Context-Aware Entity Disambiguation in Text Using Markov Chains

Lei Zhang, Achim Rettinger and Patrick Philipp

Institute AIFB, Karlsruhe Institute of Technology (KIT), Germany
{l.zhang, rettinger, patrick.philipp}@kit.edu

***Abstract*—In recent years, the amount of entities in large knowledge bases has been increasing rapidly. Such entities can help to bridge unstructured text with structured knowledge and thus be beneficial for many entity-centric applications. The key issue is to link entity mentions in text with entities in knowledge bases, where the main challenge lies in mention ambiguity. Many methods have been proposed to tackle this problem. However, most of the methods assume certain characteristics of the input mentions and documents, e.g., only named entities are considered. In this paper, we propose a context-aware approach to collective entity disambiguation of the input mentions in text with different characteristics in a consistent manner. We extensively evaluate the performance of our approach over 9 datasets and compare it with 14 state-of-the-art methods. Experimental results show that our approach outperforms the existing methods in most cases.**

I. Introduction

In recent years, large repositories of structured knowledge publicly available on the Web, such as Wikipedia, DBpedia, Freebase and YAGO, have become valuable resources in many areas, such as natural language processing (NLP), information retrieval (IR) and knowledge extraction. In this regard, *entity disambiguation*, which leverages such knowledge bases to disambiguate the referent entities of the input mentions in natural language text, has emerged as a topic of major interest. The main challenge lies in the ambiguity of the textual entity mentions. Many methods [1], [2], [3], [4], [5], [6], [7], [8] have been proposed to tackle this problem, where the goal is to map each input mention given in text to the corresponding entity in knowledge bases. If there is no existing matching entity in the knowledge base, NIL will be returned. The knowledge base adopted in this work is DBpedia, a crowd-sourced community effort to extract structured information from Wikipedia.

In general, entities can be grouped into named entities and nominal entities. While named entities have proper names, nominal entities do not have a proper name but are referenced typically by a noun phrase, which has a noun as its head word. For instance, given the sentence "US President Barack Obama will land in India for a visit.", the mentions "*Barack Obama*" and "*India*" refer to the named entities Barack_Obama and India, while the mentions "*US President*" and "*visit*" refer to the nominal entities President_of_the_United_States and State_visit. In NLP area, recognizing named entities (NER) in natural language text has been extensively addressed [9], where the output is labeled noun phrases. However, these are not entities explicitly and uniquely denoted in a knowledge base. Recently, a lot of research has focused on *named entity disambiguation* that goes one step beyond NER, where the task is to disambiguate mentions of named entities in natural language text by linking them to their corresponding entities in a knowledge base [2], [10], [11]. On the other hand, in computational linguistics *word sense disambiguation* (WSD) is a task aimed at assigning meanings to word occurrences within text, where such words usually refer to nominal entities [12], [13]. In addition, some other work focuses on *Wikification*, commonly referred to as disambiguation to Wikipedia (D2W), a task of identifying entities in text and disambiguating them into the corresponding Wikipedia pages [14], [8], [15]. In this work, we do not assume any specific entity types in our entity disambiguation task, where the entities to be disambiguated could be named entities, nominal entities or both of them. We also argue that the nominal entities in the given text can help with disambiguating named entities and vice versa.

The main contributions of this paper are as follows: (1) the introduction of a context-aware approach to collective entity disambiguation for different kinds of input mentions in text in a consistent manner; (2) the contextual entity detection based on a set of predefined part-of-speech (POS) tag patterns, which provides the context to help with entity disambiguation for the given input mentions; (3) the collective disambiguation using a class of algorithms for estimating the relative importance of candidate entities in the constructed disambiguation graph based on Markov chains; and (4) an extensive evaluation of the performance of our approach over 9 datasets and an empirical comparison with 14 state-of-the-art methods using GERBIL [16], a general entity annotation benchmark.

The rest of this paper is organized as follows. We present the overall approach in Sec. II. The details of contextual entity detection and disambiguation graph construction are provided in Sec. III and Sec. IV, respectively. Based on that, we discuss the collective disambiguation using Markov chains in Sec. V. Evaluation results are then presented in Sec. VI. Finally, we survey the related work in Sec. VII and conclude in Sec. VIII.

II. Overview

In this section, we first formally formulate the task of entity disambiguation and then briefly describe our approach.

Definition 1 (Entity Disambiguation): Given a set of *input mentions* $M_I = \{m_1, m_2, \ldots, m_p\}$ in a document D, where each mention m is encoded by an integer pair $\langle p, l \rangle$ with p as the occurrence position of m in D and l as the length of m, and a *knowledge base* KB containing a set of entities

Example 1	**Text:** The novel begins in the Shire, where the hobbit Frodo Baggins inherits the Ring from Bilbo and undertakes the quest to destroy it. **Input mentions:** $\{m_1 = \langle 24, 5\rangle, m_2 = \langle 48, 13\rangle, m_3 = \langle 75, 4\rangle, m_4 = \langle 85, 5\rangle\}$
	Ref. entities: $\{m_1.e$=Shire_(Middle-earth), $m_2.e$=Frodo_Baggins, $m_3.e$=One_Ring, $m_4.e$=Bilbo_Baggins$\}$
Example 2	**Text:** The novel begins in the Shire, where the hobbit Frodo Baggins inherits the Ring from Bilbo and undertakes the quest to destroy it. **Input mentions:** $\{m_1 = \langle 4, 5\rangle, m_2 = \langle 41, 6\rangle, m_3 = \langle 110, 5\rangle\}$
	Ref. entities: $\{m_1.e$=Novel, $m_2.e$=Hobbit, $m_3.e$=Quest$\}$
Example 3	**Text:** The novel begins in the Shire, where the hobbit Frodo Baggins inherits the Ring from Bilbo and undertakes the quest to destroy it. **Input mentions:** $\{m_1 = \langle 4, 5\rangle, m_2 = \langle 48, 13\rangle, m_3 = \langle 106, 3\rangle\}$
	Ref. entities: $\{m_1.e$=Novel, $m_2.e$=Bilbo_Baggins, $m_3.e$=NIL$\}$

TABLE I: Some examples of the entity disambiguation task, where the *input mentions* and the *contextual* mentions in the given text are highlighted and shadowed, respectively.

$E = \{e_1, e_2, \ldots, e_n\}$, the task of *entity disambiguation* is to find a function $\mu : M_I \rightarrow E \cup \{NIL\}$, which maps each input mention m to an entity e in KB, denoted by $m.e$, or to NIL if the mention cannot be linked to any entity in KB.

For each given input mention $m \in M_I$, we first retrieve a set of *candidate entities* E_m using a dictionary collected from different structures in Wikipedia, which contains each pair of entity and *surface form*, i.e., a word or phrase that can be used to refer to the corresponding entity. Then the objective of entity disambiguation is to determine which entity $e \in E_m$ is the mostly likely entity referred to by m, also called *referent entity*. Besides the given input mentions in M_I for a document D, a set of mentions M_C containing the mentions $m \notin M_I$ in D, called *contextual mentions*, that can refer to some entities in the knowledge base, called *contextual entities*, could also help with the entity disambiguation task. While the input mentions are explicitly given, the contextual mentions have to be derived by our approach, which will be discussed in Sec. III.

Some examples of entity disambiguation for different types of input mentions are shown in Table I. For instance, only the input mentions for named entities are given in Example 1, which corresponds to the typical *named entity disambiguation*. Most existing approaches [2], [10], [5] to this task take into account only the named entities but ignore the nominal entities, such as Hobbit referred to by the contextual mention "hobbit", which can indeed help with named entity disambiguation since such contextual entities are related to the actual referent entities of the input mentions. In Example 2, some individual words referring to nominal entities are given as input mentions. This is similar to the *word sense disambiguation* task, where the goal is to identify which sense of a word (i.e. meaning) is used in the given text. Based on the lexical knowledge bases, such as WordNet, knowledge-based approaches are able to obtain good performance [17]. Instead of lexical knowledge bases, large structured knowledge bases, such as DBpedia, can also be employed, such that the contextual entities appearing in the given document can be utilized for the disambiguation of word senses as entities in such knowledge bases. In Example 3, three input mentions, i.e., "novel", "Frodo Baggins" and "the", are given and the actual referent entities include the nominal entity Novel, the named entity Frodo_Baggins and NIL. Similarly, the contextual entities in the given document can be beneficial to disambiguating all the input mentions. Even for NIL corresponding to "the", which could also refer to some entities according to our dictionary, such as the entity THE_multiprogramming_system, the contextual entities can help to return NIL, because they are not related to any candidate entities of the input mention "the".

Besides the above examples, the input mentions for entity disambiguation can be yielded by many other ways, e.g., they can cover only salient entities in the given document annotated based on voter agreement or determined by domain experts. A description of 9 datasets used in our experiments will show different characteristics of the input mentions and documents. In order to address the problem of entity disambiguation for such input mentions and documents in a consistent way, we propose a framework with the following three modules:

- **Contextual Entity Detection.** The entity disambiguation task critically depends on the specific context in a given document D, which is crucial in solving the problem of entity ambiguity. In this module, we propose a new approach to *contextual entity detection* based on a set of predefined POS patterns. The goal is to select contextual entities representing the context of D, which can help to disambiguate the entities for the input mentions.
- **Disambiguation Graph Construction.** By combining the candidate entities of input mentions and the contextual entities detected in the given document, we construct the *disambiguation graph* in this module, which captures both the local mention-entity compatibility and the global entity-entity coherence as its graph structure. In this way, the constructed disambiguation graph allows us to encode different types of dependencies.
- **Collective Entity Disambiguation.** We then consider the *collective entity disambiguation* over the disambiguation graph as a stochastic process based on Markov chains. The intuition is that the actual referent entity of an input mention m should be more relevant in the disambiguation graph in the sense that it tends to have more relations to other candidate entities and contextual entities, than the rest of candidate entities of m, which should have less relations on average and be more isolated.

III. Contextual Entity Detection

Given the input document, we need to derive the contextual entities, which can be either named entities or nominal entities. For instance, in Example 3 of Table I, "Bilbo" and "hobbit" can refer to the named entity Bilbo_Baggins and the nominal entity Hobbit respectively, both of which can help with the entity disambiguation for the input mentions. To obtain these contextual entities, it is essential to first detect their mentions.

We firstly present the extraction process of our dictionary used to map surface forms to their corresponding DBpedia entities. We have exploited several structures in Wikipedia. As each Wikipedia article describes an entity in DBpedia, article titles, redirect pages and link anchors in Wikipedia can be used to refer to the corresponding entity. For each DBpedia entity, we extract its surface forms using these sources. Besides that,

Pattern Name	POS Tag Pattern	Example
Noun 1 (NP1)	(NN|NNP|NNS|NNPS)+	Kobe Bryant, Basketball
Noun 2 (NP2)	NP1 • (CD)+	Windows 10, ISO 8
Noun 3 (NP3)	(CD)+ • NP1	2014 World Cup
Noun (NP)	NP1 | NP2 | NP3	
Description 1 (DP1)	(JJ|JJS|JJR)+	Military (Operation)
Description 2 (DP2)	(VBG|VBN)+	Judging (Day), Linked (Data)
Description 3 (DP3)	NP3 • POS+	NBA's (Player)
Description (DP)	(DP1|DP2 |DP3)	
Compound Noun 1 (CNP1)	DP* • NP	Australian Prime Minister Linked Open Data NBA's All-time Scoring List
Conjunction (CP)	(CC|IN)	of, in, and, with
Compound Noun 2 (CNP2)	CNP1 • CP • CNP1	Police in Sweden First Minister of Scotland
Contextual Mention	**CNP1 |CNP2**	

TABLE II: POS patterns in regular expressions, where symbols *, +, |and • denote any number of occurrences, one or more occurrences, alternation and concatenation, respectively; NN: singular noun; NNP: proper singular noun; NNS: plural noun; NNPS: proper plural noun; CD: cardinal digit; JJ: adjective; JJS: superlative adjective; JJR: comparative adjective; VBG: present participle of verb; VBN: past participle of verb; CC: conjunction; IN: preposition; POS: possessive 's or '.

we also derive the co-occurrence relations between entities and terms, where we utilize the terms that co-occur with an entity in its surrounding sentences in Wikipedia. In addition, the link frequency between each pair of entity and surface form and the co-occurrence frequency between each pair of entity and term are also extracted, which are used for node weighting of the disambiguation graph discussed in Sec. IV. More details about the dictionary construction can be found in [18], [19].

Next we introduce two methods that have been widely used for mention detection based on N-gram and NER, and discuss their limitations, which serve as the motivation of our proposed method based on POS analysis.

Some existing work on mention detection [14], [8] firstly gathers all n-grams from the given document and the extracted n-grams matching surface forms of entities are then selected as entity mentions. These methods can detect both named entities and nominal entities but could also generate a lot of noise, i.e., mentions without actual referent entities. For instance, in Example 3 of Table I, "begins" and "from" can also refer to Battle_of_France and Etymology based on our dictionary. Such entities will be considered in the module of collective entity disambiguation, which are not helpful and might even result in degraded performance.

In some other work [10], named entity recognition (NER) has been performed on the input text to detect named entities, which are then used for entity disambiguation and linking. Due to the limitation of selected algorithms and training data, NER systems usually only focus on several types of named entities, e.g., Person, Location and Organization, such that the entities in other types cannot be detected. More importantly, all the nominal entities that might be important contextual entities and be beneficial to entity disambiguation are just ignored. In Example 3 of Table I, the contextual entity Hobbit, which is crucial for the given entity disambiguation task, cannot be detected by the NER based method.

To address the problems of N-gram and NER methods, we propose a POS tagging based method for detecting mentions of contextual entities. Given the input document D, we firstly perform the POS tagging on D and then extract all sequences conforming to a set of predefined POS patterns, denoted by P, as shown in Table II. The extracted sequences based on the POS patterns serve as the mentions of contextual entities, which have to satisfy two conditions: (1) they can refer to some entities in DBpedia based on our dictionary containing the set of surface forms of all entities, denoted by SF; (2) they are not contained in the set of input mentions M_I. Then we obtain the set of contextual entity mentions M_C as follows

$$M_C = \{m | \forall sq_m \in SQ_D : sq_m \in P \wedge m.s \in SF \wedge m \notin M_I\} \quad (1)$$

where SQ_D represents the set of all possible sequences of POS tags generated by performing POS tagging on the given document D, sq_m and $m.s$ denote a sequence of POS tags and an entity surface form w.r.t. the mention m, respectively.

Based on our dictionary, we generate the set of contextual entities E_m for each mention $m \in M_C$ and the set of all contextual entities is then just the union of E_m for all mentions in M_C defined as $E_C = \cup_{m \in M_C} E_m$.

IV. Disambiguation Graph Construction

In this module, we retrieve the set of candidate entities E_m for each input mention $m \in M_I$ based on our dictionary and the set of all candidate entities is defined as $E_I = \cup_{m \in M_I} E_m$. We then build a directed weighted graph $G = \{V, R\}$, called *disambiguation graph*, where $V = E_I \cup E_C$ is the union of contextual and candidate entities, and R is the set of directed edges representing entity relations, where an edge between two entities e_i and e_j will be added into R if the following conditions are satisfied: (1) e_i is linked to e_j in KB, i.e., $e_i \rightarrow e_j$; (2) e_i and e_j have different mentions, i.e., $e_i \in E_m$, $e_j \in E_{m'}$ and $m \neq m'$. Our approach then employs several features to assign weights to nodes and edges in G.

A. Node Weighting

For each mention m, we first calculate its *prior importance* $PI(m)$ that captures how likely the surface form $m.s$ is used as an entity mention as follows

$$PI(m) = \frac{count_{link}(m.s)}{count_{link}(m.s) + count_{text}(m.s)} \quad (2)$$

where $count_{link}(s)$ denotes the number of articles that contain s as anchor text of links and $count_{text}(s)$ denotes the number of articles where s appears as raw text without links.

For each pair of mention m and its associated entity e, we calculate their semantic similarity $SS(m, e)$ that represents the local *mention-entity compatibility* between m and e as

$$SS(m, e) = \alpha \cdot LP(m, e) + (1 - \alpha) \cdot CS(m, e) \quad (3)$$

where $LP(m, e)$ denotes the link probability of e for m and $CS(m, e)$ denotes the context similarity between m and e, α

is a tunable parameter. The link probability $LP(m, e)$ captures how likely $m.s$ refers to e, which can be calculated as

$$LP(m, e) = \frac{count_{\text{link}}(e, m.s)}{\sum_{e_i \in E_s} count_{\text{link}}(e_i, m.s)} \quad (4)$$

where $count_{\text{link}}(e, s)$ denotes the number of links using s as anchor text pointing to e as destination and E_s is the set of entities that have the surface form s. The context similarity $CS(m, e)$ between m and e can be calculated using cosine similarity on the term vectors of the context of m and e as

$$CS(m, e) = cos(m.\mathbf{c}, e.\mathbf{c}) = \frac{\langle m.\mathbf{c}, e.\mathbf{c} \rangle}{|m.\mathbf{c}| \cdot |e.\mathbf{c}|} \quad (5)$$

where $m.\mathbf{c}$ is the frequency vector of terms that contained in the surrounding sentences of m and $e.\mathbf{c}$ is the frequency vector of terms that co-occur with e extracted from Wikipedia.

Using the prior mention importance as the initial evidence and the mention-entity compatibility capturing the most likely entity behind the mention, we calculate the score of each $v \in V$ corresponding to entity e that has the mention m as

$$S(v) = PI(m) \cdot SS(m, e) \quad (6)$$

Based on that, the probability $p(v)$ serving as the weight of each node $v \in V$ can be calculated as follows

$$p(v) = \frac{S(v)}{\sum_{u \in V} S(u)} \quad (7)$$

B. Edge Weighting

The module of collective entity disambiguation relies on the global *entity-entity coherence*, which reflects the intuition that entities appearing in the same document are more likely to be related. Therefore, we calculate the semantic relatedness between each pair of connected entities e_i and e_j in G by adopting the Wikipedia link based measure [20] as

$$SR(e_i, e_j) = 1 - \frac{\log(\max(|E_i|, |E_j|)) - \log(|E_i \cap E_j|)}{\log(|E|) - \log(min(|E_i|, |E_j|))} \quad (8)$$

where E_i and E_j are the sets of entities that link to e_i and e_j in KB respectively, and E is the set of all entities in KB.

Based on the entity relatedness, we calculate the transition probability for each edge from u to v in G as follows

$$p(v|u) = \begin{cases} \frac{SR(u,v)}{\sum_{w \in OUT_u} SR(u,w)} & \text{if } (u, v) \in R \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where OUT_u is the set of entity nodes such that for each node $w \in OUT_u$, there is an edge from u to w in G.

V. Collective Entity Disambiguation

Based on the constructed disambiguation graph, we consider collective entity disambiguation as a stochastic process, more specifically, a first-order Markov chain model. Intuitively, it can be interpreted as a process where a single "random walker" traverses a graph in a stochastic manner for an infinitely long time and the fraction of time that the walker spends at a single node, i.e., the stationary distribution of the Markov chain, can then be considered as being proportional to an estimate of the importance of this node relative to others in the graph.

For the disambiguation graph G, where nodes represent both candidate entities and contextual entities in a given document D and edges correspond to relations between these entities, the Markov analogy could be seen as an *ad infinitum* stream of thought that refers to the interconnection in a sequence of entities thought by the author for writing the document D.

There is a class of algorithms that have been proposed for estimating relative importance of nodes in a graph based on Markov chains. To address the entity disambiguation problem, we start with the simple method of eigenvector centrality [21], and then discuss the well-known PageRank [22] and HITS algorithms [23] as well as their extensions with prior bias [24].

A. Eigenvector Centrality

Eigenvector centrality [21] provides a principled method to combine the importance of a node in a graph with its neighbors in ranking. The scores correspond to the likelihood of arriving in each node by traversing through the graph with a random starting node, where the decision to take a particular path is based on the weighted edges. Given the disambiguation graph G, eigenvector centrality of nodes in G can be defined as the principle eigenvector of the transition matrix $\mathbf{T}$ constructed from the weights of edges in G. The equation of the principle eigenvector $\mathbf{c}$ is defined as $\mathbf{c} = \mathbf{T} \cdot \mathbf{c}$, where the maximal eigenvalue λ corresponding to $\mathbf{c}$ is 1, since $\mathbf{T}$ is a square stochastic adjacency matrix. Each entry $T(u, v)$ in $\mathbf{T}$ specifies the transition probability $p(v|u)$ from node u to v in G, which is defined in Eq. 9, and each entry $c(v)$ in $\mathbf{c}$ represents the eigenvector centrality of node v, which is proportional to the sum of eigenvector centrality of all nodes connected to v. It can be estimated through the iterative calculation as

$$c^{i+1}(v) = \sum_{u \in IN_v} p(v|u) \cdot c^i(u) \quad (10)$$

where IN_v is the set of entity nodes such that for each node $u \in IN_v$, there is an edge from u to v in G. For each mention m having a set of candidate entities E_m, we choose the entity with the maximal $c(v)$ as the predicted linking entity, i.e., $e_m^* = \arg\max_{v \in E_m} c(v)$.

Based on the Perron-Frobenius theorem [25], an irreducible and aperiodic Markov chain can be guaranteed to converge to a unique stationary distribution. If a Markov chain has reducible or periodic components, a random walker may get stuck in these components and never visit the other parts of the graph. To solve this problem, PageRank [22] suggests reserving some probability for jumping to any node in the graph, such that the random walker can "escape" from periodic or disconnected components, which makes the graph irreducible and aperiodic. We will discuss this issue in the following.

B. PageRank

PageRank [22] is the most well-known example of Markov chains for ranking Web pages in search engine results, where the Markov analogy is defined as a "random surfer" surfing the Web based on the hyperlinks between Web pages. In the

traditional PageRank, a uniform probability is assigned to any node in the Web hyperlink graph in case of random jumps of a surfer. Given the disambiguation graph $G = (V, R)$, we first define a $|V| \times 1$ vector $\mathbf{p}_V$, whose elements are $\frac{1}{|V|}$. With the uniform prior probability $p(v)$ in $\mathbf{p}_V$ attached to each node v and the probability $p(v|u)$ of transitioning from all nodes u linked to v, as defined in Eq. 9, the iterative probability equation of v in a Markov chain can be defined as follows

$$\pi^{(i+1)}(v) = (1-d)\cdot(\sum_{u\in IN_v} p(v|u)\cdot\pi^{(i)}(u)) + d\cdot p(v) \quad (11)$$

where IN_v is the set of entity nodes such that for each node $u \in IN_v$, there is an edge from u to v in G and d is the damping factor, which determines how often a surfer jumps back to node v with probability $d\cdot p(v)$ and is typically chosen in the interval $[0.1, 0.2]$.

In [26], [27], PageRank has been extended to generate "personalized" ranks, called *personalized* PageRank, where the prior probability of nodes are non-uniform such that it can effectively bias the resulting ranks to prefer certain kinds of nodes. In this regard, we replace the uniform distribution $p(v) = \frac{1}{|V|}$ for each $v \in V$ with the non-uniform prior probability $p(v)$ defined in Eq. 6. This is analogous to adding a set of weighted outgoing edges for all the nodes in G. Intuitively, this creates a small probability for a random walk to go to some other nodes in G, although it may not have been initially connected to the current node. After convergence of the Markov chain, each node v will achieve a stationary probability $\pi(v)$. For each mention m having a set of candidate entities E_m, we choose the entity with the maximal $\pi(v)$ as the predicted linking entity, i.e., $e_m^* = \arg\max_{v\in E_m} \pi(v)$.

C. HITS

Besides PageRank, another seminal contribution to ranking nodes in Web graph is HITS [23], where two kinds of scores, namely hub and authority, are assigned to nodes in the graph depending on the topology of Web graph. In [24], HITS has been extended by fitting it into a more Markov fashion, where prior probabilities are assigned to nodes to permit random jumps. Given the disambiguation graph G, we incorporate the prior probability vector $\mathbf{p}_V$ for nodes in G into the extended HITS algorithm. Similar to PageRank, the prior probability $p(v)$ in $\mathbf{p}_V$ can be defined as uniform distribution, i.e., $p(v) = \frac{1}{|V|}$, or non-uniform according Eq. 6. This yields the following iterative equation for both hub and authority scores of each node v

$$a^{(i+1)}(v) = (1-d)\cdot(\sum_{u\in IN_v})\frac{p(v|u)\cdot h^{(i)}(u)}{H^{(i)}} + d\cdot p(v) \quad (12)$$

$$h^{(i+1)}(v) = (1-d)\cdot(\sum_{u\in OUT_v})\frac{p(u|v)\cdot a^{(i)}(u)}{A^{(i)}} + d\cdot p(v) \quad (13)$$

where IN_v (OUT_v) is the set of entity nodes such that for each $u \in IN_v$ ($u \in OUT_v$) there is an edge from u to v (v to u) and d is the damping factor similar to PageRank. $H^{(i)}$ and $A^{(i)}$ are defined as

$$H^{(i)} = \sum_{v\in V}\sum_{u\in IN_v} p(v|u)\cdot h^{(i)}(u) \quad (14)$$

$$A^{(i)} = \sum_{v\in V}\sum_{u\in OUT_v} p(u|v)\cdot a^{(i)}(u) \quad (15)$$

Datasets	#Doc.	#Ent.	Avg. Ent./Doc.	Avg. Word/Doc.
ACE2004	57	253	4.44	459
AIDA/CoNLL	231	4485	19.42	213
AQUAINT	50	727	14.54	320
DBpedia Spotlight	58	330	5.69	32
IITB	103	11242	109.15	763
KORE50	50	143	2.82	14
MSNBC	20	650	32.5	688
N^3 RSS-500	500	590	1.18	34
N^3 Reuters-128	128	637	4.98	140

TABLE III: Features of the datasets, including the numbers of documents and ground truth entities as well as the average numbers of ground truth entities and words per document.

After convergence of the algorithm, each node v corresponding to a candidate entity gets a hub score $h(v)$ and an authority score $a(v)$. Given the set of candidate entities E_m of a mention m, we choose the entity with the maximal authority score $a(v)$ as the predicted linking entity, i.e., $e_m^* = \arg\max_{v\in E_m} a(v)$.

Regarding the NIL entity problem, we use a threshold τ to determine whether we return the predicted entity e_m^* for a mention m or return NIL for all the algorithms including eigenvector centrality, traditional PageRank, PageRank with priors, traditional HITS and HITS with priors.

VI. Experiments

We conducted extensive experiments to assess our approach using GERBIL [16], a general entity annotation benchmark. In the following, we firstly discuss the experimental settings and then present the evaluation results.

A. Experimental Settings

In the experiments, we use DBpedia 2014[1] as the knowledge base. The experiments were carried out on 9 different datasets. An overview of these datasets is shown in Table III. In the following, we briefly describe these datasets and their features.

ACE2004 This dataset introduced by [28] is a subset of the ACE co-reference dataset, where the annotations are obtained by asking annotators on Amazon's Mechanical Turk to link the first mention of each co-reference chain to Wikipedia.

AIDA/CoNLL This dataset introduced by [10] is divided into 3 chunks: Training, TestA and TestB, where only named entities are annotated. In [10], the first two chunks are used for training and tuning, only TestB, made up of 231 documents, is used for testing. In our experiments, we also use Training and TestA for parameter learning and tuning, and use TestB for assessing the performance of our approach.

AQUAINT This dataset introduced by [8] consists of 50 newswire texts, where instead of annotating all occurrences of entities, only some important entities and their first mentions are retained to mimic the hyperlink structure in Wikipedia.

DBpedia Spotlight This dataset produced in [1] contains quite short texts, where the mentions of both named entities and nominal entities are annotated.

IITB This dataset presented by [29] contains 103 Web documents, where almost all mentions for broad types of entities including the not highly relevant ones are annotated.

[1] http://wiki.dbpedia.org/Downloads2014

	ACE2004	*AIDA/CoNLL*	*AQUAINT*	*DBpedia Spotlight*	*IITB*	*KORE50*	*MSNBC*	*N^3 RSS-500*	*N^3 Reuters-128*	*ACE2004*	*AIDA/CoNLL*	*AQUAINT*	*DBpedia Spotlight*	*IITB*	*KORE50*	*MSNBC*	*N^3 RSS-500*	*N^3 Reuters-128*
Systems	Micro F1									Macro F1								
ADGISTIS	0.63	0.47	0.51	0.27	0.47	0.32	0.65	0.61	**0.64**	0.77	0.5	0.49	0.28	0.48	0.3	0.61	0.61	0.7
AIDA	0.09	0.4	0.08	0.22	0.18	0.64	0.25	0.43	0.37	0.42	0.41	0.08	0.19	0.19	0.59	0.23	0.38	0.3
Babelfy	0.52	0.54	0.68	0.53	0.37	**0.74**	0.64	0.45	0.45	0.69	0.5	0.68	0.52	0.35	**0.71**	0.59	0.39	0.39
DBpedia Spotlight	0.47	0.42	0.53	0.71	0.3	0.43	0.37	0.2	0.33	0.67	0.44	0.51	0.69	0.28	0.39	0.36	0.17	0.26
Dexter	0.52	0.4	0.52	0.29	0.21	0.2	0.35	0.37	0.36	0.67	0.38	0.51	0.26	0.21	0.14	0.37	0.3	0.31
Entityclassifier.eu	0.49	0.41	0.42	0.25	0.14	0.29	0.45	0.34	0.37	0.66	0.41	0.38	0.2	0.16	0.26	0.44	0.32	0.34
FOX	0	0.45	0	0.15	0.02	0.29	0.02	0.56	0.55	0.37	0.44	0	0.12	0.02	0.25	0.02	0.54	0.58
FRES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.36	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.37
FREME NER	0	0	0	0	0	0	0	0	0	0.37	0	0	0.02	0	0	0	0	0
KEA	0.64	0.52	0.77	**0.74**	0.48	0.59	0.7	0.44	0.51	0.76	0.52	0.76	**0.73**	0.46	0.53	0.67	0.39	0.46
NERD-ML	0.56	0.45	0.58	0.55	0.43	0.32	0.54	0.38	0.41	0.72	0.45	0.56	0.53	0.42	0.26	0.54	0.31	0.35
TagMe 2	0.67	0.47	0.71	0.67	0.37	0.57	0.57	0.47	0.43	0.78	0.46	0.69	0.66	0.36	0.49	0.57	0.39	0.36
WAT	0.64	0.58	0.72	0.66	0.41	0.59	0.62	0.44	0.51	0.76	0.59	0.72	0.67	0.39	0.48	0.57	0.37	0.43
Wikipedia Miner	0.69	0.45	0.77	0.69	0.44	0.42	0.5	0.41	0.47	0.79	0.45	0.75	0.67	0.42	0.34	0.48	0.37	0.39
NC+PRankP	0.66	0.76	0.65	0.4	0.48	0.52	0.69	0.49	0.45	0.77	0.74	0.66	0.39	0.47	0.52	0.73	0.59	0.49
NER+PRankP	0.71	0.76	0.70	0.52	0.51	0.55	**0.71**	0.56	0.53	0.81	0.75	0.72	0.54	0.50	0.55	**0.75**	0.65	0.59
N-gram+PRankP	0.65	**0.78**	**0.8**	0.51	0.52	0.51	0.57	0.63	0.54	0.8	0.79	**0.8**	0.47	0.51	0.5	0.62	0.7	0.63
***POS+PRankP**	**0.78**	**0.78**	0.79	0.58	**0.54**	0.54	0.65	**0.64**	**0.64**	**0.86**	**0.8**	0.79	0.64	**0.54**	0.53	0.71	**0.71**	**0.71**
POS+EigenC	0.25	0.31	0.34	0.26	N/A	0.18	0.34	0.34	0.31	0.53	0.34	0.34	0.26	N/A	0.22	0.36	0.5	0.41
POS+HITS	0.17	0.28	0.11	0.33	0.07	0.43	0.16	0.4	0.22	0.61	0.37	0.12	0.34	0.07	0.43	0.22	0.55	0.42
POS+HITSP	0.68	0.69	0.62	0.44	0.47	0.5	0.63	0.59	0.56	0.8	0.73	0.62	0.42	0.46	0.49	0.66	0.66	0.66
POS+PRank	0.75	0.77	0.71	0.49	0.52	0.54	**0.71**	0.6	0.58	0.84	0.78	0.72	0.44	0.51	0.53	0.73	0.67	0.65

TABLE IV: Comparison of 8 variants of our approach and 14 state-of-the-art approaches on 9 datasets using Micro F1 and Macro F1 (best results formatted in bold), where if a system provides no results or errors, we report them as N/A (not available).

KORE50 This dataset [10] aims for hard disambiguation task with very ambiguous mentions. 50 hand-crafted, difficult sentences from different domains are comprised in this dataset.

MSNBC This dataset is presented by [30], in which all mentions of named entities are annotated in 20 news articles. It focuses on disambiguating named entities after running NER and co-reference resolution systems on newsire text.

N^3 RSS-500 This dataset is one of the N^3 datasets [31], where 500 sentences selected from crawled RSS feeds for a wide range of topics are annotated by domain experts.

N^3 Reuters128 This is another N^3 dataset [31], which contains 128 economic news articles, where the annotations of entities and mentions are determined by two domain experts.

Based on the TestA chunk and the Training chunk of the AIDA/CoNLL dataset, the parameter α in Eq. 3 has been tuned and we learn the threshold τ to determine whether we return the predicted entity e_m^* for a mention m as the target entity or return NIL. Regarding NER and POS based contextual entity detection, we employ Stanford Named Entity Recognizer[2] and POS Tagger[3]. For N-gram based contextual entity detection, we extract all n-grams with $n \leq 20$.

B. Evaluation Results

We extensively evaluated various variants of our approach to entity disambiguation based on different combinations of the methods for contextual entity detection (including *NC*, *NER*, *N-gram* and *POS*, where *NC* denotes the method without using any contextual entities such that the disambiguation graph contains only candidate entities of the input mentions and the others denote the methods using NER, N-gram and POS based contextual mention detection, respectively) and the algorithms for collective entity disambiguation (including *EigenC*, *PRank*, *PRankP*, *HITS* and *HITSP*, which denote the algorithms of Eigenvector Centrality, traditional PageRank, PageRank with Priors, traditional HITS and HITS with Priors, respectively). In the experiments, we employ the measures of micro F1 and macro F1 as the quality criteria.

The experimental results show that our approach with the combination of *POS* and *PRankP*, denoted by *POS+PRankP*, achieves the best results on most datasets compared to other combinations. Due to the limitation of space, we focus the following discussion on 8 variants of our approach, where *POS* or *PRankP* is involved. We compare our approach against 14 state-of-the-art approaches using GERBIL and the results will be discussed in Sec. VI-B1. In addition, the impact of different contextual entity detection methods and collective entity disambiguation algorithms in our approach will be discussed in Sec. VI-B2 and Sec. VI-B3, respectively.

1) Comparison with State-of-the-Art Methods: As shown in Table IV, we compare our approach with 14 state-of-the-art approaches on 9 datasets. The best variant of our approach *POS+PRankP* outperforms all 14 state-of-the-art approaches on 7 out of 9 datasets for both micro F1 and macro F1. Besides *POS+PRankP*, some other variants can also achieve relatively good results compared to the state-of-the-art approaches. For example, *NER+PRankP*, *N-gram+PRankP* and *POS+PRank* outperforms all state-of-the-art approaches on 4 datasets for micro F1 and on 5 datasets for macro F1, respectively.

We observe that our approach doesn't work well for two datasets, i.e., *DBpedia Spotlight* and *KORE50*, where *KEA* and *Babelfy* achieve the best results for each dataset, respectively. The reason could be that the documents in these two datasets

[2]http://nlp.stanford.edu/software/CRF-NER.html

[3]http://nlp.stanford.edu/software/tagger.html

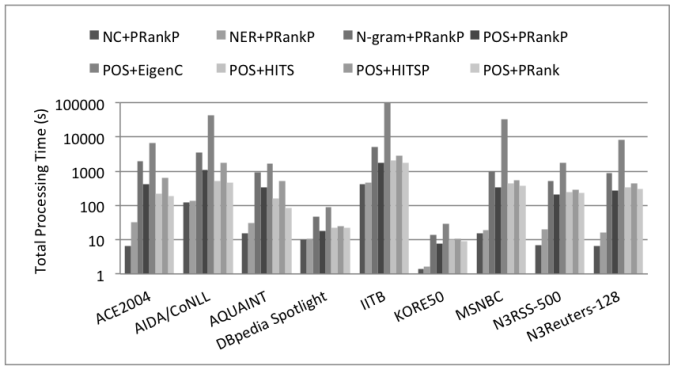

Fig. 1: Total processing time (s) of 8 variants of our approach.

are very short and also contain very ambiguous mentions such that our approach doesn't have enough context to perform the collective entity disambiguation. For such kind of documents, the context should be extracted not only from the given document itself but also from other external resources.

2) Analysis of Contextual Entity Detection: Among the variants of our approach based on different contextual entity detection methods, *POS+PRankP* apparently achieves the best results in most cases. According to both measures of micro F1 and macro F1, it obtains the best results on 5 out of 9 datasets. Compared with *POS+PRankP*, *N-gram+PRankP* yields the best results on 2 datasets for micro F1 and 1 dataset for macro F1, and *NER+PRankP* gets the best results on 1 dataset for both micro F1 and macro F1. In general, the variants of our approach using contextual entity detection, i.e., *NER+PRankP*, *N-gram+PRankP* and *POS+PRankP*, considerably outperform *NC+PRankP* that doesn't use any contextual entities.

Note that *NC+PRankP* also achieves very good results on the *AIDA/CoNLL* and *MSNBC* datasets, where it outperforms all 14 state-of-the-art approaches. The reason could be that each document in these datasets contains quite a lot of input mentions of named entities, such that these input mentions result in more candidate entities that can be utilized by the collective disambiguation. Although the *IITB* dataset has a much higher average number of input mentions per document, many of them refer to entities that are not relevant and thus cannot be beneficial to collective entity disambiguation, such that *NC+PRankP* doesn't perform very well on *IITB*.

In addition, we investigate the impact of different contextual entity detection methods on the runtime performance of our approach to entity disambiguation. Fig. 1 illustrates the total time for processing 9 datasets using different variants of our approach. We observe that *N-gram+PRankP* requires more time than *POS+PRankP*, which in turn, takes more time than *NER+PRankP* and *NC+PRankP*. This reflects the fact that *N-gram* results in more contextual entities than *POS*, which have to be taken into account by the collective disambiguation algorithms. Similarly, *POS* results in more contextual entities than *NER*. Since *NC* doesn't yield any contextual entities, it achieves the best runtime performance.

3) Analysis of Collective Entity Disambiguation: We now analyze the impact of different collective entity disambiguation algorithms, where *POS* is assumed to be the contextual entity detection method. As shown in Table IV, the variant using *PRankP* clearly outperforms the others. While *PRank* and *HITSP* yield relatively good results, the variants with *EigenC* and *HITS* show really poor performance.

Regarding the runtime performance as illustrated in Fig. 1, we observe that the variant with *EigenC* takes substantially more time, where the processing of the *IITB* dataset did not stop after running for one day such that we manually stopped it, while the variants with the other collective disambiguation algorithms exhibit only minor differences.

VII. Related Work

In this section, we review the state-of-the-art approaches to entity disambiguation, which have been empirically compared with our approach in the experiments.

DBpedia Spotlight [1] is one of the first approaches by combining named entity recognition and disambiguation based on DBpedia. By employing a vector space model, each entity is represented as a vector in a multidimensional word space, where term frequency (TF) and inverse document frequency (IDF) are utilized to model the relevance and importance of words. In addition, the inverse candidate frequency (ICF) is used to weight words according to their ability to distinguish between candidate entities.

Wikipedia Miner [8] is one of the oldest tools widely used for entity disambiguation and linking based on Wikipedia. It provides useful statistics about anchor texts and links in Wikipedia and defines an entity relatedness measure using Wikipedia link structures. Based on a classifier using different features, e.g., prior probability, context relatedness and quality, an entity disambiguator and a link detector are provided.

NERD [4] has been proposed for recognizing and extracting entities from tweets. Using a conditional random fields (CRF) model, entity types can be classified based on a rich feature vector composed of several linguistic features. In addition, a set of NER extractors are supported by the NERD Framework The follow-up, NERD-ML [6] improved the classification task by redesigning the selection of the features.

TagMe 2 [3] utilizes a set of links, pages and an in-link graph from Wikipedia to annotate entities in natural language text. It first recognizes named entities by matching terms with Wikipedia anchor texts and then disambiguates the detected mentions using the in-link graph and page information from Wikipedia. Furthermore, the identified named entities that are considered as non-coherent to the rest of the entities in the given text are then pruned by TagMe 2.

WAT [7] is the successor of TagMe including a re-design of all its components, i.e., the spotter, the disambiguator and the pruner, where two sets of algorithms have been introduced: the graph-based algorithms for collective entity linking and the vote-based algorithms for local entity disambiguation, and SVM linear models are used to tune the spotter and the pruner.

AGDISTIS [5] is a pure entity disambiguation framework, which aims at increasing the accuracy of entity disambiguation by combining some measures for calculating string similarity,

a label expansion strategy for co-referencing and the HITS algorithm for graph-based disambiguation. According to this combination, the correctness of entities detected in a given document can be significantly improved.

AIDA [10] only focuses on named entities and adopts the YAGO knowledge base as the entity collection to perform entity disambiguation. It relies on coherence graph building and dense subgraph algorithms, which aims at maximizing the coherence among the selected annotations.

KEA NER/NED [32] considers heterogeneous text sources created by automated multimedia analysis as context, which have different levels of accuracy, completeness, granularity and reliability. Ambiguity is solved by selecting candidate entities with the highest probability according to the context.

Babelfy [33] is based on random walk models and a densest subgraph algorithm to tackle both word sense disambiguation and entity linking tasks in a multilingual setting depending on the BabelNet semantic network.

Dexter [34] is an open-source framework with the aim of simplifying the implementation of entity disambiguation and linking such that it allows to replace single parts of the system, where several methods have been integrated.

VIII. Conclusions

In this paper, we proposed a context-aware approach to collective entity disambiguation for the input mentions with different characteristics in a consistent manner. By leveraging the contextual entities derived from the given document and the algorithms of collective disambiguation based on Markov chains, our approach achieves promising results on various types of input mentions. Through the extensive experiments conducted on 9 different datasets, we show that our approach outperforms 14 state-of-the-art methods in most cases. The experimental results also show the limitation of our approach for short text with very ambiguous mentions. In future work, we would like to incorporate other contexts extracted from external resources into the collective disambiguation to address the challenges of ambiguous mentions in short text.

IX. Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 611346.

References

[1] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, "Dbpedia spotlight: shedding light on the web of documents," in *I-SEMANTICS*, 2011, pp. 1–8.

[2] X. Han, L. Sun, and J. Zhao, "Collective entity linking in web text: a graph-based method," in *SIGIR*, 2011, pp. 765–774.

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

[4] M. van Erp, G. Rizzo, and R. Troncy, "Learning with the web: Spotting named entities on the intersection of NERD and machine learning," in *#MSM*, 2013, pp. 27–30.

[5] R. Usbeck, A. N. Ngomo, M. Röder, D. Gerber, S. A. Coelho, S. Auer, and A. Both, "AGDISTIS - graph-based disambiguation of named entities using linked data," in *ISWC*, 2014, pp. 457–471.

[6] G. Rizzo, M. van Erp, and R. Troncy, "Benchmarking the extraction and disambiguation of named entities on the semantic web," in *LREC*, 2014, pp. 4593–4600.

[7] F. Piccinno and P. Ferragina, "From tagme to WAT: a new entity annotator," in *ERD@SIGIR*, 2014, pp. 55–62.

[8] D. N. Milne and I. H. Witten, "Learning to link with wikipedia," in *CIKM*, 2008, pp. 509–518.

[9] J. R. Finkel, T. Grenager, and C. D. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *ACL*, 2005.

[10] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, "Robust disambiguation of named entities in text," in *EMNLP*, 2011, pp. 782–792.

[11] W. Shen, J. Wang, P. Luo, and M. Wang, "LINDEN: linking named entities with knowledge base via semantic knowledge," in *WWW*, 2012, pp. 449–458. [Online]. Available: http://doi.acm.org/10.1145/2187836.2187898

[12] R. Navigli, "Word sense disambiguation: A survey," *ACM Comput. Surv.*, vol. 41, no. 2, 2009.

[13] ——, "A quick tour of word sense disambiguation, induction and related approaches," in *SOFSEM*, 2012, pp. 115–129.

[14] R. Mihalcea and A. Csomai, "Wikify!: linking documents to encyclopedic knowledge," in *CIKM*, 2007, pp. 233–242.

[15] X. Cheng and D. Roth, "Relational inference for wikification," in *EMNLP*, 2013, pp. 1787–1796.

[16] R. Usbeck, M. Röder, A. N. Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann, P. Ferragina, C. Lemke, A. Moro, R. Navigli, F. Piccinno, G. Rizzo, H. Sack, R. Speck, R. Troncy, J. Waitelonis, and L. Wesemann, "GERBIL: general entity annotator benchmarking framework," in *WWW*, 2015, pp. 1133–1143.

[17] E. Agirre and A. Soroa, "Personalizing pagerank for word sense disambiguation," in *EACL*, 2009, pp. 33–41.

[18] L. Zhang, M. Färber, and A. Rettinger, "xlid-lexica: Cross-lingual linked data lexica," in *LREC*, 2014, pp. 2101–2105.

[19] L. Zhang, A. Rettinger, and S. Thoma, "Bridging the gap between cross-lingual nlp and dbpedia by exploiting wikipedia," in *NLP&DBpedia*, 2014, Inproceedings.

[20] I. Witten and D. Milne, "An effective, low-cost measure of semantic relatedness obtained from wikipedia links," in *WIKIAI*, 2008, pp. 25–30.

[21] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972.

[22] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 30, no. 1-7, pp. 107–117, 1998.

[23] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[24] S. White and P. Smyth, "Algorithms for estimating relative importance in networks," in *KDD*, 2003, pp. 266–275.

[25] E. Seneta, *Non-negative matrices and Markov chains; rev. version*, ser. Springer series in statistics. New York, NY: Springer, 2006.

[26] T. H. Haveliwala, "Topic-sensitive pagerank," in *WWW*, 2002, pp. 517–526.

[27] G. Jeh and J. Widom, "Scaling personalized web search," in *WWW*, 2003, pp. 271–279.

[28] L. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to wikipedia," in *ACL*, 2011, pp. 1375–1384.

[29] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, "Collective annotation of wikipedia entities in web text," in *KDD*, 2009, pp. 457–466.

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

[31] M. Röder, R. Usbeck, S. Hellmann, D. Gerber, and A. Both, "N^3 - A collection of datasets for named entity recognition and disambiguation in the NLP interchange format," in *LREC*, 2014, pp. 3529–3533.

[32] N. Steinmetz and H. Sack, "Semantic multimedia information retrieval based on contextual descriptions," in *ESWC*, 2013, pp. 382–396.

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

[34] D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani, "Dexter: an open source framework for entity linking," in *ESAIR*, 2013, pp. 17–20.