**	**0.89**	**0.89**	**0.89**
L2R	**0.91**	**0.92**	0.91	**0.88**	**0.90**	**0.90**	**0.85**	**0.88**	0.89			

Table 5.1: Accuracy results of all methods on the 4 public benchmarks.

Table 5.1 gives the results of L2R and L2R-CoNLL on the 4 benchmarks. As shown, both L2R and L2R-CoNLL outperform all competitors on the 4 datasets, with L2R performing the best overall. Another observation is that our ranking model trained on AIDA-CoNLL is quite competitive on *all* other datasets, and sometimes superior to the model trained on data from the particular benchmark. While not surprising (as all benchmarks come from the same domain—news), these results mean that the model trained on AIDA-CoNLL can be seen as an effective off-the-shelf ranking model. Another general observation is that there is quite a lot of variability in the relative ordering of the previous methods across benchmarks, except for RI and our methods. This somewhat surprising lack of robustness in a few approaches may have been caused by over-tuning for the development datasets, resulting in poor generalization when tested on different datasets.

5.3.2 Evaluation on New Unbiased Benchmarks

From the above results on the 4 benchmarks, we can see that the PRIOR baseline is surprisingly competitive to most approaches. Considering the simplicity of this baseline, which is solely based on the *prior probability*, we think these datasets are highly biased towards entities with high prior probability, and thus, not representative of all scenarios where NED is necessary. To confirm our assumption, we

[1] https://sourceforge.net/p/lemur/wiki/RankLib/

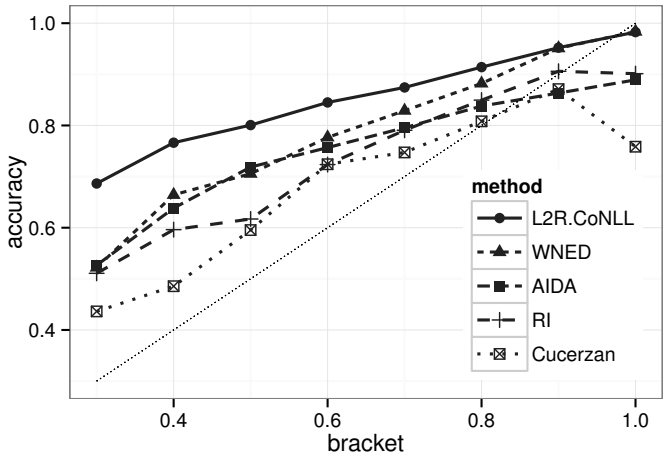
conduct a deeper analysis of the 4 benchmarks and reveal the *bias* issue of them: favoring towards popular entities while ignoring the long-tail unpopular ones.

To evaluate the robustness of our approaches using more balanced benchmarks, we propose a framework for deriving benchmarks and construct two new benchmarks from Wikipedia and the ClueWeb 2012 corpus. We use the PRIOR baseline as a proxy to measure the disambiguation difficulty of a document and construct benchmarks with varying difficulty, in which each proportion consists of documents within the same level of difficulty. Details about the benchmark analysis and the construction of new benchmarks are given in Appendix A.

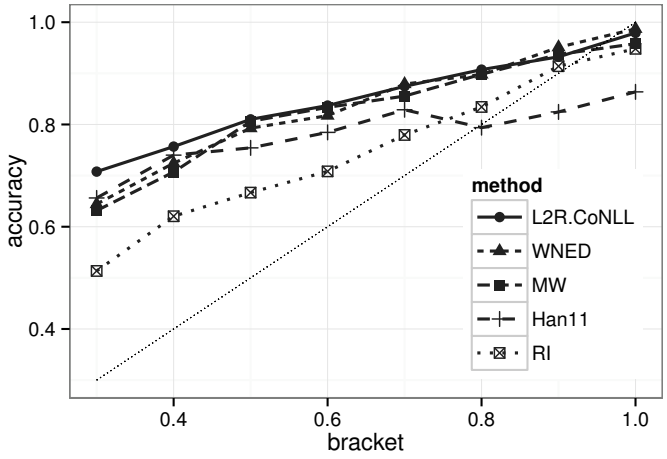
Fig. 5.1 shows the results on the new benchmarks. We plot the accuracy of the best-performing approaches for each bracket (defined by the accuracy of the PRIOR baseline). For clarity, we plot the accuracy of the best 5 approaches. For comparison, we also show the accuracy of each approach on the AIDA-CONLL benchmark. For convenience, a diagonal dotted line whose area under the curve (AUC) is 0.5 (loosely corresponding to the PRIOR baseline) is also shown. Approaches consistently above that line are expected to outperform the PRIOR baseline in practice. Table 5.2 shows the average accuracy of every approach across brackets, corresponding to the AUC in Fig. 5.1.

Method	AIDA-CONLL	Wikipedia	ClueWeb 12
PRIOR	0.57	0.56	0.57
CONTEXT	0.39	0.59	0.42
Cucerzan	0.68	0.66	0.60
M&W	0.58	0.83	0.65
Han11	0.57	0.78	0.61
AIDA	0.75	0.63	0.59
GLOW	0.61	0.69	0.57
RI	0.74	0.75	0.68
WNED	0.79	0.84	0.77
L2R-CONLL	**0.85**	**0.85**	**0.78**

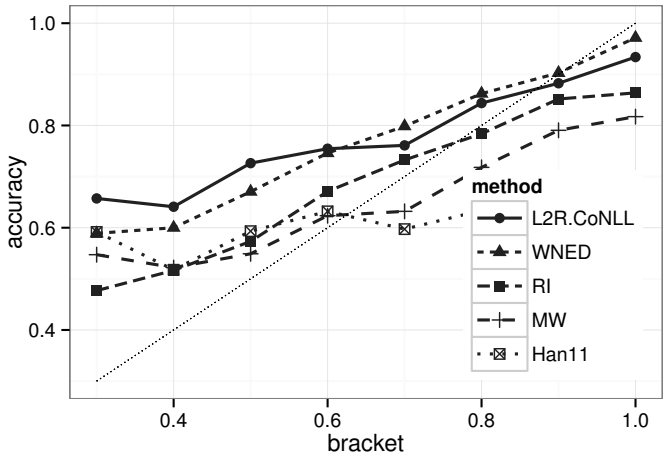
Table 5.2: Average per-bracket accuracy on large-scale benchmarks. Only those brackets with PRIOR accuracy 0.3 or higher are used.



(a) AIDA-CoNLL†.



(b) Wikipedia.



(c) ClueWeb 12.

Figure 5.1: Average accuracy of the top-5 methods on the AIDA-CoNLL, Wikipedia, and Clueweb 12 datasets grouped by the accuracy of the PRIOR baseline.

A few observations are worth mentioning here. First, the two new benchmarks complement the AIDA-CoNLL benchmark: overall, the Wikipedia benchmark is easier than AIDA-CoNLL, while the ClueWeb 12 is harder. Second, as before, the approach RI performs quite well, although not as dominantly as in the 4 public benchmarks. It also seems that the previous supervised methods tend to over-perform on their development datasets (Wikipedia for M&W and CoNLL for AIDA).

Our WNED and L2R-CoNLL approaches outperform all other competitors across all benchmarks, performing much better on the more "difficult" cases (*i.e.*, in lower brackets). In concrete terms, WNED and L2R-CoNLL exhibit, on average, 21% and 26% relative gain in accuracy over the competing approaches (excluding the baselines) on the 3 benchmarks combined, which is significant. Given that our development and tuning are done with a subset of the AQUAINT, MSNBC, and ACE2004, the strong results of WNED and L2R-CoNLL demonstrate the robustness and generality of our approach.

5.3.3 Evaluation using Public NED Framework

Table 5.3 lists the evaluation results using the GERBIL framework. As shown, L2R-CoNLL makes a further improvement over the WNED approach on most datasets, especially on the two micropost datasets on which a 3% improvement is made.

Datasets	AGDISTIS [98]	AIDA [45]	Babelfy [72]	DBpedia Spotlight [65]	FOX [92]	FREME NER [88]	Kea [93]	NERD-ML [86]	WAT [81]	xLisa [108]	PBoH [27]	WNED	L2R-CoNLL
ACE2004	0.65	0.69	0.53	0.48	0.00	0.49	0.66	0.58	0.66	0.70	0.72	**0.77**	**0.76**
	0.77	0.82	0.70	0.68	0.37	0.65	0.77	0.73	0.77	0.80	0.83	**0.88**	**0.87**
	0.66	0.80	0.61	0.58	0.00	0.58	0.76	0.67	0.76	**0.81**	0.79	**0.83**	**0.81**
	0.78	0.89	0.76	0.75	0.39	0.71	0.84	0.79	0.85	0.88	0.86	**0.91**	**0.90**
AQUAINT	0.52	0.55	0.68	0.53	0.00	0.56	0.78	0.60	0.73	0.76	**0.81**	**0.79**	**0.79**
	0.51	0.55	0.68	0.52	0.00	0.43	0.78	0.58	0.74	0.75	**0.81**	**0.79**	**0.79**
	0.73	0.57	0.70	0.55	0.00	0.58	0.81	0.62	0.75	0.79	**0.84**	**0.83**	**0.83**
	0.59	0.56	0.70	0.54	0.00	0.44	**0.80**	0.60	0.76	0.77	**0.83**	**0.83**	**0.83**
MSNBC	0.73	0.69	0.71	0.42	0.02	0.22	0.78	0.62	0.73	0.50	**0.82**	**0.88**	**0.88**
	0.73	0.65	0.68	0.44	0.02	0.16	0.77	0.64	0.73	0.50	0.82	**0.90**	**0.89**
	0.74	0.74	0.76	0.46	0.02	0.24	0.84	0.67	0.79	0.55	**0.86**	**0.89**	**0.89**
	0.73	0.70	0.73	0.48	0.02	0.18	0.84	0.70	0.80	0.57	0.85	**0.91**	**0.90**
AIDA/CoNLL-Complete	0.55	0.68	0.66	0.50	0.51	0.38	0.61	0.20	0.71	0.47	0.75	**0.76**	**0.77**
	0.53	0.66	0.60	0.50	0.48	0.29	0.57	0.12	0.68	0.45	0.75	**0.76**	**0.77**
	0.57	0.77	0.74	0.58	0.54	0.44	0.68	0.24	**0.80**	0.54	**0.80**	**0.79**	**0.80**
	0.52	0.76	0.68	0.59	0.50	0.33	0.65	0.14	**0.78**	0.52	**0.78**	**0.78**	**0.79**
AIDA/CoNLL-Test A	0.54	0.67	0.65	0.48	0.49	0.28	0.61	0.00	0.70	0.45	**0.75**	**0.76**	**0.76**
	0.50	0.62	0.59	0.47	0.45	0.23	0.56	0.00	0.66	0.41	**0.73**	**0.75**	**0.75**
	0.56	0.74	0.74	0.55	0.53	0.33	0.67	0.00	0.78	0.52	**0.80**	0.78	**0.79**
	0.49	0.71	0.68	0.55	0.47	0.25	0.64	0.00	**0.76**	0.48	**0.77**	0.75	**0.76**
AIDA/CoNLL-Test B	0.54	0.69	0.68	0.52	0.49	0.35	0.61	0.01	0.72	0.47	**0.75**	**0.75**	**0.76**
	0.54	0.68	0.62	0.51	0.48	0.22	0.61	0.00	0.70	0.46	0.75	**0.76**	**0.77**
	0.55	0.77	0.76	0.60	0.52	0.40	0.69	0.00	**0.80**	0.54	**0.80**	0.77	**0.79**
	0.54	0.78	0.70	0.60	0.51	0.26	0.70	0.01	**0.80**	0.53	**0.79**	0.78	**0.79**
AIDA/CoNLL-Training	0.55	0.69	0.65	0.50	0.52	0.39	0.61	0.28	0.71	0.48	0.75	**0.76**	**0.77**
	0.53	0.66	0.60	0.50	0.50	0.30	0.56	0.17	0.68	0.45	**0.73**	**0.77**	**0.77**
	0.57	0.77	0.74	0.58	0.55	0.45	0.69	0.33	**0.81**	0.56	**0.80**	0.79	**0.80**
	0.52	0.76	0.68	0.59	0.51	0.35	0.64	0.21	**0.79**	0.53	**0.78**	**0.78**	**0.79**
DBpediaSpotlight	0.27	0.25	0.52	0.71	0.15	0.45	0.74	0.56	0.67	0.71	**0.79**	**0.79**	**0.80**
	0.28	0.21	0.51	0.69	0.12	0.31	0.73	0.53	0.69	0.71	0.80	**0.81**	**0.82**
	0.40	0.25	0.52	0.71	0.15	0.45	0.74	0.56	0.67	0.71	**0.80**	**0.80**	**0.81**
	0.36	0.21	0.51	0.69	0.12	0.31	0.73	0.53	0.69	0.71	0.80	**0.82**	**0.83**
KORE50	0.33	**0.69**	**0.74**	0.46	0.27	0.17	0.60	0.31	0.62	0.51	0.63	0.56	0.50
	0.30	**0.64**	**0.70**	0.42	0.22	0.14	0.53	0.25	0.52	0.45	0.58	0.52	0.50
	0.33	**0.69**	**0.74**	0.46	0.27	0.17	0.60	0.31	0.62	0.51	0.63	0.56	0.50
	0.30	**0.64**	**0.70**	0.42	0.22	0.14	0.53	0.25	0.52	0.45	0.59	0.52	0.50
Microposts2014-Test	0.33	0.42	0.48	0.50	0.22	0.42	0.64	0.52	0.60	0.55	**0.73**	0.63	**0.67**
	0.60	0.59	0.63	0.66	0.49	0.60	0.76	0.67	0.74	0.68	**0.85**	0.75	**0.79**
	0.42	0.42	0.48	0.50	0.22	0.42	0.64	0.52	0.60	0.55	**0.74**	0.65	**0.69**
	0.61	0.59	0.63	0.66	0.49	0.60	0.76	0.67	0.74	0.68	**0.85**	0.76	**0.79**
Microposts2014-Train	0.42	0.51	0.51	0.48	0.31	0.46	0.65	0.52	0.63	0.59	**0.71**	0.64	**0.67**
	0.61	0.61	0.61	0.61	0.48	0.56	0.74	0.63	0.73	0.67	**0.81**	0.74	**0.76**
	0.51	0.51	0.51	0.48	0.31	0.46	0.65	0.52	0.63	0.59	**0.73**	0.67	**0.70**
	0.63	0.61	0.61	0.61	0.48	0.56	0.74	0.63	0.73	0.67	**0.82**	0.75	**0.78**
N3-RSS-500	0.61	0.45	0.44	0.20	0.56	0.28	0.44	0.38	0.44	0.45	0.53	**0.69**	**0.68**
	0.61	0.39	0.38	0.16	0.54	0.20	0.39	0.30	0.37	0.38	0.53	**0.69**	**0.68**
	0.52	**0.66**	0.64	0.32	0.50	0.44	0.62	0.57	0.64	**0.65**	0.55	**0.65**	0.63
	0.52	**0.64**	0.63	0.41	0.49	0.45	0.61	0.58	0.63	**0.66**	0.48	0.62	0.61
N3-Reuters-128	**0.66**	0.47	0.45	0.33	0.54	0.24	0.51	0.41	0.52	0.39	**0.65**	0.63	0.64
	0.72	0.38	0.39	0.27	0.57	0.16	0.46	0.35	0.44	0.34	**0.72**	**0.63**	**0.63**
	0.64	0.57	0.55	0.41	0.52	0.31	0.61	0.51	0.63	0.49	**0.69**	0.62	**0.65**
	0.68	0.51	0.55	0.41	0.54	0.29	0.60	0.51	0.59	0.52	**0.72**	0.60	0.60
OKE 2015 Task 1 evaluation dataset	0.59	0.56	0.59	0.31	0.56	0.32	**0.63**	0.61	0.57	**0.62**	**0.63**	**0.62**	**0.62**
	0.60	0.55	0.58	0.27	0.53	0.26	**0.63**	0.60	0.56	0.61	**0.63**	**0.62**	**0.62**
	0.62	0.63	0.66	0.36	0.60	0.38	**0.71**	**0.70**	0.65	**0.71**	0.68	0.65	0.65
	0.61	0.62	0.65	0.30	0.56	0.28	**0.71**	0.68	0.62	**0.70**	0.67	0.64	0.65
OKE 2015 Task 1 example set	**1.00**	0.60	0.4	0.22	**0.78**	0.25	0.55	0.00	0.60	0.5	0.50	0.67	0.67
	1.00	0.72	0.65	0.44	0.67	0.44	0.69	0.33	0.72	0.69	0.67	**0.75**	**0.75**
	1.00	**0.86**	0.57	0.50	0.80	0.40	0.75	0.00	**0.86**	0.80	0.67	0.75	0.75
	1.00	**0.89**	0.80	0.33	**0.89**	0.50	0.82	0.33	**0.89**	**0.89**	0.78	0.82	0.82
OKE 2015 Task 1 gold standard sample	0.62	0.67	0.71	0.25	0.54	0.41	**0.78**	**0.77**	0.72	0.75	0.76	**0.78**	**0.78**
	0.64	0.65	0.68	0.20	0.49	0.32	0.76	0.74	0.69	0.73	0.76	**0.78**	**0.77**
	0.64	0.71	0.75	0.27	0.56	0.44	**0.81**	**0.81**	0.77	0.79	0.80	**0.82**	**0.82**
	0.64	0.67	0.72	0.22	0.53	0.35	**0.79**	0.77	0.73	0.76	0.78	**0.80**	**0.79**

Table 5.3: Results reported by GERBIL. The rows in each cell report the F1@Micro, F1@Macro, InKB F1@Micro, and InKB F1@Macro, in which **red** marks the highest F1 and **blue** marks the second highest F1.

5.4 Summary

In this chapter, we proposed a supervised approach for NED using a learning to rank algorithm to combine various local and global features, and experimentally evaluated the performance of our approach on 4 public benchmarks and a public NED evaluation framework. Our L2R approach was shown to outperform many state-of-the-arts, including our unsupervised approach WNED. More importantly, we found that the ranking model trained on one dataset (AIDA-CoNLL) also achieved high accuracy on other datasets. To further evaluate the robustness of our approaches, we constructed two balanced benchmarks from large corpora and the evaluation on them further demonstrated the robustness of our learning to rank approach.

While one of our goals is to exploit the available training data to learn a robust ranking model for the NED task, we only experimented with a limited number of features in our approach. There are a lot more features that are important to represent mentions and entities. For example, the neighboring entities in the entity graph and statistically connections between entities are important graph features that could capture some semantics of entities. Fine-grained types of mentions and entities are also useful features. Exploring these features to improve NED will be worth the effort, as shown in Chapter 6.

Chapter 6

Scaling out NED

In this chapter, we address the scalability issue of our random walk-based NED approaches. We propose to approximate the semantic signatures of entities using a small set of pre-selected landmark entities instead of all entities in the large disambiguation graph. We also explore other features derived from entity graphs and employ MinHash [54] to improve the efficiency.

6.1 Overview

While our approaches WNED and L2R can achieve the state-of-the-art accuracy, their efficiency is only moderate with the highest costs lying in building the disambiguation graphs and performing the random walks, which prevent the system from scaling out. Actually, many global approaches face major trade-offs between efficiency and accuracy like ours.

To improve the efficiency, many approaches turn to simple lexical and statistical features. For example, Spotlight [65] uses only word-level context and prior probability. Tagme [81] introduces semantic relatedness in their approach, but only focuses on short text with a limited number of mentions, thus cannot efficiently handle long text such as news articles. AIDA-light [76] uses a simplified semantic relatedness computed from the domain information of entities (*i.e.*, Wikipedia categories), which is efficient but less competitive compared to AIDA [45]. All these approaches, as can be seen, sacrifice one for the other between accuracy and efficiency.

Our approach follows the paradigm of global approaches aiming to find the assignment with the maximum global coherence. We assume that the semantics of an entity mentioned in a document can be captured by its relatedness to a set of pre-selected entities that are independent of the document, so that we can pre-compute these semantic signatures of entities offline without using the online constructed disambiguation graph. We explore various features from this semantic signature and the entity graphs. Through the entity graphs, we infer a set of graph-based semantic relatedness from the connection strength between entities and the entities in the direct neighborhood. We also represent entities using their describing documents and attributed categories. All these features can be pre-processed offline and loaded for online NED, which can support parallel computations on clusters because of their independence, thus can help scale out an NED system.

We continue to employ the iterative disambiguation algorithm and the learning to rank algorithm used in our previous approaches, which have been shown to be effective and robust. Recall that in WNED and L2R, we approximate the semantic relatedness between a candidate entity e_i and the assignment Γ using the similarity between the semantic signatures of e_i and the document d, as follows:

$$\psi(e_i, \Gamma) = \psi(e_i, \mathbf{d})$$

However, we cannot use this approximation anymore since our approach avoids the online computation of semantic signatures for d now. Instead, we measure the semantic relatedness between entity e_i and document d by summing up the relatedness between e_i and each entity mentioned in d, as defined below:

$$\psi(e_i, \Gamma) = \sum_{e_j \in E_d} \psi(e_i, e_j)$$

in which E_d is the set of entities representing the document d, referred to as the *semantic representation* of the document.

The main focus of this chapter is to find effective semantic representations of a document and semantic signatures of entities that can help measure semantic relatedness. Below we describe the details of features in our ranking model and various representations and signatures used to derive these features.

6.2 Local Features

We use 3 local features in our previous L2R approach: prior probability $prior(m, e)$, context similarity $ctxSim(m, e)$, and name similarity $nameSim(m, e)$

6.3 Global Features

Our global features mainly consist of semantic relatedness between entities and semantic relatedness between an entity and a document. We first describe 3 representations of documents, and then the features built on them.

6.3.1 Semantic Representation of Documents

A document can be represented in different ways, such as using bag-of-words, fine-grained categories, keyphrases, and named entities. To facilitate measures of semantic relatedness, we choose to use named entities to represent documents so that we can use the semantic signatures of entities to measure the global coherence. Below are 3 different document representations we have explored.

Entities from Unambiguous Mentions. Our first document representation, denoted as $E1_d$, is a set of entities referred to by unambiguous mentions in a given document, which have been shown to be effective in improving the accuracy of NED [32], [68].

Entities from PRIOR Baseline. The second document representation $E2_d$ consists of the entities resulted from the PRIOR baseline. As shown in our previous evaluation, the PRIOR baseline is competitive and can outperform a few competing NED approaches, thus should be able to generate results to well represent a document. Also, its efficiency is superior for our scalability requirement since prior probability is pre-computed and can be used for ranking without further processing.

Entities from Disambiguated Mentions. The last set of document representation $E3_d$ is the set of entities disambiguated up to each iteration. Our approach

uses an iterative algorithm which disambiguates mentions one by one. Thus in each iteration, we have a set of entities linked by mentions that are already disambiguated. This set is updated whenever a new mention is disambiguated, and is used as a semantic representation of the document. In the training phase, since we do not perform NED when constructing training instances, we use the entities in the ground truth up to each iteration as the document representation.

6.3.2 Semantic Signature of Entities

To measure the semantic relatedness $\psi(e_i, e_j)$ between entities, we explore a set of semantic signatures of entities from various sources, including two entity graphs, the describing documents of entities, and the attributed categories of entities from a KB. We first describe the entity graphs used to derive these signatures.

Entity Graph We use the two variants of EG described in Chapter 1: *PageLink Graph* and *Co-occurrence Graph*. Figure 6.1 gives a Co-occurrence graph around entities in the examples about *Jeff Malone* and *Karl Malone*, in which each edge is weighted by the total number of co-occurrences of two entities.

Semantic Signature using Landmarks

To solve the efficiency issue from the online graph construction and random walks in WNED and L2R, we propose to use a set of prominent entities in an entity graph to represent entities. We refer to these entities as *landmark nodes*, examples of which are shown in *gray* boxes in Figure 6.1. Using a set of landmarks $LM \subseteq E$ from the EG, we can represent each entity using a semantic signature (a vector of dimension $M = |LM|$) with each element gives the relatedness between the entity and the corresponding landmark. For example, the semantic signature of *Karl Malone* is a vector of relatedness between Karl Malone and each landmark.

To compute the relatedness between an entity and a landmark, we use an approximate Personalized PageRank (PPR) with restart from the landmark through a fast PPR algorithm [58]. Instead of measuring the probability (PPR) that a source node reaches to a target node (the method employed in WNED and L2R), this fast

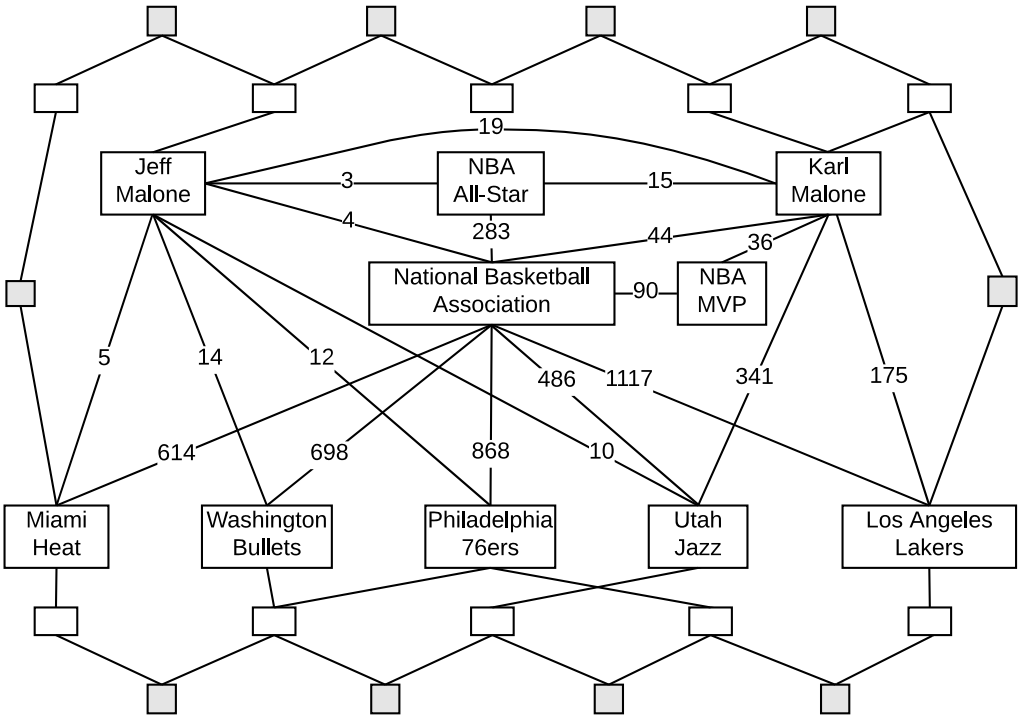


Figure 6.1: An example entity graph, in which nodes are entities and edges are the relations between entities. The weight on edges shows the connection strength between the connected entities, which is the number of co-occurrences in this example.

PPR algorithm estimates PPR using a bidirectional estimator which combines the probability from a source node s and a target node t to an intermediate set of nodes. The algorithm first finds a frontier set of nodes $\mathsf{FRONTIER}(t)$ and computes a reverse probability from t to each node in $\mathsf{FRONTIER}(t)$. It then performs a random walk with restart from the source node s to estimate the probability that s reaches to any nodes in $\mathsf{FRONTIER}(t)$. The reverse probability from t and forward probability from s to nodes in $\mathsf{FRONTIER}(t)$ are then combined to estimate the PPR from s to t. With each landmark as a source node, we run the fast PPR algorithm for each landmark and use the PPR score as a measure of semantic relatedness between a landmark and each entity. We then aggregate the semantic relatedness between an entity and all landmarks to build a semantic signature for each entity.

Ideally, landmarks should be representative entities in an EG that are discriminable enough to model the semantics of entities. Identifying these entities, however, is rather difficult and also subjective to applications. In this work, we explore

two different ways for the landmark selection. The first one is to select entities based on their global PageRank score (from a random walk over the entire entity graph), and the second is to select entities based on their types [1] with the top K highest ranked entities (by their PageRank score) under each type.

With the selected landmarks, we can pre-compute the semantic signatures of all entities (denoted as $Landmark(e)$), which will significantly improve the disambiguation efficiency.

Semantic Signature using Neighboring Entities

While the landmark-based semantic signature models the relatedness between an entity and other indirectly connected entities (landmarks), neighboring entities explicitly define the semantics between an entity and its directly connected entities [69]. Through these connections, we can build another semantic signature for each entity and use it to measure semantic relatedness.

From the *PageLink graph* and *Co-occurrence graph*, we build two different semantic signatures $PageLinkNbr(e_i)$ and $CooccurNbr(e_i)$ for each entity e_i, both consisting of the neighboring entities of e_i in the graph. For example, the semantic signature of *Karl Malone* in the EG in Figure 6.1 would be $\{$*Jeff Malone, NBA All-Star, National Basketball Association, NBA MVP, Utah Jazz, Los Angeles Lakers*$\}$.

To improve the efficiency of computing the semantic relatedness using the neighboring entities based semantic signature, we further process the semantic signatures (a set of entity names) using MinHash [54] and convert each of them into a vector (whose dimension is the number of hashing functions). Besides the efficiency improvement through pre-processing, another advantage of using MinHash is to alleviate the name variation issue through the shingles. For example, *NBA All-Star* and *NBA All-Star Challenge*, which do not have exact string matching, would match partially through their common shingle *NBA All-Star*. Furthermore, we also expand the name set of entities using their aliases from the alias dictionary. Overall, we construct 4 semantic signatures using the neighboring entities: $PageLinkNbr(e_i)$, $PageLinkNbrAlias(e_i)$, $CooccurNbr(e_i)$, and

[1] We used the 112 fine-grained types from the FIGER system [57]

$CooccurNbrAlias(e_i)$.

Other Semantic Signatures

Besides the semantic signatures derived from entity graphs, we also explore two more semantic signatures from the Wikipedia articles and categories.

Wikipedia Article. In Wikipedia, each article gives a detailed description of an entity. Thus the words and named entities in an article could be used to represent the corresponding entity. We use bag-of-words from Wikipedia articles to generate semantic signatures of entities and convert them to vectors through MinHash. As with neighboring entity representations, we pre-process all Wikipedia articles and cache them for efficient retrieval.

Wikipedia Category. Categories in Wikipedia are designed to help group articles on similar subjects, and are organized in a taxonomy-like structure (not strictly enforced). Each Wikipedia article may have multiple categories which can be treated as semantic tags. We use these categories associated with each entity as another semantic signature of entities. Again, each signature is converted to a vector through MinHash for efficiency purpose.

6.3.3 Semantic Features

With the semantic representations of documents and semantic signatures of entities, we can derive a set of semantic features using the semantic relatedness between entities. In this approach, we use the Jensen-Shannon Divergence [26] to compute the similarity between the landmark-based semantic signature of entities since it is a preferred measure for the probability distribution. For other MinHash based signatures, we use the Jaccard similarity coefficient [54] to compute the semantic relatedness.

Semantic Relatedness in Entity Graphs

While many approaches [44], [68] use semantic signatures of entities to measure semantic relatedness, they commonly neglect the semantic relatedness within EGs.

If we represent each EG as a matrix A, we can then build two matrices $A_{PageLink}$ and $A_{Co-occur}$ from the two EGs, with the semantic relatedness A_{ij} defined using a connection strength between e_i and e_j: the number of links in the PageLink Graph and the number of co-occurrences in the Co-occurrence Graph. We can use this connection strength between entities as a measure of semantic relatedness. So in the example EG shown in Figure 6.1, the semantic relatedness between *Jeff Malone* and *Utah Jazz* is 1 in $A_{PageRank}$ and 10 in $A_{Co-occur}$, while the semantic relatedness between *Karl Malone* and *Utah Jazz* is 1 in $A_{PageRank}$ and 341 in $A_{Co-occur}$.

6.3.4 Summary of Global Features

	Notation	Description
Semantic Representations of Documents	$E1_d$	Entities from unambiguous mentions
	$E2_d$	Entities from PRIOR baseline
	$E3_d$	Entities from disambiguated mentions
Semantic Relatedness	$Landmark(e)$	Using landmark based semantic signatures
	$PageLinkNbr(e)$	Using name of neighboring entities from PageLink graph
	$PageLinkNbrAlias(e)$	Using aliases of neighboring entities from PageLink graph
	$CooccurNbr(e)$	Using name of neighboring entities from Co-occurrence graph
	$CooccurNbrAlias(e)$	Using aliases of neighboring entities from Co-occurrence graph
	$WikiDoc(e)$	Using bag-of-words from Wikipedia documents
	$Category(e)$	Using attributed categories from Wikipedia
	$A_{PageLink}$	Using connection strength between entities in PageLink graph
	$A_{Co-occur}$	Using connection strength between entities in Co-occurrence graph

Table 6.1: 3 semantic representations of documents and 9 semantic relatedness.

Table 6.1 summarizes the global features we have explored, including 3 semantic representations of documents and 9 different semantic relatedness measures. For each document representation, 9 global coherence scores are computed between a candidate entity and the document, resulting in 27 global features. With the 3 local features, we have 30 features in total.

6.4 Iterative NED

Our NED approach uses an iterative disambiguation algorithm similar to the one used in WNED and L2R, which iterates over the list of mentions in a document and applies a learned ranking model on each mention using features described above. The main difference between our algorithm and the algorithms in WNED and L2R is that we pre-compute and cache the semantic signatures of entities so that our feature extraction is simply computing the semantic relatedness between entities using these semantic signatures, while algorithms in WNED and L2R have to compute the signature signatures and semantic relatedness online in each iteration.

6.5 Experimental Evaluation

We use the 4 public datasets and the 2 new benchmarks built in Chapter 5 for our evaluation. We use accuracy to measure the effectiveness and running time (milliseconds) to measure the efficiency. For the Wikipedia and ClueWeb datasets, the accuracy is reported up to 3 digits after the decimal point for a better comparison of the results. We also apply Bootstrap [20] with re-sampling and report the average accuracy with confidence intervals[2]. All results are reported with the number of re-samplings set to 100,000 and the confidence level set to 95%. For the implementation, we use the Stanford SNAP[3] for the random walk with restart. We use 500 hash functions for the MinHash implementation and the RankLib[4] for the learning to rank algorithms.

6.5.1 Evaluation of Learning to Rank Algorithms

We first evaluate different learning to rank algorithms and how each optimization metric and feature normalization method affect the accuracy of NED.

[2]We report the overall accuracy of all brackets for readability. Readers can refer to the corresponding tables in Appendix B for the detailed results on each bracket.

[3]`http://snap.stanford.edu/snap/index.html`

[4]`https://sourceforge.net/p/lemur/wiki/RankLib/`

Learning to Rank Algorithms

We compare a few learning to rank algorithms, including pointwise approaches (MART and CoordinateAscent), pairwise approaches (LambdaMART and RandomForest), and listwise approaches (AdaRank and ListNet). For the experiment setting, we use *NDCG@10* as the optimization metric, *linear* as the feature normalization method, and other default parameters in RankLib[5]. The results are reported on the Wikipedia dataset using all 30 features.

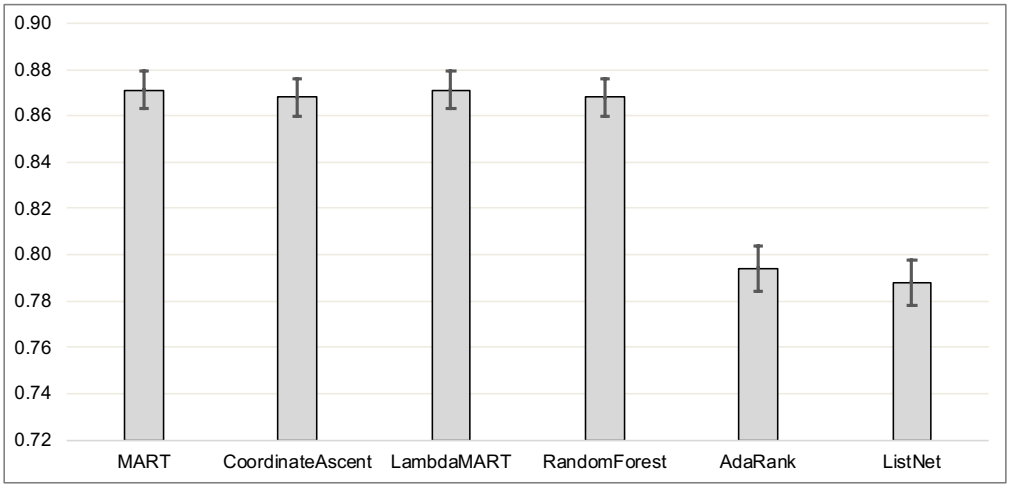


Figure 6.2: Accuracy of different learning to rank algorithms on the Wikipedia dataset.

As shown in Figure 6.2 (details in Table B.1), there is no winning algorithm on all buckets of the Wikipedia dataset. Among the 3 categories of learning to rank algorithms, listwise approaches, which highly rely on the ranking structure of training instances, get much lower accuracy than approaches in the other two categories. This result indicates that the ranking structure may play an important role in training models; thus high-quality ranking instances are required to make better use of the listwise approaches. We also find that the accuracy of pointwise approaches is quite similar to that of the pairwise approaches. One reason could be that training for the optimal ordering of instances is more important in our case when partial rankings (instead of full rankings) of training instances is used. In this work, we choose the LambdaMART algorithm to train our ranking model. We believe that other learning to rank algorithms can achieve similar accuracy.

[5] `https://sourceforge.net/p/lemur/wiki/RankLib/`

Optimization Metrics

The second experiment evaluates different optimization metrics used to learn ranking models. Figure 6.3 (details in Table B.2) shows the accuracy of ranking models trained using 6 different optimization metrics.

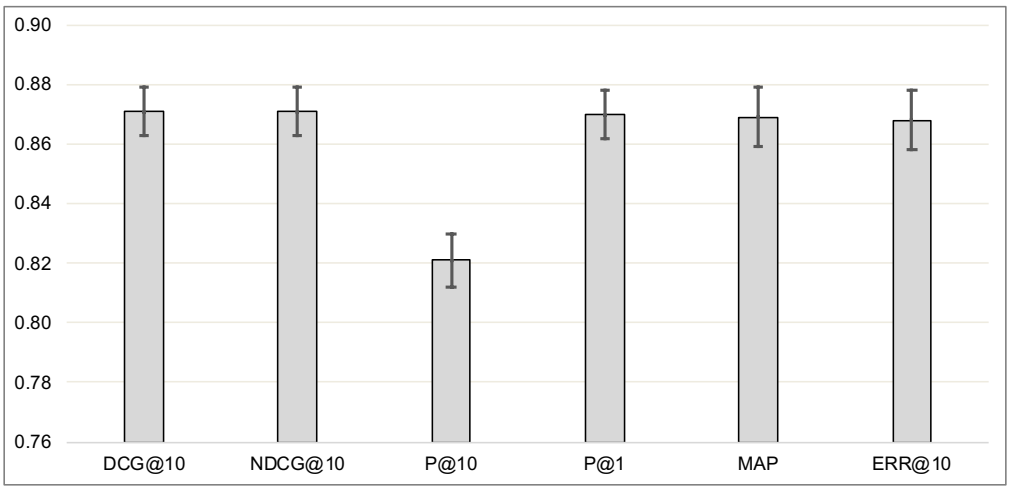


Figure 6.3: Accuracy of LambdaMART models using different optimization metrics on the Wikipedia dataset.

The result for P@1 is reported using a ranking model optimized for P@1 which is the same as our evaluation metric *accuracy*. As seen, most optimization metrics, except P@10, do not make much difference on the accuracy of ranking models. Here we simply pick NDCG@10 as our optimization metric for the model training. Other metrics, however, are also strong metrics.

Feature Normalization Methods

Most features used in our approach are independent of each other and their ranges of values vary widely, which could make some algorithms not working properly if not normalized. Here we evaluate 3 different feature normalization methods: *sum* which normalizes each feature by its summed value, *linear* which normalizes each feature by its max value, and *zscore* which normalizes each feature by its mean and standard deviation.

Figure 6.4 (details in Table B.3) shows the accuracy of our system using LambdaMART. Again, all feature normalization methods are comparable with each other

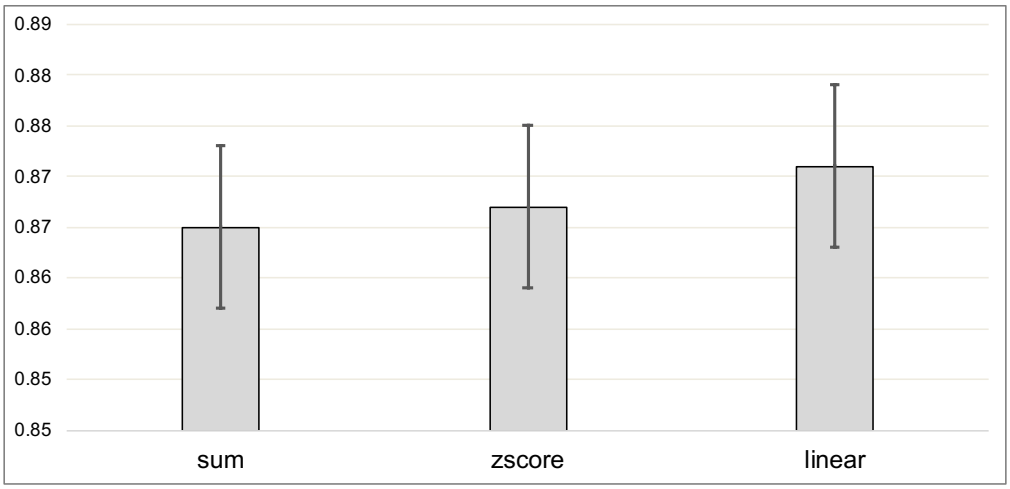


Figure 6.4: Accuracy of different feature normalization methods using LambdaMART and NDCG@10 on the Wikipedia dataset.

on the Wikipedia dataset. Although using linear achieves the best overall accuracy, the results are not statistically better than the other two. In the following experiments, we choose linear as our feature normalization method.

6.5.2 Evaluation of Features

In the following experiments, we use *LambdaMART* as our algorithm, *NDCG@10* as the optimization metric, and *linear* as the feature normalization method.

Semantic Representation of Documents

Figure 6.5 and Figure 6.6 (details in Table B.4 and Table B.5) show the accuracy of our NED approach using different document representations, in which *Unambiguous* represents $E1_d$ – *entities from unambiguous mentions*, *Prior* represents $E2_d$ – *entities from* PRIOR *baseline*, and *Disambiguated* represents $E3_d$ – *entities from disambiguated mentions*. Rows starting with "-" (*e.g.*, *-representation*) are combinations excluding a specific *representation*. For example, *-Unambiguous* is the combination of all representations except *Unambiguous*.

Out of the 3 representations, *Unambiguous* achieves the best accuracy on both Wikipedia and ClueWeb datasets. Its significance is further confirmed by the accuracy drop when *Unambiguous* is removed, which is the largest drop among all on both Wikipedia and ClueWeb datasets. This result somehow indicates that most

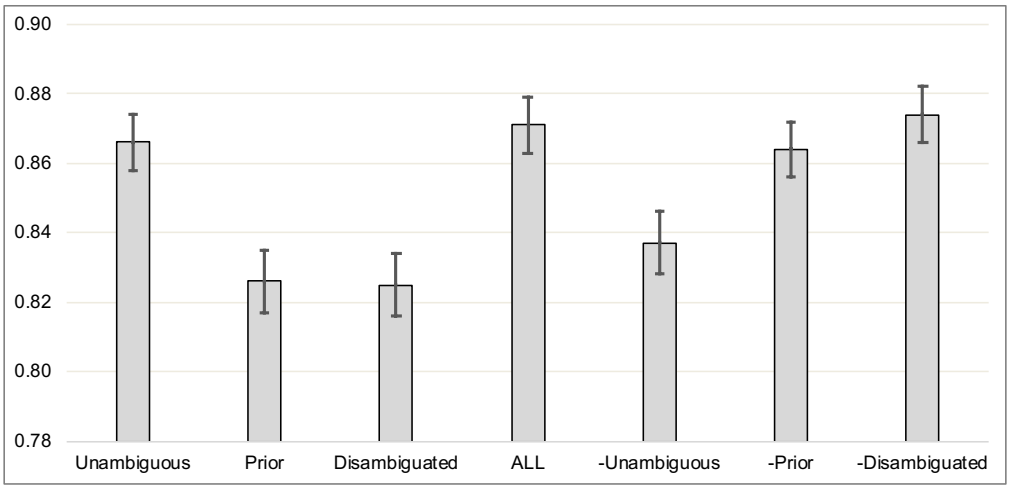


Figure 6.5: Accuracy with different document representations on the Wikipedia dataset.

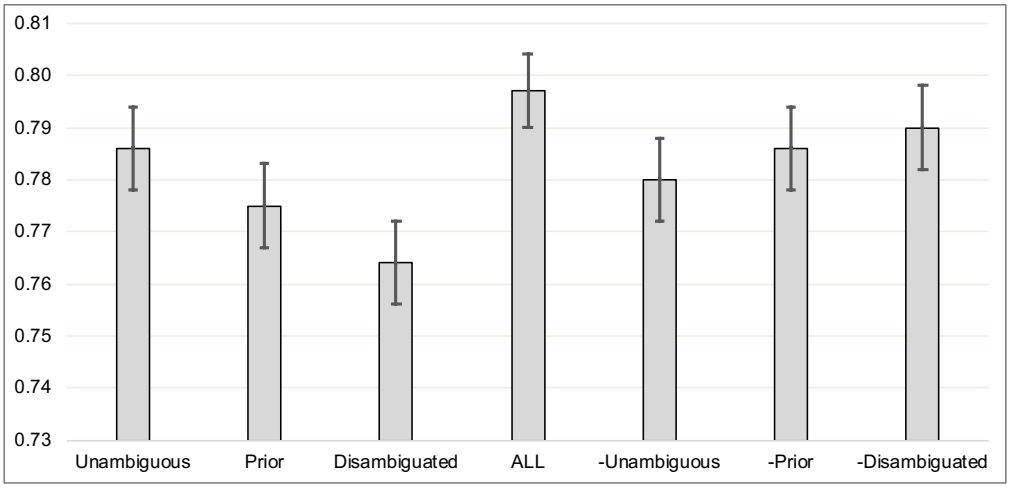


Figure 6.6: Accuracy with different document representations on the ClueWeb dataset.

unambiguous mentions are more related to the central topic of a document and thus are more coherent with entities of other mentions. One interesting result is that removing *Disambiguated* can help achieve better accuracy on the Wikipedia dataset than that using all 3 representations. This could be caused by errors in these disambiguated mentions. Results on the ClueWeb dataset also confirm the relative significance of each document representation on the Wikipedia dataset, except that the difference between different combinations on the ClueWeb dataset is much smaller.

Semantic Signature of Entities

We first evaluate the impact of different landmark selection strategies on the accuracy of NED. The experiment is performed using LambdaMART on all features and document representations with different sets of landmarks.

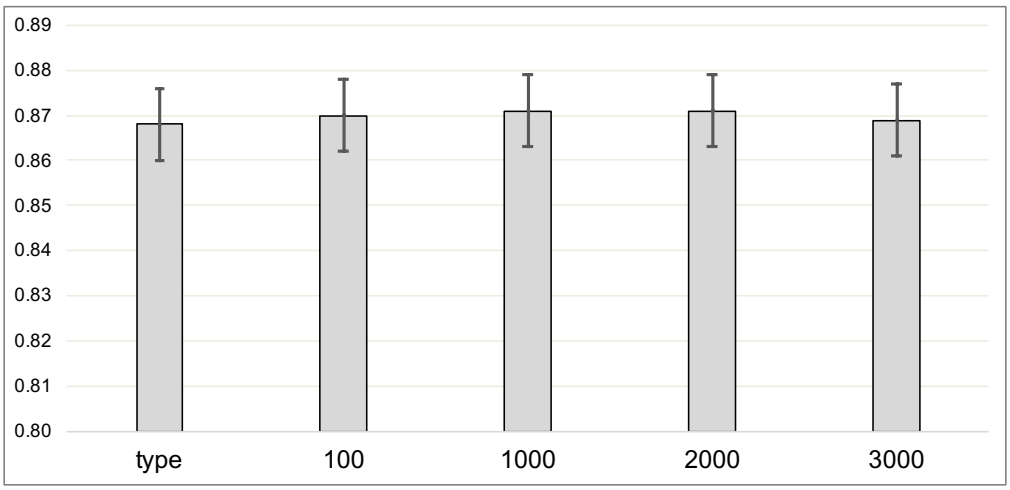


Figure 6.7: Accuracy of NED using different landmark strategies on the Wikipedia dataset.

Figure 6.7 (details in Table B.6) gives the accuracy using different set of landmarks, in which *type* represents the landmarks based on the entity types (952 landmarks in total), *100, 1000, 2000*, and *3000* represent landmarks with the top K highest PageRank values. As shown, different landmark selection strategies do not affect the results much. Also changing the number of landmarks does not result in any loss or improvement of the accuracy. In our approach, we choose 1000 landmarks to derive any related features of entities.

Feature evaluation Figure 6.8 and Figure 6.9 (details in Table B.7 and Table B.8) give the accuracy of our NED approach using features computed from different semantic signatures of entities. The first two bars show the results of WNED and L2R and the rest show the results of our approach using different features, in which *LOCAL* is the combination of the 3 local features. *Landmark* represents features computed using landmarks, *PL_NBR* and *PL_NBR_ALS* represent neighboring entities from the PageLink graph using name and aliases respectively, *CO_NBR* and *CO_NBR_ALS* represent neighboring entities from the Co-occurrence graph, *Wiki-*

Doc and *Category* represent features computed from the MinHash of Wikipedia documents and categories respectively, and *PL_SR* and *CO_SR* correspond to the semantic relatedness between entities using connection strength in the PageLink graph and Co-occurrence graph.

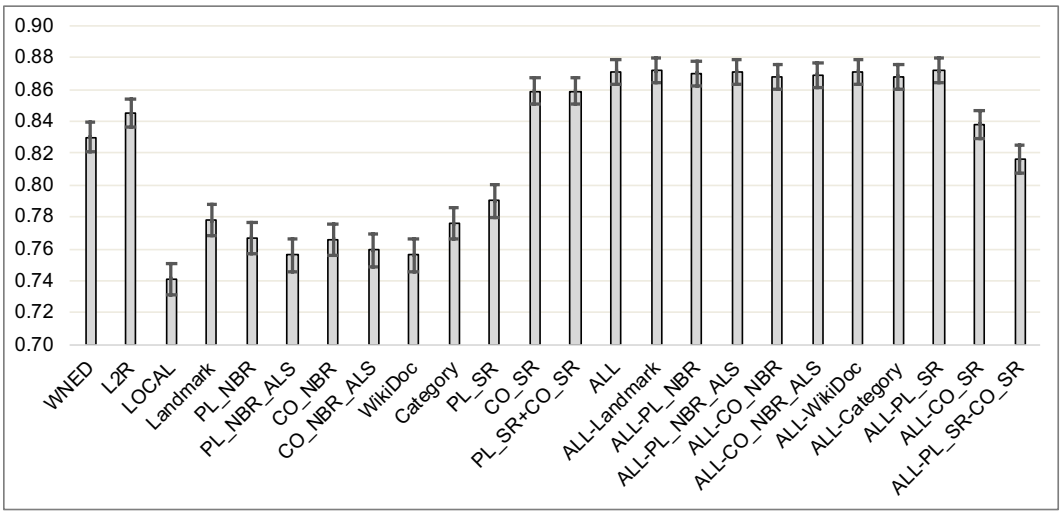


Figure 6.8: Accuracy of NED using features from different semantic signatures of entities on the Wikipedia dataset.

There are a few results on the Wikipedia dataset worth noting here. First, as expected, *LOCAL* performs the worst among all combinations. Second, *Landmark* gets higher accuracy than other MinHash-based semantic signatures (*i.e.*, neighboring entities from the PageLink and Co-occurrence graph, and the Wikipedia document) except *Categories*. Third, using the connection strength in the Co-occurrence graph as semantic relatedness can help get much better results than other features and outperforms WNED and L2R. Its importance can also be confirmed by the accuracy drop when removed from the whole feature set (the last 2 bars: *ALL-CO_SR*, and *ALL-PL_SR-CO_SR*). Last but not least, our model combining all 30 features (bar *ALL*) can outperform WNED by 4.1% and L2R by 2.6% on the Wikipedia dataset.

On the ClueWeb dataset, landmarks and MinHash-based features get comparable accuracy. One noticeable result is that *Category* performs much better than other MinHash-based features on both Wikipedia and ClueWeb datasets, which indicates that the categories annotated through crowd-source are good representations

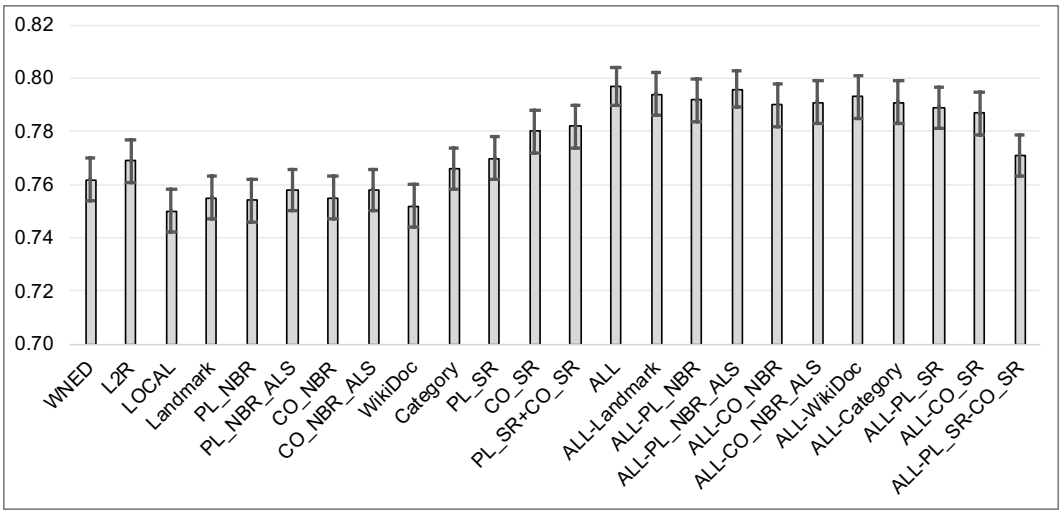


Figure 6.9: Accuracy of NED using features from different semantic signatures of entities on the ClueWeb dataset.

of entities. Similar to that on the Wikipedia dataset, using connection strength from the Co-occurrence graph as semantic relatedness also contributes most to the overall accuracy on the ClueWeb dataset. This advantage over other features on this non-Wikipedia dataset further confirms the effectiveness of using the connection strength between entities as a semantic relatedness measure. This simple measure with surprisingly high accuracy somehow shows that the human understanding of semantic relatedness between entities is actually well annotated and captured through the WikiLinks and associated documents of entities. Obtaining this co-occurrence on other non-Wikipedia corpora, though challenging, is worth exploring.

We then evaluate our approach on the 4 public datasets, presented in Figure 6.10 (details in Table B.9). As shown, most features can help achieve high accuracy, which demonstrates that these public datasets are easy to disambiguate. Similar to the results on the Wikipedia and ClueWeb datasets, using connection strength as semantic relatedness can help improve the overall accuracy on most datasets.

6.5.3 Evaluation of Efficiency

Our last experiment is to evaluate the efficiency of our two models using GERBIL: NED-SR which combines the LOCAL features with the semantic relatedness mea-

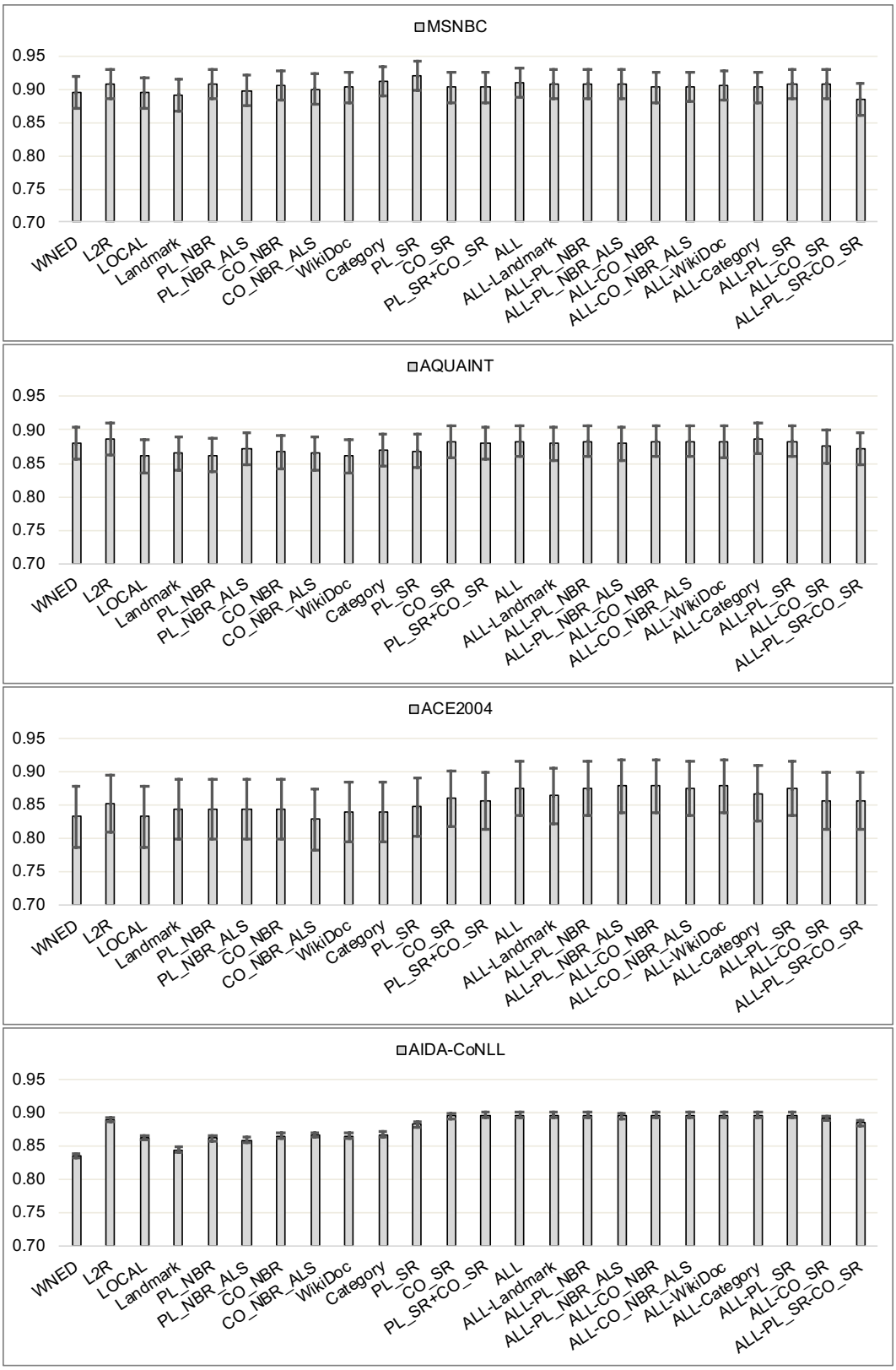


Figure 6.10: Accuracy of NED using features from different semantic signatures of entities on 4 public datasets.

sured using the connection strength, and NED-ALL which uses all 30 features. Both models are trained using the AIDA-CoNLL dataset.

Table 6.2 shows the results of our 4 systems and other 11 systems in GERBIL. As shown, NED-SR is quite competitive among all NED systems on most datasets and NED-ALL can further improve the results.

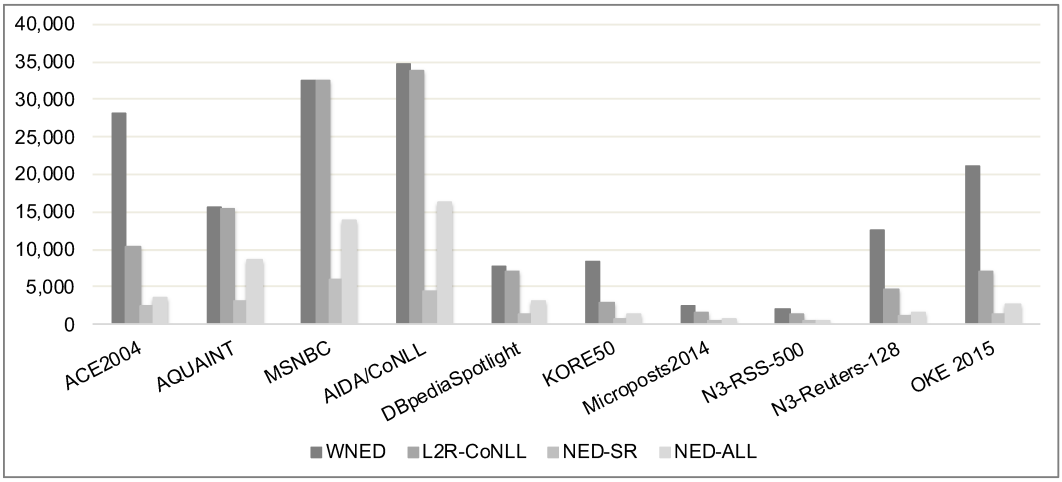


Figure 6.11: Average running time (milliseconds) per document in each benchmark of our 4 NED systems reported by GERBIL.

Figure 6.11 (details in Table B.10) reports the average running time per document of our 4 systems on each dataset (for AIDA/CoNLL, Microposts2004, and OKE 2015, we only report the result on one of their datasets). All experiments are run on a server with 8-core 2.20 GHz CPU and 64G memory, and the reported time is averaged over 10 repeated runnings. For WNED and L2R-CoNLL, a minimum of 55G memory has to be allocated for the evaluation because of the large size of entity graphs. While NED-SR and NED-ALL require only 5G memory for the evaluation. No parallel algorithms or frameworks are used in all 4 systems.

As shown in the figure, our NED-ALL requires only $1/3$ to $1/2$ running time of L2R-CoNLL while achieving comparable accuracy. The NED-SR, which uses only the connection strength between entities, can further reduce the running time by half on most datasets with very high accuracy (as shown in Table 6.2). With the reduced memory consumption and improved running efficiency, our NED system can be easily scaled out to handle large-scale datasets.

Datasets	AGDISTIS [98]	AIDA [45]	Babelfy [72]	DBpedia Spotlight [65]	FOX [92]	FREME NER [88]	Kea [93]	NERD-ML [86]	WAT [81]	xLisa [108]	PBoH [27]	WNED	L2R-CoNLL	NED-SR	NED-ALL
ACE2004	0.65	0.69	0.53	0.48	0.00	0.49	0.66	0.58	0.66	0.70	0.72	0.77	0.76	**0.82**	**0.83**
	0.77	0.82	0.70	0.68	0.37	0.65	0.77	0.73	0.77	0.80	0.83	**0.88**	0.87	**0.91**	**0.91**
	0.66	0.80	0.61	0.58	0.00	0.58	0.76	0.67	0.76	0.81	0.79	0.83	0.81	**0.85**	**0.86**
	0.78	0.89	0.76	0.75	0.39	0.71	0.84	0.79	0.85	0.88	0.86	0.91	0.90	**0.92**	**0.93**
AQUAINT	0.52	0.55	0.68	0.53	0.00	0.56	0.78	0.60	0.73	0.76	**0.81**	**0.79**	**0.79**	0.77	0.77
	0.51	0.55	0.68	0.52	0.00	0.43	0.78	0.58	0.74	0.75	**0.81**	**0.79**	**0.79**	0.77	0.77
	0.73	0.57	0.70	0.55	0.00	0.58	0.81	0.62	0.75	0.79	**0.84**	**0.83**	**0.83**	0.81	0.81
	0.59	0.56	0.70	0.54	0.00	0.44	0.80	0.60	0.76	0.77	**0.83**	**0.83**	**0.83**	**0.81**	**0.81**
MSNBC	0.73	0.69	0.71	0.42	0.02	0.22	0.78	0.62	0.73	0.50	0.82	**0.88**	**0.88**	**0.86**	**0.86**
	0.73	0.65	0.68	0.44	0.02	0.16	0.77	0.64	0.73	0.50	0.82	**0.90**	**0.89**	0.88	**0.89**
	0.74	0.74	0.76	0.46	0.02	0.24	0.84	0.67	0.79	0.55	0.86	**0.89**	**0.89**	**0.87**	**0.87**
	0.73	0.70	0.73	0.48	0.02	0.18	0.84	0.70	0.80	0.57	0.85	**0.91**	**0.90**	0.88	0.89
AIDA/CoNLL-Complete	0.55	0.68	0.66	0.50	0.51	0.38	0.61	0.20	0.71	0.47	0.75	0.76	0.77	**0.82**	**0.83**
	0.53	0.66	0.60	0.50	0.48	0.29	0.57	0.12	0.68	0.45	0.75	0.76	0.77	**0.81**	**0.82**
	0.57	0.77	0.74	0.58	0.54	0.44	0.68	0.24	0.80	0.54	0.80	0.79	0.80	**0.84**	**0.85**
	0.52	0.76	0.68	0.59	0.50	0.33	0.65	0.14	0.78	0.52	0.78	0.78	0.79	**0.82**	**0.83**
AIDA/CoNLL-Test A	0.54	0.67	0.65	0.48	0.49	0.28	0.61	0.00	0.70	0.45	0.75	**0.76**	**0.76**	**0.80**	**0.80**
	0.50	0.62	0.59	0.47	0.45	0.23	0.56	0.00	0.66	0.41	0.73	0.75	0.75	**0.78**	**0.79**
	0.56	0.74	0.74	0.55	0.53	0.33	0.67	0.00	0.78	0.52	**0.80**	0.78	0.79	**0.83**	**0.83**
	0.49	0.71	0.68	0.55	0.47	0.25	0.64	0.00	0.76	0.48	0.77	0.75	0.76	**0.79**	**0.80**
AIDA/CoNLL-Test B	0.54	0.69	0.68	0.52	0.49	0.35	0.61	0.01	0.72	0.47	0.75	0.75	0.76	**0.81**	**0.83**
	0.54	0.68	0.62	0.51	0.48	0.22	0.61	0.00	0.70	0.46	0.75	0.76	0.77	**0.80**	**0.82**
	0.55	0.77	0.76	0.60	0.52	0.40	0.69	0.00	0.80	0.54	0.80	0.77	0.79	**0.83**	**0.85**
	0.54	0.78	0.70	0.60	0.51	0.26	0.70	0.01	0.80	0.53	0.79	0.78	0.79	**0.80**	**0.83**
AIDA/CoNLL-Training	0.55	0.69	0.65	0.50	0.52	0.39	0.61	0.28	0.71	0.48	0.75	0.76	0.77	**0.82**	**0.83**
	0.53	0.66	0.60	0.50	0.50	0.30	0.56	0.17	0.68	0.45	0.73	0.77	0.77	**0.81**	**0.82**
	0.57	0.77	0.74	0.58	0.55	0.45	0.69	0.33	0.81	0.56	0.80	0.79	0.80	**0.85**	**0.86**
	0.52	0.76	0.68	0.59	0.51	0.35	0.64	0.21	0.79	0.53	0.78	0.78	0.79	**0.82**	**0.83**
DBpediaSpotlight	0.27	0.25	0.52	0.71	0.15	0.45	0.74	0.56	0.67	0.71	**0.79**	**0.79**	**0.80**	0.74	0.74
	0.28	0.21	0.51	0.69	0.12	0.31	0.73	0.53	0.69	0.71	0.80	**0.81**	**0.82**	0.75	0.76
	0.40	0.25	0.52	0.71	0.15	0.45	0.74	0.56	0.67	0.71	**0.80**	**0.80**	**0.81**	0.75	0.75
	0.36	0.21	0.51	0.69	0.12	0.31	0.73	0.53	0.69	0.71	0.80	**0.82**	**0.83**	0.76	0.77
KORE50	0.33	**0.69**	**0.74**	0.46	0.27	0.17	0.60	0.31	0.62	0.51	0.63	0.56	0.50	0.56	0.61
	0.30	**0.64**	**0.70**	0.42	0.22	0.14	0.53	0.25	0.52	0.45	0.58	0.52	0.50	0.51	0.63
	0.33	**0.69**	**0.74**	0.46	0.27	0.17	0.60	0.31	0.62	0.51	0.63	0.56	0.50	0.57	0.64
	0.30	**0.64**	**0.70**	0.42	0.22	0.14	0.53	0.25	0.52	0.45	0.59	0.52	0.50	0.52	0.61
Microposts2014-Test	0.33	0.42	0.48	0.50	0.22	0.42	0.64	0.52	0.60	0.55	**0.73**	0.63	**0.67**	0.64	0.64
	0.60	0.59	0.63	0.66	0.49	0.60	0.76	0.67	0.74	0.68	**0.85**	0.75	**0.79**	0.78	0.78
	0.42	0.42	0.48	0.50	0.22	0.42	0.64	0.52	0.60	0.55	**0.74**	0.65	**0.69**	0.67	0.67
	0.61	0.59	0.63	0.66	0.49	0.60	0.76	0.67	0.74	0.68	**0.85**	0.76	**0.79**	0.78	0.78
Microposts2014-Train	0.42	0.51	0.51	0.48	0.31	0.46	0.65	0.52	0.63	0.59	**0.71**	0.64	**0.67**	**0.67**	**0.67**
	0.61	0.61	0.61	0.61	0.48	0.56	0.74	0.63	0.73	0.67	**0.81**	0.74	**0.76**	**0.76**	**0.76**
	0.51	0.51	0.51	0.48	0.31	0.46	0.65	0.52	0.63	0.59	**0.73**	0.67	**0.70**	**0.70**	**0.70**
	0.63	0.61	0.61	0.61	0.48	0.56	0.74	0.63	0.73	0.67	**0.82**	0.75	**0.78**	**0.78**	**0.78**
N3-RSS-500	0.61	0.45	0.44	0.20	0.56	0.28	0.44	0.38	0.44	0.45	0.53	**0.69**	0.68	**0.70**	**0.70**
	0.61	0.39	0.38	0.16	0.54	0.20	0.39	0.30	0.37	0.38	0.53	**0.69**	0.68	**0.70**	**0.70**
	0.52	**0.66**	0.64	0.32	0.50	0.44	0.62	0.57	0.64	**0.65**	0.55	**0.65**	0.63	0.64	0.64
	0.52	**0.64**	0.63	0.41	0.49	0.45	0.61	0.58	0.63	**0.66**	0.48	0.62	0.61	0.62	0.62
N3-Reuters-128	**0.66**	0.47	0.45	0.33	0.54	0.24	0.51	0.41	0.52	0.39	**0.65**	0.63	0.64	0.63	0.64
	0.72	0.38	0.39	0.27	0.57	0.16	0.46	0.35	0.44	0.34	**0.72**	**0.63**	**0.63**	**0.63**	**0.63**
	0.64	0.57	0.55	0.41	0.52	0.31	0.61	0.51	0.63	0.49	**0.69**	0.62	**0.65**	0.63	0.64
	0.68	0.51	0.55	0.41	0.54	0.29	0.60	0.51	0.59	0.52	**0.72**	0.60	0.60	0.59	0.60
OKE 2015 Task 1 evaluation dataset	0.59	0.56	0.59	0.31	0.56	0.32	**0.63**	0.61	0.57	**0.62**	**0.63**	**0.62**	**0.62**	**0.62**	**0.62**
	0.60	0.55	0.58	0.27	0.53	0.26	**0.63**	0.60	0.56	0.61	**0.63**	**0.62**	**0.62**	**0.62**	**0.62**
	0.62	0.63	0.66	0.36	0.60	0.38	**0.71**	**0.70**	0.65	**0.71**	0.68	0.65	0.65	0.65	0.65
	0.61	0.62	0.65	0.30	0.56	0.28	**0.71**	0.68	0.62	**0.70**	0.67	0.64	0.65	0.65	0.65
OKE 2015 Task 1 example set	**1.00**	0.60	0.4	0.22	0.78	0.25	0.55	0.00	0.60	0.50	0.50	0.67	0.67	0.82	**0.83**
	1.00	0.72	0.65	0.44	0.67	0.44	0.69	0.33	0.72	0.69	0.67	**0.75**	**0.75**	0.58	**0.75**
	1.00	0.86	0.57	0.50	0.80	0.40	0.75	0.00	0.86	0.80	0.67	0.75	0.75	0.89	**0.90**
	1.00	**0.89**	0.80	0.33	**0.89**	0.50	0.82	0.33	**0.89**	**0.89**	0.78	0.82	0.82	0.60	0.82
OKE 2015 Task 1 gold standard	0.62	0.67	0.71	0.25	0.54	0.41	**0.78**	**0.77**	0.72	0.75	0.76	**0.78**	**0.78**	**0.77**	**0.77**
	0.64	0.65	0.68	0.20	0.49	0.32	0.76	0.74	0.69	0.73	0.76	**0.78**	**0.77**	**0.77**	**0.77**
	0.64	0.71	0.75	0.27	0.56	0.44	**0.81**	**0.81**	0.77	0.79	0.80	**0.82**	**0.82**	**0.81**	**0.81**
	0.64	0.67	0.72	0.22	0.53	0.35	**0.79**	0.77	0.73	0.76	0.78	**0.80**	**0.79**	0.78	0.78

Table 6.2: Results reported by GERBIL. The rows in each cell report the F1@Micro, F1@Macro, InKB F1@Micro, and InKB F1@Macro, in which **red** marks the highest F1 and **blue** marks the second highest F1.

6.6 Summary

In this chapter, we explored various features derived from the local context, entity graphs, describing documents, and attributed categories to build an efficient NED system. We mainly focused on the representations of documents and semantic signatures of entities. We found that the entities of unambiguous mentions were better representations for a document than others, with much higher accuracy on both the Wikipedia and ClueWeb datasets. For the landmark-based entity representation, its accuracy, though slightly better than a few other representations, is much worse than that of WNED and L2R which use the semantic signatures computed on the disambiguation graph. Another notable finding is that the connection strength between entities in the Co-occurrence graph is an effective measure of semantic relatedness and performs much better than other features. At last, we evaluated our approach using the GERBIL framework and demonstrated that our new approaches required much less memory and can greatly reduce the processing time per document compared to our previous approaches WNED and L2R.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we investigated the named entity ambiguity problem and presented approaches to address the challenges arising in building an accurate, robust, and scalable NED system. Our main contributions are in the methods to measure semantic relatedness and the algorithms to disambiguate mentions, both related to the global coherence – the core notion of global approaches.

For semantic relatedness, we mainly measured it using semantic signatures of entities. We first proposed a novel method to compute semantic signatures of entities and documents using random walks with restart on a disambiguation graph curated for a document. The semantic signature has two advantages: (1) it incorporates the relatedness from indirectly connected entities in the disambiguation graph, which overcomes the feature sparsity issue for entities with little context; and (2) it gives a uniform representation for both entities and documents, which provides a novel way to measure the global coherence. Experiments showed that this semantic signature can help achieve state-of-the-art accuracy.

We then proposed another method to compute semantic signatures using the random walk model, but with a set of pre-selected landmarks instead of the disambiguation graph curated for each document. Since the landmarks are selected independently from the document, the semantic signature of entities can then be pre-computed offline, so that we can avoid the expensive online random walks and thus improve the scalability of our NED system. Besides, we also explored other

semantic signatures to represent entities using neighboring entities, describing documents, and attributed categories. We experimentally demonstrated the efficiency of our NED system using these semantic signatures for semantic relatedness measure in NED. Particularly, we found that using the connection strength between entities in the Co-occurrence graph as the semantic relatedness measure was surprisingly effective, with high accuracy and a large gain in efficiency.

For disambiguation, we presented an iterative algorithm that incorporated the interdependency between mentions in a document into an iterative process. In each iteration, we take the disambiguation results from previous iterations into the measure of the semantic relatedness between candidate entities and the document, and use a ranking model to disambiguate the mention. We first explored a hand-tuned ranking model, which was simple but not robust to the change of datasets and features. We then employed a learning to rank algorithm to improve the robustness. For the global coherence, we measured it in two ways. The first way is to use the semantic relatedness between the uniform semantic signatures of the candidate entity and the document, and the second is to use an aggregated semantic relatedness between the candidate entity and the set of entities representing the document.

Overall, from our evaluation, we found that the semantic relatedness from lexical representations can only achieve mediocre accuracy, while the statistical information from large corpora, such as the *prior probability* and *connection strength*, can achieve high accuracy, especially on datasets with popular entities. The semantic signatures derived from entity graphs, on the other hand, can complement them by providing semantic information for entities. These findings indicate that in many cases, statistical information from *larger* corpora is enough for NED to win easy victories. In the uphill battles towards higher accuracy, it would be useful to exploit semantic information like the semantic signatures proposed in this thesis.

Another contribution of this thesis is revealing the biased issue of most public benchmarks, for which we constructed two benchmarks from large Web corpora (*i.e.*, Wikipedia and ClueWeb 2012) with documents of balanced difficulty (*i.e.*, they have the same number of documents in each difficulty class). We demonstrated the robustness of our NED approaches using the two new benchmarks.

7.2 Future Work

7.2.1 Improving Candidate Selection

In the error analysis in Chapter 4, around 54% disambiguation errors are from candidate selection, caused by co-reference resolution mistakes, incomplete alias dictionaries, or aggressive pruning criteria. Thus improving the candidate selection would be a direction to boost the accuracy of an NED system. For alias dictionaries, we can improve the coverage by extracting alternative names of entities from plain text, such as using information extraction techniques to extract nicknames from sentence *"... Malone, nicknamed "The MailMan"..."*. For candidate pruning, exploring semantic information such as entity types would be worth exploring. Moreover, as more features are introduced into candidate selection, many computations would be repeated in both candidate selection and mention disambiguation. Therefore, we can unify the two steps together in an iterative way: use the results from mention disambiguation to improve candidate selection and limit the disambiguation on a more relevant candidate set from candidate selection.

7.2.2 Joint NED with Entity Typing and Relation Extraction

NED is commonly designed as a standalone task in the pipeline of a KBP system, separated from other tasks such as NER, RE, and entity typing. While these tasks are performed independently, they can also mutually benefit from each other if done jointly. For example, fine-grained types of a mention would provide information that can help with both candidate selection and mention disambiguation. Knowing that *Los Angeles* refers to a *basketball team*, we would prune the city *Los Angeles* and teams of other sports from the candidate set. On the other hand, disambiguating a mention to its referent entity in a KB will directly retrieve the types of entities, which not only helps the entity typing itself but also provides training datasets for the task. Besides entity typing, relation extraction can also provide additional information for NED as shown by Cheng and Roth [15] and benefit from NED. Therefore, jointly solving problems in NED, entity typing, and relation extraction would be a direction worth exploring.

7.2.3 Enriching WikiLinks in Wikipedia

As shown in the thesis, the connections between entities not only help derive semantic signatures of entities, but also can be used as an effective measure of the semantic relatedness between entities. Since repeated links to the same entity are not encouraged [1] in Wikipedia, entities in the graphs constructed using WikiLinks are actually not connected as well as they are supposed to. Thus enriching WikiLinks in Wikipedia would result in a much better connected entity graph, which can further improve the semantic representation or the connection strength measure. One method for the WikiLink enrichment is to chain mentions that refer to the same entity through co-reference resolution and create a WikiLink for each of them. We can also enrich WikiLinks through an iterative process on the Wikipedia corpus: (1) apply an NED model on each document to find new WikiLinks; (2) regenerate the entity graph using these new WikiLinks and update the NED model; (3) repeat the process until no new WikiLinks can be found. This iterative method for link enrichment could also be applied on other corpora such as the ClueWeb to enrich entity graphs using connections outside of Wikipedia.

7.2.4 NED on Other Data Sources

Besides documents from Wikipedia and news corpora, documents from other sources also provide rich facts and relations about named entities. Examples are microposts from social sites and online reviews of entities (*e.g.*, products and restaurants) from E-commerce sites. Different from news articles, these documents are short and casual with little context, which makes the NED more challenging. Web tables, on the other hand, are knowledge-rich sources with no textual context. However, entities in a table follow a *coherence assumption* in the sense that entities in a column or a row are of the same type or share the same relationship. Thus applying the methods and algorithms developed in the thesis on these data sources would be an interesting direction to explore.

[1] `https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking`

7.2.5 Combining Random Walk Model with Deep Neural Network for NED

Deep Neural Network (DNN) based approaches have been explored to address the NED problem with promising results [24], [34], [42]. Most of those approaches use the embeddings of entities learned from the lexical and statistical knowledge in large corpora such as Wikipedia. These embeddings encode the semantics of entities into low-dimensional continuous vector spaces and are used to measure the semantic coherence of assignments. Recent knowledge graph embeddings [103] incorporate the graph structure of knowledge graphs into the embeddings of entities, and applying them on NED will be interesting. However, most DNN-based approaches are not able to handle the indirect connections to explore the potential of the large graph structure like our random walk-based models do. Thus combining our random walk model with DNN models would be another promising direction to further improve NED. One way is to use models such as our L2R approach to combine the representations from the two independent models with each representing a set of features. Another idea is to learn embeddings of entities by incorporating the random walk into the training of DNN models using approaches like DeepWalk [79], which can learn embeddings of entities that capture the semantics from indirect connections in our entity graphs.

References

[1] Ivo Anastácio, Bruno Martins, and Pável Calado, "Supervised learning for linking named entities to knowledge base entries," in *Proceedings of the Fourth Text Analysis Conference, TAC 2011, November 14-15, 2011, Gaithersburg, Maryland, USA*, 2011.

[2] Nguyen Bach and Sameer Badaskar, "A survey on relation extraction," *Language Technologies Institute, Carnegie Mellon University*, 2007.

[3] Amit Bagga and Breck Baldwin, "Entity-based cross-document coreferencing using the vector space model," in *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference.*, 1998, pp. 79–85.

[4] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann, "Dbpedia - A crystallization point for the web of data," *J. Web Sem.*, vol. 7, no. 3, pp. 154–165, 2009.

[5] Jens Bleiholder and Felix Naumann, "Data fusion," *ACM Comput. Surv.*, vol. 41, no. 1, 1:1–1:41, 2008.

[6] Ronald J. Brachman and Hector J. Levesque, *Knowledge Representation and Reasoning*. Elsevier, 2004.

[7] Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[8] Razvan C. Bunescu and Raymond J. Mooney, "A shortest path dependency kernel for relation extraction," in *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, HLT/EMNLP 2005, 6-8 October 2005, Vancouver, British Columbia, Canada*, The Association for Computational Linguistics, 2005, pp. 724–731.

[9] Razvan C. Bunescu and Marius Pasca, "Using encyclopedic knowledge for named entity disambiguation," in *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*, 2006.

[10] Zhiyuan Cai, Kaiqi Zhao, Kenny Q. Zhu, and Haixun Wang, "Wikification via link co-occurrence," in *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, October 27 - November 1, 2013, San Francisco, CA, USA*, 2013, pp. 1087–1096.

[11] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li, "Learning to rank: From pairwise approach to listwise approach," in *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), June 20-24, 2007, Corvallis, Oregon, USA*, 2007, pp. 129–136.

[12] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell, "Toward an architecture for never-ending language learning," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, July 11-15, 2010, Atlanta, Georgia, USA*, 2010.

[13] Vitor R. Carvalho, Yigit Kiran, and Andrew Borthwick, "The intelius nickname collection: Quantitative analyses from billions of public records," in *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, 2012, pp. 607–610.

[14] Jinxiu Chen, Dong-Hong Ji, Chew Lim Tan, and Zheng-Yu Niu, "Relation extraction using label propagation based semi-supervised learning," in *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 17-21 July 2006, Sydney, Australia*, 2006.

[15] Xiao Cheng and Dan Roth, "Relational inference for wikification," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA*, 2013, pp. 1787–1796.

[16] Ronan Collobert and Jason Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), June 5-9, 2008, Helsinki, Finland*, ser. ACM International Conference Proceeding Series, vol. 307, ACM, 2008, pp. 160–167.

[17] Silviu Cucerzan, "Large-scale named entity disambiguation based on wikipedia data," in *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, 2007, pp. 708–716.

[18] Hong-Jie Dai, Chi-Yang Wu, Richard Tzong-Han Tsai, and Wen-Lian Hsu, "From entity recognition to entity linking: A survey of advanced entity linking techniques," in *Proceedings of the 26th Annual Conference of the Japanese Society for Artificial Intelligence*, 2012, pp. 110–120.

[19] Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin, "Entity disambiguation for knowledge base population," in *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, 2010, pp. 277–285.

[20] Bradley Efron and Robert J. Tibshirani, *An Introduction to the Bootstrap*, ser. Monographs on Statistics and Applied Probability 57. Boca Raton, Florida, USA: Chapman & Hall/CRC, 1993.

[21] Anthony Fader, Stephen Soderland, and Oren Etzioni, "Identifying relations for open information extraction," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK.*, ACL, 2011, pp. 1535–1545.

[22] Paolo Ferragina and Ugo Scaiella, "Fast and accurate annotation of short texts with wikipedia pages," *IEEE Software*, vol. 29, no. 1, pp. 70–75, 2012.

[23] Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, 2005, pp. 363–370.

[24] Matthew Francis-Landau, Greg Durrett, and Dan Klein, "Capturing semantic similarity for entity linking with convolutional neural networks," in *The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016, June 12-17, 2016, San Diego, California, USA.*, The Association for Computational Linguistics, 2016, pp. 1256–1261.

[25] Jerome H Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[26] Bent Fuglede and Flemming Topsoe, "Jensen-shannon divergence and hilbert space embedding," in *International Symposium on Information Theory*, 2004, p. 31.

[27] Octavian-Eugen Ganea, Marina Ganea, Aurélien Lucchi, Carsten Eickhoff, and Thomas Hofmann, "Probabilistic bag-of-hyperlinks model for entity linking," in *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, April 11 - 15, 2016, Montreal, Canada*, 2016, pp. 927–938.

[28] M. R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[29] Stephen Guo, Ming-Wei Chang, and Emre Kiciman, "To link or not to link? A study on end-to-end tweet entity linking," in *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, 2013, pp. 1020–1030.

[30] Yuhang Guo, Guohua Tang, Wanxiang Che, Ting Liu, and Sheng Li, "HIT approaches to entity linking at TAC 2011," in *Proceedings of the Fourth Text Analysis Conference, TAC 2011, November 14-15, 2011, Gaithersburg, Maryland, USA*, 2011.

[31] Zhaochen Guo and Denilson Barbosa, "Entity linking with a unified semantic representation," in *23rd International World Wide Web Conference, WWW '14, April 7-11, 2014, Seoul, Republic of Korea, Companion Volume*, 2014, pp. 1305–1310.

[32] Zhaochen Guo and Denilson Barbosa, "Robust entity linking via random walks," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, November 3-7, 2014, Shanghai, China*, 2014, pp. 499–508.

[33] Zhaochen Guo and Denilson Barbosa, "Robust named entity disambiguation with random walks," *Semantic Web*, vol. Preprint, pp. 1–21, 2017.

[34] Nitish Gupta, Sameer Singh, and Dan Roth, "Entity linking via joint encoding of types, descriptions, and context," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, September 9-11, 2017, Copenhagen, Denmark*, Association for Computational Linguistics, 2017, pp. 2671–2680.

[35] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran, "Evaluating entity linking with wikipedia," *Artif. Intell.*, vol. 194, pp. 130–150, 2013.

[36] Xianpei Han and Le Sun, "A generative entity-mention model for linking entities with knowledge base," in *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, 2011, pp. 945–954.

[37] Xianpei Han and Le 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, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, 2012, pp. 105–115.

[38] Xianpei Han, Le Sun, and Jun Zhao, "Collective entity linking in web text: A graph-based method," in *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, July 25-29, 2011, Beijing, China*, 2011, pp. 765–774.

[39] Xianpei Han and Jun Zhao, "Named entity disambiguation by leveraging wikipedia semantic knowledge," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, November 2-6, 2009, Hong Kong, China*, 2009, pp. 215–224.

[40] Sanda M. Harabagiu, Steven J. Maiorano, and Marius Pasca, "Open-domain textual question answering techniques," *Natural Language Engineering*, vol. 9, no. 3, pp. 231–267, 2003.

[41] Taher H. Haveliwala, "Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 784–796, 2003.

[42] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang, "Learning entity representation for entity disambiguation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, The Association for Computer Linguistics, 2013, pp. 30–34.

[43] Marti A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *14th International Conference on Computational Linguistics, COLING 1992, August 23-28, 1992, Nantes, France*, 1992, pp. 539–545.

[44] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum, "KORE: keyphrase overlap relatedness for entity disambiguation," in *21st ACM International Conference on Information and Knowledge Management, CIKM'12, October 29 - November 02, 2012, Maui, HI, USA*, 2012, pp. 545–554.

[45] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum, "Robust disambiguation of named entities in text," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK*, 2011, pp. 782–792.

[46] Thad Hughes and Daniel Ramage, "Lexical semantic relatedness with random graph walks," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2007, June 28-30, 2007, Prague, Czech Republic*, 2007, pp. 581–589.

[47] Stephanie Husby and Denilson Barbosa, "Topic classification of blog posts using distant supervision," in *Proceedings of the Workshop on Semantic Analysis in Social Media*, Association for Computational Linguistics, 2012, pp. 28–36.

[48] Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian, "Overview of tac-kbp2015 tri-lingual entity discovery and linking," in *Proceedings of the Text Analysis Conference*, 2015.

[49] Yohan Jo and Alice H. Oh, "Aspect and sentiment unification model for online review analysis," in *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, February 9-12, 2011, Hong Kong, China*, Irwin King, Wolfgang Nejdl, and Hang Li, Eds., ACM, 2011, pp. 815–824.

[50] Grzegorz Kondrak, "N-gram similarity and distance," in *String Processing and Information Retrieval*, ser. Lecture Notes in Computer Science, vol. 3772, Springer Berlin Heidelberg, 2005, pp. 115–126.

[51] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti, "Collective annotation of wikipedia entities in web text," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, June 28 - July 1, Paris, France*, 2009, pp. 457–466.

[52] John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi, "LCC approaches to knowledge base population at TAC 2010," in *Proceedings of the Third Text Analysis Conference, TAC 2010, November 14-15, 2011, Gaithersburg, Maryland, USA*, 2010.

[53] Douglas B. Lenat, "CYC: A large-scale investment in knowledge infrastructure," *Commun. ACM*, vol. 38, no. 11, pp. 32–38, 1995.

[54] Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman, *Mining of Massive Datasets, 2nd Ed*. Cambridge University Press, 2014.

[55] Hang Li, *Learning to Rank for Information Retrieval and Natural Language Processing*, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2011.

[56] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti, "Annotating and searching web tables using entities, types and relationships," *PVLDB*, vol. 3, no. 1, pp. 1338–1347, 2010.

[57] Xiao Ling and Daniel S. Weld, "Fine-grained entity recognition," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012.

[58] Peter Lofgren, Siddhartha Banerjee, and Ashish Goel, "Personalized pagerank estimation and search: A bidirectional approach," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, February 22-25, 2016, San Francisco, CA, USA*, 2016, pp. 163–172.

[59] Aibek Makazhanov, Davood Rafiei, and Muhammad Waqar, "Predicting political preference of twitter users," *Social Netw. Analys. Mining*, vol. 4, no. 1, p. 193, 2014.

[60] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008.

[61] Olena Medelyan, Ian H. Witten, and David Milne, "Topic indexing with wikipedia," in *Proceedings of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, 13 July, 2008, Chicago, USA*, 2008, pp. 19–24.

[62] Edgar Meij, Krisztian Balog, and Daan Odijk, "Entity linking and retrieval," in *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, July 28 - August 01, 2013, Dublin, Ireland*, 2013, p. 1127.

[63] Edgar Meij, Krisztian Balog, and Daan Odijk, "Entity linking and retrieval for semantic search," in *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, February 24-28, 2014, New York, NY, USA*, 2014, pp. 683–684.

[64] Edgar Meij, Wouter Weerkamp, and Maarten de Rijke, "Adding semantics to microblog posts," in *Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, February 8-12, 2012, Seattle, WA, USA*, 2012, pp. 563–572.

[65] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer, "Dbpedia spotlight: Shedding light on the web of documents," in *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, September 7-9, 2011, Graz, Austria*, 2011, pp. 1–8.

[66] Rada Mihalcea and Andras Csomai, "Wikify!: Linking documents to encyclopedic knowledge," in *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, November 6-10, 2007, Lisbon, Portugal*, 2007, pp. 233–242.

[67] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, 2013, pp. 3111–3119.

[68] David N. Milne and Ian H. Witten, "Learning to link with wikipedia," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, October 26-30, 2008, Napa Valley, California, USA*, 2008, pp. 509–518.

[69] David Milne and Ian H. Witten, "An effective, low-cost measure of semantic relatedness obtained from wikipedia links," in *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI, 2008), 13 July, 2008, Chicago, USA*, 2008, pp. 25–30.

[70] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky, "Distant supervision for relation extraction without labeled data," in *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, The Association for Computer Linguistics, 2009, pp. 1003–1011.

[71] Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Tanti Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm

Greaves, and Joel Welling, "Never-ending learning," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, AAAI Press, 2015, pp. 2302–2310.

[72] Andrea Moro, Alessandro Raganato, and Roberto Navigli, "Entity linking meets word sense disambiguation: A unified approach," *TACL*, vol. 2, pp. 231–244, 2014.

[73] David Nadeau and Satoshi Sekine, "A survey of named entity recognition and classification," *Linguisticae Investigationes*, vol. 30, no. 1, 2007.

[74] Roberto Navigli, "Word sense disambiguation: A survey," *ACM Comput. Surv.*, vol. 41, no. 2, 10:1–10:69, 2009.

[75] Vincent Ng, "Supervised noun phrase coreference research: The first fifteen years," in *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, 2010, pp. 1396–1411.

[76] Dat Ba Nguyen, Johannes Hoffart, Martin Theobald, and Gerhard Weikum, "Aida-light: High-throughput named-entity disambiguation," in *Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014), April 8, 2014, Seoul, Korea*, 2014.

[77] Thien Huu Nguyen, Nicolas Fauceglia, Mariano Rodriguez-Muro, Oktie Hassanzadeh, Alfio Massimiliano Gliozzo, and Mohammad Sadoghi, "Joint learning of local and global features for entity linking via neural networks," in *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, ACL, 2016, pp. 2310–2320.

[78] Bo Pang and Lillian Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2007.

[79] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena, "Deepwalk: Online learning of social representations," in *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, 2014, pp. 701–710.

[80] Minh C. Phan, Aixin Sun, Yi Tay, Jialong Han, and Chenliang 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, CIKM 2017, November 06 - 10, 2017, Singapore.*, ACM, 2017, pp. 1667–1676.

[81] Francesco Piccinno and Paolo Ferragina, "From tagme to WAT: a new entity annotator," in *ERD'14, Proceedings of the First ACM International Workshop on Entity Recognition & Disambiguation, July 11, 2014, Gold Coast, Queensland, Australia*, 2014, pp. 55–62.

[82] Danuta Ploch, Leonhard Hennig, Ernesto William De Luca, and Sahin Albayrak, "DAI approaches to the TAC-KBP 2011 entity linking task," in *Proceedings of the Fourth Text Analysis Conference, TAC 2011, November 14-15, 2011, Gaithersburg, Maryland, USA*, 2011.

[83] William Radford, Joel Nothman, Matthew Honnibal, James R. Curran, and Ben Hachey, "Document-level entity linking: CMCRC at TAC 2010," in *Proceedings of the Third Text Analysis Conference, TAC 2010, November 15-16, 2010, Gaithersburg, Maryland, USA*, 2010.

[84] Lev-Arie Ratinov and Dan Roth, "GLOW TAC-KBP2011 entity linking system," in *Proceedings of the Fourth Text Analysis Conference, TAC 2011, November 14-15, 2011, Gaithersburg, Maryland, USA*, 2011.

[85] Lev-Arie Ratinov, Dan Roth, Doug Downey, and Mike Anderson, "Local and global algorithms for disambiguation to wikipedia," in *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, 2011, pp. 1375–1384.

[86] Giuseppe Rizzo, Marieke van Erp, and Raphaël Troncy, "Benchmarking the extraction and disambiguation of named entities on the semantic web," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, May 26-31, 2014, Reykjavik, Iceland.*, 2014, pp. 4593–4600.

[87] Sunita Sarawagi, "Information extraction," *Foundations and Trends in Databases*, vol. 1, no. 3, pp. 261–377, 2008.

[88] Felix Sasaki, Tatiana Gornostay, Milan Dojchinovski, Michele Osella, Erik Mannens, Giannis Stoitsis, Phil Ritchie, Thierry Declerck, and Kevin Koidl, "Introducing FREME: deploying linguistic linked data," in *Proceedings of the Fourth Workshop on the Multilingual Semantic Web (MSW4) co-located with 12th Extended Semantic Web Conference (ESWC 2015), June 1, 2015, Portorož, Slovenia.*, 2015, pp. 59–66.

[89] Wei Shen, Jianyong Wang, and Jiawei Han, "Entity linking with a knowledge base: Issues, techniques, and solutions," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 443–460, 2015.

[90] Avirup Sil, Ernest Cronin, Penghai Nie, Yinfei Yang, Ana-Maria Popescu, and Alexander Yates, "Linking named entities to any database," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, 2012, pp. 116–127.

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

[92] René Speck and Axel-Cyrille Ngonga Ngomo, "Named entity recognition using FOX," in *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, October 21, 2014, Riva del Garda, Italy.*, 2014, pp. 85–88.

[93] Nadine Steinmetz and Harald Sack, "Semantic multimedia information retrieval based on contextual descriptions," in *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, May 26-30, 2013, Montpellier, France. Proceedings*, 2013, pp. 382–396.

[94] Ang Sun, Ralph Grishman, and Satoshi Sekine, "Semi-supervised relation extraction with large-scale word clustering," in *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, 2011, pp. 521–529.

[95] Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang, "Modeling mention, context and entity with neural networks for entity disambiguation," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, AAAI Press, 2015, pp. 1333–1339.

[96] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan, "Fast random walk with restart and its applications," in *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China*, 2006, pp. 613–622.

[97] William Tunstall-Pedoe, "True knowledge: Open-domain question answering using structured knowledge and inference," *AI Magazine*, vol. 31, no. 3, pp. 80–92, 2010.

[98] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both, "AGDISTIS - agnostic disambiguation of named entities using linked open data," in *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, 2014, pp. 1113–1114.

[99] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann, "GERBIL: general entity annotator benchmarking framework," in *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, May 18-22, 2015, Florence, Italy*, 2015, pp. 1133–1143.

[100] Petros Venetis, Alon Y. Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, Gengxin Miao, and Chung Wu, "Recovering semantics of tables on the web," *PVLDB*, vol. 4, no. 9, pp. 528–538, 2011.

[101] Nina Wacholder, Yael Ravin, and Misook Choi, "Disambiguation of proper names in text," in *5th Applied Natural Language Processing Conference, ANLP 1997, March 31 - April 3, 1997, Marriott Hotel, Washington, USA*, 1997, pp. 202–208.

[102] Jingjing Wang, Haixun Wang, Zhongyuan Wang, and Kenny Qili Zhu, "Understanding tables on the web," in *Conceptual Modeling - 31st International Conference ER 2012, Florence, Italy, October 15-18, 2012. Proceedings*, 2012, pp. 141–155.

[103] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, 2017.

[104] Ian H. Witten, Frank Eibe, and Mark A. Hall, *Data mining: practical machine learning tools and techniques, 3rd Edition*. Morgan Kaufmann, Elsevier, 2011.

[105] Qiang Wu, Christopher J. C. Burges, Krysta Marie Svore, and Jianfeng Gao, "Adapting boosting for information retrieval measures," *Inf. Retr.*, vol. 13, no. 3, pp. 254–270, 2010.

[106] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu, "Probase: A probabilistic taxonomy for text understanding," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, May 20-24, 2012, Scottsdale, AZ, USA*, 2012, pp. 481–492.

[107] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji, "Joint learning of the embedding of words and entities for named entity disambiguation," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, August 11-12, 2016, Berlin, Germany*, ACL, 2016, pp. 250–259.

[108] Lei Zhang and Achim Rettinger, "X-lisa: Cross-lingual semantic annotation," *PVLDB*, vol. 7, no. 13, pp. 1693–1696, 2014.

[109] Tao Zhang, Kang Liu, and Jun Zhao, "The nlpr_tac entity linking system at TAC 2011," in *Proceedings of the Fourth Text Analysis Conference, TAC 2011, November 14-15, 2011, Gaithersburg, Maryland, USA*, 2011.

[110] Wei Zhang, Yan Chuan Sim, Jian Su, and Chew Lim Tan, "Entity linking with effective acronym expansion, instance selection, and topic modeling," in *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, July 16-22, 2011, Barcelona, Catalonia, Spain*, 2011, pp. 1909–1914.

[111] Wei Zhang, Jian Su, Chew Lim Tan, and Wenting Wang, "Entity linking leveraging automatically generated annotation," in *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, 2010, pp. 1290–1298.

[112] Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu, "Learning to link entities with knowledge base," in *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, 2010, pp. 483–491.

[113] Yiping Zhou, Lan Nie, Omid Rouhani-Kalleh, Flavian Vasile, and Scott Gaffney, "Resolving surface forms to wikipedia topics," in *23rd International Conference on Computational Linguistics, COLING 2010, 23-27 August 2010, Beijing, China*, 2010, pp. 1335–1343.

Appendix A

New Benchmarks for NED

In this chapter, we first present a deep analysis of 4 public benchmarks: MSNBC, AQUAINT, ACE2004, and AIDA-CoNLL. We then describe a framework for deriving unbiased benchmarks and the steps to construct two benchmarks using the framework.

A.1 Analysis of the Public Benchmarks

To analyze the benchmarks, we break down the documents in each dataset by their level of accuracy achieved by PRIOR (*i.e.*, the brackets are determined by the overall accuracy of all mentions in a document). As shown in Table A.1, the vast majority of documents in these benchmarks are not particularly challenging. In fact, PRIOR produces perfect results on as many as 20% of documents in AQUAINT and AIDA-CoNLL and 31% of documents in ACE2004. It follows that these benchmarks are dated and unlikely to lead to further significant improvements in the area.

A desirable feature of any thorough evaluation that is not necessarily fulfilled by any of these benchmarks is that of *representativeness*. Namely, it would be ideal to have a mix of mentions or documents with *different levels of difficulty* in equal proportions (say on a 10-point scale from "easy" to "hard"). Without such equity, the effectiveness metrics reported in the literature (which aggregate at the mention or document level) may not be good predictors of actual performance in real applications. For instance, if a large fraction of mentions in a benchmark are "too easy" compared with real datasets, the metrics will overestimate the true accuracy.

Accuracy	MSNBC		AQUAINT		ACE2004		AIDA-CoNLL	
	#docs	#mentions	#docs	#mentions	#docs	#mentions	#docs	#mentions
0.0 – 0.1	0 (0%)	0	0 (0%)	0	0 (0%)	0	5 (0.4%)	5.0
0.1 – 0.2	0 (0%)	0	0 (0%)	0	0 (0%)	0	35 (2.5%)	40.4
0.2 – 0.3	0 (0%)	0	0 (0%)	0	0 (0%)	0	29 (2.1%)	20.2
0.3 – 0.4	0 (0%)	0	0 (0%)	0	0 (0%)	0	62 (4.5%)	17.4
0.4 – 0.5	2 (10%)	51.5	0 (0%)	0	0 (0%)	0	61 (4.4%)	30.0
0.5 – 0.6	3 (15%)	45.7	0 (0%)	0	0 (0%)	0	100 (7.2%)	22.5
0.6 – 0.7	3 (15%)	37.0	1 (2%)	8.0	5 (14.3%)	10.8	164 (11.8%)	21.7
0.7 – 0.8	4 (20%)	29.8	12 (24%)	15.3	5 (14.3%)	10.8	210 (15.1%)	26.8
0.8 – 0.9	3 (15%)	53.0	16 (32%)	14.4	12 (34.3%)	8.5	267 (19.2%)	28.3
0.9 – 1.0	3 (15%)	25.0	11 (22%)	15.0	2 (5.7%)	12.0	164 (11.8%)	43.5
1.0	2 (10%)	17.5	10 (20%)	13.9	11 (31.4%)	6.4	291 (21.0%)	13.2

Table A.1: Breakdown of the public benchmarks by the accuracy of the PRIOR method; #docs and #mentions are, respectively, the number of documents and the average number of mentions per document in each bracket; the number in parenthesis is the fraction of the entire benchmark covered by each bracket.

Of course, to fine-tune the difficulty of disambiguating mentions in a benchmark one needs a reliable indicator of "difficulty" that can be applied to a large number of documents. Manual annotations are undesirable here, and so is *crowdsourcing*: the number of annotations needed might prove prohibitive, and even if resources are not a concern this leads to a *single* benchmark (*i.e.*, if more documents are needed, more annotations would be required).

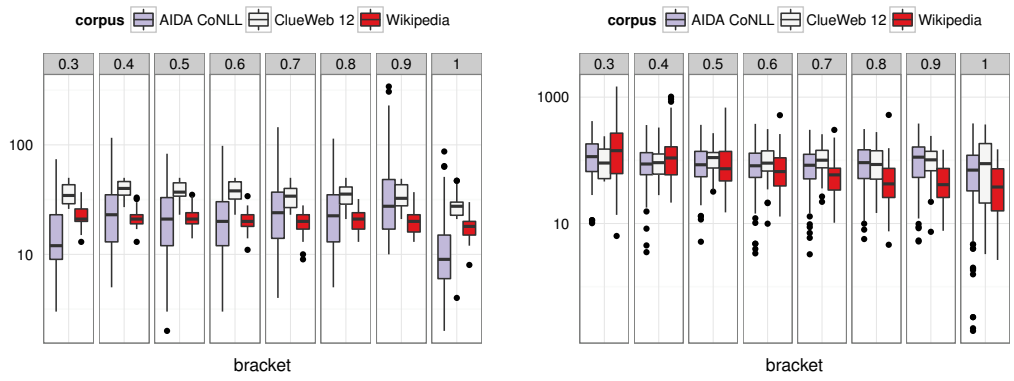
A.2 Benchmarks with Varying Difficulty

One of the challenging problems for this benchmark is to measure the difficulty of disambiguating a mention. As shown above, whether the right entity of a given mention is the most popular one among the candidates is an important indicator for a disambiguation, and the *prior probability* is one of the important measures. Therefore, we consider the PRIOR baseline as a proxy for the true difficulty of a mention and use it to help construct new and balanced benchmarks.

We start from large publicly annotated corpora, such as ClueWeb [1] and Wikipedia, and obtain the datasets by sampling from documents in these corpora. In this way, we can quickly collect large corpora of annotated documents and retain as many as needed while tuning the disambiguation difficulty to the desired proportion.

[1] `http://lemurproject.org/clueweb12/`

More precisely, we apply PRIOR on all documents of Wikipedia (the 20130606 dump) and the FACC1 annotated ClueWeb 2012 dataset [2], and measure the disambiguation accuracy on each document. We then group documents by the resulting average accuracy of all mentions in the document, and randomly pick 40 documents for each bracket (the level of accuracy as shown in Table A.1). Also, we further restrict the benchmarks to documents in which PRIOR achieved 0.3 or higher accuracy as we observe that below that threshold, the quality of the annotations in the ClueWeb dataset is quite low. Finally, we control the number of mentions per document: for the Wikipedia corpus we have the mean at 20.8 ($\sigma = 4.9$) and for the ClueWeb 2012 we have the mean at 35.5 ($\sigma = 8.5$).



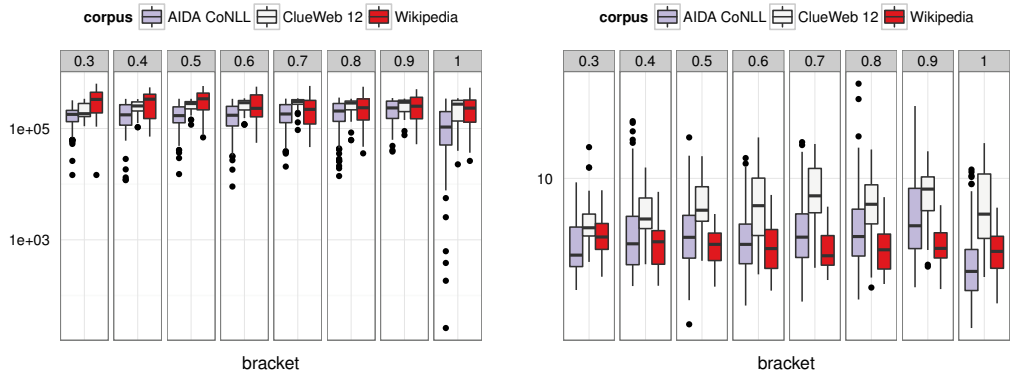
(a) Mentions per document.

(b) Candidate entities per mention.

Figure A.1: Corpus statistics.

Here are some statistics about the proposed benchmarks: Fig. A.1a shows the average number of mentions per document and Fig. A.1b shows the average number of candidates per mention. For the sake of comparison, we also report the same statistics from the documents in the AIDA-CONLL dataset in the respective accuracy brackets. Fig. A.2 shows statistics about the disambiguation graphs built by our method (which, as discussed in Chapter 4, depend both on the number of candidates per mention and on how densely connected they are in the disambiguation graph). Fig. A.2a shows the average graph sizes (in terms of the number of nodes) and Fig. A.2b shows the average node degree.

[2] http://lemurproject.org/clueweb12/FACC1/



(a) Number of nodes.

(b) Average node degree.

Figure A.2: Disambiguation graph statistics.

As one can see, the variability in our datasets is considerably smaller compared with AIDA-CoNLL, particularly when it comes to clear outliers (indicated as individual dots in the charts). More details about the datasets can be found in `http://dx.doi.org/10.7939/DVN/10968`

Appendix B

Tables with Detailed Evaluation Results

B.1 Evaluation of Learning to Rank Algorithms

	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0	1.0	Overall
MART	0.779±0.027	0.789±0.026	0.821±0.025	0.844±0.024	0.898±0.021	0.919±0.018	0.961±0.013	0.997±0.004	0.871±0.008
CoordinateAscent	0.716±0.029	0.807±0.025	0.822±0.025	0.856±0.023	0.906±0.021	0.924±0.018	0.961±0.013	0.999±0.003	0.868±0.008
LambdaMART	0.757±0.028	0.806±0.026	0.806±0.026	0.860±0.023	0.888±0.022	0.935±0.016	0.956±0.014	0.999±0.003	0.871±0.008
RandomForest	0.745±0.028	0.774±0.027	0.820±0.025	0.866±0.022	0.902±0.021	0.929±0.017	0.958±0.014	0.999±0.003	0.868±0.008
AdaRank	0.627±0.031	0.559±0.032	0.739±0.029	0.776±0.027	0.870±0.024	0.920±0.018	0.945±0.016	0.999±0.003	0.794±0.010
ListNet	0.662±0.031	0.669±0.030	0.731±0.029	0.784±0.027	0.853±0.025	0.846±0.024	0.893±0.021	0.914±0.020	0.788±0.010

Table B.1: Accuracy of different learning to rank algorithms on the Wikipedia dataset.

	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0	1.0	Overall
DCG@10	0.757±0.028	0.806±0.026	0.806±0.026	0.860±0.023	0.888±0.022	0.935±0.016	0.956±0.014	0.999±0.003	0.871±0.008
NDCG@10	0.757±0.028	0.806±0.026	0.806±0.026	0.860±0.023	0.888±0.022	0.935±0.016	0.956±0.014	0.999±0.003	0.871±0.008
P@10	0.646±0.031	0.725±0.029	0.755±0.028	0.813±0.025	0.871±0.024	0.884±0.021	0.938±0.017	0.999±0.003	0.821±0.009
P@1	0.758±0.028	0.798±0.026	0.812±0.026	0.855±0.023	0.889±0.022	0.932±0.017	0.955±0.014	0.999±0.003	0.870±0.008
MAP	0.757±0.028	0.790±0.026	0.812±0.026	0.854±0.023	0.893±0.022	0.931±0.017	0.959±0.014	0.999±0.003	0.869±0.008
ERR@10	0.760±0.028	0.794±0.026	0.803±0.026	0.854±0.023	0.890±0.022	0.931±0.017	0.955±0.014	0.999±0.003	0.868±0.008

Table B.2: Accuracy of LambdaMART using different optimization metrics on the Wikipedia dataset.

	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0	1.0	Overall
sum	0.762±0.027	0.796±0.026	0.791±0.027	0.846±0.023	0.897±0.021	0.922±0.018	0.956±0.014	0.995±0.005	0.865±0.008
zscore	0.768±0.027	0.797±0.026	0.810±0.026	0.837±0.024	0.891±0.022	0.924±0.018	0.956±0.014	0.995±0.005	0.867±0.008
linear	0.757±0.028	0.806±0.025	0.806±0.026	0.860±0.022	0.888±0.022	0.935±0.016	0.956±0.014	0.999±0.003	0.871±0.008

Table B.3: Accuracy of different feature normalization methods using LambdaMART and NDCG@10 on the Wikipedia dataset.

B.2 Evaluation of Features

	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0	1.0	Overall
Unambiguous	0.746±0.028	0.775±0.027	0.830±0.025	0.846±0.024	0.895±0.022	0.917±0.018	0.963±0.013	0.999±0.003	0.866±0.008
Prior	0.661±0.030	0.700±0.030	0.758±0.028	0.832±0.024	0.857±0.025	0.913±0.019	0.951±0.015	0.999±0.003	0.826±0.009
Disambiguated	0.681±0.030	0.722±0.029	0.749±0.029	0.822±0.025	0.848±0.025	0.892±0.021	0.941±0.016	0.999±0.003	0.825±0.009
ALL	0.757±0.028	0.806±0.026	0.806±0.026	0.860±0.023	0.888±0.022	0.935±0.016	0.956±0.014	0.999±0.003	0.871±0.008
-Unambiguous	0.676±0.030	0.745±0.028	0.779±0.027	0.818±0.025	0.858±0.025	0.916±0.018	0.956±0.014	0.999±0.003	0.837±0.009
-Prior	0.748±0.028	0.769±0.027	0.809±0.026	0.849±0.023	0.893±0.022	0.935±0.016	0.959±0.014	0.999±0.003	0.864±0.008
-Disambiguated	0.762±0.027	0.793±0.026	0.825±0.025	0.862±0.022	0.899±0.021	0.936±0.016	0.959±0.014	0.999±0.003	0.874±0.008

Table B.4: Accuracy with different document representations on the Wikipedia dataset.

	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0	1.0	Overall
Unambiguous	0.670±0.025	0.599±0.025	0.728±0.022	0.766±0.021	0.806±0.021	0.861±0.018	0.911±0.015	0.997±0.003	0.786±0.008
Prior	0.628±0.026	0.593±0.025	0.714±0.023	0.745±0.022	0.790±0.022	0.879±0.017	0.901±0.016	1.000±0.000	0.775±0.008
Disambiguated	0.610±0.026	0.608±0.025	0.674±0.024	0.731±0.022	0.787±0.022	0.861±0.018	0.898±0.016	0.999±0.002	0.764±0.008
ALL	0.693±0.025	0.671±0.024	0.740±0.022	0.760±0.021	0.792±0.022	0.871±0.017	0.900±0.016	1.000±0.000	0.797±0.007
-Unambiguous	0.644±0.026	0.633±0.025	0.717±0.023	0.745±0.022	0.790±0.022	0.867±0.017	0.896±0.016	0.999±0.002	0.780±0.008
-Prior	0.681±0.025	0.629±0.025	0.706±0.023	0.768±0.021	0.790±0.022	0.857±0.018	0.909±0.015	0.997±0.003	0.786±0.008
-Disambiguated	0.663±0.026	0.639±0.025	0.739±0.022	0.759±0.021	0.794±0.021	0.870±0.017	0.903±0.016	1.000±0.000	0.790±0.008

Table B.5: Accuracy with different document representations on the ClueWeb dataset.

	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0	1.0	Overall
type	0.752±0.028	0.792±0.026	0.810±0.026	0.851±0.023	0.891±0.022	0.939±0.016	0.958±0.014	0.999±0.003	0.868±0.008
100	0.754±0.028	0.791±0.026	0.812±0.026	0.858±0.023	0.893±0.022	0.936±0.016	0.960±0.014	0.999±0.003	0.870±0.008
1000	0.757±0.028	0.806±0.026	0.806±0.026	0.860±0.022	0.888±0.022	0.935±0.016	0.956±0.014	0.999±0.003	0.871±0.008
2000	0.746±0.028	0.811±0.025	0.805±0.026	0.860±0.022	0.894±0.022	0.935±0.016	0.961±0.013	0.999±0.003	0.871±0.008
3000	0.747±0.028	0.800±0.026	0.801±0.026	0.859±0.023	0.891±0.022	0.935±0.016	0.961±0.013	0.999±0.003	0.869±0.008

Table B.6: Accuracy of NED using different landmark strategies on the Wikipedia dataset.

	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0	1.0-1.1	Overall
WNED	0.644±0.031	0.727±0.029	0.784±0.027	0.818±0.025	0.881±0.023	0.902±0.020	0.952±0.015	0.988±0.008	0.830±0.009
L2R	0.709±0.029	0.758±0.028	0.796±0.027	0.839±0.024	0.877±0.023	0.911±0.019	0.936±0.017	0.978±0.011	0.845±0.009
LOCAL	0.514±0.032	0.594±0.032	0.651±0.031	0.696±0.030	0.775±0.029	0.838±0.024	0.944±0.016	0.999±0.003	0.741±0.010
LOCAL+Landmark	0.602±0.032	0.665±0.030	0.700±0.030	0.737±0.029	0.795±0.028	0.866±0.023	0.933±0.017	0.999±0.003	0.778±0.010
LOCAL+PL_NBR	0.599±0.032	0.599±0.032	0.684±0.031	0.735±0.029	0.791±0.029	0.865±0.023	0.941±0.016	0.999±0.003	0.767±0.010
LOCAL+PL_NBR_ALS	0.565±0.032	0.617±0.031	0.668±0.031	0.715±0.029	0.774±0.029	0.847±0.024	0.945±0.016	0.999±0.003	0.756±0.010
LOCAL+CO_NBR	0.603±0.032	0.611±0.031	0.679±0.031	0.718±0.029	0.792±0.029	0.861±0.023	0.940±0.016	0.999±0.003	0.766±0.010
LOCAL+CO_NBR_ALS	0.586±0.032	0.615±0.031	0.668±0.031	0.719±0.029	0.769±0.030	0.860±0.023	0.935±0.017	0.999±0.003	0.759±0.010
LOCAL+WikiDoc	0.559±0.032	0.607±0.031	0.675±0.031	0.719±0.029	0.774±0.030	0.859±0.023	0.936±0.017	0.999±0.003	0.756±0.010
LOCAL+Category	0.584±0.032	0.644±0.031	0.694±0.030	0.748±0.028	0.796±0.028	0.872±0.022	0.943±0.016	0.999±0.003	0.776±0.010
LOCAL+PL_SR	0.595±0.032	0.654±0.031	0.727±0.029	0.779±0.027	0.811±0.028	0.878±0.022	0.944±0.016	0.999±0.003	0.790±0.010
LOCAL+CO_SR	0.743±0.028	0.775±0.027	0.797±0.026	0.844±0.024	0.885±0.022	0.922±0.018	0.953±0.015	0.999±0.003	0.859±0.008
LOCAL+PL_SR+CO_SR	0.733±0.029	0.795±0.026	0.796±0.027	0.832±0.024	0.891±0.022	0.920±0.018	0.955±0.014	0.999±0.003	0.859±0.008
ALL	0.757±0.028	0.806±0.026	0.806±0.026	0.860±0.023	0.888±0.022	0.935±0.016	0.956±0.014	0.999±0.003	0.871±0.008
ALL-Landmark	0.754±0.028	0.803±0.026	0.813±0.026	0.854±0.023	0.897±0.022	0.937±0.016	0.960±0.014	0.999±0.003	0.872±0.008
ALL-PL_NBR	0.756±0.028	0.816±0.025	0.797±0.027	0.853±0.023	0.895±0.022	0.929±0.017	0.956±0.014	0.999±0.003	0.870±0.008
ALL-PL_NBR_ALS	0.769±0.027	0.792±0.026	0.813±0.026	0.859±0.023	0.891±0.022	0.931±0.017	0.955±0.014	0.999±0.003	0.871±0.008
ALL-CO_NBR	0.754±0.028	0.795±0.026	0.815±0.026	0.847±0.023	0.891±0.022	0.932±0.017	0.956±0.014	0.999±0.003	0.868±0.008
ALL-CO_NBR_ALS	0.761±0.027	0.794±0.026	0.809±0.026	0.851±0.023	0.891±0.022	0.935±0.016	0.958±0.014	0.999±0.003	0.869±0.008
ALL-WikiDoc	0.756±0.028	0.807±0.026	0.814±0.026	0.854±0.023	0.890±0.022	0.936±0.016	0.956±0.014	0.999±0.003	0.871±0.008
ALL-Category	0.758±0.028	0.794±0.026	0.806±0.026	0.848±0.023	0.890±0.022	0.936±0.016	0.958±0.014	0.999±0.003	0.868±0.008
ALL-PL_SR	0.771±0.027	0.797±0.026	0.811±0.026	0.855±0.023	0.891±0.022	0.935±0.016	0.958±0.014	0.999±0.003	0.872±0.008
ALL-CO_SR	0.710±0.029	0.741±0.028	0.780±0.027	0.815±0.025	0.851±0.025	0.905±0.019	0.950±0.015	0.999±0.003	0.838±0.009
ALL-PL_SR-CO_SR	0.654±0.031	0.701±0.030	0.756±0.028	0.809±0.026	0.829±0.027	0.892±0.021	0.946±0.016	0.999±0.003	0.816±0.009

Table B.7: Accuracy of NED using features from different semantic signatures of entities on the Wikipedia dataset.

	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0	1.0-1.1	Overall
WNED	0.581±0.027	0.604±0.025	0.672±0.024	0.748±0.022	0.799±0.021	0.864±0.018	0.899±0.016	0.978±0.009	0.762±0.008
L2R	0.657±0.026	0.644±0.025	0.719±0.023	0.751±0.022	0.757±0.023	0.849±0.018	0.879±0.017	0.937±0.014	0.769±0.008
LOCAL	0.615±0.026	0.541±0.025	0.673±0.023	0.719±0.023	0.767±0.022	0.843±0.019	0.904±0.016	0.999±0.002	0.750±0.008
LOCAL+Landmark	0.618±0.026	0.551±0.025	0.699±0.023	0.737±0.022	0.751±0.023	0.841±0.019	0.901±0.016	0.997±0.003	0.755±0.008
LOCAL+PL_NBR	0.630±0.026	0.580±0.025	0.660±0.024	0.721±0.022	0.755±0.023	0.848±0.018	0.897±0.016	0.999±0.002	0.754±0.008
LOCAL+PL_NBR_ALS	0.633±0.026	0.547±0.025	0.681±0.023	0.741±0.022	0.768±0.022	0.859±0.018	0.898±0.016	1.000±0.000	0.758±0.008
LOCAL+CO_NBR	0.594±0.027	0.518±0.026	0.698±0.023	0.740±0.022	0.797±0.021	0.854±0.018	0.898±0.016	0.997±0.003	0.755±0.008
LOCAL+CO_NBR_ALS	0.596±0.026	0.586±0.025	0.678±0.023	0.742±0.022	0.765±0.023	0.854±0.018	0.901±0.016	0.999±0.002	0.758±0.008
LOCAL+WikiDoc	0.623±0.026	0.554±0.025	0.691±0.023	0.721±0.023	0.753±0.023	0.834±0.019	0.896±0.016	1.000±0.000	0.752±0.008
LOCAL+Category	0.671±0.025	0.594±0.025	0.692±0.023	0.716±0.023	0.760±0.023	0.853±0.018	0.898±0.016	0.999±0.002	0.766±0.008
LOCAL+PL_SR	0.636±0.026	0.604±0.025	0.692±0.023	0.741±0.022	0.774±0.022	0.865±0.018	0.907±0.015	0.997±0.003	0.770±0.008
LOCAL+CO_SR	0.663±0.026	0.608±0.025	0.726±0.022	0.744±0.022	0.793±0.021	0.855±0.018	0.907±0.016	1.000±0.000	0.780±0.008
LOCAL+PL_SR+CO_SR	0.659±0.026	0.616±0.025	0.731±0.022	0.736±0.022	0.798±0.021	0.864±0.018	0.904±0.016	1.000±0.000	0.782±0.008
ALL	0.693±0.025	0.671±0.024	0.740±0.022	0.760±0.021	0.792±0.022	0.871±0.017	0.900±0.016	1.000±0.000	0.797±0.007
ALL-Landmark	0.679±0.025	0.652±0.024	0.728±0.022	0.776±0.021	0.798±0.021	0.865±0.017	0.901±0.016	1.000±0.000	0.794±0.008
ALL-PL_NBR	0.674±0.025	0.657±0.024	0.729±0.022	0.761±0.021	0.790±0.021	0.868±0.017	0.906±0.016	1.000±0.000	0.792±0.008
ALL-PL_NBR_ALS	0.695±0.025	0.667±0.024	0.731±0.022	0.768±0.021	0.791±0.022	0.866±0.017	0.901±0.016	0.999±0.002	0.796±0.007
ALL-CO_NBR	0.676±0.025	0.652±0.024	0.727±0.022	0.759±0.021	0.788±0.022	0.869±0.017	0.901±0.016	1.000±0.000	0.790±0.008
ALL-CO_NBR_ALS	0.695±0.025	0.637±0.025	0.739±0.022	0.748±0.022	0.790±0.022	0.873±0.017	0.898±0.016	1.000±0.000	0.791±0.008
ALL-WikiDoc	0.692±0.025	0.654±0.024	0.730±0.022	0.760±0.021	0.788±0.022	0.871±0.017	0.900±0.016	1.000±0.000	0.793±0.008
ALL-Category	0.679±0.025	0.644±0.024	0.728±0.022	0.765±0.021	0.784±0.022	0.873±0.017	0.902±0.016	1.000±0.000	0.791±0.008
ALL-PL_SR	0.678±0.025	0.644±0.024	0.728±0.022	0.759±0.021	0.788±0.022	0.862±0.018	0.898±0.016	1.000±0.000	0.789±0.008
ALL-CO_SR	0.677±0.025	0.644±0.024	0.708±0.023	0.764±0.021	0.787±0.022	0.863±0.018	0.904±0.016	1.000±0.000	0.787±0.008
ALL-PL_SR-CO_SR	0.664±0.026	0.603±0.025	0.698±0.023	0.730±0.022	0.777±0.022	0.858±0.018	0.895±0.016	1.000±0.000	0.771±0.008

Table B.8: Accuracy of NED using features from different semantic signatures of entities on the ClueWeb dataset.

	MSNBC	AQUAINT	ACE2004	AIDA-CoNLL
WNED	0.895±0.024	0.880±0.024	0.833±0.046	0.836±0.004
L2R	0.909±0.022	0.886±0.023	0.852±0.043	0.890±0.004
LOCAL	0.895±0.023	0.861±0.025	0.833±0.046	0.863±0.004
LOCAL+Landmark	0.892±0.024	0.865±0.025	0.844±0.044	0.845±0.004
LOCAL+PL_NBR	0.909±0.022	0.862±0.025	0.844±0.044	0.862±0.004
LOCAL+PL_NBR_ALS	0.899±0.023	0.872±0.024	0.844±0.044	0.859±0.004
LOCAL+CO_NBR	0.907±0.022	0.867±0.025	0.844±0.044	0.866±0.004
LOCAL+CO_NBR_ALS	0.901±0.023	0.865±0.025	0.829±0.046	0.867±0.004
LOCAL+WikiDoc	0.904±0.023	0.861±0.025	0.840±0.045	0.866±0.004
LOCAL+Category	0.912±0.022	0.869±0.024	0.840±0.045	0.868±0.004
LOCAL+PL_SR	0.921±0.021	0.868±0.025	0.848±0.044	0.883±0.004
LOCAL+CO_SR	0.904±0.023	0.882±0.023	0.860±0.042	0.896±0.004
LOCAL+PL_SR+CO_SR	0.904±0.023	0.880±0.024	0.856±0.043	0.897±0.004
ALL	0.910±0.022	0.883±0.023	0.875±0.040	0.897±0.004
ALL-Landmark	0.909±0.022	0.879±0.024	0.864±0.042	0.897±0.004
ALL-PL_NBR	0.909±0.022	0.883±0.023	0.875±0.040	0.897±0.004
ALL-PL_NBR_ALS	0.909±0.022	0.879±0.024	0.879±0.040	0.896±0.004
ALL-CO_NBR	0.904±0.023	0.883±0.023	0.879±0.040	0.897±0.004
ALL-CO_NBR_ALS	0.905±0.022	0.883±0.023	0.875±0.040	0.897±0.004
ALL-WikiDoc	0.907±0.022	0.882±0.024	0.879±0.040	0.897±0.004
ALL-Category	0.904±0.023	0.887±0.023	0.868±0.041	0.897±0.004
ALL-PL_SR	0.909±0.022	0.883±0.023	0.875±0.040	0.897±0.004
ALL-CO_SR	0.909±0.022	0.875±0.024	0.856±0.043	0.892±0.004
ALL-PL_SR-CO_SR	0.886±0.024	0.871±0.024	0.856±0.043	0.885±0.004

Table B.9: Accuracy of NED using features from different semantic signatures of entities on 4 public datasets.

B.3 Evaluation of Efficiency

Datasets	WNED	L2R-CoNLL	NED-SR	NED-ALL
ACE2004	28137	10296	2368	3617
AQUAINT	15593	15409	3039	8655
MSNBC	32541	32613	6011	13853
AIDA/CoNLL-Test A	34632	33937	4481	16352
DBpediaSpotlight	7831	7120	1320	3122
KORE50	8317	2972	693	1409
Microposts2014-Test	2523	1666	435	622
N3-RSS-500	2107	1309	413	556
N3-Reuters-128	12528	4572	1135	1704
OKE 2015 Task1 evaluation dataset	21029	7040	1409	2760

Table B.10: Average running time (milliseconds) per document in each benchmark of our 4 NED systems reported by GERBIL.