EREL: an Entity Recognition and Linking algorithm
Cuong Duc Nguyen\(^a\) and Trong Hai Duong\(^b\)

\(^a\)Faculty of Information Technology, Ton Duc Thang University, HoChiMinh city, Vietnam; \(^b\)Institute of 4.0, Nguyen Tat Thanh University, HoChiMinh City, Vietnam

**ABSTRACT**
This paper introduces the EREL algorithm that integrates Entity Recognition, Co-reference Resolution (CR) and Disambiguation. The algorithm recognizes entity mentions as the longest name based on the name dictionary constructed from the Wikipedia data. The CR is integrated into the algorithm to improve the performance in processing short-form or abbreviated names. The algorithm employs a new approach in disambiguation entities using new features as entity-level context information and case-sensitive data about the mention in disambiguation. Tested on four benchmark data sets in the GERBIL framework, EREL outperforms the current Entity Linking methods. EREL achieves the micro f-score as 0.83 in both tasks Disambiguate to Wikipedia and Annotate to Wikipedia.

**ARTICLE HISTORY**
Received 5 June 2017
Accepted 24 August 2017

**KEYWORDS**
Entity Recognition; Entity Linking; Co-reference Resolution; information retrieval; text analysis; Disambiguation

---

1. Introduction

Recognizing entity mentions in a text and linking them to entities in a knowledge base are two fundamental tasks in text analysis. In the knowledge extraction pipeline, a Named Entity Recognition (NER) system is often used to recognize mentions of named entities in text and then an Entity Linking (EL) system is executed to link recognized mentions to entities in a knowledge base like Wikipedia (Rao, McNamee, & Dredze, 2013). Because NER systems focus on identifying named entities, such as people, organizations and locations, EL is often considered as only linking named entities and not able processing nominal entities (Moro, Raganato, & Navigli, 2014).

Instead of using lexical and semantic features to identify the boundary of an entity mention as NER, Milne and Witten (2008) introduced the Link Detection problem, in which a short-and-meaningful sequence of terms is identified as a relevant mention of an entity if there is a highly statistical relation between the term sequence and the mention. In this problem, the identification of mentions is driven by the knowledge base, in which an entity can be a named entity or a nominal entity. Compared to using NER to identify entities, Link Detection can recognize named entities, common nouns and other entities as adjectives or gerunds.

---

**CONTACT**
Cuong Duc Nguyen
nguyenduccuong@tdt.edu.vn

© 2018 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
This paper introduces the EREL algorithm that integrates Entity Recognition (ER), Co-reference Resolution (CR) and Disambiguation. The ER follows the Link Detection approach, in which EREL searches for the longest Wikipedia entity name in the text as mentions. The CR step is applied to make co-references from short-name mentions and abbreviated names to the full-name entities to improve the linking performance these co-referenced entities. The Entity Disambiguation step applies the entity-level context information and other features to find most appropriate entity for each detected mention. The last two stages in EREL apply new techniques compared with other algorithms in EL.

The rest of the paper is organized as follows: Section 2 reviews related works. Section 3 presents the proposed EREL algorithm. The experimental result is shown in Section 4. Section 5 is the discussion. Finally, Section 6 makes a conclusion.

2. Related works

Nadeau and Sekine (2007) and Marrero, Urbano, Sánchez-Cuadrado, Morato, and Gómez-Berbis (2013) provide comprehensive surveys about NER. The surveys show that existing studies in NER focus on named entities, such as persons, organizations and places. The lack of using an external knowledge base in NER systems is one of the important observations on NER systems.

Shen, Wang, and Han (2015) provide a deep review of EL with a knowledge base. In the description of EL, this review considers that the EL task only works on named entities (proper nouns). The review also states that the EL system often uses a preceded NER system, which focuses on recognizing named entities, to discover entity mentions.

There are three main approaches in EL:

- Collective EL: using an assumption about the single topic of a document, algorithms in this approach facilitate the common topical coherence between the analysed and reference documents to find the highest ranked entity as the linked entity.
- Graph-based Collective EL: methods in this approach use graphs to model documents and then compare the context similarity between the analysed document and the reference document.
- Vector-space model: techniques in this approach construct a vector-space model of context information to disambiguate linkable entities to a mention.

Collective EL uses topical coherence between the analysed document and the reference document to find the highest ranked entity as the linked entity. Cucerzan (2007) exploits the agreement between categories of the input document and the reference document. Kulkarni, Singh, Ramakrishnan, and Chakrabarti (2009) utilize local hill-climbing, rounding integer linear programs and local optimization within pre-clustering entities to maximize the global coherence between entities. Ratinov, Roth, Downey, and Anderson (2011) apply local and global algorithms to optimize the topical coherence function. Guo, Chang, and Kiciman (2013) utilize the structural Support Vector Machine (SVM) algorithm on tweet data with several features as the bag of words from the surrounding text, the entity popularity, the entity type, the top-k tf-idf words, the Jaccard distance between Wikipedia pages, etc. LINDEN (Marrero et al., 2013) uses the SVM algorithm to learn a
ranking function with four features: entity popularity, semantic associativity, semantic similarity and global topical coherence between mapping entities.

Wikipedia Miner (Milne & Witten, 2008) consists of two main functions: ‘Learning to detect links’ and ‘Learning to disambiguation links’. In ‘Learning to detect links’, Wikipedia Miner applies the machine-learning algorithm, including Naïve Bayes, C4.5, SVM and Bagged C4.5, to decide which term in the text can be linked to a Wikipedia entity. Features of a term are extracted from the surrounding context, including its link probability, relatedness, disambiguation confidence, generality, location and spread. Terms classified by the learning algorithms are considered as entity mentions, so that only mentions with a high probability to be able linked to Wikipedia entities are recognized. In ‘Learning to disambiguation links’, Wikipedia Miner also applies mentioned classification algorithms to classify which Wikipedia entity can be linked to a link detected by ‘Learning to detect links’. Features for the classification are extracted from the surrounding context of a detected link, consisting of commonness, relatedness and context quality of that link.

Tagme (Ferragina & Scaiella, 2010) focusses on EL for short texts, such as snippets of search-engine results, tweets and news. Tagme detects text anchors (entity mentions) by using a pre-construct anchor dictionary and identifies anchor boundaries by the anchor probability. Tagme disambiguates text anchors by using a disambiguation scoring function, which considers features as the collective agreement between anchors, the average relatedness between linkable entities, the significance of a linkable entity and the sum of votes given by other anchors. The scoring function is used by a classifier (C4.5, Bagged C4.5 or SVM) or a threshold filter to identify which entities can be linked to text anchor. After the entity disambiguation, Tagme uses a pruner to discard ‘bad’ anchors. The pruner takes into account only two features: the link probability of a text anchor and the coherence of its candidate entity.

WAT (Piccinno & Ferragina, 2014) enhances Tagme in entity disambiguation with voting, graph-based and optimization technologies. WAT applies a voting algorithm to select the most appropriate entity for an anchor from the output of entity disambiguation from several classifiers. WAT develops a Mention-Entity graph in which mention nodes are linked to a set of candidate entities. An edge of the graph is weighted by (i) identity (always 1), (ii) commonness or (iii) context similarity. The entity disambiguation applies graph analysis algorithms, such as PageRank, Personalized PageRank, Hypertext-Induced Topic Search (HITS) and SALSA, on the developed graph to select a suitable entity for a mention. WAT also uses an optimizer to perform a second disambiguation pass using the base-nt method that favours the entity voted by the ‘right’ majority of other linkable entity. Finally, WAT also applies a binary classification algorithm, the SVM method, to filter useless annotation.

Graph-based Collective EL uses graphs to model documents and develops an algorithm traversing on the constructed graph to find linked entities. Han, Sun, and Zhao (2011) use the Referent Graph to represent mentions and entities and introduce the collective inference algorithm to traverse on the represented graph to find the linked entity. Hoffart et al. (2011) build a weighted graph of mentions and candidate entities and compute a dense subgraph that approximates the best joint mention-entity mapping. Babelfy (Guo & Barbosa, 2014) constructs a graph-based semantic interpretation of the input document and develops an algorithm to identify densest subgraphs.
AGDISTIS (Usbeck et al., 2014) applies a three-phase process: NER, candidate choosing and entity disambiguation. In NER, AGDISTIS constructs a set of entity surface forms from DBpedia and then applies string normalization and expansion policy to identify entity mentions from text documents. AGDISTIS chooses entity mentions by using a trigram similarity which is an n-gram similarity with n = 3. In entity disambiguation, AGDISTIS constructs a disambiguation graph from the knowledge base. AGDISTIS employs the HITS algorithm on the disambiguation graph to calculate authoritative values and hub values. AGDISTIS uses calculated values to identify the correct entity for an entity mention.

DBpedia Spotlight (Mendes, Jakob, García-Silva, & Bizer, 2011) uses a cector space model of context information to disambiguate linkable entities to a mention. Each entity is represented by a vector of words, which are weighted by the Term Frequency (TF) and the proposed Inverse Candidate Frequency values. The cosine similarity measurement between vectors of the reference document and the current document is used to rank linkable entities. The highest ranked linkable entity is selected as the linked entity.

Some researchers propose the combination of NER and EL in one task. Pu, Hassanzadeh, Drake, and Miller (2010) introduce a system that executes online ER and EL on text streams, and demands a continuous extraction phrase and a linking to top-k relevant entities. Guo et al. (2013) propose a method that focuses the combined NER and EL task on unstructured texts as tweets. NEREL (Sil & Yates, 2013) uses a maximum-entropy model with the vectors of lexical, links and topical features. KODA (Mrabet, Gardent, Foulonneau, Simperl, & Ras, 2015) uses a statistical parser to find noun phrases, selects the noun phrase with the highest TF-IDF as an entity mention and then uses a co-occurrence maximization process to select the linked entity.

Traditional EL methods require a given set of mentions to link them to entities, but several aforementioned EL algorithms can also recognize mentions of entities. DBpedia Spotlight searches the input text for mentions that are extracted from Wikipedia anchors, titles and redirects, by using the LingPipe Exact Dictionary-Based Chunker. TagMe and WAT search the input text for mentions that are also defined by the set of Wikipedia page titles, anchors and redirects. Wikipedia Miner applies a machine-learning approach to detect entity mentions.

CR is the task of identifying mentions that refer to the same real-world entity existing in the same document. CR often uses the set of entity mentions from NER systems as its input. CR is also used as a pre-processing and independent tool of other semantic techniques. For example, in LinkPeople (Garcia & Gamallo, 2014), the input text is pre-processed by CR before applied Open Information Extraction techniques.

Existing CR techniques focus on named entities, such as human names and organization names (Clark & Manning, 2015; Garcia & Gamallo, 2014; Hajishirzi, Zilles, Weld, & Zettlemoyer, 2013; Stoyanov & Eisner, 2012). There are two main issues in existing CR techniques:

- Reference to an ambiguous entity: CR methods assume a referenced mention linking to a single person (Garcia & Gamallo, 2014) or a single named entity (Clark & Manning, 2015; Raghunathan et al., 2010) and then facilitate the entity’s information to select the suitable co-reference. Real-world entities are often ambiguous so that this assumption needs to be seriously reconsidered.
Using ambiguous data to solve ambiguous reference: CR techniques collect other text information and other entity mentions in the same document to select the most appropriate reference.

This paper introduces the EREL algorithm that discovers and links entity mentions to entities in Wikipedia. Instead of using lexical rules to recognize entity mentions as NER, EREL applies the Link Detection approach by searching entity mentions as the longest Wikipedia entity names. EREL uses CR techniques on the entity mentions to discover co-referenced entities. Finally, EREL applies a newly developed disambiguation technique to link recognized mentions to entities in Wikipedia.

3. The EREL algorithm

The EREL algorithm has four main steps as in Figure 1. Step 1 – Document Structure Analysing examines the document structure and splits the document into paragraphs. Step 2 – Entity Mention Scanning skims over a paragraph text to identify entity mentions that are able to link to entities in Wikipedia. Step 3 – Co-reference Resolution discovers co-referencing entities that can refer to an aforementioned entity. Step 4 – Disambiguation ranks linkable entities of a mention and selects the most appropriate entity as the linked entity of that mention. The final set of ‘mention-entity’ links is return as the result of the algorithm.

The content of EREL is shown in Figure 2. The four main steps of EREL are described in the following sections.

3.1. Step 1 – Document Structure Analysis

Step 1 analyses the document structure to find the main elements of the document, such as title, authors, place and source. In the context of available data sets, almost all documents are in the form of news or simple web pages. The exemplary structure of a news
The EREL algorithm:
Input: A text document
Output: A set of (mention, entity)
Main content:

**Step 1 - Document Structure Analyzing:**
- Determine document header parts and the starting point of the main content
- Split the document text into paragraphs

**Step 2 - Entity Mention Scanning:**
For each paragraph:
- Split the paragraph text into words and assign a Part-Of-Speech tag for each word
- For each length \( l \) from \( \text{max\_length} \) down to 1: Scanning from beginning to the end of the paragraph text:
  - Form the *Temporary Mention Candidate Set (TMCS)* as the combination of \( l \) continuous uncovered words
  - Query the existing of each mention candidate of TMCS on the Knowledge Base (KB)
  - If a candidate exists in KB,
    - Mark “covered” for all words of the mention candidate
  - Add the candidate to *Mention Candidate Set (MCS)*
- If length \( l \) equals to 1 and no mention candidate of TMCS exists in the KB
  - Add the original mention candidate to the *Unrecognized Mention Candidate Set (UMCS)*

**Step 3 - Co-reference Resolution (CR):**
CR for abbreviated terms
- For each entity \( e_i \) in MCS or UMCS:
  - If there is entity \( e_j \) in MCS where \( j < i \), \( e_i \) and \( e_j \) not in the header part, \( e_i \) is the abbreviated name of \( e_j \); \( e_i \) is co-referencing entity of \( e_j \)
  - If there is entity \( e_j \) where \( e_i \) is in the header part, \( e_j \) not in the header part of the document, \( e_i \) is the abbreviated name of \( e_j \); \( e_i \) is co-referencing entity of \( e_j \)

CR for short-name entities
- For each entity \( e_i \) in MCS or UMCS:
  - If there is entity \( e_j \) where \( j < i \), \( e_i \) and \( e_j \) not in the header part, POS-tag of \( e_i \) is NNP or NNPS, the mention of \( e_i \) is a part of the mention of \( e_j \); \( e_i \) is co-referencing entity of \( e_j \)
  - If there is entity \( e_j \) where \( e_i \) is in the header part, \( e_j \) not in the header part of the document, POS-tag of \( e_i \) is NNP or NNPS, the mention of \( e_i \) is a part of the mention of \( e_j \); \( e_i \) is co-referencing entity of \( e_j \)

**Step 4 – Entity Disambiguation:**
- For each entity candidate \( e_i \) in both MCS and the header part:
  - Filter the set of linkable entities by the entity type
- For each entity candidate \( e_i \) in MCS:
  - If \( e_i \) is not a co-referencing entity:
    - Calculate the linked measure of each linkable entity
    - Select the entity with the highest linked measure as the linked entity of \( e_i \)

---

**Figure 2.** The EREL algorithm.

document is shown in Figure 3. The structure of a web document is often known when the document is crawled from the Internet.

The Document Structure Analysis brings two advantages for EREL. Firstly, an entity mention in the Title part of a news document is often in the short form of an entity in the main content. For instance, entity mention ‘Clinton’ in a title ‘Clinton …’ is the shortening form of entity ‘Bill Clinton’ in the main content of the document. This writing style of ‘mention first, explain later in main content’ is often used to draw the attention of readers.
in news documents. The entity disambiguation on ‘Bill Clinton’ is obviously easier than that on ‘Clinton’, so that, if ‘Clinton’ can be co-referenced to ‘Bill Clinton’ in the content, entity mention ‘Clinton’ can be linked to the same entity with ‘Bill Clinton’. Therefore, entity mentions recognized on the document’s title, appear earlier in the document, will be co-referenced to entity mentions recognized in the main content, which locates later in the document, by Step 3 – Co-reference. Secondly, from recognized elements, the type of corresponding entity candidates can be specified. The entity type is used in Step 3 to filter the set of linkable entities of a mention. For instance, ‘AFP’ can be linked to 24 Wikipedia entities but if it is a news agency, ‘AFP’ can be only linked to ‘Agence France-Presse’. The entity-type filtering reduces the set of linkable entities for an entity mention by using domain knowledge, so that the performance of Step 4 – Entity Disambiguation can be improved.

3.2. Step 2 – Entity Mention Scanning

In Step 2, the algorithm uses the Stanford POS-Tagger (Toutanova, Klein, Manning, & Singer, 2003) to split the paragraph text into sentences and assigns a Part-Of-Speech tag for each word in a sentence. The algorithm repeatedly scans each sentence from beginning to the end by a string with \( l \) uncovered words. Length \( l \) is consequently decreased from max_length to 1. In this paper, parameter max_length is selected as 10 to cover the longest mention in Wikipedia.

With a length \( l \), if the last word is a noun in the plural form, its singular and plural forms are joined with the previous \((l – 1)\) words to create mention candidates that are stored in the Temporary Mention Candidate Set (TMCS). Otherwise, \( l \) words are jointed to create one mention candidate in TMCS.

The algorithm consequently searches each mention candidate in TMCS in the Wikipedia. If there is a mention candidate existing in the dictionary of mentions and satisfying the pre-defined case-sensitive constraint, the algorithm considers that mention candidate as a good mention candidate and stops searching candidates in TMCS. The algorithm marks label ‘covered’ for all words in the matching mention candidate and adds the matching mention candidate to the Mention Candidate Set (MCS). Each candidate in MCS consists of the recognized mention and links to at least one entity in Wikipedia.

In the pre-defined case-sensitive constraint, if a mention candidate is started with a lower-case word, there must be at least one linkable entity that has its case-sensitive name in the lower-case form. This constraint prevents the algorithm from matching
words in the text with entities as movies, songs or novels, which are freely named from common nouns, such as ‘Adaptation’, ‘Across the Bridge’ and ‘After Many Days’.

In EREL, each entity has two names: Wikipedia and case-sensitive names. The Wikipedia name of an entity always has the first character in the upper-case form and is unique. In addition, the Wikipedia name of an entity can have some explaining part that is put in the round bracket, such as ‘Alice (novel)’. The case-sensitive name of a Wikipedia entity is the mention that should be appeared in a text. For instance, in a document, the case-sensitive names of ‘Apple’ and ‘Alice (novel)’ are ‘apple’ and ‘Alice’, respectively. The case-sensitive name of an entity is constructed from the Wikipedia name and its lower-/upper-case form is determined by scanning the content of the reference Wikipedia document.

The dictionary of mentions is constructed in advance. For each entity, its Wikipedia name, re-directed name and case-sensitive name are added to the dictionary. EREL does not use the anchor text, the surface name of an outbound URL link in a reference document, as the mention of an entity as in several other papers.

When \( l \) equals to 1, EREL only considers words as nouns or adjectives (specifying by the assigned Part-Of-Speech tag) to add to \( TMCS \). If all mention candidates in \( TMCS \) do not exist in the KB, the algorithm adds the mention candidate to the Unrecognized Mention Candidate Set (UMCS), which consists of mention candidates unable linking to any entity in Wikipedia. With more than 4 million entities, Wikipedia covers almost of English words and human names, so that an unrecognized mention is likely a new entity name or an abbreviated word.

3.3. Step 3 – Co-reference

CR, which identifies the references between entities in the same document, has not been integrated to EL. In a document, an entity is often in the full-name form, such as ‘Cameron Diaz’, in the first occurrence and in the shortening form, such as ‘Diaz’ in later occurrences in the same document. In general, the disambiguation of entity ‘Diaz’ is much difficult than that of entity ‘Cameron Diaz’, so that EL algorithms can use co-reference techniques to reference ‘Diaz’ to ‘Cameron Diaz’ and then only disambiguate entity ‘Cameron Diaz’. The using of co-reference techniques can increase the performance of EL algorithms in the case of shortening name as ‘Diaz’.

In Step 3 – Co-reference Resolution, the EREL algorithm uses two types of CR: CR for abbreviated terms and CR for short-name entities. In EREL, CR techniques only work on named entities.

In CR for abbreviated terms, self-defined abbreviated terms inside the document are linked to a named entity with the full name. A self-defined abbreviated term of a named entity is created by extracting the first capitalized character of each word in the whole term. For instance, in the MSNBC data set, with entity ‘Institute for Supply Management’, which is recognized in \( MCS \), its abbreviated term ‘ISM’ is created by extracting the first capitalized character of each word in the whole term. If entity candidate ‘ISM’ exists in the \( UMCS \), it references to entity ‘Institute for Supply Management’ in \( MCS \).

In CR for short-name entities, each candidate of \( UMCS \) and \( MCS \) is tried to refer to a candidate in \( MCS \), which is previously mentioned in the input document. In this CR, each mention candidate is checked whether it is the short name of a candidate in \( MCS \). A candidate is the short name of another entity if both entities are named entities and the entity
mention of the first candidate is a part of the entity mention of the second candidate. This type of CR is widely used in news document or Wikipedia document about persons. For example, in the Wikipedia document of entity ‘Bill Clinton’, the whole name is mentioned in the first paragraph and entity mention ‘Clinton’ is used more than 500 times later in the text.

3.4. Entity disambiguation

The last task in the algorithm is the Disambiguation of candidates in MCS. So far, each candidate in MCS has a set of linkable entities in KB. There are two sub-tasks in Disambiguation: Disambiguation by entity type and Disambiguation by ranking.

In Disambiguation by entity type, if the candidate type is specified in analysing the document structure, the set of linkable entities is filtered by the candidate type stored in the KB. For example, in the ACE data set, candidate ‘AFP’ relates to 24 entities in Wikipedia. If its entity type is a news agency, the set of related entities is filtered to one entity ‘Agence France-Presse’. The list of entities for some main entity types can be extracted from Wikipedia. For example, the list of news agencies is retrieved at address https://en.wikipedia.org/wiki/Category:News_agencies. The list of cities is retrieved at address https://en.wikipedia.org/wiki/Category:City.

In Disambiguation by ranking, for each candidate in MCS, EREL ranks linkable entities of that candidate and selects the entity with the highest linked measurement as the linked entity. The ranking function between a mention \( m \) and a linkable entity \( e \) is shown in Figure 4. The linked measurement is the total of three main parts: the co-existing entity measurement \( M_e \), the case-sensitive measurement \( M_c \) and the entity significance \( M_s \). These parts will be explained in the following paragraphs.

The co-existing entity measurement \( M_e \) shows the appropriateness of a linkable entity to a mention candidate by the co-existing entities in the same document. In a document, co-existing entities ‘iPhone 4S’ and ‘iPhone 5’ indicate that entity ‘Apple Inc.’ is the most appropriate entity to link to mention ‘Apple’. The co-existing entity indicator is the number of jointed entities between the set of co-existing entities of the document and the set of indicated entities of each linkable entity. For the analysed document, the set of co-existing entities is formed by the all linkable entities of all mention candidates in MCS so far in the Disambiguation process. It is worth to mention that the set of co-existing entities

\[
M(m, e) = M_e + M_c + M_s
\]

With

- \( M_e \) = the standardized number of jointed entity between the co-existing entity set of mention \( m \) and the indicated entity set of linked entity \( e \)
- \( M_c \) = 1.0 if both the mention and the case-sensitive name of the linkable entity are in the same case
- \( M_s \) = the significance of linked entity \( e \)

Figure 4. The ranking function for Disambiguation.
entities is continuously reduced during the Disambiguation process. For a linkable entity, the set of indicated entities is formed by the set of out-linked entities in the reference document. For each mention candidate in MCS, the co-existing entity indicator is standardized by the set of linkable entities.

The case-sensitive measurement $M_c$ describes the relatedness on the case sensitivity between the mention of a candidate and the real name of a related entity. The case-sensitive measurement is the bonus on the linkable entity having the real name that has the same case with the mention. After collecting the bonus of all potential linked entities, $M_c$ is also standardized.

The entity significance $M_s$ is pre-calculated value and stored in DB. Because of the lack of data corpus that annotated on the entity level, the prior probability of an entity with known its mention cannot be calculated. For example, with entity mention ‘Apple’, the conditional probabilities $p(\text{Apple Inc.} | \text{Apple})$ or $p(\text{Apple Bank} | \text{Apple})$ cannot be rightly calculated by available data corpus. The paper proposes to use the entity significance that is defined as ‘length of the reference document of that entity’. If an entity is popular, its reference document is likely to be long and have several facts. Thus, the significance is somehow corresponding with the conditional probability. After counting the length of the reference document, the entity significance is lightly adjusted. The significance of the entity that has the identical name with the entity mention is modified to the max significance value. The entity significance is standardized on the set of related entities for each mention candidate in MCS.

The complexity of EREL is estimated based on the number of queries on the KB in Step 2 – Scanning entity and the number of entities in Step 3 – Co-reference Resolution. With $doc_length$ being the number of words in the text document and $number\_of\_named\_entities$ being the number of entities in MCS, EREL has a complexity as $O(max\_length * doc\_length + number\_of\_named\_entities^2)$. In the CR step, EREL only works on named entities, which appear about 5–7 times per document, the complexity can be considered as linear with $doc\_length$.

4. Experiments

4.1. Setting up

The experiments are executed in the GERBIL (Usbeck et al., 2015) framework, the upgraded version of the BAT-framework (Cornolti, Ferragina, & Ciaramita, 2013). Because of the reimplementation of EL algorithms is quite difficult due to the complexity of the task and the advanced data preparation, GERBIL is developed as an independent evaluation framework for semantic entity annotation. In GERBIL, an EL algorithm is implemented by its authors and provided as a Web service for other programs. For each sentence, GERBIL sends a request including the text of that sentence and a list of corresponding mentions to the provided service that returns the linking results back to GERBIL. To evaluate an EL algorithm on a benchmarking data set, GERBIL sends a list of requests to the provided Web service of that EL algorithm, receives the linking results, compares received linking results to the expected values in the data set and then calculates performance indicators, such as precision, recall or f-score. GERBIL totally controls the loading of the tested data sets and the expected results. GERBIL does not allow the tested EL algorithm access to
the expected results. With such architecture, the separation between the accessing to tested data sets and the execution of EL algorithms provides a fair comparison of the linking performances. GERBIL has an online site\(^1\) to execute experiments on EL tasks with different benchmarking data sets.

In GERBIL, the configuration of each EL algorithm is set up by its authors to achieve best performances on benchmarking data sets. GERBIL does not provide an explicit mechanism to change the configuration of EL algorithms.

Major EL methods and benchmarking data sets are available in this framework. GERBIL provides its source codes and benchmarking data sets in the GitHub repository.\(^2\) In the experiments in this section, EREL is compared with four EL algorithms, AGDISTIS, DBpedia Spotlight, WAT and Wikipedia Miner.

Since the beginning of November 2015, the GERBIL framework is majorly upgraded from version 1.1.4 to version 1.2.0, and then 1.2.1. The current version of GERBIL (1.2.1) uses URI matching and is a lack of features that are often used in the popular EL testing. We set up a local GERBIL 1.2.1 with some self-developed functions that provide the normal testing environment for EL, similar as GERBIL version 1.1.4, such as:

- Annotations with a chosen Wikipedia entity as ‘null’ or an empty string will be removed.
- Annotations with a chosen Wikipedia entity that does not exist in the current DBpedia corpus will be removed.
- Annotations with a chosen Wikipedia entity that is a disambiguation page will be removed.
- An annotation with a chosen Wikipedia entity that is a redirecting entity will be replaced by the re-directed entity.

Regarding the approaches in EL, Cornolti et al. (2013) proposed a classification of entity-annotation systems. In six classified systems of the BAT-framework, Disambiguate to Wikipedia (D2W) is the traditional EL problem and Annotate to Wikipedia (A2W) is the combined problem of Link Detection and EL system. In GERBIL, D2W and A2W are called D2KB and A2KB, respectively.

All tested algorithms are evaluated on four benchmarking data sets ACE 2004, AQUAINT, MSNBC and IITB. These benchmarking data sets are available in the GERBIL framework on GitHub.\(^3\) Table 1 shows the statistical information on entities of these four data sets.

The Wikipedia dump (version December 8, 2014) is processed and stored in an MS SQL server as the KB for EREL. From KB, EREL constructs the dictionary of mentions that was previously described in Section 3.2. The co-existing entity measurement \(M_e\) and the

| Data Set   | Number of documents | Document type | Number of annotations on a document |
|------------|---------------------|---------------|-------------------------------------|
| ACE 2004   | 57                  | News          | 4.44                                |
| AQUAINT    | 50                  | News          | 14.54                               |
| MSNBC     | 20                  | News          | 32.50                               |
| IITB       | 103                 | Web page      | 109.22                              |

Table 1. Statistical information on entities of benchmark data sets.
entity significance $M_s$, which are previously defined in Section 3.4, are pre-calculated based on KB.

### 4.2. Testing on the D2W task

In the D2W task, a tested annotator is given a set of mentions on a text by GERBIL and after linking, it gives the annotated results back to GERBIL. To test with the D2W task, EREL is lightly modified as follows:

- The given set of mentions is used to initialize the set of entity mentions. Step 2 only scans on the remaining uncovered text segments to recognize mentions.
- For each entity created by the given mention, EREL searches for a full mention, a part of mention, an expanded mention up to three words to the left and right of the given mention.
- Step 4 is only applied on the set of entities created by the given mentions to speed up the EL process.

The experiment results on task D2W are shown in Table 2. The EREL algorithm does not require any input parameter so that it only provides one micro f-score value per data set. The traditional precision–recall curve is not available for EREL. Due to the diversified characteristics of tested data sets, the micro f-score is statistically enough to show the algorithm performance and no other test is needed.

In Table 2, EREL achieves the best result on all four benchmark data sets and also the best average result. With the data sets having high numbers of annotations on a document as MSNBC and IITB, the performance of EREL is significantly better than those of all other tested algorithms. On the IITB data set, EREL achieves a high result as 0.8235, which is nearly 0.20 higher than the performances of other tested algorithms and higher than the previously reported highest result as 0.71 on this data set (Shen et al., 2015).

In the D2W task, the GERBIL framework provides an entity mention for annotators and expects to receive one corrected linking result. If with each provided entity mention, an annotator returns exactly one linking result, its recall value should be equal to the precision value. If an annotator is not confident about its linking result on a provided entity mention and then refuses to return a linking result, so that the number of return linking results is smaller than the number of provided mentions sent by GERBIL. The refusal of returning linking results of an annotator increases its precision value but decrease its recall value.

AGDISTIS applies a recognition step to identify named entity in provided mentions so that it cannot process common-noun entities. Thus, it has poor performances on ACQUAINT and IITB, in which there are several common-noun entities.

DBpedia Spotlight finds entity mention candidates by applying a string matching algorithm with the longest case-insensitive match and then uses a candidate selection phase to reduce the number of candidates. In entity disambiguation, DBpedia Spotlight uses a disambiguation confidence as 0.7 to filter incorrect annotations. This filtering makes its precision much higher than its recall.

WAT applies a pruner to filter useless annotations by using a pre-defined threshold and a binary SVM algorithm. This pruning also makes its precision much higher than its recall.
| Algorithm    | ACE2004 Micro Prec. | ACE2004 Micro Recall | ACE2004 Micro F1 | AQUAINT Micro Prec. | AQUAINT Micro Recall | AQUAINT Micro F1 | MSNBC Micro Prec. | MSNBC Micro Recall | MSNBC Micro F1 | IITB Micro Prec. | IITB Micro Recall | IITB Micro F1 | Average Micro F1 |
|--------------|---------------------|----------------------|------------------|---------------------|---------------------|---------------------|-------------------|-------------------|------------------|------------------|------------------|------------------|-----------------|
| AGDISTIS     | 0.6652              | 0.6507               | 0.6578           | 0.5192              | 0.5075              | 0.5133              | 0.7325            | 0.7235            | 0.728            | 0.3435           | 0.3395           | 0.3415           | 0.5602           |
| DBpedia Spotlight | 0.8824            | 0.7860               | 0.8314           | 0.8325              | 0.7538              | 0.7912              | 0.7753            | 0.6467            | 0.7052           | 0.6965           | 0.6548           | 0.6702           | 0.7495           |
| WAT          | 0.8529              | 0.7598               | 0.8037           | 0.8095              | 0.7147              | 0.7592              | 0.8636            | 0.7296            | 0.791            | 0.7291           | 0.6542           | 0.6896           | 0.7609           |
| Wikipedia Miner | 0.7975            | 0.7470               | 0.7714           | 0.7865              | 0.7345              | 0.7596              | 0.7323            | 0.5723            | 0.6425           | 0.7261           | 0.4911           | 0.5859           | 0.6986           |
| EREL         | 0.8565              | 0.8341               | **0.8451**       | 0.8115              | 0.7823              | **0.7966**          | 0.8762            | 0.8694            | **0.8728**       | 0.8397           | 0.8079           | **0.8235**       | **0.8345**       |
Wikipedia Miner only accepts an entity mention with a high probability to be linked to an entity. With ACE2004 and ACQUAINT, Wikipedia Miner provides reasonable performance. However, with two remaining data sets, the performance of Wikipedia Miner is clearly decreased.

For each provided entity mention, EREL greedily searches for entity candidates by the longest Wikipedia name matching and return a linking result, so that its precision and recall are close together. Without candidate filtering, EREL recognizes the highest number entity candidates. However, the highest f-score of EREL shows the efficiency of its CR and disambiguation steps.

4.3. Testing on the A2W task

In the A2W task, a tested annotator is given a text by GERBIL and after annotating, it gives the annotated results back to GERBIL.

The current evaluation measurement in GERBIL is proposed by Cornolti et al. (2013), in which the number of false-positive cases is the number of returned annotations that are not existed in the benchmark data set. This evaluation does not properly measure the performance of annotators on the false-positive cases due to the characteristics of the benchmark data sets, such as:

- The annotation style: In ACE 2004 and ACQUAINT, only the first occurrence of an entity is annotated. In MSNBC and IITB, all occurrences of an entity are annotated. Thus, with ACE 2004 and ACQUAINT, annotations on the not-the-first-occurrence mention are counted as false-positive cases so this judgment is not consistent with MSNBC and IITB.
- The details of annotated information: In ACE 2004, ACQUAINT and MSNBC, only the most important entities are retained. IITB is the most detailed dataset since almost all mentions are annotated (Moro et al., 2014). The ‘importance’ of entities is difficult to define and in several documents in the selected benchmark data sets, the selection of annotations is not consistent. For instance, in file 20000715_AFP_ARB.0072.eng of ACE 2004, only entities ‘Iran’ and ‘Moscow’ are annotated but entities ‘Israel’ and ‘Palestinian’ are discarded, although all of them are important entities of the document. Counting annotations on ‘Israel’ and ‘Palestinian’ are false-positive are not right in this case.

We proposed the evaluation measurement on the false-positive case as:

a returned annotation is considered as false-positive if its mention is matched or overlapped with a mention of an expected annotation in the benchmark data sets but its annotated entity is different with the annotated entity of that expected annotation.

The proposed evaluation measurement only considers a resulted annotation of an annotator if and only if it is overlapped with an annotation provided by the benchmark data set. The proposed evaluation measurement is independent with the annotation style and the type of annotated entities in benchmark data sets. It also uses all available testing information provided by the benchmark data set.

With the implementation of the proposed evaluation measurement, the experiment results on task A2W are shown in Table 3. Similar to the results in task D2W, the EREL
algorithm achieves the best result on all four benchmark data sets and also the best average result.

In the A2W task, the GERBIL framework only provides text documents without entity mentions. Annotators have to recognize entity mentions and link them to Wikipedia entities. Because of its freedom, A2W requires annotators filter unnecessary candidates, identify correct borders for entity mentions and prune unnecessary annotations. It is worth to mention that annotators filter common-noun entities more often than named entities.

For data sets mainly consisting of named entities, such as ACE2004 and ACQUAINT, EREL has the lowest precisions due to the largest number of annotations creating by its greedy Wikipedia name matching. However, EREL achieves the highest recalls, up to 95–96%, so that, for correct recognized entity mentions, the entity disambiguation of EREL is quite efficient. Combining the precisions and recalls into f-score, EREL still achieves best results.

Data set MSNBC consists of both named and common-noun entities, EREL has an equivalent precision but its superior recall makes EREL achieve the best f-score result, which is much higher than those of other algorithms. Data set IITB has a large percentage of common-noun entities, so that the filtering mechanism of tested algorithms makes them have the very low recall. EREL maintains a very high recall as 0.8544 make them outperform other tested algorithm.

4.4. Evaluation the effect of CR

Integrating CR into EL is the most significant point of EREL. This section examines the effect of CR on the performance of EREL.

This experiment compares the performances of four EREL versions: EREL – the full algorithm, EREL1 – EREL without CR, EREL2 – EREL with ‘CR for abbreviated terms’ only and EREL3 – EREL with ‘CR for short-name entities’ only.

Table 4 shows the performances of four EREL versions in both D2W and A2W tasks. In four test data sets, the CR only makes a great effect on MSNBC due to its writing style. Specifically, in performances on MSNBC, CR makes EREL’s f-scores gain 0.12 and 0.09 in tasks D2W and A2W, respectively. In two types of CR, ‘CR for short-name entities’ produces a bigger gain than ‘CR for abbreviated terms’.

5. Discussion

The mention scanning mechanism of EREL is simple and efficient in recognizing the occurrence of entities in text. Long and complex mentions can be identified by the algorithm. The checking on mention acceptance facilitates the case-sensitive characteristics of entities to prevent the false-acceptance on movie, song or book names.

The CR step of EREL is very efficient in data sets MSNBC and IITB, where about 5% of annotations are resolved by Co-reference techniques.

The disambiguation of EREL uses entity-level context information, case-sensitive characteristic and significance in selecting the most appropriate linked entity for a mention. The significance of an entity is estimated by the length of the reference document with some adjustment on the default entity. Therefore, EREL facilitates three new features in disambiguation when compared with existing algorithms.
### Table 3. Performances (Micro Precision, Micro Recall and Micro F-measure) of the tested algorithms on four benchmark data sets in task A2W.

| Algorithm          | ACE2004 Micro Prec. | ACE2004 Micro Recall | ACE2004 F1 | AQUAINT Micro Prec. | AQUAINT Micro Recall | AQUAINT F1 | MSNBC Micro Prec. | MSNBC Micro Recall | MSNBC F1 | IITB Micro Prec. | IITB Micro Recall | IITB F1 | Average Micro F1 |
|--------------------|--------------------|----------------------|-------------|--------------------|----------------------|-------------|-------------------|-------------------|----------|----------------|-------------------|----------|------------------|
| DBpedia Spotlight  | 0.8482             | 0.802                | 0.8244      | 0.8244             | 0.5828               | 0.6828      | 0.798             | 0.5389            | 0.6433   | 0.7809         | 0.3362            | 0.57     | 0.6801           |
| WAT                | 0.8490             | 0.7951               | 0.8212      | 0.7899             | 0.4677               | 0.5875      | 0.8288            | 0.6928            | 0.7548   | 0.7422         | 0.1481            | 0.2469   | 0.6032           |
| Wikipedia Miner    | 0.7889             | 0.8294               | 0.7714      | 0.8015             | 0.7219               | 0.7596      | 0.8089            | 0.5329            | 0.6775   | 0.7632         | 0.4755            | 0.5859   | 0.7134           |
| EREL               | 0.7349             | 0.9683               | 0.8356      | 0.707              | 0.9503               | 0.8108      | 0.8018            | 0.9601            | 0.8739   | 0.8045         | 0.8544            | 0.8287   | 0.8373           |
Table 4. Performances of four EREL versions.

| Algorithm | ACE2004 | AQUAINT | MSNBC | IITB | Average |
|-----------|---------|---------|--------|------|---------|
|           | Micro Prec. | Micro Recall | Micro F1 | Micro Prec. | Micro Recall | Micro F1 | Micro Prec. | Micro Recall | Micro F1 | Micro Prec. | Micro Recall | Micro F1 | Micro Prec. | Micro Recall | Micro F1 | Micro F1 |
| (a) Task D2W |         |         |         |      |         |         |         |         |         |         |         |         |         |         |         |         |         |
| EREL1     | 0.8739  | 0.8472  | 0.8603  | 0.8159 | 0.7853  | 0.8003  | 0.7887  | 0.7281  | 0.8415  | 0.8135  | 0.8273  | 0.8084  |
| EREL2     | 0.8739  | 0.8472  | 0.8603  | 0.8159 | 0.7853  | 0.8003  | 0.7927  | 0.7343  | 0.7624  | 0.8408  | 0.8084  | 0.8243  | 0.8097  |
| EREL3     | 0.8565  | 0.8341  | 0.8451  | 0.8115 | 0.7823  | 0.7966  | 0.8727  | 0.8633  | 0.8680  | 0.8395  | 0.8076  | 0.8232  |
| EREL      | 0.8565  | 0.8341  | 0.8451  | 0.8115 | 0.7823  | 0.7966  | 0.8762  | 0.8694  | 0.8728  | 0.8397  | 0.8079  | 0.8235  | 0.8345  |
| (b) Task A2W |         |         |         |      |         |         |         |         |         |         |         |         |         |         |         |         |         |
| EREL1     | 0.7379  | 0.9632  | 0.8356  | 0.7108 | 0.9487  | 0.8127  | 0.7150  | 0.8659  | 0.7832  | 0.8075  | 0.8540  | 0.8301  | 0.8203  |
| EREL2     | 0.7349  | 0.9683  | 0.8356  | 0.7098 | 0.9505  | 0.8127  | 0.7115  | 0.9554  | 0.8156  | 0.8074  | 0.8549  | 0.8305  | 0.8325  |
| EREL3     | 0.7379  | 0.9632  | 0.8356  | 0.7080 | 0.9485  | 0.8108  | 0.8127  | 0.8801  | 0.8450  | 0.8042  | 0.8535  | 0.8281  | 0.8370  |
| EREL      | 0.7349  | 0.9683  | 0.8356  | 0.7070 | 0.9503  | 0.8108  | 0.8018  | 0.9601  | 0.8739  | 0.8045  | 0.8544  | 0.8287  | 0.8373  |
However, the disambiguation of EREL can be improved further. The entity-context information is calculated by the reference Wikipedia document that is sparsely annotated. The significance should be better estimated when an appropriate corpus is available.

6. Conclusion

The paper introduced the EREL algorithm that combines ER, CR and Disambiguation in one method. In the ER step, EREL searches for longest entity names existing in texts. The algorithm applied co-reference techniques on short-name and abbreviation. The disambiguation of the algorithm applied three new features as entity-level context information, case-sensitive characteristic and significance in selecting the most appropriate linked entity for a mention.

Tested on four benchmark data sets in the benchmarking Gerbil environment, EREL outperforms four EL methods with f-scores as 0.83 in both tasks D2W and A2W. These results are clearly better than the performance of current EL methods.

Notes

1. http://gerbil.aksw.org/gerbil/
2. https://github.com/dice-group/gerbil/wiki
3. https://github.com/dice-group/gerbil/wiki/Licences-for-datasets

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Cuong Duc Nguyen is currently a lecturer of Faculty of Information Technology, Ton Duc Thang University. He received Ph.D. degree from Cardiff University, United Kingdom. From 2007 to 2014, he is the Dean of School of Computer Science and Engineering, the International University - VNU HCMC, Vietnam. His research interests are Machine Learning, Data Mining, Search Engine and Information Retrieval.

Trong-Hai Duong is currently the Director of Institute Science and Technology of Industry 4.0. He received M.Sc. and Ph.D. degrees from the School of Computer and Information Engineering, Inha University, 2012. He received a Bachelor of Science with a certificate for the most excellent student at research from the Department of Informatics of Hue University, Vietnam, 2004. He received the Golden Globe awards and Innovative Youth badge in 2016. He was one of the 37 students of all Vietnam universities who won a scholarship certificate from the Ministry of Postal Service of Vietnam for the most excellent student of information technology, 2003. He has experience in Information Engineering, especially Artificial Intelligence. His research interest focuses on Semantics and Bigdata. He has more than 60 publications.

References

Clark, K., & Manning, C. D. (2015, July). Entity-centric co-reference resolution with model stacking. Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing, Beijing, China (Vol. 1, pp. 1405–1415). Association for Computational Linguistics.
Cornolti, M., Ferragina, P., & Ciaramita, M. (2013). A framework for benchmarking entity-annotation systems. Proceedings of the 22nd international conference on World Wide Web, Rio de Janeiro, Brazil (pp. 249–260). International World Wide Web Conferences Steering Committee.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on Wikipedia data. Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), Prague, Czech Republic (Vol. 7, pp. 708–716). Association for Computational Linguistics.

Ferragina, P., & Scaiella, U. (2010, October). Tagme: On-the-fly annotation of short text fragments (by Wikipedia entities). Proceedings of the 19th ACM international conference on information and knowledge management, Toronto, ON, Canada (pp. 1625–1628). ACM.

Garcia, M., & Gamallo, P. (2014). Entity-centric conference resolution of person entities for open information extraction. Procesamiento del Lenguaje Natural, 53, 25–32. Retrieved from http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/5049

Guo, S., Chang, M. W., & Kiciman, E. (2013). To link or not to link? A study on end-to-end tweet entity linking. Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies, Atlanta, GA, USA (pp. 1020–1030). Association for Computational Linguistics.

Guo, Z., & Barbosa, D. (2014). Entity linking with a unified semantic representation. Proceedings of the companion publication of the 23rd international conference on World Wide Web companion, Seoul, Republic of Korea (pp. 1305–1310). International World Wide Web Conferences Steering Committee.

Hajishirzi, H., Zilles, L., Weld, D. S., & Zettlemoyer, L. S. (2013). Joint coreference resolution and named-entity linking with multi-pass sieves. Proceedings of the 2013 conference on empirical methods in natural language processing (EMNLP 2013), Seattle, WA, USA (pp. 289–299). Association for Computational Linguistics.

Han, X., Sun, L., & Zhao, J. (2011). Collective entity linking in web text: A graph-based method. Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval, Beijing, China (pp. 765–774). ACM.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., & Weikum, G. (2011). Robust disambiguation of named entities in text. Proceedings of the conference on empirical methods in natural language processing (EMNLP ’11), Edinburgh, UK (pp. 782–792). Association for Computational Linguistics.

Hajishirzi, H., Zilles, L., Weld, D. S., & Zettlemoyer, L. S. (2013). Joint coreference resolution and named-entity linking with multi-pass sieves. Proceedings of the 2013 conference on empirical methods in natural language processing (EMNLP 2013), Seattle, WA, USA (pp. 289–299). Association for Computational Linguistics.

Kulkarni, S., Singh, A., Ramakrishnan, G., & Chakrabarti, S. (2009). Collective annotation of wikipedia entities in web text. Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, Paris, France (pp. 457–466). ACM.

Marrero, M., Urbano, J., Sánchez-Cuadrado, S., Morato, J., & Gómez-Berbís, J. M. (2013). Named entity recognition: Fallacies, challenges and opportunities. Computer Standards & Interfaces, 35(5), 482–489. doi:10.1016/j.csi.2012.09.004

Mendes, P. N., Jakob, M., García-Silva, A., & Bizer, C. (2011). Dbpedia spotlight: Shedding light on the web of documents. Proceedings of the 7th international conference on semantic systems, Graz, Austria (pp. 1–8). ACM.

Milne, D., & Witten, I. H. (2008). Learning to link with Wikipedia. Proceedings of the 17th ACM conference on information and knowledge management, Napa Valley, CA, USA (pp. 509–518). ACM.

Moro, A., Raganato, A., & Navigli, R. (2014). Entity linking meets word sense disambiguation: A unified approach. Transactions of the Association for Computational Linguistics, 2, 231–244. Retrieved from https://transacl.org/ojs/index.php/tacl/article/view/291

Mrabet, Y., Gardent, C., Foulonneau, M., Simperl, E., & Ras, E. (2015, January). Towards knowledge-driven annotation. Proceedings of the 20th international conference on artificial intelligence, Austin, TX, USA (pp. 2425–2431). AAAI Press.

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. Lingvisticae Investigationes, 30(1), 3–26. doi:10.1075/li.30.1.03nad

Piccinno, F., & Ferragina, P. (2014). From TagME to WAT: A new entity annotator. Proceedings of the first international workshop on entity recognition & disambiguation, Gold Coast, QLD, Australia (pp. 55–62). ACM.
Pu, K. Q., Hassanzadeh, O., Drake, R., & Miller, R. J. (2010). *Online annotation of text streams with structured entities*. Proceedings of the 19th ACM international conference on information and knowledge management, Toronto, ON, Canada (pp. 29–38). ACM.

Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., & Manning, C. (2010). *A multi-pass sieve for coreference resolution*. Proceedings of the 2010 conference on empirical methods in natural language processing, Cambridge, MA, USA (pp. 492–501). Association for Computational Linguistics.

Rao, D., McNamee, P., & Dredze, M. (2013). *Entity linking: Finding extracted entities in a knowledge base*. In T. Poibeau, H. Saggion, J. Piskorski, & R. Yangarber (Eds.), *Multi-source, multilingual information extraction and summarization* (pp. 93–115). Berlin Heidelberg: Springer.

Ratinov, L., Roth, D., Downey, D., & Anderson, M. (2011). *Local and global algorithms for disambiguation to Wikipedia*. Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, Portland, OR, USA (Vol. 1, pp. 1375–1384). Association for Computational Linguistics.

Shen, W., Wang, J., & Han, J. (2015). *Entity linking with a knowledge base: Issues, techniques, and solutions*. *IEEE Transactions on Knowledge and Data Engineering*, 27(2), 443–460. Retrieved from http://ieeexplore.ieee.org/abstract/document/6823700/

Sil, A., & Yates, A. (2013). *Re-ranking for joint named-entity recognition and linking*. Proceedings of the 22nd ACM international conference on information & knowledge management, San Francisco, CA, USA (pp. 2369–2374). ACM.

Stoyanov, V., & Eisner, J. (2012). *Easy-first coreference resolution*. Proceedings of COLING 2012, Mumbai, India (pp. 2519–2534). Association for Computational Linguistics.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). *Feature-rich part-of-speech tagging with a cyclic dependency network*. Proceedings of the 2003 conference of the North American chapter of the Association for Computational Linguistics on human language technology, Edmonton, Canada (Vol. 1, pp. 173-180). Association for Computational Linguistics.

Usbeck, R., Ngomo, A. C. N., Röder, M., Gerber, D., Coelho, S. A., Auer, S., & Both, A. (2014). *AGDISTIS-graph-based disambiguation of named entities using linked data*. Proceedings of 13th international semantic web conference – ISWC 2014, Riva del Garda, Italy (pp. 457–471). Springer.

Usbeck, R., Röder, M., Ngonga Ngomo, A. C., Baron, C., Both, A., Brümmer, M., … Ferragina, P. (2015, May). *Gerbil: General entity annotator benchmarking framework*. Proceedings of the 24th international conference on World Wide Web, Florence, Italy (pp. 1133–1143). International World Wide Web Conferences Steering Committee.