GRAPH-BASED APPROACHES TO RESOLVE ENTITY AMBIGUITY

by

Maria Pershina

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

May, 2016

Professor Ralph Grishman

Dedication

To my family.

Acknowledgments

Doing a PhD has been a valuable and exciting experience. I am very thankful to my advisor, Ralph Grishman, for his patience and continuous support throughout my graduate career. Ralph provided me with good advice and guidance, and introduced me to Proteus research group. I had a lot of freedom to explore different research topics and to get my first industrial experience. Feeling his positive attitude and support have always been a great source of inspiration for me. I will be always grateful for his help.

During my PhD I was fortunate to work and co-author with many brilliant people. I was lucky to meet Bonan Min and Wei Xu who inspired and shaped my first paper. I was very happy to work with Yifan He. It was a great pleasure to be a part of Proteus project and meet Thien Huu Ngyuen, Lisheng Fu, Xiang Li, Kai Cao, and Miao Fan. I am thankful to Satoshi Sekine and Adam Meyers who were always giving me a useful feedback during our weekly lunch meetings.

I owe special thanks to Mohamed Yakout and Kaushik Chakrabarti, my mentors at Microsoft Research, who exposed me to the research in industry. Their expertise and support made my internship at MSR in Redmond in 2013 a very productive one.

ACKNOWLEDGMENTS

I am thankful to many faculty and students at New York University who inspired and supported me on the way through my graduate study.

Abstract

Information Extraction is the task of automatically extracting structured information from unstructured or semi-structured machine-readable documents. One of the challenges of Information Extraction is to resolve ambiguity between entities either in a knowledge base or in text documents. There are many variations of this problem and it is known under different names, such as coreference resolution, entity disambiguation, entity linking, entity matching, etc. For example, the task of coreference resolution decides whether two expressions refer to the same entity; entity disambiguation determines how to map an entity mention to an appropriate entity in a knowledge base (KB); the main focus of entity linking is to infer that two entity mentions in a document(s) refer to the same real world entity even if they do not appear in a KB; entity matching (also record deduplication, entity resolution, reference reconciliation) is to merge records from databases if they refer to the same object.

Resolving ambiguity and finding proper matches between entities is an important step for many downstream applications, such as data integration, question answering, relation extraction, etc. The Internet has enabled the creation of a growing number of large-scale knowledge bases in a variety of domains, posing a

ABSTRACT

scalability challenge for Information Extraction systems. Tools for automatically aligning these knowledge bases would make it possible to unify many sources of structured knowledge and to answer complex queries. However the efficient alignment of large-scale knowledge bases still poses a considerable challenge.

Various aspects and different settings to resolve ambiguity between entities are studied in this dissertation. A new scalable domain-independent graph-based approach utilizing Personalized Page Rank is developed for entity matching across large-scale knowledge bases and evaluated on datasets of 110 million and 203 million entities. A new model for entity disambiguation between a document and a knowledge base utilizing a document graph and effectively filtering out noise is proposed; corresponding datasets are released. A competitive result of 91.7% in microaccuracy on a benchmark AIDA dataset is achieved, outperforming the most recent state-of-the-art models. A new technique based on a paraphrase detection model is proposed to recognize name variations for an entity in a document. Corresponding training and test datasets are made publicly available. A new approach integrating a graph-based entity disambiguation model and this technique is presented for an entity linking task and is evaluated on a dataset for the Text Analysis Conference Entity Discovery and Linking 2014 task.

Table of contents

Dedication **iii**

Acknowledgments **iv**

Abstract **vi**

List of Figures **xii**

List of Tables **xiii**

1 Introduction **1**

1.1 Challenges & Contributions 5

1.2 Overview 7

2 Related Work **9**

3 Entity Matching **14**

3.1 Introduction 14

3.2 Pairs Graph Construction 17

3.2.1 Seed Pairs Generation 19

3.2.2 Expanding the Pairs Graph 19
3.3 Random Surfer Model 22
3.3.1 Personalized PageRank 22
3.3.2 Holistic Similarity In a Knowledge Graph 22
3.3.3 Optimization 23
3.4 Pipeline 26
3.5 Experiments 26
3.5.1 Data 26
3.5.2 Graph construction and score propagation 26
3.5.3 Models and Evaluation 27
3.5.4 Results 28
3.6 Conclusion 29

4 Entity Disambiguation **32**
4.1 Introduction 32
4.2 Related Work 35
4.3 The Graph Model 36
4.3.1 Vertices 37
4.3.2 Edges 38
4.4 The Challenge 39
4.4.1 Personalized PageRank 40
4.4.2 Coherence and Constraints 40
4.4.3 PPRSim 42
4.5 Experiments and Results 44

4.5.1 Data 44
4.5.2 Evaluation 45
4.5.3 PPR 45
4.5.4 Baselines 45
4.5.5 Results 46
4.6 Conclusion 46

5 Entity Linking **48**
5.1 Introduction 48
5.2 Related Work 51
5.3 Document Graph 52
5.3.1 Candidates 52
5.3.2 Edges 53
5.3.3 Initial Similarity 53
5.4 Name Variations as Paraphrases 54
5.5 ParaLink 57
5.6 Experiments and Results. 59
5.6.1 Data 59
5.6.2 Evaluation 61
5.6.3 Baselines 62
5.6.4 Results 62
5.6.5 Discussion 63
5.7 Conclusion 63

TABLE OF CONTENTS

6 Future Work **65**

Bibliography **67**

List of Figures

3.1 Toy knowledge databases represented as a) knowledge graphs; b) *(s,p,o)* triples; c) graph of entity pairs. 18
3.2 Algorithm 1 for graph expansion. 30
3.3 Pipeline for HolisticEM framework. 31

4.1 A toy document graph for three entity mentions: *United F.C., Lincolnshire, Devon White*. Candidates and their initial similarity scores are generated for each entity mention. 37

5.1 Performance of ASOBEK model with different feature sets applied to name variation task. 56
5.2 Examples of ParaLink refining and clustering steps I, II, III. 57
5.3 ParaLink diagram with refining and clustering steps I, II, III. 60

List of Tables

1.1 Main characteristics of Entity Matching, Disambiguation, and Linking tasks. 4

3.1 Freebase and IMDB datasets statistics. 27

3.2 Precision, Recall and F-score for (1) Seed Pairs; (2) pairs in Pairs Graph with Simple and GraphBased initial scores; (3) pairs in Pairs Graph with PPR-propagated Simple (HolisticEM+S) and GraphBased (HolisticEM+GB) scores. 28

4.1 Performance of PPRSim compared to baselines and state-of-the-art models on AIDA dataset. Baselines iSim and PPR choose a candidate with the highest initial similarity or coherence correspondingly. 43

4.2 Performance of PPRSim with different initial similarities and constraints. 46

List of Tables

5.1 Performance of ParaLink in $\mathbf{B}^3$+**F** score compared to the baseline and state-of-the-art models on TAC EDL 2014 train/test datasets. NYU (PR): PageRank with one-name-per-cluster name clustering; PPRSim: Personalized PageRank as described in (Pershina et al., 2015c); PPRSim+I/II/III: Combining PPRSim separately with steps in ParaLink; ParaLink: PPRSim with all steps I,II,III; ParaLink*: ParaLink scored on manually corrected TAC answer key. 62

Chapter 1

Introduction

Traditionally, Information Extraction is associated with extraction of event information from natural language text. This was a popular task of the Message Understanding Conferences (MUC) in the late eighties and nineties (Sundheim, 1992). MUC was the first large scale effort to boost research into automatic information extraction and it would define the research field for the decades to come. According to (Bunescu and Paşca, 2006) Information Extraction involves the processing of natural language text to produce structured knowledge, suitable for storage in a database for later retrieval or automated reasoning. Cowie and Lehnert, 1996, Cowie and Lehnert (1996) see information extraction as a process that involves the extraction of fragments of information from natural language texts and linking of these fragments into a coherent framework. They define the goal of information extraction as "to build systems that find and link relevant information while ignoring extraneous and irrelevant information".

Text is not the only source of information. The Internet has enabled the cre-

ation of a growing number of large-scale knowledge bases in a variety of domains, containing complementary information. Tools for automatically aligning these knowledge bases would make it possible to unify many sources of structured knowledge and to answer complex queries. However, the efficient alignment of large-scale knowledge bases still poses a substantial challenge and is the core problem of the entity matching task.

The document-centric view of information extraction has received considerable attention. However, the end result, a group of entities and relations, often are not the only structured knowledge product. In a development environment, new extractions must be merged with previously extracted information, often stored in a structured information database, a knowledge base (KB). This last step is critical for automatic knowledge base population, which requires linking mentions in text to entries in a KB, determining information duplication between the text and KB, exploiting existing knowledge in improving information extraction, and detecting when to create new entries in the knowledge base. These challenges are addressed by entity disambiguation and entity linking tasks.

To the discerning human eye, the "Bush" in "Mr. Bush left for the Zurich environment summit in Air Force One." is clearly the US president. Further context may reveal him to be the 43rd president, George W. Bush, and not the 41st president, George H. W. Bush. The ability to disambiguate a polysemous entity mention or infer that two orthographically different mentions are the same entity mention is crucial in updating an entity's KB record. This task has been variously called entity disambiguation, record linkage, or entity linking. When per-

formed without a KB, entity disambiguation reduces to the traditional document coreference resolution problem, in which entity mentions, either within the same document or across multiple documents, are clustered together, where hopefully each cluster corresponds to a single real world entity. The emergence of large scale publicly available KBs like Wikipedia and DBpedia has spurred an interest in linking textual entity references to their entries in these public KBs. Bunescu and Pasca (2006) and Cucerzan (2007) presented important pioneering work in this area, but suffer from several limitations including Wikipedia specific dependencies, scale, and the assumption of a KB entry for each entity.

In the last decade, a growing number of large-scale knowledge bases have been created online. Domains include music, movies, publications and biological data[1]. As these knowledge bases sometimes contain both overlapping and complementary information, there has been growing interest in attempting to merge them by aligning their common elements. This alignment could have important uses for information retrieval and question answering. For example, one could be interested in finding a scientist with expertise on certain related protein functions - information which could be obtained by aligning a biological database with a publication one. This task is known as entity matching (also referred to as duplicate identification, record linkage, entity resolution, reference reconciliation, etc.) and is challenging to automate as different knowledge bases generally use different terms to represent their entities, and the space of possible matchings grows exponentially with the number of entities. Entity matching is a crucial step for data integration and data cleaning problems (Cohen et al., 1999; Hernandez and Stolfo,

1. Such as MusicBrainz, IMDb, DBLP, UnitProt.

1995; Rahm and Do, 2000).

We summarize the similarities and differences of three tasks in Table 1.1.

Task	Entity		
	Matching	Disambiguation	Linking
Data	structured: one or more databases	unstructured: text documents	
Knowledge Base	no	yes	yes (optional)
Goal	identify duplicate entries	map entity mentions to KB entries	cluster corefered entities
Scale	100-200M entities per database	1-200 entity mentions, 1-20k candidates per document	
Evaluation	precision/recall/ f-score	microaccuracy (precision@1) macroaccuracy	clustering metrics, e.g. B^3+F
Challenges	scalability	a noisy document graph	
		balance local similarity vs relatedness	
			NIL clustering

Table 1.1: Main characteristics of Entity Matching, Disambiguation, and Linking tasks.

Here we study various aspects and different settings to resolve ambiguity between data entries. We present models to address a problem of large-scale entity matching across knowledge graphs, and to solve tasks of entity disambiguation and entity linking between text documents and a knowledge base. We design a graph representation for every setting. Graph edges represent relations between different nodes. Our ultimate goal is to quantify the importance of one node to another. Personalized Page Rank provides a natural measure of the relatedness

between nodes.

The PageRank algorithm (Brin and Page, 1998; Page et al., 1999) considers random walk on a graph, where at each step with probability ϵ (teleport probability) we jump to a randomly selected node on a graph, and with probability $1-\epsilon$ we follow a random outgoing edge of the current node. Stationary distribution of this walk gives PageRank weights associated with each node. Personalized PageRank (PPR) is the same as PageRank, except that all teleports are made to the same source node, for which we are personalizing the PageRank. Intuitively, pairwise weights $PPR(s \rightarrow e)$ represent relationships between nodes in the graph: the higher the weight is, the more relevant endpoint e is for the source s. Thus PPR naturally measures the importance of e for s (Brin and Page, 1998). Our graph-based approaches utilizing PPR do not require training and perform competitively on benchmark datasets.

1.1 Challenges & Contributions

We address different challenges for entity matching, disambiguation, and linking problems. Our models have the following properties:

For all problems

- We start from a graph representation of the problem;
- Our models are based on a random walk algorithm, they do not require training;

- Our models benefit from both relational information between entities in a knowledge graph and from the local information for the node in question;
- Score propagation scheme is scalable for large graphs; it can be efficiently implemented on MapReduce (Sections 3.2.2, 3.3.3);

Entity Matching

- Generic: domain independent, robust to incomplete data, applicable to one or more datasets;
- Does not propagate errors by doing *simultaneous* resolution for all nodes (Section 3.3.2);
- Experiments on Microsoft Knowledge Graphs validate the effectiveness and scalability of our approach by accurately resolving 1.6M matching pairs;

Entity Disambiguation

- Our method is able to better utilize the local similarity between a candidate and a KB node, unlike previous PageRank based approaches in Named Entity Disambiguation (Alhelbawy and Gaizauskas, 2014) which mainly rely on global coherence;
- We tailor the Personalized PageRank algorithm to only focus on one high-confidence entity at a time to reduce the impact of noisy candidates (Section 4.4.3);
- Our model achieves a precision of 91.7% on a benchmark AIDA dataset;

Entity Linking

- We adopt the paraphrase model to measure the similarity between entity mention strings and address the problem of name variations;
- We show how to apply this model for NIL-clustering and efficiently combine it with a graph-based entity disambiguation technique to improve Entity Linking (Section 5.5);
- We achieve the competitive result of 80.5% in B^3+F score on a dataset for the diagnostic Text Analysis Conference Entity Discovery and Linking 2014 task.

1.2 Overview

The rest of this thesis is organized as follows.

The review of prior work in related areas is in Chapter 2.

In Chapter 3 we discuss the problem of entity matching across knowledge graphs. We review prior work in this area and discuss challenges that were not addressed by previous approaches. This is followed by a graph representation of the problem and a motivation for using Personalized Page Rank on this graph. The precision and recall results are presented for databases of 110M and 203M entities.

Chapter 4 is devoted to the problem of entity disambiguation for text documents. We discuss a procedure for building graph representation of a document. We then devise an algorithm that efficiently combines an initial similarity of a can-

didate and its relatedness to the document, and filters out noise from a document graph.

The problem of entity linking and the related problem of name variations are discussed in Chapter 5. We draw analogies between name variations and paraphrase identification problems. We then show how to integrate two state-of-the-art models — for entity disambiguation and paraphrase identification — into a new approach for the entity linking task.

We conclude the thesis and discuss future work in Chapter 6.

Chapter 2

Related Work

Information extraction is concerned with both identifying structured information in text and disambiguating extracted information and entities. The ambiguity of entity names, especially in large corpora like the Web or citations in scholarly articles, has served to motivate research on entity resolution. To address ambiguity in personal name search, (Mann and Yarovskiy, 2003) disambiguates person names using biographic facts, like birth year, occupation and affiliation. When present in a text, biographic facts extracted using regular expressions help disambiguation. More recently, the Web People Search Task clustered web pages for entity disambiguation (Artiles et al., 2008).

The related task of cross-document coreference resolution has been addressed by several researchers starting from (Bagga and Baldwin, 1998). (Poesio et al., 2007) built a cross-document coreference system using features from encyclopedic sources like Wikipedia. This continues to be a popular task (Huang et al., 2010; Popescu, 2010). Entity linking has been scaled to consider hundreds of thousands of unique

entities, whereas operating on this scale is a challenge for cross-document coreference resolution. Recent approaches to scaling this task have included distributed graphical models over a compute cluster (Singh et al., 2011) and a streaming coreference algorithm (Rao et al., 2010). Successful coreference resolution is insufficient for correct entity linking, as the coreference chain must still be correctly mapped to the proper KB entry.

By comparison, research in entity disambiguation began only recently. The earliest work done by (Bunescu and Paşca, 2006) and (Cucerzan, 2007) aims to link entity mentions to their corresponding topic pages in Wikipedia. These authors do not use the term entity disambiguation and they take different approaches. Cucerzan uses heuristic rules and Wikipedia disambiguation markup to derive mappings from surface forms of entities to their Wikipedia entries. For each entity in Wikipedia, a context vector is derived as a prototype for the entity and these vectors are compared (via dot-product) with the context vectors of unknown entity mentions. His work assumes that all entities have a corresponding Wikipedia entry, but this assumption fails for a significant number of entities in news articles and even more for other genres, like forums and blogs. Bunescu and Pasca (2006), on the other hand, suggest a simple method to handle entities not in Wikipedia by learning a threshold to decide if the entity is not in Wikipedia. Both works mentioned rely on Wikipedia specific annotations, such as category, hierarchies and disambiguation links. (Milne and Witten, 2008) use machine learning to identify significant terms within unstructured text and built their system on (Cucerzan, 2007).

Several different techniques for entity disambiguation have been used in recent work. Han and Sun (2011) combine different forms of disambiguation knowledge evidence from mention-entity associations and entity popularity in the KB, and context similarity. Ratinov et al. (2011) use a mixture of local and global features to train the coefficients of a linear ranking SVM to rank different NE candidates. Shirakawa et al. (2011) cluster related textual mentions and assign a concept to each cluster using probabilistic taxonomy. Han et al. (2011) use local dependency between NE mention and the candidate entity, and semantic relatedness between candidate entities to construct a referent graph, proposing a collective inference algorithm to infer the correct reference node in the graph. Hoffart et al. (2011) pose the problem as one of finding a dense sub-graph, which is infeasible in a huge graph.

The entity linking problem aims to cluster together entities that refer to the same real world object. It is often done by aligning entities in a document with a corresponding entries in a knowledge base. However a substantial challenge is presented by entities that do not appear in the KB, called NIL entities, e.g. people names on forum data. These names are often ambiguous, misspelled, or incomplete and should be handled differently. Since the Text Analytics Conference on Knowledge Base Population (TAC-KBP) included the task of entity linking (McNamee et al., 2009), the task has grown in popularity with many different approaches (Ji and Grishman, 2011; Zhang et al., 2010). Examples include the use of information retrieval techniques, such as query expansion (Gottipati and Jiang, 2011), for retrieving the correct KB entry; generative clustering models for

entities in text based on KB entries (Han and Sun, 2011); and graph partitioning, Markov-Chain Monte Carlo and centroid models to obtain optimal clustering for both linked and unlinked (NIL) entities (Monahan et al., 2014).

The entity matching problem was originally defined in 1959 by (Newcombe et al., 1959) and was formalized by (Fellegi and Sunter, 1969) 10 years later. Since then it has been considered under various facets and from different communities, including the AI research community, the DB research community, and industry. Numerous approaches have been proposed for entity matching especially for structured data. Due to the large variety of data sources and entities to match there is no single "best" match algorithm. A single match approach typically performs very differently for different domains and match problems. For example, it has been shown that there is no universally best string similarity measure (Guha et al., 2004; Sarawagi and Kirpal, 2004). Instead it is often beneficial and necessary to combine several methods for improved matching quality, e.g. to consider the similarity of several attributes or to take into account relationships between entities. For large datasets it is popular to apply blocking strategies to reduce the search space for entity matching and achieve sufficiently fast execution times. There are several entity matching frameworks that have recently been developed which support multiple approaches for blocking and matching as well as their combination (Baxter et al., 2003; Bilenko et al., 2006; Kenig and Gal, 2013; Michelson and Knoblock, 2006; Vries et al., 2009; Whang et al., 2009).

A significant amount of research has been done in this area — particularly under the umbrella term of ontology matching (Choi et al., 2006; Kalfoglou and

Schorlemmer, 2003; Shvaiko and Euzenat, 2013; Suchanek et al., 2011). An ontology is a formal collection of world knowledge and can take different structured representation. Despite the large body of literature in this area, most of the work on ontology matching has been demonstrated only on fairly small datasets of the order of a few hundred entities. In particular, (Shvaiko and Euzenat, 2013) identified large-scale evaluation as one of the main challenges for the field of ontology matching.

Entities to be resolved may reside in distributed, typically heterogeneous data sources or in a single data source, e.g. in a database or a search engine store. They may be physically materialized or dynamically requested from sources, e.g. by database queries or keyword searches (Chiang et al., 2014; Papadakis et al., 2013).

The conceptual unifying property of entity matching, disambiguation and linking tasks is their goal — to identify data points (database entries, entities, etc) that refer to the same real world object. We show how to represent each of these problems as a graph and how to use Personalized Page Rank on this graph to find correct matches.

Chapter 3

Entity Matching

3.1 Introduction

A common prerequisite for knowledge discovery is accurately combining data from multiple, heterogeneous sources into a unified, mineable knowledge graph. An important step in creating such a graph is entity matching. Entity matching is the problem of determining if two entities in a data set refer to the same real-world object. It is a complex and ubiquitous problem, that appears in numerous application domains including information extraction, data integration, and language processing.[1]

As an example consider the two toy knowledge graphs in Figure 3.1. Nodes in these graphs are actors, movies, characters, performance entities, and their attributes. Edges between nodes correspond to Resource Description Framework

1. This chapter is a revised version of (Pershina et al., 2015a) and patent application (Yakout et al., 2014).

(RDF) triples (s, p, o) and are annotated with predicates p. RDF is a simple yet very powerful triple-based representation for semantic web. It was defined by World Wide Web Consortium (W3C) in 1998 (Lassila and Swick, 1998). For example, triple (p1, played_by, a1) represents an edge between performance p1 and actor a1, meaning that performance p1 was played by an actor a1. The goal is to learn that actors a2 and a4 are the same (*Tom Cruise*), movies m1 and m2 are likely to be different (*Mission Impossible* vs *Mission Impossible III*), as well as actors a1 and a3 (*Douglas* played *Kiev Room Agent* in *Mission Impossible* vs *Douglas* played *IMF Agent* in *Mission Impossible III*).

A growing body of work has shown that incorporating global information can improve the entity matching performance. For example, global information is employed in simultaneous coreference in (Singla and Domingos, 2006), jointly modeled record and field coreference in (Culotta and McCallum, 2005), dirichlet process for modeling interactions between dataset entities in (Bhattacharya and Getoor, 2006; Hall et al., 2008), distribution of wrong entries in input datasets for data fusion in (Dong et al., 2014), probabilistic ontology alignment in (Suchanek et al., 2011). The main limitations of the above techniques are requirements for prior domain knowledge for modeling, data for training, and/or probabilistic inference, which makes these methods computationally infeasible for large data sets.

Missing or incomplete information in the database is another challenge in the entity matching process, e.g. actors a1 and a3 in Figure 3.1 have the same name "Douglas" but correspond to different people - actors Sam Douglas and Douglas Price. Only the dissimilarity of neighboring pairs can help to properly resolve

this pair, e.g. different names for characters (c1,c3) should impact the score for (a1, a3). The intuition behind our algorithm is that matching nodes should have similar nodes in their neighborhood, thus our approach ranks pair (a2,a4) higher than pair (a1,a3) in Figure 3.1.

Greedy iterative approaches (Bohm et al., 2012; Dong et al., 2005; Guo and Barbosa, 2014; Lacoste-Julien et al., 2013; Whang et al., 2009) process nodes sequentially, e.g. by using a priority queue: the highest scored node is resolved as a match, triggering updates for other nodes, and so on. The drawback of this process is the propagation of erroneous decisions, accepted earlier. For example, different actors (a1,a3) would be mistakenly resolved as a match by a greedy approach, since they have exactly the same attributes. This decision would later boost the similarity score between movies (m1,m2) that are clearly different.

In the era of big data, the sources to merge may comprise millions of nodes of tens of different types and will require scalable techniques to resolve matches. There are many approaches, such as blocking, clustering, bootstrapping (Bhattacharya and Getoor, 2007; Cohen and Richman, 2002; McCallum et al., 2000; Rastogi et al., 2011; Whang et al., 2009), constrained deduplication (Arasu et al., 2009), duplicate detection (Herschel and Naumann, 2008), proposed to avoid the quadratic number of comparisons between all pairs of entities to make it scalable. Many entity matching techniques strive for scalability and implicitly use the graph of potential matching candidate pairs to propagate similarity scores. In this paper we present a new method to construct such a graph.

There are primary and relationship entities in the knowledge graphs. Primary

entities can have non-reference attributes, such as *name*, etc (e.g. actor a2). Relationship entities serve to connect primary entities and to describe this connection, e.g. performance p1 in Figure 3.1ab shows that actor a1 played character c1 in movie m1. Relationship entities do not have any non-reference attributes, making it very difficult to match them. Our approach achieves a high F-score of 98% when resolving relationship entity matches.

The main properties of our entity matching framework can be summarized as follows:

- Generic: domain independent, robust to incomplete data, accurate for primary and relationship entities;
- Scalable: both graph construction and score propagation scheme are scalable for large datasets;
- Efficient: does not require training, probabilistic inference, intermediate local models; avoids propagating errors by doing *simultaneous* resolution for all nodes.

3.2 Pairs Graph Construction

Constructing the pairs graph with all possible pairs of entities is unnecessary and often not feasible when we integrate very large knowledge graphs. However, we want the pairs graph to have at least all the matching entities. A pair of entities is a potential match if: (1) their attribute values overlap (or their corresponding

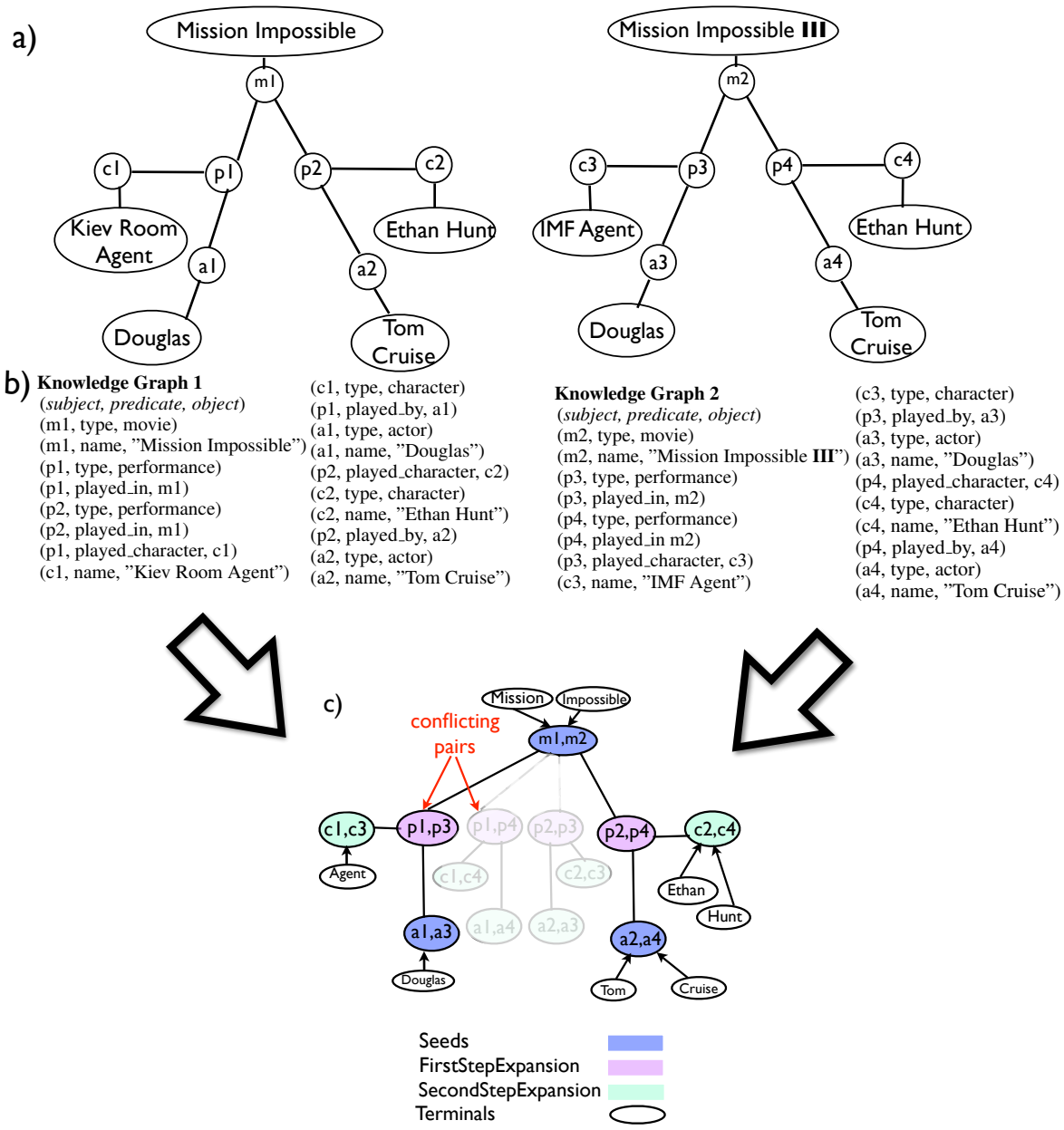

Figure 3.1: Toy knowledge databases represented as a) knowledge graphs; b) *(s,p,o)* triples; c) graph of entity pairs.

bag-of-word encodings overlap); and/or (2) they are connected to entities in their corresponding graphs that are likely to match.

First, we construct potentially matching Seed Pairs based on direct attributes similarity. Second, we expand Seed Pairs, using their corresponding connected entities, and add the necessary new entity pairs and edges to the graph.

3.2.1 Seed Pairs Generation

In the generation of Seed Pairs we rely only on the attribute values of the entities, encoded into a bag of words. We compute IDF (or Inverse Document Frequency) scores for words with respect to the source graph and calculate cosine similarity between entities. First, we organize encoded entities e for each input graph into a schema $\langle Word, e, idf \rangle$. Then we compute IDF scores for individual words $idf(w)$ with respect to the source graph, and finally obtain initial similarity for the pair:

$$\langle\ e_1,\ e_2,\quad sim(\langle e_1, e_2\rangle) = \frac{1}{||e_1|| \cdot ||e_2||} \sum_{w \in e_1 \cap e_2} idf_1(w) \times idf_2(w)\ \ \rangle \tag{3.1}$$

In practice, we generate Seed Pairs separately for each entity type and do additional optimization and pruning, e.g. we discard words with very low IDF score (stop words), and keep only the top $k = 100$ potential matches per entity.

3.2.2 Expanding the Pairs Graph

Given two knowledge graphs our goal is to identify pairs of entities that match. We need to build graph of pairs that captures the influence of each pair of enti-

ties on the neighborhood entities. Thus, the similarity of a pair of entities will contribute to the similarity of its neighborhood pairs of entities. For example, the similarity of a pair of actors contribute to the similarity of their corresponding pairs of performances and movies. The final similarity score of a pair of entities depends on both the similarity of their primitive attribute values and the similarity of their connected entities. The graph has two types of nodes: (i) a node that represents a pair of entities; and (ii) a node that represents a word. An edge between nodes, that are pairs of entities, represents the dependency between their corresponding similarities. For example, the similarity of two performances depends on the similarity of their corresponding movie and actor entity pairs. An edge between a pair node and a word node means that the word is shared between the pair of entities. For example, the word "Tom" is shared between the pair of actors (a2,a4).

Given only seed pairs we need to generate additional pairs and edges to add connectivity to the graph and to include relationship entities, that do not have any atomic attribute values to be considered at the seed generation step. We do two-step expansion for seed pairs $(a_s, b_s) \in$ Seed Pairs, generated for graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$.

First Step. For triples $(a_s, p, a) \in E_1$ and $(b_s, p, b) \in E_2$ we add an entity pair node (a, b), if both a and b are entities of the same type. Then we add an undirected edge connecting it to (a_s, b_s). Similar step is performed for triples $(a, p, a_s) \in E_1$ and $(b, p, b_s) \in E_2$. For example, we add a new pair (p1,p3) and an edge connecting it to the seed (m1,m2) (Figure 3.1c).

Second Step. At this step we expand relationship entities (a, b) that were added

at the first step. Thus, we expand a pair of performances (p1,p3) by adding a pair of characters (c1,c3) and a corresponding edge between these two pairs. This two-step process is illustrated in Figure 3.1c. There can be slightly different definitions of relationship entities. We use entities that do not have *name* attribute.

There are many conflicting pairs produced during the expansion process, e.g. a pair of movies (m1,m2) generates four performance pairs (p1,p3), (p1,p4), (p2,p3), (p2,p4). We assume, that there are no duplicates within the original knowledge graphs and thus pairs (p1,p3) and (p1,p4) cannot coexist together with respect to their parent (m1,m2). To resolve conflicts between nodes we use local information, such as shared terminal attributes, and *stable marriage* heuristics (Gale and Shapley, 1962) to keep only relevant pairs. For example, performance node p1 can "marry" either p3 or p4, and the same is true about node p2. Node p1 would "prefer" p3 since their immediate neighbor pairs (c1,c3) and (a1,a3) have some shared terminals. Similarly, p2 would "prefer" p4 over p3. As a result, the stable marriage algorithm maps p1 to p3 and p2 to p4. Thus nodes (p1,p4) and (p2,p3) and their further extensions, (c1,c4), (a1,a4), (c2,c3), (a2,a3), are removed from the graph (shaded nodes in Figure 3.1c).

Terminals. Lastly, we add shared terminals to all nodes. Thus, string *Douglas* is attached to the pair (a1,a3), strings *Tom* and *Cruise* are attached to the pair (a2,a4), etc.

The individual seed expansion step is independent from other seeds expansion, so it is parallelizable and is done efficiently with MapReduce. The graph construction process is summarized in Algorithm 3.2.

3.3 Random Surfer Model

The graph of pairs captures the influence of each pair of entities on the neighborhood entities (Figure 3.1). We need a holistic approach to quantify the influence of terminal value similarity on entity pair similarity and the influence of one entity pair on another. Personalized PageRank provides a natural way to measure this influence.

3.3.1 Personalized PageRank

Consider a directed weighted graph $G(V, E)$. Its edges represent relations between nodes. PageRank is the stationary distribution of a random walk on G, where at each step with probability ϵ (teleport probability) we jump to a randomly selected node on a graph, and with probability $1 - \epsilon$ we follow a random outgoing edge from the current node. Personalized PageRank (PPR) is the same as PageRank but all teleports are made to the same source node, for which we personalize the PageRank. Thus, for every source node v and landing node u there is an associated PPR weight denoted as $PPR(v \to u)$.

3.3.2 Holistic Similarity In a Knowledge Graph

The contribution of each word to the similarity of a pair of entities should consider the word's popularity. Popular words across entities will less likely influence the matching decision. Therefore each word will have a weight. The well established measure for this purpose is the IDF which is well known in the Information

Retrieval area. The lower the IDF score of a word is, the more entities this word is shared between. Hence, the less distinguishing such a word is to its entities.

The contribution of a word w to the similarity of a pair p is conversely proportion to the degree of the node w in the graph (i.e. the number of node pairs connected to node w). This is similar to the popularity notion, which the IDF weight is trying to capture.

Another way to explain a word's contributions to the pairs similarity is by considering a random surfer, walking in the graph. Let us consider a random surfer that continuously starts its trip at the word node w. Then the landing probability of the surfer at node $\langle e_1, e_2 \rangle$ is essentially the amount of contribution of word w to the similarity of pair $\langle e_1, e_2 \rangle$. Summing these contributions over all words w we obtain

$$sim(\langle e_1, e_2 \rangle) = \sum_{\forall w} PPR(w \rightarrow \langle e_1, e_2 \rangle) \tag{3.2}$$

The above reasoning about the contribution of the words to entities similarity using a random surfer can be extended further to the contribution of entity pairs towards entity pairs.

3.3.3 Optimization

Equation (3.2) requires computing PPR weights for all primitive words w in the graph to calculate $sim(\langle e_1, e_2 \rangle)$. Thus, to compute the contribution of a pair of actors (a2,a4) from Figure 3.1c) to the similarity of a pair of movies (m1,m2) we would have to start a random surfer at each shared primitive value for (a2,a4): names *Tom* and *Cruise*. One can optimize this by computing summary of words at

the pair level as pair's initial similarity, and then propagate it using PPR weights. We use two strategies to compute this initial similarity.

Simple Initial Similarity. Pairs Graph construction filters out irrelevant pairs, entities, and their attributes. It induces two subgraphs - one in each source. These are the subgraphs that were used to build Pairs Graph. We can compute initial similarity scores $iSim(\langle e_1, e_2\rangle)$ for every pair node $\langle e_1, e_2\rangle$ with respect to these two subgraphs in a similar fashion as in Section 3.2.1, Equation (3.1): first, compute $idf(w)$ for words w with respect to the source subgraphs, then calculate cosine similarity between entities in each pair to produce $iSim(\langle e_1, e_2\rangle)$.

GraphBased Initial Similarity. Let us denote as V_w all pairs of entities in the graph $G(V, E)$ that share primitive value w. These pairs are immediate neighbors of w, so $|V_w| = degree(w)$. Random surfer, started at node w, is either teleported with probability ϵ or makes one step in a random direction with probability $1 - \epsilon$. There are $|degree(w)|$ possible directions, so after one iteration random surfer lands at any of the nodes $v \in V_w$ with probability $P(v) = \frac{1-\epsilon}{degree(w)}$, and then the process resumes. Thus

$$\begin{aligned} PPR(w \to \langle e_1, e_2\rangle) &= \sum_{v \in V_w} P(v) \cdot PPR(v \to \langle e_1, e_2\rangle) \\ &\sim \sum_{v \in V_w} \frac{1}{deg(w)} \cdot PPR(v \to \langle e_1, e_2\rangle) \end{aligned}$$

Then similarity of a pair $\langle e_1, e_2\rangle$ from Equation (3.2) can be rewritten as following

$$\begin{aligned} sim(\langle e_1, e_2\rangle) &= \sum_w PPR(w \to \langle e_1, e_2\rangle) \\ &\sim \sum_w \frac{1}{deg(w)} \sum_{v \in V_w} PPR(v \to \langle e_1, e_2\rangle) \end{aligned}$$

Every vertex $v = \langle e'_1, e'_2 \rangle$ appears in this sum as many times as many primitive values w pair $\langle e'_1, e'_2 \rangle$ has. Let us denote this set of shared primitive values for vertex v as W_v. Changing the order of summation and combining terms for the same vertices we get

$$\begin{aligned} sim(\langle e_1, e_2 \rangle) \sim & \sum_v PPR(v \to \langle e_1, e_2 \rangle) \sum_{w \in W_v} \frac{1}{deg(w)} \\ = & \sum_v PPR(v \to \langle e_1, e_2 \rangle) \cdot iSim(v), \\ \text{where } iSim(v) = & \sum_{w \in W_v} \frac{1}{deg(w)} \end{aligned} \tag{3.3}$$

denotes initial similarity of node v and is equal to the sum of degree reciprocals of primitive values for node v.

This computation justifies the idf logic, described in Section **??**. Namely, the contribution of a word $w \in W_v$ into initial similarity of a pair v is conversely proportion to the degree of the node w in the graph (i.e. the number of reference pairs sharing word w).

The optimization step (3.3) subsumes all shared primitive values $w \in W_v$ of every pair v into initial similarity $iSim(v)$.

These strategies for summarizing primitive values allow us to remove all terminal nodes from the pairs graph and thus drastically reduce its size. Moreover, this step significantly simplifies further computation by reducing starting points for random surfer to only non-terminal (reference) nodes:

$$sim(\langle e_1, e_2 \rangle) = \sum_{\substack{v \in non\text{–}terminal \\ nodes}} PPR(v \to \langle e_1, e_2 \rangle) \cdot iSim(v) \tag{3.4}$$

3.4 Pipeline

Our pipeline for Holistic Entity Matching is in Figure 3.3. Its input consists of two knowledge graphs. The goal is to identify duplicate entries in these graphs. There are following steps in this pipeline: (1) generate Seeds; (2) construct Pairs Graph; (3) compute initial similarity for pairs in the Pairs Graph; (4) propagate initial scores via PPR; (5) resolve final scores.

3.5 Experiments

3.5.1 Data

For our experiments we use two datasets: Freebase[2] and an IMDB dataset from an internal data warehouse[3]. Given two sources with more than 110M and 203M entities correspondingly (Table 3.1), we focus on several entity types of interest to avoid space and time limitations. Namely, we pick actors and movies, as they are closely related and thus may benefit from each other. During graph construction step these entities will introduce new nodes, such as performance pairs, allowing us to validate our technique for relationship entities.

3.5.2 Graph construction and score propagation

We generated 5M seeds, 3.2M actor pairs and 1.8M movie pairs, and built graph of pairs as described in Section 3.2. We adopt the Monte Carlo approach

2. www.freebase.com
3. Internal Microsoft IMDB dataset is obtained from imdb pages.

Dataset	Freebase	IMDB
Total Entities	110.6M	203.9M
Actors	425.9K	2.7M
Movies	240.8K	2.5M
Performances	1.2M	5.8M

Table 3.1: Freebase and IMDB datasets statistics.

(Fogaras and Balazs, 2004) for computing Personalized PageRank. It performs a number of independent random walks for every source node and takes an empirical distribution of ending nodes to obtain PPR weights with respect to the source. We initialized 4,000 random walks for every source node, performed 5 steps of PPR at each node with teleport probability $\epsilon = 0.2$, and computed final scores according to (3.4).

3.5.3 Models and Evaluation

Seed scores from Section 3.2.1, and initial scores from Section 3.3.2, are our two baselines; we compare their precision and recall with HolisticEM in Table 3.2. In addition to the ground truth matches, available from internal data warehouse, that covers about 95% of all matches between Freebase and IMDB datasets, we perform additional manual evaluation of uniformly sampled 1000 unmatched entities for each type. We use thresholds obtained on a development dataset for every entity type to resolve HolisticEM scores.

Models	Movies P/R/F	Actors P/R/F	Performances P/R/F
Seeds	68.2 / 82.4 74.6	66.4 / 79.1 72.2	N/A
Simple	92.0 / 76.1 83.3	83.2 / 78.2 80.6	N/A
GraphBased	88.2 / 86.4 87.3	86.9 / 83.2 85.0	N/A
Holistic+Simple	93.1 / 92.5 92.8	95.8 / 93.5 94.7	93.1 / 95.9 94.4
Holistic+GraphBased	99.3 / 96.9 **98.1**	98.9 / 97.9 **98.4**	98.6 / 97.4 **98.0**

Table 3.2: Precision, Recall and F-score for (1) Seed Pairs; (2) pairs in Pairs Graph with Simple and GraphBased initial scores; (3) pairs in Pairs Graph with PPR-propagated Simple (HolisticEM+S) and GraphBased (HolisticEM+GB) scores.

3.5.4 Results

Holistic Entity Matching improves F-score over Seeds and both initial scores - Simple and GraphBased. It is interesting that pairs generated in Pairs Graph and scored with Simple or GraphBased routines are already a good baseline achieving on movies an F-score of 83% and 87% correspondingly. Propagating initial scores with PPR further improves the performance achieving a very competitive results with F-score of 98.1% on movies and F-score of 98.4% on actors. The GraphBased initial scores perform better than Simple initial scores, proving that optimization (3.3) properly captures contribution of terminal values with respect to the graph structure.

This result compares favorably with the most recent state-of-the-art greedy approach SiGMa (Lacoste-Julien et al., 2013) that achieves an F-score of 97% on

movies when merging smaller datasets, also derived from IMDB and Freebase, with 3.1M and 474K entities correspondingly and with ground truth of 255K movies. This justifies the efficiency of HolisticEM, that is able to handle 60 times bigger datasets of 203M and 110M entities with 6 times bigger ground truth of 1.6M.

In addition to primary entities HolisticEM efficiently resolves relationship entities matches achieving high F-score of 98.0% on performances (Table 3.2).

3.6 Conclusion

We propose a novel scalable framework for collective entity matching across knowledge graphs. We describe a new way of constructing a graph of potential matching pairs and propose a new scheme to propagate similarity between pairs in this graph. Building the subgraph from seeds, adding necessary connections and controlling its expansion can be a very efficient graph sampling technique for dense graphs, where considering quadratic number of all possible pairs makes any further computations infeasible. By propagating scores via Personalized Page Rank we significantly simplified the entity matching routine. Our PPR-based framework does not require any prior domain knowledge, training, probabilistic inference; it scales to large datasets and has a competitive performance on both primary and relationship entities. This approach can be implemented on MapReduce to efficiently handle industrial size datasets.

Algorithm 1 : Pairs Graph Construction

Input: graphs $G_1(V_1, E_1(s,p,o))$, $G_2(V_2, E_2(s,p,o))$
Phase 1: generate Seeds $=\{(s_1, s_2) | s_1 \in V_1, s_2 \in V_2\}$
Phase 2: expand Seeds

table Edges(pair(s_1, s_2), pair(o_1, o_2));
// One step expansion for all seeds.
for $(a_s, b_s) \in$ Seeds **do**
    Edges=Edges $\cup$ SingleStep(a_s, b_s);
end for
table newNodes =select pairs $(s_1, s_2), (o_1, o_2)$
                from Edges;
// Second step expansion for relationship entities.
for $\{(a,b) \in$ newNodes & isRelationshipEnt$(a,b)\}$ **do**
    Edges=Edges $\cup$ SingleStep(a, b);
end for

makeBidirectionalEdges(Edges);
table AllNodes(s_1, s_2)=select $(s_1, s_2), (o_1, o_2)$
                from Edges, Seeds;
// Find shared terminal attributes of the same type $p_1 = p_2$.
table TerminalEdges(terminal o, pair(s_1, s_2))=
  AllNodes combine E_1 on AllNodes.$s_1 = E_1.s_1$
        combine E_2 on AllNodes.$s_2 = E_2.s_2$
        where $E_1.p_1 = E_2.p_2$
        and $E_1.o_1 = E_2.o_2 = o$ is terminal;
return Edges $\cup$ TerminalEdges;

function SingleStep(pair (a, b))
    // case 1: find edges to object pairs
    table Expand$_{\mathbf{O}}$(pair(a,b), pair(o_1, o_2)) =
        E_1 combine E_2 on $p_1 = p_2, s_1 = a, s_2 = b$;
    // case 2: find edges from subject pairs
    table Expand$_{\mathbf{S}}$(pair(s_1, s_2), pair(a,b)) =
        E_1 combine E_2 on $p_1 = p_2, o_1 = a, o_2 = b$;
    // Remove conflicting nodes via StableMarriage algo
    NewEdges = StableMarriage(Expand$_{\mathbf{O}}$ $\cup$ Expand$_{\mathbf{S}}$);
    return NewEdges;
end function

Figure 3.2: Algorithm 1 for graph expansion.

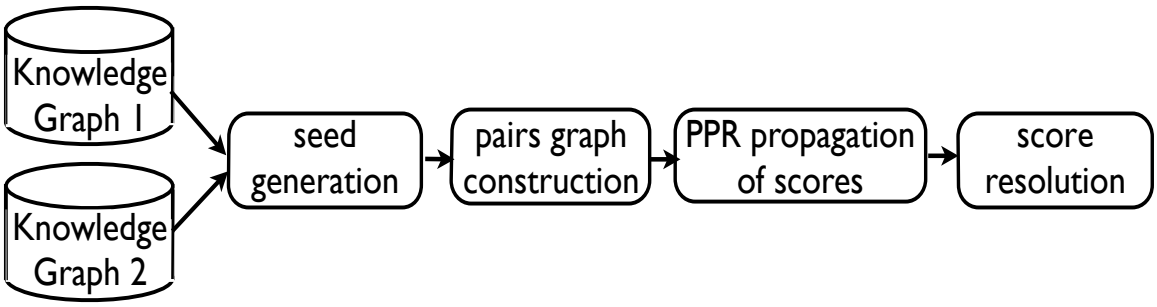

Figure 3.3: Pipeline for HolisticEM framework.

Chapter 4

Entity Disambiguation

4.1 Introduction

In Chapter 3 we studied the entity matching problem and presented a graph-based approach to find duplicates in databases. A database is a structured input defined by its schema, and it can be naturally presented as a graph, where graph-ranking techniques, such as Personalized PageRank, can be applied. It turns out that similar methods can be used for Named Entity Disambiguation (NED) - the task to map textual entity mentions in a document with structured data in a Knowledge Base (KB).[1]

While input for NED has a different nature - combination of unstructured and structured data - it is still possible to represent it as a graph and to apply graph-ranking techniques similar to those used for Entity Matching. There are several additional steps in NED that are needed to process this input before it

1. This chapter is a revised version of (Pershina et al., 2015c).

can be converted to a graph. First, named entity recognition is applied to detect named entity mentions in a document. Second, a list of KB candidates should be generated for every entity mention. Third, a notion of node has to be introduced. Forth, edges have to be constructed to represent relations between nodes in a graph.

Entity matching and NED tasks have different scales since a database may comprise hundred millions of entities while an average news document has at most several hundreds of entities. Different scales pose different challenges for these tasks. A practical method for entity matching has to be efficient and scalable for large input. On the other hand the large dense graph for entity matching is a more accurate snapshot of relations in databases and thus is more robust to noise. The much smaller scale of NED input requires a cherry-picking approach that would properly filter out noise and can efficiently combine all available information about other nodes in a graph.

Finally, for the NED task it is often implied that every entity mention has a correct corresponding entry in a knowledge base. Thus, a natural accuracy measure for this task is *precision*@1.0 - fraction of entities disambiguated correctly assuming that candidates pool always include the true answer. We will discuss a more realistic setting for this problem in the next Chapter 5 devoted to entity linking. Namely, we will omit an assumption that every entity has a correct entry in a knowledge base and propose a technique to cluster together entities that refer to the same real world object and are not mapped to any valid entry in a KB. For entity matching task this assumption does not hold either and thus precision and

recall are used to measure the accuracy of the model.

NED is both useful on its own, and serves as a valuable component in larger Knowledge Base Construction systems (Mayfield, 2014). Since the surge of large, publicly available knowledge bases (KB) such as Wikipedia, the most popular approach has been linking text mentions to KB nodes (Bunescu and Paşca, 2006). In this paradigm, the NED system links text mentions to the KB, and quite naturally utilizes information in the KB to support the linking process. Recent NED systems (Alhelbawy and Gaizauskas, 2014; Cucerzan, 2007; Ratinov et al., 2011) usually exploit two types of KB information: *local* information, which measures the similarity between the text mention and the candidate KB node; and *global* information, which measures how well the candidate entities in a document are connected to each other, with the assumption that entities appearing in the same document should be coherent. There is a trade-off between local and global views since both types of features have their strengths and drawbacks: local features better encode similarity between a candidate and a KB node, but overlook the coherence between entities; global features are able to exploit interlinking information between entities, but can be noisy if they are used on their own, without considering information from the text and the KB.

In this chapter, we propose to disambiguate NEs using a Personalized PageRank (PPR) random walk algorithm. Given a document and a list of entity mentions within the document, we first construct a graph whose vertices are linking candidates and whose edges reflect links in Wikipedia. We run the PPR algorithm on this graph, with the constraint that we only allow the highest scored candidate

for each entity to become the start point of a hop. As all candidates but the correct one are erroneous and probably misleading, limiting the random walk to start from the most promising candidates effectively filters out potential noise in the Personalized PageRank process.

Our method has the following properties: 1) as our system is based on a random walk algorithm, it does not require training model parameters ; 2) unlike previous PageRank based approaches in NED (Alhelbawy and Gaizauskas, 2014) which mainly rely on global coherence, our method is able to better utilize the local similarity between a candidate and a KB node (Section 4.3); and 3) we tailor the Personalized PageRank algorithm to focus on a single high-confidence entity at a time to reduce noise (Section 4.4).

4.2 Related Work

Early attempts at the NED tasks use local and surface level information. (Bunescu and Paşca, 2006) first utilize information in a knowledge base (Wikipedia) to disambiguate names, by calculating the similarity between the context of a name mention and the taxonomy of a KB node.

Later research, such as (Cucerzan, 2007) and (Milne and Witten, 2008) extends this line by exploring richer feature sets, such as coherence features between entities. Global coherence features have therefore been widely used in NED research (see e.g. (Cheng and Roth, 2013; Hoffart et al., 2011; Ratinov et al., 2011)) and have been applied successfully in TAC shared tasks (Cucerzan, 2011). These methods often involve optimizing an objective function that contains both local and global

terms, and thus requires training on an annotated or distantly annotated dataset.

Our system performs collective NED using a random walk algorithm that does not require supervision. Random walk algorithms such as PageRank (Page et al., 1999) and Personalized PageRank (Jeh and Widom, 2003) have been successfully applied to NLP tasks, such as Word Sense Disambiguation (WSD: (Agirre and Soroa, 2009; Sinha and Mihalcea, 2007)).

(Alhelbawy and Gaizauskas, 2014) successfully apply the PageRank algorithm to the NED task. Their work is the closest in spirit to ours and performs well without supervision. We try to further improve their model by using a PPR model to better utilize local features, and by adding constraints to the random walk to reduce noise.

4.3 The Graph Model

We construct a graph representation $G(V, E)$ from the document D with pre-tagged named entity textual mentions $M = \{m_1, ..., m_k\}$. For each entity mention $m_i \in M$ there is a list of candidates in the KB, $C_i = \{c_1^i, ..., c_{n_i}^i\}$. Vertices V are defined as pairs

$$V = \{\ (m_i, c_j^i) \mid\ \ m_i \in M, c_j^i \in C_i\ \},$$

corresponding to the set of all possible KB candidates for different mentions in M. Edges are undirected and exist between two vertices if the two candidates are directly linked in the knowledge base, but no edge is allowed between candidates for the same named entity. Every vertex (m, c) is associated with an initial similarity

score between entity mention m and candidate c (Figure 4.1).

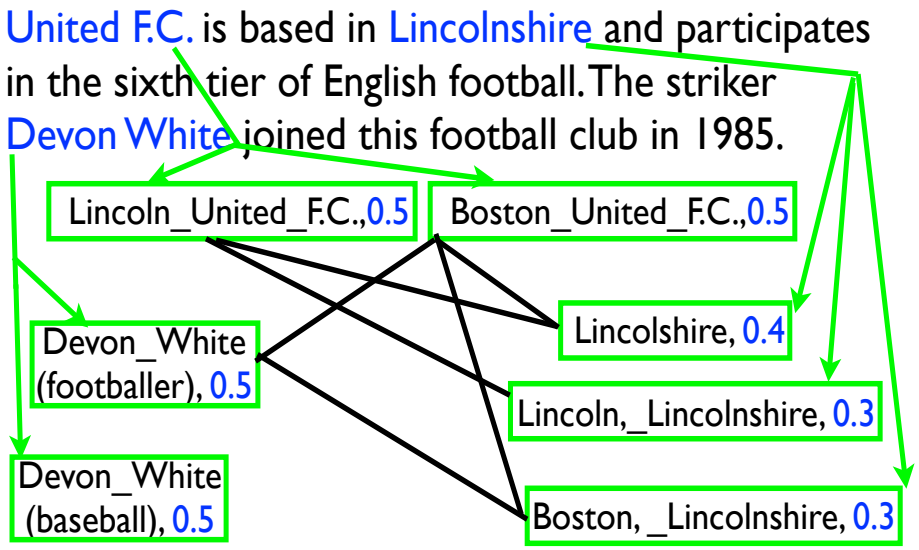

Figure 4.1: A toy document graph for three entity mentions: *United F.C., Lincolnshire, Devon White*. Candidates and their initial similarity scores are generated for each entity mention.

4.3.1 Vertices

4.3.1.1 Candidates

Given named entity mentions M in the document, we need to generate all possible candidates for every mention $m \in M$. We first perform coreference resolution on the whole document and expand m to the longest mention in the coreference chain. We then add a Wikipedia entry c to the candidate set C_i for mention m_i if 1) the title of c is the same as the expanded form of m_i, or 2) string m_i redirects to page c, or 3) c appears in a disambiguation page with title m_i.

4.3.1.2 Initial Similarity

Initial similarity $iSim$ for vertex (m, c) describes how similar entity mention m to candidate c is. It is independent from other candidates in the graph G. We experiment with the local measure (localSim), based on the local information about the entity in the text, and the global measure (popSim), based on the global importance of the entity. Initial similarity scores of all candidates for a single named entity mention are normalized to sum to 1.

- **localSim**: The local similarity score is produced by a MaxEnt model trained on the TAC2014 EDL training data (LDC2014E15). MaxEnt features include string similarity between the title of the Wikipedia entry and the entity mention, such as edit distance, whether the text mention starts or ends with the Wikipedia title, etc; and whether they have the same type (e.g. person, organization, location, etc).

- **popSim**: We use the Freebase popularity as an alternative similarity measure. The Freebase popularity is a function of entity's incoming and outgoing link counts in Wikipedia and Freebase.[2]

4.3.2 Edges

Edges in our graph model represent relations between candidates. We insert an edge between two candidates if the Wikipedia entry corresponding to either of

2. https://developers.google.com/freebase/v1/search

the two candidates contains a link to the other candidate. We assume that this relation is bidirectional and thus this edge is undirected.

There is a toy document graph in Figure 4.1 with three entity mentions and seven candidates: three candidates generated for *Lincolnshire*, and two candidates generated for *United F.C.* and *Devon White* each. Each graph node $e(m, c)$ is a pair of an entity mention m and a candidate c; every node is assigned an initial score, normalized across all candidates for the same entity. An edge is drawn between two candidates for different entities whenever there is a link from the Wikipedia page for one candidate to the Wikipedia page for another. There is no edge between candidates competing for the same entity.

4.4 The Challenge

A successful entity disambiguation algorithm would benefit from both the initial similarity between candidate and entity, as well as the coherence among entities in the same document. We assume that every entity can refer to at most one in the list of possible candidates, so all candidates except for the correct one for each entity are erroneous and will introduce noise into the document graph. Based on this observation, we contend that the typical random walk approach, which computes coherence of one candidate to the whole graph, is not suitable for our scenario. To address this problem, we propose to consider pairwise relations between every two nodes, given by PPR scores, compute the contribution of every node to the coherence of the other, and impose *aggregation constraints* to avoid redundant contributions.

4.4.1 Personalized PageRank

The PageRank algorithm considers random walk on a graph, where at each step with probability ϵ (teleport probability) we jump to a randomly selected node on a graph, and with probability $1 - \epsilon$ we follow a random outgoing edge of the current node. Stationary distribution of this walk gives PageRank weights associated with each node. Personalized PageRank is the same as PageRank, except that all teleports are made to the same source node, for which we are personalizing the PageRank.

4.4.2 Coherence and Constraints

The *coherence* of the node e to the graph G quantifies how well node e "fits" into this graph. Intuitively, pairwise weights $PPR(s \rightarrow e)$ represent relationships between nodes in the graph: the higher the weight is, the more relevant endpoint e is for the source s. Candidate nodes in the graph have different quality, measured by their initial similarity $iSim$. Thus, coherence of the node e to the graph G due to the presence of node s is given by

$$coh_s(e) = PPR(s \rightarrow e) \cdot iSim(s), \tag{4.1}$$

where relevance e for s is weighted by the $iSim(s)$, which is the similarity between entity e and candidate s. We experiment with a MaxEnt-trained local score and the Freebase popularity as the $iSim$ in Section 4.5.

We observe that summing the contributions $coh_s(e)$ for all nodes $s \in V$ would accumulate noise, and therefore impose two *aggregation constraints* to take into

account this nature of document graph G. Namely, to compute coherence $coh(e)$ of the node $e(m, c)$, corresponding to the entity mention m and the candidate c, to the graph G we enforce:

(c1) ignore contributions from candidate nodes competing for an entity m;
(c2) take only one, highest contribution from candidate nodes, competing for an entity $m' \neq m$;

The first constraint **(c1)** means that alternative candidates $\bar{e}(m, \bar{c})$, generated for the same entity mention m, should not contribute to the coherence of $e(m, c)$, as only one candidate per entity can be correct. For the same reason the second constraint **(c2)** picks the single candidate node $s(m', c')$ for entity $m' \neq m$ with the highest contribution $coh_s(e)$ towards e. So these constraints guarantee that exactly *one* and the *most relevant* candidate per entity will contribute to the coherence of the node e. Thus, the set of contributors towards $coh(e)$ is defined as

$$CONTR_{e(m,c)} = \{\ (m',\ \underset{c}{argmax}\ coh_{(m',c)}(e)\ ) \in V,\ m' \neq m\ \} \tag{4.2}$$

Then coherence of the node e to graph G is given by

$$coh(e) = \sum_{s \in CONTR_{e(m,c)}} coh_s(e) \tag{4.3}$$

Consider the example in Figure 4.1, which has two connected components. Candidate Devon_White_(baseball) is disconnected from the rest of the graph and can neither contribute towards any other candidate nor get contributions from other nodes. So its coherence is zero. All other candidates are connected, i.e. belong to the same connected component. Thus, the random walker, started from

any node in this component, will land at any other node in this component with some positive likelihood.

Let us consider the $CONTR_{e(m,c)}$ for entity mention m = *Lincolnshire* and candidate c = Lincolnshire, 0.4,. Without our constraints, nodes Devon_White_(footballer), 0.5, Lincoln_United_F.C., 0.5, Boston_United_F.C., 0.5, Lincoln_Lincolnshire, 0.3, Boston_Lincolnshire, 0.3 can all potentially contribute towards coherence of Lincolnshire, 0.4.

However, **(c1)** and **(c2)** will eliminate contribution from some of the candidates: Constraint **(c1)** does not allow Lincoln_Lincolnshire, 0.3 and Boston_Lincolnshire, 0.3 to contribute, because they compete for the same entity mention as candidate Lincolnshire, 0.4; constraint **(c2)** will allow only one contribution from either Lincoln_United_F.C., 0.5 or Boston_United_ F.C., 0.5 whichever is bigger, since they compete for the same entity mention *United F.C.*. Therefore, set $CONTR_{e(m,c)}$ for entity mention m = *Lincolnshire* and candidate c = Lincolnshire, 0.4, will contain only two contributors: candidate Devon_White_(footballer), 0.5, for entity mention *Devon_White,* and exactly one of the candidates for entity mention *United F.C.*

4.4.3 PPRSim

Our goal is to find the best candidate for every entity given a candidate's coherence and its initial similarity to the entity. To combine the coherence score $coh(e)$ with $iSim(e)$, we weight the latter with an average value of PPR weights used in

Models	Microaccuracy	Macroaccuracy
Cucerzan	51.03	43.74
Kulkarni	72.87	76.74
Hoffart	81.82	81.91
Shirakawa	82.29	83.02
Alhelbawy	87.59	84.19
iSim	62.61	72.21
PPR	85.56	85.86
PPRSim	91.77	89.89

Table 4.1: Performance of PPRSim compared to baselines and state-of-the-art models on AIDA dataset. Baselines iSim and PPR choose a candidate with the highest initial similarity or coherence correspondingly.

coherence computation (4.3) across all nodes in the document graph $G(V, E)$:

$$PPR_{avg} = \frac{\sum_{e \in V} \sum_{s \in CONTR_e} PPR(s \to e)}{|V|} \tag{4.4}$$

Thus, the final score for node e is a linear combination

$$score(e) = coh(e) + PPR_{avg} \cdot iSim(e) \tag{4.5}$$

If the document graph has no edges then PPR_{avg} is zero and for any node e its coherence $coh(e)$ is zero as well. In this case we set $score(e)$ to its initial similarity $iSim(e)$ for all nodes e in the graph G.

Finally, PPRSim disambiguates entity mention m with the highest scored candidate $c \in C_m$:

$$disambiguate(m) = \underset{c \in C_m}{\mathrm{argmax}} \;\; score(m, c) \tag{4.6}$$

To resolve ties in (4.6) we pick a candidate with the most incoming Wikipedia links.

Thus, candidate Devon_White_(footballer), 0.5 in Figure 4.1 will get higher overall score than its competitor, Devon_White_(baseball), 0.5. Their initial scores are the same, 0.5, but the latter one is disconnected from other nodes in the graph and thus has a zero coherence. So, entity mention *Devon White* will be correctly disambiguated with the candidate Devon_White_(footballer), 0.5. This candidate is directly connected to Boston_United_F.C., 0.5 and has a shortest path of length 3 to Lincolnshire_United_F.C., 0.5, and therefore contributes more towards Boston_United_F.C., 0.5, and boosts its coherence to make it the correct disambiguation for *United F.C.* Similarly, *Lincolnshire* is correctly disambiguated with Boston, Lincolnshire, F.C., 0.3.

4.5 Experiments and Results

4.5.1 Data

For our experiments we use dataset AIDA[3]. All textual entity mentions are manually disambiguated against Wikipedia links (Hoffart et al., 2011). There are 34,965 annotated mentions in 1393 documents. Only mentions with a valid entry in the Wikipedia KB are considered (Hoffart et al., 2011), resulting in a total of 27,816 mentions. We use a Wikipedia dump from June 14, 2014, as the reference KB. Our set of candidates is publicly available for experiments[4].

3. http://www.mpi-inf.mpg.de/yago-naga/aida/
4. https://github.com/masha-p/PPRforNED

4.5.2 Evaluation

We use two evaluation metrics: (1) Microaccuracy is the fraction of correctly disambiguated entities; (2) Macroaccuracy is the proportion of textual mentions, correctly disambiguated per entity, averaged over all entities.

4.5.3 PPR

We adopt the Monte Carlo approach (Fogaras and Balazs, 2004) for computing Personalized PageRank. It performs a number of independent random walks for every source node and takes an empirical distribution of ending nodes to obtain PPR weights with respect to the source. We initialized 2,000 random walks for every source node, performed 5 steps of PPR, and computed PPR weights from all iterations dropping walks from the first one. The teleport probability is set to 0.2.

4.5.4 Baselines

We performed a set of experiments using initial similarity and Personalized PageRank weights. Model iSim uses only Freebase scores and achieves microaccuracy of 62.61% (Table 5.1). PPR model picks a candidate with highest coherence, computed in (4.3), where no initial similarity is used ($iSim \equiv 1.0$) and no constraints are applied. It has a microaccuracy of 85.56%. This is a strong baseline, proving that coherence (4.3), solely based on PPR weights, is very accurate. We also reimplemented the most recent state-of-the-art approach by (Alhelbawy and Gaizauskas, 2014) based on the PageRank. We ran it on our set of candidates

PPRSim	Micro	Macro
$iSim \equiv 1.0$	85.56	85.86
$iSim =$ localSim	87.01	86.65
$iSim =$ popSim	90.26	88.98
+(c1)	90.52	89.21
+(c2)	91.68	89.78
+(c1),(c2)	91.77	89.89

Table 4.2: Performance of PPRSim with different initial similarities and constraints.

4.5.5 Results

We observe that PPR combined with global similarity popSim achieves a microaccuracy of 90.2% (Table 4.2). Adding constraints into the coherence computation further improves the performance to 91.7%. Interestingly, (c2) is more accurate than (c1). When put together, (c1)+(c2) performs better than each individual constraint (Table 4.2). Thus, combining coherence and initial similarity via (5.4) improves both micro- and macroaccuracy, outperforming state-of-the-art models (Table 5.1).

4.6 Conclusion

In this chapter we devise a new algorithm for collective named entity disambiguation based on Personalized PageRank. We show how to incorporate pairwise constraints between candidate entities by using PPR scores and propose a new robust scheme to compute coherence of a candidate entity to a document. Our approach outperforms state-of-the-art models and opens up many opportunities to

employ pairwise information in NED.

Chapter 5

Entity Linking

5.1 Introduction

In Chapter 4 we considered Named Entity Disambiguation task that maps textual entity mentions to corresponding entries in a knowledge base. A usual assumption for this problem is to expect a correct disambiguation KB entry for every entity mention in a text. The current chapter is devoted to Entity Linking (EL) task, where the ultimate goal is to cluster together textual entity mentions, that refer to the same real world object. The first step of this process links entity mentions to entries of some knowledge base. After linked entities are grouped together if they refer to the same KB entry, unlinked entities (NILs) have to be clustered as well. Unlike the NED task, we do not assume that every entity links to a KB entry. Thus this setting is a more realistic one since knowledge base is often incomplete and can miss some entities (Min et al., 2013). [1]

1. This chapter is a revised version of (Pershina et al., 2016).

So far we ignored local differences between entity mention strings. Therefore misspelled or corrupted names of the same named entity will never end up in the same cluster. To improve both mapping and clustering steps we address the problem of name variations - when same named entity is represented by different strings. This is particularly important for clustering NIL entities, that do not have a correct entry in the knowledge base and thus can be clustered only based on their string representation.

To evaluate our approach we use data provided by Entity Linking (EL) track at NIST Text Analysis Conference Knowledge Base Population (TAC-KBP) (Ji et al., 2014). This dataset is very different from the AIDA collection of documents, that we used in Chapter 4 for NED. First, it includes documents from different genres such as newswire, web data, discussion forum posts, and local news, while AIDA has only Reuters newswire articles. Second, about 35% of entity mentions in this dataset are NILs - they do not have a correct corresponding entry in the provided knowledge base. This makes it different from NED task, where we assumed that such an entry always exists.

As opposed to Entity Matching and Named Entity Disambiguation tasks, the evaluation metric for Entity Linking is designed to judge the quality of obtained clusters but not the accuracy for individual entities. Thus, the final evaluation depends on both the accuracy of intermediate disambiguation step against knowledge base as well as further clustering of NIL entities. Various clustering metrics are presented in (Bagga and Baldwin, 1998; Luo, 2013). We compute the B^3+F score that was used in TAC EDL 2014 competition.

Linking raw entity mentions in a document to real world entities is useful on its own and serves as a valuable component in larger Knowledge Base Construction systems (Mayfield, 2014), e.g. the Cold Start track of TAC KBP program where the goal is to develop an automatic system to construct a KB from scratch (Ji et al., 2014). In the Wikification community (Bunescu and Paşca, 2006), text mentions are linked to Wikipedia, a large and publicly available knowledge base.

There are two paradigms to solve the EL problem: local, non-collective approaches for Entity Linking resolve one mention at a time, relying on a context and local features, while collective approaches try to disambiguate the set of relevant mentions simultaneously, assuming that entities appearing in the same document should be coherent. (Alhelbawy and Gaizauskas, 2014; Cassidy et al., 2012; Cucerzan, 2007, 2011; Fernandez et al., 2010; Ferragina and Scaiella, 2010; Guo et al., 2011; Han and Zhao, 2009; Han and Sun, 2011; Hoffart et al., 2011; Huang et al., 2014; Kulkarni et al., 2009; Liu et al., 2013; Pennacchiotti and Pantel, 2009; Pershina et al., 2015c; Radford et al., 2010; Ratinov et al., 2011; Shen et al., 2013). We follow the second paradigm and present a collective approach, which is based on PPRSim for NED discussed in Chapter 4.

On the other hand, Nevertheless, we still try to capture the local similarity between the entity mention and its candidates in our model. To measure the similarity between entity mention strings we propose to use an approach which was proven effective for paraphrase detection. For this purpose we adopt the state-of-the-art ASOBEK paraphrase model (Eyecioglu and Keller, 2015). It was developed for paraphrase identification in Twitter and was ranked first among 19

teams on the Paraphrase In Twitter (PIT) 2015 task. It uses six simple character and word features and trains an SVM. This universal system is trained on pairs of entity name variations, which we make publicly available[2], and provides an accurate similarity measure between entity mention strings.

In this chapter we make the following contributions: 1) we use the paraphrase model to measure the similarity between entity mention strings and provide publicly available training data for this model; 2) we efficiently incorporate this model into a state-of-the-art entity disambiguation technique applied to the Entity Linking task and achieve the competitive result of 80.5% in $\mathbf{B}^3$+**F** score on the diagnostic TAC EDL 2014 dataset.

5.2 Related Work

Traditionally, there were two paradigms to solve Entity Linking problem: non-collective approaches (Guo et al., 2013; Han and Sun, 2011; Mihalcea and Csomai, 2007; Milne and Witten, 2008), and collective ones (Alhelbawy and Gaizauskas, 2014; Cassidy et al., 2012; Cucerzan, 2007, 2011; Fernandez et al., 2010; Ferragina and Scaiella, 2010; Guo et al., 2011; Han and Zhao, 2009; Han and Sun, 2011; Hoffart et al., 2011; Huang et al., 2014; Kulkarni et al., 2009; Liu et al., 2013; Medelyan et al., 2008; Pennacchiotti and Pantel, 2009; Pershina et al., 2015c; Radford et al., 2010; Ratinov et al., 2011; Shen et al., 2013).

Traditionally the EDL task starts by detecting entity mentions in text. Once the entities have been extracted, the EDL task relies on the systems developed for

2. `https://github.com/masha-p/paraphrase_flavor`

entity linking and NIL clustering.

The NIL Clustering task was added to TAC KBP in 2011, attracting the attention of many researchers. Entity Linking has been evaluated at several TAC conferences (2009-2013), and an overview of existing techniques can be found in (Ji and Grishman, 2011; Ji et al., 2014). Most efficient systems capture the interplay of Entity Linking and NIL clustering tasks (Monahan et al., 2014). While Entity Linking can scale linearly with the number of entities, clustering is a much more expensive operation (Singh et al., 2011). We address the problem of scalability by using a greedy clustering approach based on a simple paraphrase model. Its worst case running time is $O(mn)$ where m is the number of NILs in a document and n is the number of no-NILs.

5.3 Document Graph

5.3.1 Candidates

Given a document with pre-tagged named entity textual mentions M, we generate all possible candidates for every entity mention $m \in M$. First, we perform coreference resolution on the whole document and expand m to the longest mention in the coreference chain. We then add a Wikipedia entry c to the candidate set C_i for mention m_i in one of three cases: 1) the title of c is the same as the expanded form of m_i; 2) string m_i redirects to page c; 3) c appears in a disambiguation page with title m_i.

5.3.2 Edges

To represent relations between candidates we insert an edge between two candidates if the Wikipedia entry corresponding to either of the two candidates contains a link to the other candidate. We assume that information can flow in either direction and thus this edge is undirected.

We construct a graph representation $G(V, E)$ from the document D with pre-tagged named entity textual mentions M=$\{m_1, ..., m_k\}$. For each entity mention $m_i \in M$ there is a list of candidates in the KB $C_i = \{c_1^i, ..., c_{n_i}^i\}$. Vertices V are defined as pairs

$$V = \{(m_i, c_j^i) | m_i \in M, c_j^i \in C_i\},$$

corresponding to the set of all possible KB candidates for different mentions in M. Every vertex (m, c) has an initial similarity score $iSim(m, c)$ between m and c.

5.3.3 Initial Similarity

We split m and c into sets of tokens T_m and T_c and recognize two cases: 1) if T_m and T_c have any tokens in common then their similarity is 1.0; 2) otherwise it is a reciprocal of the edit distance between m and c:

$$iSim(m, c) = \begin{cases} 1.0\ , & \text{if}\ \ T_m \cap T_c \neq \phi \\ \frac{1}{edit(m,c)}, & otherwise \end{cases} \tag{5.1}$$

Thus, the pairwise initial similarity for "*Buenos Aires*" vs "*Buenos Aires Wildlife Refugee*" and for "*Buenos Aires*" vs "*University of Buenos Aires*" equals to 1.0. This simple metric does not use any external resources and is applicable to all entity

mentions even if they do not appear in a Freebase and Wikipedia, as opposed to the freebase popularity metric used in (Alhelbawy and Gaizauskas, 2014; Pershina et al., 2015c). We show in Section 5.5 that combining (5.1) with PPRSim can efficiently utilize the document graph, that represents other entities, and perform competitively on the TAC EDL 2014 data.

5.4 Name Variations as Paraphrases

Depending on the text genre there can be different variations of the same named entity. Official sources such as newswire are more strict and more likely to use official titles to address people and organizations. The forum data, on the opposite, does not have such standards and may use interchangeably "Hillary Clinton" vs "Hitlery Clinton" , "richardsdenni" vs "Rich Dennison", "mich state fair" vs "Michigan st Fair", "the blond demon" vs "le demon blond", etc. Edit distance is not a reliable clue to detect these kind of differences. For example, the above pairs have edit distance of 4, 12, 11, and 15 correspondingly.

One can view name variations as paraphrases of the same entity mention. There is no strict definition of a paraphrase (Bhagat and Hovy, 2013) and in linguistic literature paraphrases are most often characterized by an approximate equivalence of meanings across phrases. Thus, in a broad sense, detecting whether two phrases refer to the same entity mention is a particular case of the paraphrase problem.

A growing body of research studied the problem of paraphrases in Twitter (Guo and Diab, 2012; Guo et al., 2013; Socher et al., 2011; Xu et al., 2015a), in bilingual data (Bannard and Callison-Burch, 2005), and even paraphrases between idioms

(Pershina et al., 2015b). Finally, there was a new **P**araphrase **I**n **T**witter track (PIT) proposed in SemEval 2015 (Xu et al., 2015b). Most paraphrase models are tailored for a data set that they will be applied to. Thus, Twitter paraphrase models often make use of hashtags, timestamps, geotags, or require topic and anchor words (Xu et al., 2015a). None of this is applicable to named entity mentions.

Based on this observation, we focus on a holistic ASOBEK approach (Eyecioglu and Keller, 2015) for paraphrase identification in entity linking. The ASOBEK model uses simple character and word features and trains a linear SVM. This work is motivated by set theory and every phrase is considered a set of either character uni/bi-grams (C_1, C_2), or word uni/bi-grams (W_1, W_2). There are three types of features derived from these sets: 1) count of elements in a set, e.g. $|C_1|$ (length); 2) count of elements in the set overlap, e.g. $|C_1^{phrase_1} \cap C_1^{phrase_2}|$; 3) count of elements in the set union, e.g. $|C_1^{phrase_1} \cup C_1^{phrase_2}|$. (Eyecioglu and Keller, 2015) reported best performance using just six features:

$$
\begin{aligned}
&|C_2^{phrase_1} \cap C_2^{phrase_2}|,\\
&|C_2^{phrase_1} \cup C_2^{phrase_2}|,\\
&|W_1^{phrase_1} \cap W_1^{phrase_2}|,\\
&|W_1^{phrase_1} \cup W_1^{phrase_2}|,\\
&|C_2^{phrase_1}|,\\
&|C_2^{phrase_2}|.
\end{aligned}
\tag{5.2}
$$

We adopt this model for our task for detecting name variations. Namely, we built our training data set of name variation pairs, extracted ASOBEK best features, and trained a linear SVM (Joachims, 2006)[3] on this data.

We tested the ASOBEK model for three different feature sets that were explored in original paper: 1) feature set that performed best (ASOBEK), six features (5.2) total; 2) same as above plus length in words $|W_1^{phrase_1}|, |W_1^{phrase_2}|$, eight features total; 3) same as above plus unigram features, twelve total. We plot precision-recall curves for these three variations (Figure 5.1). First feature set performs slightly better confirming the result of Eyecioglu and Keller, 2015; all three achieve maximal F-score around 92% with precision of 96% and recall 88%. For our experiments we use the first feature set that was proven to be the best in original paper.

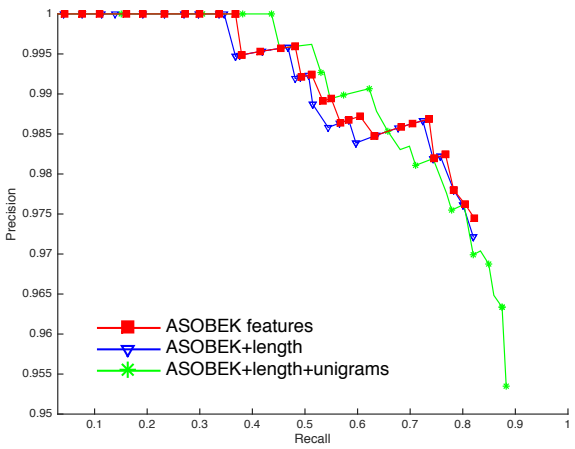

Figure 5.1: Performance of ASOBEK model with different feature sets applied to name variation task.

3. https://www.cs.cornell.edu/people/tj/svm_light/

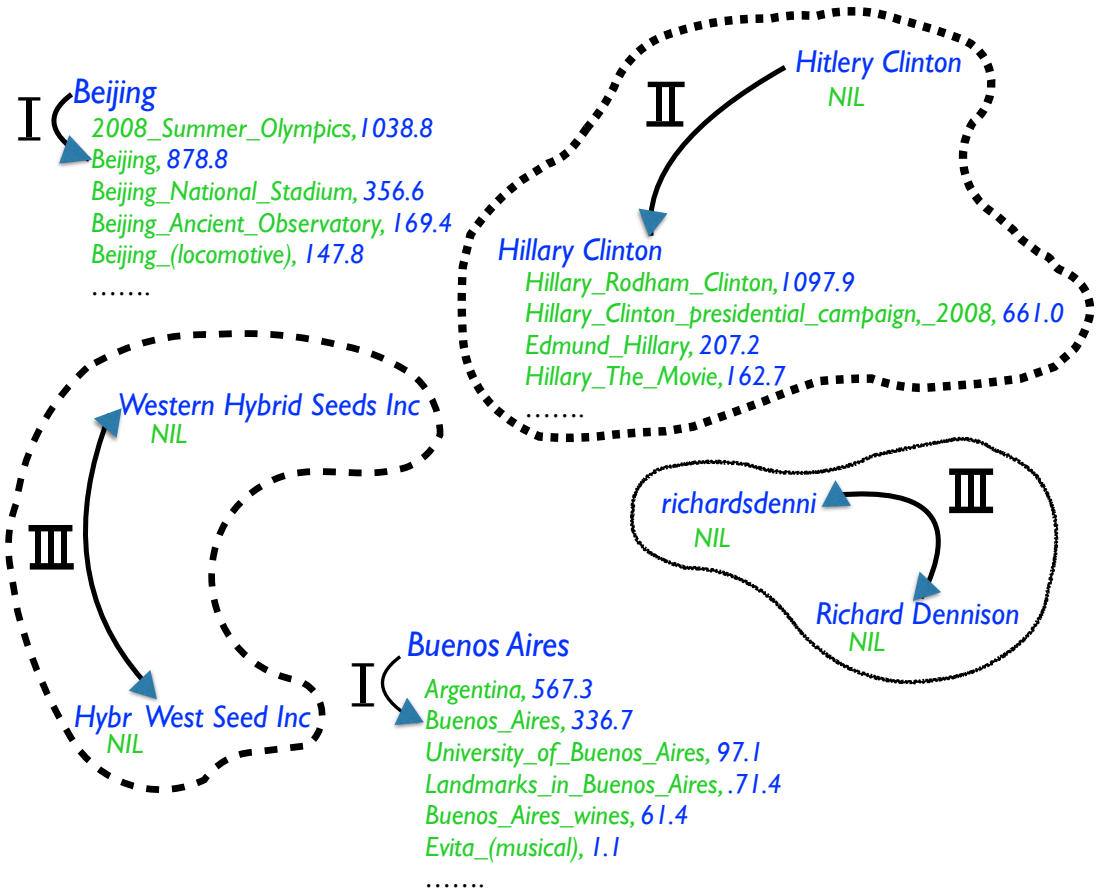

Figure 5.2: Examples of ParaLink refining and clustering steps I, II, III.

5.5 ParaLink

The most recent state-of-the-art entity disambiguation model PPRSim (Pershina et al., 2015c) runs Personalized PageRank (PPR) on the document graph and is based on intuition that pairwise weight $PPR(s \rightarrow e)$ measures how relevant endpoint e is for the source s. Then coherence of the node e to the graph G due

to the presence of node s is computed as

$$coh_s(e) = PPR(s \rightarrow e) \cdot iSim(s) \tag{5.3}$$

Since there can be only one correct candidate per entity, PPRSim imposes aggregation constraints to take only the highest contribution from candidate nodes, competing for the same entity. Finally, the total score for the node e is

$$score(e) = coh(e) + PPR_{avg} \cdot iSim(e) \tag{5.4}$$

where total coherence $coh(e)$ of node e to the graph is computed with respect to aggregation constraints and initial similarity score $iSim(e)$ is weighted by an average value of PPR weights used in coherence computation.

However, this approach often ranks higher a popular candidate connected to many nodes in a graph over the correct but less popular one. In fact, running PPRSim on the AIDA dataset yields a precision of 91.7% while the correct disambiguation link is contained within the top three ranked candidates for more than 99% of entity mentions[4].

For example, the top candidate for mention *Buenos Aires* is the incorrect entity *Argentina*, generated from the disambiguation page. It is winning over the correct one *Buenos Aires*, ranked second, due to a larger amount of incoming links (56K vs 12K) and thus a better connected neighborhood in a document graph (34 vs 26 edges). These candidates are top ranked by PPRSim on a document graph. However, the second candidate is a perfect paraphrase of the textual entity mention,

4. `https://github.com/masha-p/PPRforNED`

while the first one is not. Thus, using the similarity between the entity mention string and the KB entry title to select among the top-scoring candidates found by PPRSim can solve this problem (step I).

Entity disambiguation models usually assume that every entity mention has a valid KB entry and do not handle explicitly NIL entities. Thus NILs get clustered using the default one-name-per-cluster strategy. So, "Hitlery Clinton" will be clustered separately from "Hillary Clinton", "richardsdenni" will be separate from "Rich Dennison", etc. We propose to cluster every NIL candidate together with the most similar already linked entity mention if their paraphrase similarity is above a certain threshold obtained on a development dataset (step II).

Finally, NIL candidates, that were not assigned a link at the previous step, get clustered with the most similar NILs or constitute a singleton NIL cluster if no similar mentions can be found (step III). Thus ParaLink combines PPRSim with three additional refining steps based on paraphrase similarity between entity mention strings (Figure 5.2,5.3).

5.6 Experiments and Results.

5.6.1 Data

For our experiments we use the diagnostic TAC EDL 2014 dataset. Its training part consists of 158 documents with 5966 pretagged entity mentions; the test set contains 138 documents with 5234 pretagged entity mentions. All entity mentions are manually disambiguated against Wikipedia links, all NIL entities are clustered.

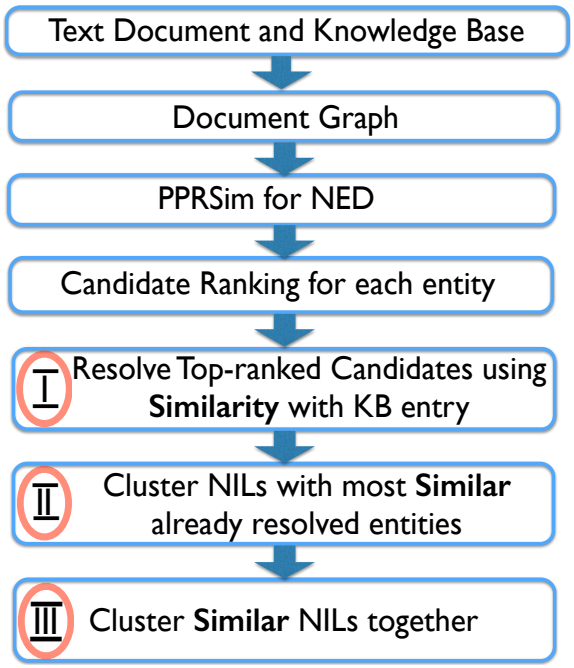

Figure 5.3: ParaLink diagram with refining and clustering steps I, II, III.

To train an ASOBEK model we extract name variations from the training data for TAC EDL 2014 task. Given entity clusters we pick pairs of entity mentions from the same cluster to create a set of name variations that refer to the same real world entity and we pair entity mentions from different clusters to have negative examples. Our training data consists of 1143 positive pairs and 1500 negative pairs, our test has 511 positive pairs and 1168 negative pairs. It is publicly available for future experiments.[5] We use TAC training data to tune for an optimal threshold for each step I,II,III.

5. https://github.com/masha-p/paraphrase_flavor

5.6.2 Evaluation

We use the standard TAC EDL clustering metric $\mathbf{B}^3$+**F** to evaluate baseline and ParaLink models. This metric compares the gold partitioning G and the one from the entity linking system S.

B-cubed cluster scoring compares clusters in the gold and response partition Bagga and Baldwin, 1998. The B-cubed cluster precision is the weighted average of a per-element precision score. Precision of an element A is the following:

$$\mathbf{B}^3 Precision(A, goldPartition, resPartition) =$$

$$\frac{|cluster(goldPartition, A) \cap cluster(resPartition, A)|}{|cluster(resPartition, A)|}$$

where $cluster(partition, A)$ is the cluster in the partition containing the element A; in other words, this is A's equivalence class and contains the set of all elements equivalent to A in the partition. Then each cluster in the gold partition is weighted equally, and each element is weighted equally within a cluster:

$$\mathbf{B}^3 ClusterPrecision(goldPartition, resPartition) =$$

$$\sum_a \frac{\mathbf{B}^3 Precision(a, goldPartition, resPartition)}{|goldPartition| * |cluster(goldPartition, a)|}$$

Recall is defined dually by switching the roles of gold and response partitions, and the F1-measure is defined in the usual way.

A brief analysis of the answer key revealed some mistakes in the TAC annotation. By fixing the answer link for 6 mentions in the training data (from the total of 5966) and for 22 mentions in a test data (from the total of 5234) we improved our $\mathbf{B}^3$+**F** by 0.1 and 0.2 correspondingly (Table 5.1). Our corrected answer keys are publicly available.

5.6.3 Baselines

We compare our model with several graph-based approaches. Our baseline is a faithful re-implementation of the NYU 2014 entity linking system based on PageRank (Alhelbawy and Gaizauskas, 2014; Ji et al., 2014; Nguyen et al., 2014) and ranked 4th in TAC EDL 2014. We compare it with the state-of-the-art PPRSim model for named entity disambiguation (Pershina et al., 2015c).

Models	Train data	Test data
NYU(PR)	76.2	76.3
PPRSim	78.4	78.9
PPRSim+I	79.1	80.0
PPRSim+II	79.5	80.3
PPRSim+III	79.2	80.2
ParaLink	79.7	80.5
ParaLink*	79.8	80.7

Table 5.1: Performance of ParaLink in $\mathbf{B}^3$+**F** score compared to the baseline and state-of-the-art models on TAC EDL 2014 train/test datasets. NYU (PR): PageRank with one-name-per-cluster name clustering; PPRSim: Personalized PageRank as described in (Pershina et al., 2015c); PPRSim+I/II/III: Combining PPRSim separately with steps in ParaLink; ParaLink: PPRSim with all steps I,II,III; ParaLink*: ParaLink scored on manually corrected TAC answer key.

5.6.4 Results

We observe that the refined disambiguation process for PPRSim (step I) improves the performance from 78.4% to 79.1% on training, and from 78.9% to 80.0% on test datasets. Adding paraphrase clustering (step II and III) further improves the $\mathbf{B}^3$+**F** score to achieve 79.7% and 80.5% on training and test datasets. Thus,

we show that paraphrase similarity can be efficiently incorporated into the entity linking pipeline and improve the performance (Table 5.1).

5.6.5 Discussion

Interestingly, performance of PageRank is about the same on both training and test data, while ParaLink achieves a better result on test dataset than on training one. The reason is that the fraction of discussion forum posts is slightly higher in test data than in training - about 20% vs 15%. ParaLink is particularly efficient for this type of data since it combines the power of disambiguation PPRSim model with ability to efficiently cluster misspelled and corrupted names, that are typical for forum posts. Thus it achieves a better performance on a dataset with more informal documents.

5.7 Conclusion

In this chapter we discuss the problem of name variations for the entity linking task. We show how to adopt ASOBEK paraphrase model to solve this problem and how to incorporate it into the entity linking pipeline. Using paraphrase paradigm for the name variations problem opens new perspectives for future research in Information Extraction.

For the future work we will further explore the problem of name variations and will extend our graph-based approach for better NIL detection in cases when only incorrect candidates are generated for the named entity. We plan to investigate

other clustering techniques for NIL and non-NIL entities.

Chapter 6

Future Work

This thesis focuses on the important problem of resolving entity ambiguity in various settings. It is an important step for many downstream applications, such as data integration, question answering, relation extraction, etc. Our methods are based on a graph representation of data and ranking techniques utilizing Personalized Page Rank. This approach introduces new challenges but also new potentials for resolving entity ambiguity.

Our future work on Entity Matching will be focused on different scenarios for constructing pairs graph and on a more sophisticated score propagation schemes. We plan to explore different strategies for filtering out noise when aggregating contributions from different nodes in order to further improve the score propagation.

We showed encouraging results and demonstrated the potential of modeling data as a graph but we skip the question of coverage/recall for both entity matching across knowledge graphs and entity linking for text documents. This is an important characteristics that can downgrade the performance if not properly ad-

dressed and tracked. Being unsupervised, our graph-based ranking method may require a more thorough study and/or labeled data in order to choose a true match from the ranked list of candidates. Our paraphrase component solves this problem to some extent leaving room for further improvement. Our current routine for entity linking also does not handle the case when none of proposed candidates is a true match and a NIL label should be assigned instead of the highest ranked candidate.

A clustering step for Entity Linking task can be performed differently and can be either before or after the linking step. For the future work we plan to further explore the problem of name variations and experiment with other features and other models for paraphrase identification. Also there can be several clustering strategies applied at the same time. It is an interesting and challenging topic on its own and is one of the directions to investigate and to further improve the entity linking. Another question to answer is whether our graph-based approach can be extended to identify NILs when proposed candidates do not contain a true match.

Bibliography

Agirre, Eneko and Soroa, Aitor (2009). "Personalizing PageRank for Word Sense Disambiguation". In: *Proceedings of the 12th Conference of European Chapter of the Association for Computational Linguistics (EACL)*, pp. 33–41.

Alhelbawy, Ayman and Gaizauskas, Robert (2014). "Graph Ranking for Collective Named Entity Disambiguation". In: *Proceedings of the 52nd Meeting of Association for Computational Linguistics (ACL)*, pp. 75–80.

Arasu, Arvind, Re, Christopher, and Suciu, Dan (2009). "Large-Scale Deduplication with Constraints using Dedupalog". In: *Proceedings of the International Conference on Data Engineering (ICDE)*, pp. 952–963.

Artiles, Javier, Sekine, Satoshi, and Gonzalo, Julio (2008). "Web people search: results of the first evaluation and the plan for the second". In: *Proceedings of the International Conference on World Wide Web (WWW)*, pp. 1071–1072.

Bagga, Amit and Baldwin, Breck (1998). "Entity-based cross-document coreferencing using the vector space model". In: *Proceedings of the Conference on Computational Linguistics (COLING)*, pp. 79–85.

Bannard, Colin and Callison-Burch, Chris (2005). "Paraphrasing with Bilingual Parallel Corpora". In: *Proceedings of the 43th Annual Meeting on Association for Computational Linguistics (ACL)*, pp. 597–604.

BIBLIOGRAPHY

Baxter, Rohan, Christen, Peter, and Churches, Tim (2003). "A Comparison of Fast Blocking Methods for Record Linkage". In: *Proceedings of the KDD Workshop on Data Cleaning, Record Linkage and Object Consolidation*, pp. 25–27.

Bhagat, Rahul and Hovy, Eduard (2013). "What is a paraphrase?" In: *Transactions of Computational Linguistics*, pp. 463–472.

Bhattacharya, Indrajit and Getoor, Liz (2006). "A Latent Dirichlet Model for Unsupervised Entity Resolution". In: *Proceedings of the 6th Siam International Conference on Data Mining (SDM)*, pp. 47–58.

— (2007). "Collective Entity Resolution in Relational Data". In: *Transactions of ACM Journal on Knowledge Discovery from Data (TKDD)*.

Bilenko, Mikhail, Kamath, Beena, and Mooney, Raymond J. (2006). "Adaptive Blocking: Learning to Scale Up Record Linkage". In: *Sixth International Conference on Data Mining (ICDM)*, pp. 87–96.

Bohm, Christoph, Marco, Gerard de, Naumann, Felix, and Weikum, Gerhard (2012). "LINDA: Distributed Web-of-Data-Scale Entity Matching". In: *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 2104–2108.

Brin, Sergey and Page, Lawrence (1998). "The anatomy of a large-scale hypertextual Web search engine". In: *Proceedings of the Computer Networks and ISDN Systems*, pp. 107–117.

Bunescu, Razvan and Paşca, Marius (2006). "Using Encyclopedic Knowledge for Named entity Disambiguation". In: *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 9–16.

BIBLIOGRAPHY

Cassidy, Taylor, Ji, Heng, Ratinov, Lev, Zubiaga, Arkaitz, and Huang, Hongzhao (2012). "Analysis and enhancement of wikification for microblogs with context expansion". In: *Proceedings of the International Conference on Computational Linguistics (COLING)*, pp. 441–456.

Cheng, Xiao and Roth, Dan (2013). "Relational Inference for Wikification". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1787–1796.

Chiang, Yueh-Hsuan, Doan, AnHai, and Naughton, Jeffrey F (2014). "Tracking Entities in the Dynamic World: A Fast Algorithm for Matching Temporal Records". In: *Proceedings of the Very Large Data Bases Conference (PVLDB)*, pp. 469–480.

Choi, Namyoun, Song, Il-Yeol, and Han, Hyoil (2006). "A survey on ontology mapping". In: *Proceedings of the ACM SIGMOD Record*, pp. 34–41.

Cohen, William, Kautz, Henry, and McAllester, David (1999). "Hardening soft information sources". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 255–259.

Cohen, William and Richman, Jacob (2002). "Learning to match and cluster large high-dimensional data sets for data integration". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 475–480.

Cowie, Jim and Lehnert, Wendy (1996). "Information Extraction". In: *Communications of the ACM Magazine*, pp. 80–91.

Cucerzan, Silviu (2007). "Large-Scale Named Entity Disambiguation Based on Wikipedia Data". In: *Proceedings of the 2007 Joint Conference on Empirical*

Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP, CoNLL), pp. 708–716.

Cucerzan, Silviu (2011). "TAC Entity Linking by Performing Full-document Entity Extraction and Disambiguation". In: *Proceedings of the 2011 Text Analysis Conference (TAC)*, pp. 708–716.

Culotta, Aron and McCallum, Andrew (2005). "Joint Deduplication of Multiple Record Types in Relational Data". In: *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 31–48.

Dong, Xin, Halevy, Alon, and Madhavan, Jayant (2005). "Reference Reconciliation in Complex Information Spaces". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 85–96.

Dong, Xin, Gabrilovich, Evgeniy, Heitz, Geremy, Horn, Wilko, Murphy, Kevin, Sun, Shaohua, and Zhang, Wei (2014). "From Data Fusion to Knowledge Fusion". In: *Proceedings of the Very Large Data Bases Conference (PVLDB)*, pp. 881–892.

Eyecioglu, Asli and Keller, Bill (2015). "Twitter Paraphrase Identification with Simple Overlap Features and SVMs". In: *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pp. 64–69.

Fellegi, Ivan and Sunter, Alan (1969). "A theory for record linkage". In: *Journal of the American Statistics Association*, pp. 1183–1210.

Fernandez, Norberto, Fisteus, Jesus, Sanchez, Luis, and Martin, Eduardo (2010). "Webtlab: A cooccurence-based approach to KBP 2010 entity-linking task". In: *Proceedings of the Text Analysis Conference (TAC)*.

Ferragina, Paolo and Scaiella, Uga (2010). "Tagme: on-the-fly annotation of short text fragments (by wikipedia entities)". In: *Proceedings of the ACM Inter-*

national Conference on Information and Knowledge Management (CIKM), pp. 1625–1628.

Fogaras, Daniel and Balazs, Racz (2004). "Towards Scaling Fully Personalized Page Rank". In: *Proceedings of the 3rd Workshop on Algorithms and Models for the Web-Graph (WAW)*, pp. 105–117.

Gale, David and Shapley, Lloyd (1962). "College Admissions and the Stability of Marriage". In: *The American Mathematical Monthly Journal*, pp. 9–15.

Gottipati, Swapna and Jiang, Jing (2011). "Linking entities toa knowledge base with query expansion". In: *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP)*, pp. 804–813.

Guha, Sudipto, Koudas, Nick, Marathe, Amit, and Srivastava, Divesh (2004). "Merging the results of approximate match operations". In: *Proceedings of the Very Large Data Bases Conference (PVLDB)*, pp. 267–276.

Guo, Yuhang, Che, Wanxiang, Liu, Ting, and Li, Sheng (2011). "A graph-based method for entity linking". In: *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 1010–1018.

Guo, Weiwei and Diab, Mona (2012). "Modeling Sentences in the Latent Space". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 864–872.

Guo, Weiwei, Li, Hao, Ji, Heng, and Diab, Mona (2013). "Linking Tweets to News: A Framework to Enrich Short Text Data in Social Media". In: *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 239–249.

BIBLIOGRAPHY

Guo, Zhaochen and Barbosa, Denilson (2014). "Robust entity linking via random walks." In: *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*, pp. 499–508.

Hall, Robert, Sutton, Charles, and McCallum, Andrew (2008). "Unsupervised deduplication using Cross-Field Dependencies". In: *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 310–317.

Han, Xianpei and Zhao, Jun (2009). "Named Entity Disambiguation by Leveraging Wikipedia Semantic Knowledge". In: *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*, pp. 215–224.

Han, Xianpei and Sun, Le (2011). "A generative entity-mention model for linking entities with knowledge base". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (HLT-ACL)*, pp. 945–954.

Hernandez, Mauricio and Stolfo, Salvatore (1995). "The merge/purge problem for large databases". In: *Proceedings of the ACM SIGMOD Conference on Management of Data*, pp. 127–138.

Herschel, Melanie and Naumann, Felix (2008). "Scaling up duplicate detection in graph data". In: *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*, pp. 1325–1326.

Hoffart, Johannes, Yosef, Mohamed Amir, Bordino, Ilaria, Fürstenau, Hagen, Pinkal, Manfred, Spaniol, Marc, Taneva, Bilyana, Thater, Stefan, and Weikum, Gerhard (2011). "Robust Disambiguation of Named Entities in Text". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 782–792.

BIBLIOGRAPHY

Huang, Jian, Treeratpituk, Pucktada, Taylor, Sarah, and Giles, Lee (2010). "Enhancing cross document coreference of web documents with context similarity and very large scale text categorization". In: *Proceedings of the Conference on Computational Linguistics (COLING)*, pp. 483–491.

Huang, Hongzhao, Cao, Yunbo, Huang, Xiaojiang, Ji, Heng, and Lin, Chin-Yew (2014). "Collective tweet wikification based on semi-supervised graph regularization". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 380–390.

Jeh, Glen and Widom, Jennifer (2003). "Scaling Personalized Web Search". In: *Proceedings of the International Conference on World Wide Web (WWW)*, pp. 271–279.

Ji, Heng and Grishman, Ralph (2011). "Knowledge base population: Successful Approaches and Challenges". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1148–1158.

Ji, Heng, Nothman, Joel, and Hachey, Ben (2014). "Overview of TAC-KBP2014 Entity Discovery and Linking Tasks". In: *Proceedings of the Text Analysis Conference (TAC)*.

Joachims, Thorsten (2006). "Training Linear SVMs in Linear Time". In: *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 217–226.

Kalfoglou, Yannis and Schorlemmer, Marco (2003). "Ontology mapping: the state of the art". In: *Knowledge Engineering Review Journal*, pp. 1–31.

Kenig, Batya and Gal, Avigdor (2013). "MFIBlocks: An effective blocking algorithm for entity resolution". In: *Proceedings of the Information Systems Journal*, pp. 908–926.

BIBLIOGRAPHY

Kulkarni, Sayali, Singh, Amit, Ramakrishnan, Ganesh, and Chakrabart, Soumen (2009). "Collective annotation of Wikipedia entities in web text". In: *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 457–466.

Lacoste-Julien, Simon, Palla, Konstantina, Davies, Alex, Kasneci, Gjerdji, Graepel, Thore, and Ghahramani, Zoubin (2013). "SiGMa: Simple Greedy Matching for Aligning Large Knowledge Bases". In: *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 572–580.

Liu, Xiaohua, Li, Yitong, Wu, Haocheng, Zhou, Ming, Wei, Furu, and Lu, Yi (2013). "Entity Linking for Tweets". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1304–1311.

Luo, Xiaoqiang (2013). "On Coreference Resolution Performance Metrics". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pp. 25–32.

Mann, Gideon and Yarovskiy, David (2003). "Unsupervised Personal Name Disambiguation". In: *Proceedings of the Conference on Natural Language Learning at HLT-NAACL (CoNLL)*, pp. 33–40.

Mayfield, James (2014). "Cold Start Knowledge Base Population at TAC 2014". In: *Proceedings of the Text Analysis Conference (TAC)*.

McCallum, Andrew, Nigam, Kamal, and Ungar, Lyle (2000). "Efficient clustering of high-dimensional data sets with application to reference matching". In: *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 169–178.

BIBLIOGRAPHY

McNamee, Paul, Simpson, Heather, and Dang, Hoa Trang (2009). "Overview of the TAC 2009 knowledge base population track". In: *Proceedings of the Text Analysis Conference (TAC)*.

Medelyan, Olena, Witten, Ian, and Milne, David (2008). "Topic indexing with Wikipedia". In: *Proceedings of the Wikipedia and Artificial Intelligence Workshop of Association for the Advancement of Artificial Intelligence Conference (AAAI)*, pp. 19–24.

Michelson, Matthew and Knoblock, Craig (2006). "Learning Blocking Schemes for Record Linkage". In: *Proceedings of the Association for the Advancement of Artificial Intelligence Conference (AAAI)*, pp. 440–445.

Mihalcea, Rada and Csomai, Andras (2007). "Wikify!: linking documents to encyclopedic knowledge". In: *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*, pp. 233–242.

Milne, David and Witten, Ian (2008). "Learning to Link with Wikipedia". In: *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*, pp. 509–518.

Min, Bonan, Grishman, Ralph, Wan, Li, Wang, Chang, and Gondek, David (2013). "Distant Supervision for Relation Extraction with an Incomplete Knowledge Base". In: *Proceedings of the North American Chapter of Association for Computational Linguistics (NAACL-HLT)*, pp. 777–782.

Monahan, Sean, Carpenter, Dean, Gorelkin, Maxim, Crosby, Kevin, and Brunson, Mary (2014). "Populating Knowledge Base with Entities and Events". In: *Proceedings of the Text Analysis Conference (TAC)*.

Newcombe, H, Kennedy, J, Axford, S, and James, A (1959). "Automatic linkage of vital records". In: *Science Journal*, pp. 954–959.

BIBLIOGRAPHY

Page, Lawrence, Brin, Sergey, Motwani, Rajeev, and Winograd, Terry (1999). *The PageRank Citation Ranking: Bringing Order to the Web*. Tech. rep. 1999-66. Stanford InfoLab.

Papadakis, George, Ioannou, Ekaterini, Palpanas, Themis, Niederee, Claudia, and Nejdl, Wolfgang (2013). "A Blocking Framework for Entity Resolution in Highly Heterogeneous Information Spaces". In: *IEEE Transactions on Knowledge and Data Engineering*, pp. 2665–2682.

Pennacchiotti, Marco and Pantel, Patrick (2009). "Entity extraction via ensemble semantics". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 238–247.

Pershina, Maria, He, Yifan, and Grishman, Ralph (2015c). "Personalized Page Rank for Named Entity Disambiguation". In: *Proceedings of the North American Chapter of Association for Computational Linguistics (NAACL-HLT)*, pp. 238–243.

— (2015b). "Idiom Paraphrases: Seventh Heaven vs Cloud Nine". In: *Proceedings of the EMNLP workshop on Linking Models of Lexical, Sentential and Discourse-Level Semantics (EMNLP-LSDSem)*, pp. 76–82.

Pershina, Maria, Yakout, Mohamed, and Chakrabarty, Kaushik (2015a). "Holistic Entity Matching Across Knowledge Graphs". In: *Proceedings of the IEEE Conference on Granular Computing*.

Pershina, Maria, He, Yifan, and Grishman, Ralph (2016). "Entity Linking with a Paraphrase Flavor". In: *International Conference on Language Resources and Evaluation (LREC)*.

Poesio, Massimo, Day, David, Artstein, Ron, Duncan, Jason, Eidelman, Vladimir, Giuliano, Claudio, Hall, R, Hitzeman, Janet, Jern, Alan, Kabadjov, Mijail,

Yong, S, Keong, W, Mann, Gideon, Moschitti, Alessandro, Ponzetto, Simone, Smith, Jason, Steinberger, J, Strube, Michael, Su, J, Versley, Yannick, Yang, Xiaofeng, and Wick, Michael (2007). "Exploiting lexical and encyclopedic resources for entity disambiguation: Final report. Tech rep." In: *JHU CLSP Summer Workshop*.

Popescu, Octavian (2010). "Dynamic parameters for cross document coreference". In: *Proceedings of the Conference on Computational Linguistics (COLING)*, pp. 988–996.

Radford, Will, Hachey, Ben, Nothman, Joel, Honnibal, Matthew, and Curran, James R (2010). "CMCRC at TAC10: Document-level entity linking with graph-based re-ranking." In: *Proceedings of the Text Analysis Conference (TAC)*.

Rahm, Erhard and Do, Hong Hai (2000). "Data cleaning: problems and current approaches". In: *IEEE Data Engineering Bulletin*, pp. 3–13.

Rao, Delip, McNamee, Paul, and Dredze, Mark (2010). "Streaming cross document entity coreference resolution". In: *Proceedings of the Conference on Computational Linguistics (COLING)*, pp. 1050–1058.

Rastogi, Vibhor, Dalvi, Nilesh, and Garofalakis, Minos (2011). "Large-scale collective entity matching". In: *Proceedings of the Very Large Data Bases Conference (PVLDB)*, pp. 208–218.

Ratinov, Lev, Roth, Dan, Downey, Doug, and Anderson, Mike (2011). "Local and Global Algorithms for Disambiguation to Wikipedia". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pp. 1375–1384.

BIBLIOGRAPHY

Sarawagi, Sunita and Kirpal, Alok (2004). "Efficient set joins on similarity predicates". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 247–258.

Shen, Wei, Wang, Jianyong, Luo, Ping, and Wang, Min (2013). "Linking named entities in tweets with knowledge base via user interest modeling". In: *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 68–76.

Shvaiko, Pavel and Euzenat, Jerome (2013). "Ontology matching: State of the Art and Future Challenges". In: *IEEE Transactions on Knowledge and Data Engineering*, pp. 158–176.

Singh, Sameer, Subramanya, Amarnag, Pereira, Fernando, and McCallum, Andrew (2011). "A large-scale cross-document coreference using distributed inerence and hierarchical models". In: *Proceedings of the 49st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 793–803.

Singla, Parag and Domingos, Pedro (2006). "Entity Resolution with Markov Logic". In: *Sixth International Conference on Data Mining (ICDM)*, pp. 572–582.

Sinha, Ravi and Mihalcea, Rada (2007). "Unsupervised Graph-basedWord Sense Disambiguation Using Measures of Word Semantic Similarity". In: *Proceedings of the International Conference on Semantic Computing*, pp. 363–369.

Socher, Richard, Huang, Eric, Pennington, Jeffrey, Ng, Andrew, and Manning, Christopher (2011). "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection". In: *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pp. 801–809.

BIBLIOGRAPHY

Suchanek, Fabian, Abiteboul, Serge, and Senellart, Pierre (2011). "PARIS: Probabilistic Alignment of Relations, Instances, and Schema". In: *Proceedings of the Very Large Data Bases Conference (PVLDB)*, pp. 157–168.

Sundheim, Beth (1992). "Overview of the fourth message understanding evaluation and conference". In: *Proceedings of the 49st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3–21.

Nguyen, Thien, He, Yifan, Pershina, Maria, Li, Xiang, and Grishman, Ralph (2014). "New York University 2014 Knowledge Base Population Systems." In: *Proceedings of the Text Analysis Conference (TAC)*.

Vries, Timothy de, Ke, Hui, Chawla, Sanjay, and Christen, Peter (2009). "Robust Record Linkage Blocking Using Suffix Arrays". In: *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 305–314.

Lassila, Ora and Swick, Ralph (1998). "Resource Description Framework (RDF) Model and Syntax Specification". In: *World Wide Web Consortium*.

Xu, Wei, Ritter, Alan, Callison-Burch, Chris, Dolan, William, and Ji, Yangfeng (2015a). "Extracting Lexically Divergent Paraphrases from Twitter". In: *Transactions of the Association for Computational Linguistics (TACL)*, pp. 435–448.

Xu, Wei, Callison-Burch, Chris, and Dolan, William (2015b). "SemEval-2015 Task 1: Paraphrase and Semantic Similarity in Twitter (PIT)". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*.

Whang, Steven, Menestrina, David, Koutrika, Georgia, Theobald, Martin, and Garcia-Molina, Hector (2009). "Entity Resolution with Iterative Blocking". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 219–232.

BIBLIOGRAPHY

Yakout, Mohamed, Pershina, Maria, and Chakrabarty, Kaushik (2014). "Holistic Entity Matching Across Knowledge Graphs". In: *Patent Application N MS 358015.01*.

Zhang, Wei, Su, Jian, Tan, Chew Lim, and Wang, Wen Ting (2010). "Entity linking leveraging automatically generated annotation". In: *Proceedings of the Conference on Computational Linguistics (COLING)*, pp. 1290–1298.