Unsupervised Entity Linking using Graph-based Semantic Similarity

By: Ali M. Naderi

Advisors: Prof. Horacio Rodriguez, Prof. Jordi Turmo

TALP Research Center, Department of Computer Science

Technical University of Catalunya (BarcelonaTech)

This dissertation is submitted for the degree of

Doctor of Philosophy

December 2015

I would like to dedicate this thesis to my loving wife and parents who have given me everything . . .

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

By: Ali M. Naderi

December 2015

Acknowledgements

I would like to express my gratitude to Prof. Horacio Rodríguez and Prof. Jordi Turmo for their supervision, valuable Advice, and support throughout the course of this research works. Many thanks go to the people in the TALP group for their comments. Finally, the technical assistance provided by “Universitat Politecnica de Catalunya" (UPC) is gratefully acknowledged.

The research in this thesis project is carried out within the frameworks of two projects: a) *KNOW2* project (Language understanding technologies for multilingual domain-oriented information access)[1], between 4 universities and 9 EPOs (Ente Promotor Observador: Observing Promoter), and funded by the Spanish Ministry of Science and Innovation, and b) *SKATeR* project (Scenario Knowledge Acquisition by Textual Reading)[2], between 6 universities and 6 EPOs, and also funded by the Spanish Ministry of Science and Innovation.

KNOW2 Project. As a key aspect in KNOW2 project, knowledge mining is emerging as the enabling technology for new forms of Multi-Lingual Information Access (MLIA, which encompasses both terms), as it combines the last advances in text mining, knowledge acquisition, natural language processing, and semantic interpretation. Question answering, information access based on entities, cross-lingual information access, and navigation via cross-document relations are examples of new applications and tasks that are being adopted both by start-ups and consolidated companies such as Google, Yahoo and Microsoft. KNOW2

[1]KNOW2 Project (TIN2009-14715-C04-04) – `http://ixa.si.ehu.es/know2`
[2]SKATeR Project (TIN2012-38584-C06-01) – `http://nlp.lsi.upc.edu/skater/`

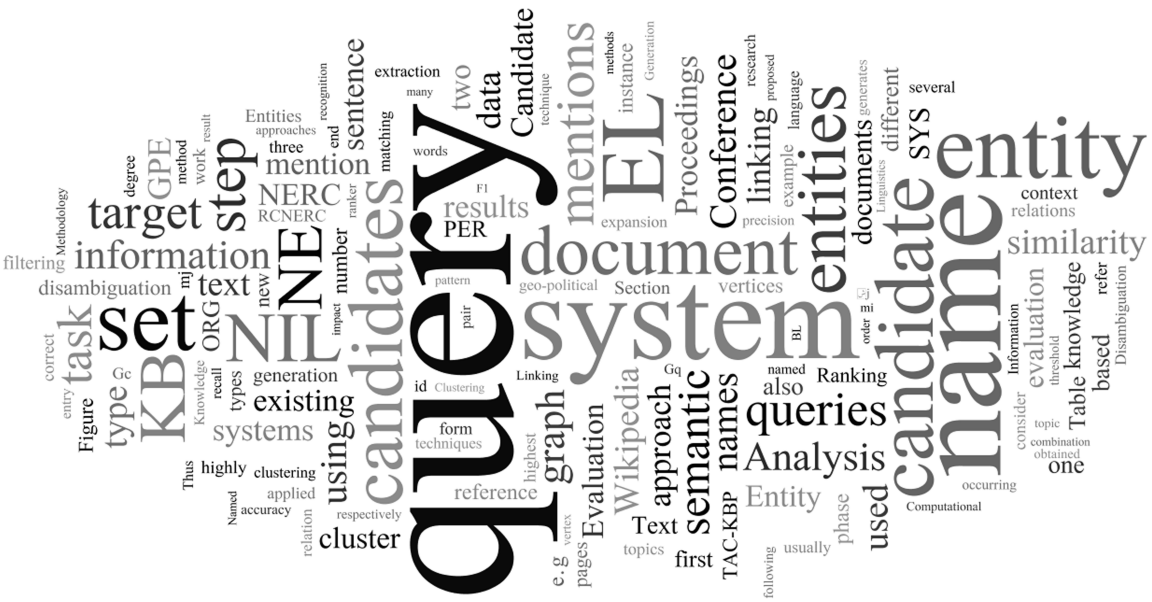
emulated and improved current MLIA systems with research to enable the construction of an integrated environment allowing the cost-effective deployment of vertical information access portals for specific domains. The predecessor of KNOW2 (*KNOW* project)[3] already enhanced Cross-Lingual Information Retrieval and Question Answering technology with improved concept-based NLP technologies.

SKATeR Project. The aim of the ongoing *SKATeR* (Scenario Knowledge Acquisition by Textual Reading) consists in advancing the state-of-the-art in the integration of textual processing, semantic interpretation, inference and reasoning, detection and generalization of events, scenario induction and its exploitation in a number of advanced content-based domain applications. Given the current state-of-the-art in those areas, it is also planned to develop intuitive collaborative interfaces which will allow communities of users to improve the knowledge for different domains, including multilingual communities.

Research in the area of Knowledge mining is orienting towards the use of Knowledge Base (KB) augmentation techniques to improve the results of MLIA systems. Entity Linking (EL) task as an important part towards the KB augmentation can be considered as the subject of significant study to advance the goals defined under the projects KNOW2 and SKATeR.

[3]KNOW Project (TIN2006-15049-C03) – `http://ixa.si.ehu.es/know`

Content Word Cloud



Abstract

Nowadays, the human textual data constitutes a great proportion of the shared information resources such as World Wide Web (WWW). Social networks, news and learning resources as well as Knowledge Bases (KBs) are just the small examples that widely contain the textual data which is used by both human and machine readers. The nature of human languages is highly ambiguous, means that a short portion of a textual context (such as words or phrases) can semantically be interpreted in different ways. A language processor should detect the best interpretation depending on the context in which each word or phrase appears. In case of human readers, the brain is quite proficient in interfering textual data. Human language developed in a way that reflects the innate ability provided by the brain's neural networks. However, there still exist the moments that the text disambiguation task would remain a hard challenge for the human readers. In case of machine readers, it has been a long-term challenge to develop the ability to do natural language processing and machine learning. Different interpretation can change the broad range of topics and targets. The different in interpretation can cause serious impacts when it is used in critical domains that need high precision. Thus, the correctly inferring the ambiguous words would be highly crucial. To tackle it, two tasks have been developed: *Word Sense Disambiguation* (WSD) to infer the sense (i.e. meaning) of ambiguous words, when the word has multiple meanings, and *Entity Linking* (EL) (also called, *Named Entity Disambiguation*–NED, *Named Entity Recognition and Disambiguation*–NERD, or *Named Entity Normalization*–NEN) which is used to explore the correct reference of *Named Entity* (NE) mentions occurring in documents. The solution

to these problems impacts other computer-related writing, such as *discourse*, improving relevance of *search engines*, *anaphora resolution*, *coherence*, and *inference*.

This document summarizes the works towards developing an unsupervised Entity Linking (EL) system using graph-based semantic similarity aiming to disambiguate Named Entity (NE) mentions occurring in a target document. The EL task is highly challenging since each entity can usually be referred to by several NE mentions (synonymy). In addition, a NE mention may be used to indicate distinct entities (polysemy). Thus, much effort is necessary to tackle these challenges. Our EL system disambiguates the NE mentions in several steps. For each step, we have proposed, implemented, and evaluated several approaches. We evaluated our EL system in TAC-KBP[4] English EL evaluation framework in which the system input consists of a set of queries, each containing a query name (target NE mention) along with start and end offsets of that mention in the target document. The output is either a NE entry id in a reference Knowledge Base (KB) or a Not-in-KB (NIL) id in the case that system could not find any appropriate entry for that query. At the end, we have analyzed our result in different aspects. To disambiguate query name we apply a graph-based semantic similarity approach to extract the network of the semantic knowledge existing in the content of target document.

Keywords: Entity Linking, Named Entity Disambiguation, Knowledge Base Population, Graph-based Semantic Similarity.

[4]Knowledge Base Population contest framework in the Text Analysis Conference.

Table of contents

List of figures

List of tables

Nomenclature

List of symbols

α	the weight assigned to each link h
$\beta_{\varphi_i,\varphi_j}$	degree of similarity between φ_i G_q and $\varphi_j \in G_c$
β_{m_i,m_j}	The degree of similarity between $m_i \in G_q$ and $m_j \in G_c$
Λ	set of all pairs
λ_i	ith pair composed by the query name q and each NE mention
$\mathscr{D}$	a collection target documents
$\mathscr{K}_T$	set of all entities around the world
φ_i	ith binary vector (a row matrix)
C	set of candidates
$C_D(v)$	degree centrality for vector v
c_i	ith candidate in set C
CLR	set of NIL clusters
clr_i	ith cluster in CLR

clr_{new} new NIL cluster

d_q target document corresponding to query q

E^* set of incoming edges

E_c set of candidate graph edges

e_i ith entity

E_q set of query graph edges

F_f Filtering function

G_c candidate graph

G_q query graph

h_i ith link

$H_{q,c}$ a set of links between G_q and each G_c

id_q query Id

id_{clr} NIL cluster Id

K Knowledge Base

l_i ith lemma in L_λ

L_T bag of lemmas of all pairs Λ

L_λ set of lemmas for λ

lev_{m_i,m_j} Leveshtein distance metric to measure differences between two strings m_i and m_j

m_i ith NE mention

M_{s_o} set of NE mentions occurring in the query sentence

NIL not-in-KB entity

q query name

q_{nil} NIL query

s_o query sentence

T_G set of topics for graph G

t_i ith topic in T_G

thr threshold

thr_{nil} NIL threshold

V_c set of candidate graph vertices

V_q set of Alternate Names (AN); set of query graph vertices

v_q^i ith AN for query name q

W_e weight of each incoming edge e

w_i ith word in W_λ

W_λ context between q and m

X_c and S_c score symbols obtained by each candidate

List of abbreviations and acronyms

ACE Automated Content Extraction

AN Alternate Name

DUC Document Understanding Conference

EL Entity Linking

ERD Entity Recognition and Disambiguation

HAC Hierarchical Agglomerative Clustering

IE Information Extraction

IR Information Retrieval

KB Kowledge Base

KBP Knowledge Base Population

KD Knowledge-based Disambiguation

LDA Latent Dirichlet Allocation

NE Named Entity

NED Named Entity Disambiguation

NER Named Entity Recognition

NERC Named Entity Recognition and Classification

NERD Named Entity Recognition and Disambiguation

NIST National Institute of Standards and Technology

RCNERC Rule-based Combination NERC

SD Supervised Disambiguation

TAC Text Analysis Conference

TREC Text REtrieval Conference

UD Unsupervised Disambiguation

WSD Word Sense Disambiguation

Chapter 1

Introduction

Recently, the needs of world knowledge for Artificial Intelligence (AI) applications are highly increasing. A system intelligence can be measured by amount of its interaction and knowledge about the real world entities. As an appearance of the world knowledge, a KB is a crucial resource to keep and categorize facts, entities and their relations. Large scale KB has been proved to be valuable for many natural language processors such as *question answering* [48], *information extraction* [73], *coreference resolution* [76] and *word sense disambiguation* [26]. The Internet is growing to be a wide and complex global KB known as the *Semantic Web*, according to the *World Wide Web Consortium* (W3C). A KB helps towards process of huge amount of information in a short span of time. A well-structured KB reduces a firm cost by decreasing the amount of human resources' time spent aiming to discover information about - among countless possibilities - trade laws or firm policies and objectives. However, the high cost of manual elicitation to create the KB, forces toward automatic acquisition from text. This requires two main abilities.

1. extracting relevant information of mentioned entities including attributes and relations between them (*Slot Filling*), and

2. linking these entities with entries in the KB (*Entity Linking*–EL).

The scope of the achieved work presented in this document is limited to the EL task.

EL has recently observed many attentions from the researchers, especially for IE purposes. A traditional IE task includes three main steps:

- *NE Recognition and Classification*: Detecting NE mentions occurring in target documents and classify them different entity types such as such as Person, Organization or GeoPolitical entities. There are several relevant work in [2, 14, 19, 50, 83], and the review article by Nadeau and Sekine [61].

- *Coreference Resolution*: Group two or more NEs and other anaphoric mentions in a document or a set of documents that refer to the same real world entity. For example, "Mr. President", "B. Obama", and "Barack Obama" occurring in a set of documents might refer to the same entity [3, 69, 89, 94].

- *Relation Extraction*: The relation extraction task is to determining the relation between two NEs occurring in the target documents. As an example, occurring two person names regarding to the football match, a relation extraction system should return sport, match, and football as answer [5, 86].

This traditional view over IE has received considerable attention. However, they are not often the only structured information. Another task that has recently observed many attention is NED. It includes disambiguation of a NE mentions occurring in target documents and linking them to the correct entities in the KB. However, the NED task alone is not enough in a new defined IE task, especially for automatic KBP tasks. In the recently defined IE task, the new extractions must be merged with previously extracted information in the reference KB. It in turn requires linking extracted information in text to entries in a KB and determining whether any duplicate exists between the information. If yes, to update the corresponding entry. In the case that there is no entry corresponding to new extraction information, a new entry should be created in the KB to locate that information.

Entity Linking (EL), also known as *record linkage* or *grounding* is an important step towards addressing the goals of KB augmentation and can also be used in other areas such as *topic detection*, *machine translation*, and *information retrieval*. It can also be viewed as an unsupervised *Named Entity Disambiguation* (NED) problem at large scale. EL is the task of referring a NE mention in a document to the unique entity within a reference KB. A NE mention is a mention that uniquely refers to an entity by its proper name, acronym, nickname, alias, abbreviation or other alternate name. Entities can have different types such as person (e.g.“`James Taylor`"), organization (e.g. “`Microsoft`"), and geo-political entities (e.g. “`New York City`"). In the EL task, new extractions must be merged with previously extracted information in KB. As an example for linking NEs to a reference KB (e.g. Wikipedia), when seeing the text “`American politician Chuck Hagel`", the NE mention “`Chuck Hagel`" should be linked the Wikipedia entity “`http://en.wikipedia.org/wiki/Chuck_Hagel`". A system input in the EL task[1] is defined as:

- a knowledge base $K = \{e_1, \ldots, e_n\} \subseteq \mathscr{K}_T$, where $e_i \in K$ is the ith entity in the K and $\mathscr{K}_T$ is the set of all entities around the world.

- a query name q occurring in a target document $d_q \in \mathscr{D}$, where $\mathscr{D}$ is the collection of target documents.

The system output is either:

- the entity e_i to which q refers, or

- *NIL* if such an entity does not exist in *KB*.

[1] Based on the EL task definition in TAC-KBP 2014

The task is formalized as a function:

$$link(q,K) = \begin{cases} e_i & \text{if } 1 \leq i \leq n \\ NIL & otherwise \end{cases} \tag{1.1}$$

where $link(q,KB)$ is the function to detect the correct entity for a query name. In other words, given a set of queries, each of which consisting of a query name (target NE mention) and a document in which the query name occurred, and the start and end offsets of the query name, the system should provide the identifier of the KB entity to which the query name refers if existing, or a NIL Id if there is no such KB entry. The EL system is also required to cluster together queries referring to the same Not-in-KB (NIL) entities and to provide a unique ID for each cluster.

1.1 EL Applications

EL is a new task in the field of NLP which has attracted many attentions in the recent years. It has a high potential for being improved with a wide range of applications. Following, some applications in the business environment are explained.

- Recently, the activity of security threats (e.g., extremist groups) is highly increasing in the virtual environments such as forums, weblogs, and social networks like Facebook and Twitter. The security agencies may gather many unstructured information about them. Manual extraction of mentions (e.g. persons, organizations, locations, and future events) from the unstructured data is however highly time consuming and is against the essence of these threats that need velocity in reaction. EL would be an appropriate

solution to automate the process of mapping necessary information from the huge amount of unstructured documents to the structured data during a short span of time.[2]

- EL systems can be used in the platform of all human-computer/robot dialogue systems. To communicate, these systems should firstly infer the speech dialogue. This in turn requires disambiguation of NE mentions in the human dialogue. As an application, an EL system can be applied in the humanoid robots and assistive machines such as diagnosis systems. In addition, it could be used in wide range of embedded systems such as natural language processors embedded within new generation of cars, tv, mobile devices.

- It can be used for all systems that use a KB. In general, a KB is not a static collection of information, but a dynamic resource that may itself have the capacity to learn, for instance, as part of an AI expert system. To this end, a KB needs continuous augmentation of its entries (update). However, manual augmentation of entries is highly time consuming. For this purpose, EL systems are highly beneficial in order to automate the elicitation of structured information from documents and help IE to create/update entries in KB.

- EL system can be used to annotate texts with semantic information. One example is the *Wikify!* [57] which automatically generates a link to Wikipedia for each disambiguated NE mentions existing in the target documents. This technique is also used by news agencies to provide significant information for their clients. Another application is in the digital libraries where the goal is to cluster and link the same authors both in papers and in citations [33].

[2]For this reason, the EL task within KBP contests in the framework of TAC is supported by U.S. Department of Defense.

- EL can be used for a broad range of applications in companies with different subjects of activity. In the companies that focus on the email services, it can be applied to process the email messages and to extract upcoming events and task along with their dates. Subsequently, it can link them to a calendar. Several companies work on knowledge discovery task focusing in the real-life entities. For instance, some financial companies used such systems to monitor events like company mergers and other financial activities like bilateral contracts and product releases.

1.2 Problem Definition

EL is the task of referring Named Entity (NE) mentions occurring in a natural language text to their correct entities (persons, organizations, and geo-political entities) in a reference KB. EL task is non-trivial due to highly ambiguous nature of human language. In the task, text processors are usually faced to many challenges in correctly linking mentions. The EL task is challenging for three main reasons:

1. *Polysemy*. One query name may be used to refer distinct entities. It can be interpreted in different ways depending on the context in which it appears. As an instance for person entities, consider the following sentence:

> "George Bush brought to the White House a dedication to traditional American values and a determination to direct them toward making the United States a kinder and gentler nation.",

"George Bush" might refer to either "George H. W. Bush", the 41st President of the United States, or "George W. Bush", 43rd President of the United States.

The polysemy may also exist given that some entities are incompletely referred. Query names can be *pseudonyms* or *nicknames*, and are often acronyms. Organization and geo-political entities are also faced to these challenges. "ABC" can be referred to more than hundred entities such as "American Broadcasting Company" or "Australian Broadcasting Company". The query name "Georgia" can be linked to either "Georgia (country)" or an American state. In addition, two NE mentions may overlap. For instance, in the following sentence:

> "The University of York, is a research-intensive plate glass university located in the city of York, England. In 2012 York joined the Russell Group in recognition of the institution's world-leading research and outstanding teaching.",

"University of York" is an overlapping mention that refers to both "University of York" as an organization and also to "York" as a geo-political entity. In addition, the second and third occurrences of "York" in the quotation above refer to a geo-political entity and to an organization entity, respectively. The ambiguity can be more challenging. For instance, in the sentence:

> "The Big Apple is hosting a famous soccer match.",

the "Big Apple" refers to "New York City". In discussion fora, e.g., blogs and other social media documents such as tweets, the texts might contain orthographic irregularities such as misspellings which make the EL even harder. For instance, in the sentence:

> "James Hatfield is working with Kirk Hammett.",

the NE mention "James Hatfield" can refer to the American author. But, the correct form of "Hatfield" is "Hetfield" referring to the main songwriter and co-founder of heavy metal band, Metallica.

2. *Synonymy*. One entity in the KB can be referred by several query names. For example, in the following sentence:

> "Former American president George W. Bush (a.k.a. Bushie, Dubya) is widely known to use nicknames to refer to journalists, fellow politicians, and members of his White House staff.",

"Bushie and "Dubya are synonym and both referring to "George W. Bush", 43rd President of the United States. Besides, *Metonymy* can sometimes be a form of synonymy by which an entity is called not by its own name but rather by the name of something associated in meaning with that entity. For example, consider the following sentence:

> "Hollywood is a neighborhood in the central region of Los Angeles, California. It is notable for its place as the home of the entertainment industry, including several of its historic studios.",

the "Hollywood" is used as a metonym for the U.S. film industry.

3. *Absence*. Many query names occurring in the target documents are referring to not-in-KB (NIL) entities. Indeed, for that query names there are not a mapping entity in the reference KB. An EL system should detect them. Each set of NIL query names referring to the same not-in-KB entity should be clustered together in a group.

These examples indicate that EL task is faced to many challenges. Tackling these challenges would be very tough without extracting the semantic knowledge from the neighboring context of those NE mentions. In next section, we will briefly describe our approach to overcome these difficulties. Consequently, in the Section 3 our approach would be explained in detail.

1.3 Hypothesis

The hypothesis behind this work is based on this fact that query names existing in a document are usually coherent. They form an inter-related semantic network and each group of mentions can be clustered by one or more topics. Furthermore, in a document with different and distinct subjects, the mentions are usually more correlated whenever their offsets in the document get closer. Thus, to disambiguate a query name we extract this network between the NE mentions existing in the target document. To this objective, we present an unsupervised approach to disambiguate NE query names. Our system generates a network of relations using a graph-based method and based on semantic similarity between the NE mentions.

1.4 The Proposal and Contributions

Recently some researchers proposed their EL systems following supervised disambiguation techniques. These approaches are however faced to lack of enough annotated training data. Semi-supervised and unsupervised techniques are alternatives to overcome this problem. To tackle the challenges mentioned in 1.2 we have developed an unsupervised EL system. It disambiguate query names occurring in the target documents in a pipeline. It includes *Document Preprocessing* step to preprocess the target document (Section 3.1), *Candidate Generation and Filtering* step to generate a set of candidates generated for each query

name and then to filter-out the least reliable (Section 3.2), *Candidate Ranking* step to rank candidates in order to find out the best matching KB candidate for each query name (Section 3.3) and *NIL clustering* step to cluster those queries without any candidate in KB (Section 3.3.3). For each step, we have proposed techniques to tackle the facing challenges.

Briefly, in the document preprocessing step we apply several techniques (described in Section 3.1) to normalize the document and expand the information in order to assist the process of disambiguation. In this step, we applied a *Rule-based Combined NERC* (RCNERC) system to distinguish query names in the target document. This is a combination of three NERC systems that is able to amend the result of named entity recognition using predefined rules. In addition, in this step the system applies other techniques such as text normalization, acronym expansion, pattern extraction to enrich the target document. In the candidate generation and filtering step the system initially generates a set of candidates for each query and it then applies a rule-based approach to filter-out noisy candidates from the set of whole candidates. It helps to obtain a discriminative set of candidates that increases the system accuracy in linking task. In the candidate ranking step we proposed our unsupervised disambiguation approach that uses graph structure towards ranking candidates. It discovers the semantic knowledge laid in the context of document. To tackle the highly ambiguous nature of EL task it is crucial to exploit the semantic relations between NE mentions in the target document. Since our method is unsupervised it does not have the defect of supervised approaches that is the lack of enough annotated data for training. Finally, for the NIL Clustering step, we have applied a term clustering approach that groups all same not-in-KB queries in a cluster indicating a new entity in the KB.

Meanwhile, our research in this area spans over several areas within the field of NLP, and we believed that a number of distinguishable contributions are contained in our work. We want to highlight them, listing them below in what we consider their order of decreasing relevance:

- *C.1: Unsupervised Disambiguation using Local Information.* We have proposed an unsupervised graph-based approach using local information occurring in the target document (henceforth, *local ranker*). In our experiments, the *local information* refers to the data existing in a sentence where the query name occurs. The hypothesis behind it, is based on this idea that a relevant semantic relation occurs often between query name and each NE mention (the pair ⟨query name, NE mention⟩) in the same sentence. The system uses these semantic relations to rank candidates. To this end, it extracts the context between each pair in the same sentence. A binary vector (a row matrix) is then assigned to the context elements (bag of lemmas) between each pair. In order to rank the candidates, the system generates a star graph for the query name and one for each candidate. The system computes the similarity between query graph and each candidate graph. The goal is to select the most similar candidate to the query name. Central vertex of query graph is labeled with the query name and central vertex of each candidate graph is labeled with the candidate name. Other vertices in the graphs are labeled with those NE mentions existing in the set of pairs. Each edge is labeled with the semantic relation existing between the linked entities. This is represented by a binary vector corresponding to each pair. The system ranks each candidate based on the degree of similarity between query graph and each candidate graph. Details are described in Section 3.3.1.

- *C.2: Unsupervised Disambiguation using Global Information.* We have proposed an Unsupervised graph-based approach that takes advantage of global information (henceforth, *global ranker*) existing in the target document. This information is the semantic knowledge not only existing in the query sentence (the sentence where the query name occurs) but also the information lied in other sentences of the target document (in our experiment, a text window of 3 sentences, the query sentence and

the previous and following ones). In this approach, we consider the fact that NE mentions existing in a document are usually coherent. They form an inter-related semantic network and each group of mentions can be clustered by one or more topics. The system exploits the semantic networks between NE mentions. The first approach (local ranker) computes the semantic similarity just from the target document. On the contrary, the second approach (global ranker) computes the semantic similarity using world knowledge, specially using DISCO[3] and based on the statistical analysis of very large text collections (in our experiment, English Wikipedia). The system generates a graph for the query name and one for each candidate. Each NE mentions occurring in the text window (three sentences) of the target document, would be a vertex in the query graph (excluding query name). Likewise, each NE mention, recognized from the first 10 sentences[4] of each candidate's document, is a vertex for this candidate graph (excluding query name). The relations (edges) are the semantic similarity (measured by DISCO) between each two vertices. The system thereupon computes the most important vertices as the topics for each graph. The topics are recognized by computing degree centrality for each vertex. Finally, the system ranks the candidates based on degree of similarity between the topics in the query graph and each candidate graph. Details are described in Section 3.3.2.

- *C.3: Combined Disambiguation approach.* In some queries, the query sentence does not contain any NE mention (the sentence just has the query name). In such cases, the system cannot apply the local ranker. To solve this problem, the system initially tries to apply the local ranker. If the query sentence contains no NE mentions (except, the query name), the global ranker will be used. We have evaluated both ranking

[3]DISCO is a NLP tools which allows to retrieve the semantic similarity between arbitrary words – `http://www.linguatools.de/disco/disco_en.html`

[4]We consider the fact that the first sentences in each candidate's document are more informative. It is a notable consideration for systems extracting information from Wikipedia pages. See, for instance, [54].

approaches, separately. We realized that the local ranker has a better performance for those queries having more than two candidates and the global ranker has better performance for the queries having two candidates. In order to improve the results, we combined these two approaches. Using this technique, the results obtained by the combination approach improved (Figure 4.6).

- *C.4: NIL clustering using Alternate Names (ANs).* A large amount of the queries refer to the entities that are not present in the reference KB (NIL queries). For those queries, the system clusters them in groups, each referring to a same Not-in-KB entity (NIL Clustering). We proposed a NIL clustering approach. The system consider the first NIL query as a NIL cluster. The next NIL queries and their properties (query name and Alternate Names–ANs) are compared with existing NIL clusters. ANs are the name variants for a query name. The comparison uses a fuzzy matching techniques based on the Dice similarity between each cluster's properties and the new query name (and its ANs). In each comparison if the dice similarity is more than a predefined threshold (in our experiment, set to 0.8), the new NIL query will be joined to that cluster. Otherwise, the system generates a new cluster for this query. Details are explained in Section 3.3.3.

- *C.5: NE Recognition and Classification using Rule-based Combination Approach (RCNERC).* In this research, we have proposed a Rule-based Combination NERC (RCNERC) system (three-phase NERC system explained in Section 3.1). The first step is called *recognition phase* and is responsible for detecting and classifying query names to PER (person), ORG (organization), GPE (geo-political entity), MISC (miscellaneous), and N/A (not-available) types using three NERC systems (Stanford, Illinois, and Senna). The second step is called *combination phase* and combines the classification results based on the majority voting between three NERC systems. Finally, the third step is called *amendment phase* and modifies the results of combination phase by

a set of predefined rules. [55] presented a combination NERC technique for linking entities. Compared with this work, we applied a further step (amendment phase). This phase takes advantage of pattern extraction and matching to modify NERC annotations. Our RCNERC system significantly improved the final results of the EL system. It demonstrated that although the candidate ranking is a crucial step in EL systems, without a reliable NERC approach, its performance would dramatically be decreased.

- *C.6: Three-phase Candidate Filtering.* The accuracy of EL systems in ranking candidates is exponentially reduced whatever the number of candidates (for each query) is increased. Thus, it requires a method to filter-out those candidates which are unlikely the correct answer. The candidate ranking step is afterward applied to the rest of candidates. We proposed a rule-based candidate filtering method to reduce the probability of selecting a wrong entity in the reference KB during the candidate ranking step. [71] and [39] presented a candidate filtering method for EL task. They applied a title matching in order to filter-out the candidates. In our filtering technique, we use a wider range of rules in three main categories: 1) Title Matching, 2) Type Matching and 3) Pattern Extraction and Matching (Section 3.2). This technique helps to obtain a discriminating set of candidates that increases the system accuracy.

1.5 Overview of this Document

The rest of this document is organized as follows: Chapter 2 gives a review of the state of the art and the development of EL systems:

- This Chapter explains about the early history of EL.
- In continue, it describes different part of the most popular EL system architecture as pipeline and presents literature review on each part of the EL system.

- And finally, several evaluation framework for EL task are described in this chapter.

Chapter 3 presents our approach and the state of the work that has been carried. This Chapter is organized in three sections:

- Document Processing section where at the outset the documents are preprocessed.
- Document Generation and Filtering step where a set of candidates for each query are generated and noisy ones are filtered.
- Candidate Ranking section where the best candidate for each query is chosen. This section itself includes three parts: a) Candidate Ranking using Local Information, b) Candidate Ranking using Global Information and c) NIL clustering where the system clusters each group of NIL queries referring to a same not-in-KB entity.

Chapter 4 describes the evaluation framework and analysis of results. This section includes:

- Introducing the evaluation framework that we used in our experiments containing a) evaluation task definition, b) evaluation metrics and c) evaluation data.
- presenting the evaluation results and analysis of them.

Finally, Chapter 5 reflects our conclusion and future works on the research.

Chapter 2

State of The Art

This section presents an overview of early history of EL (Section 2.1), the general architecture of EL systems as well as the research done towards different EL approaches (Section 2.2). A sketch of the evolution in EL evaluations is presented (Section 2.3) and finally, the Wikipedia as a valuable KB is briefly reviewed in Section 2.4.

2.1 Early History and Recent Works on EL

Recent works focused on EL in its contemporary history are inspired from the antecedent research on Word Sense Disambiguation-WSD which is the task of detecting the correct sense of a word in text. For instance, the word "bar" can be meant as an obstacle, load, rod, arrow or court. The correct sense of a word in the text is determined given its neighborhood. Even two occurrences of a word in the same document may have different senses depending their close neighborhood. Many studies achieved on WSD are quite relevant to EL. [45] presented its unsupervised WSD algorithm using machine readable dictionaries. Also, [95] proposed its algorithm for unsupervised WSD. The algorithm is based on two powerful

constraints - that words tend to have one sense per discourse and one sense per collocation - exploited in an iterative bootstrapping procedure. EL is a recent task inspired by WSD.

By analogy, both EL and WSD tasks tackle synonymy and polysemy challenges existing in the human language. However, there are differences between the tasks. In EL task, the candidates are located in the KB (e.g., Wikipedia, DBpedia). In WSD task, the candidates are placed in the sense repositories (usually WordNet-WN senses) [32]. In addition, the candidate generation step in WSD supposes that this lexical database is a complete resource. On the contrary, the EL task may deal with the queries that are not in the KB. In such cases, the query is tagged as NIL [7, 52]. The NIL clustering step is applied and the NIL query will be added to the KB as a new entry. As another difference, the NE mentions in the EL task usually vary more than lexical mentions in WSD. Therefore, the EL task requires a wider candidate generation process [44, 90]. The earliest work on the EL task were presented by [7] and [17]. Their goals were to link NEs occurring in the documents to their corresponding entries in Wikipedia. They did not use the term EL in their works and they applied different approaches. [17] used heuristic rules and Wikipedia disambiguation markup for mapping from NE mentions to their Wikipedia entries. [7] suggested an approach for not-in-KB entities by learning a NIL threshold to determine whether the entity exists in the reference KB. Besides, earlier EL studies were focused at each time on disambiguation a NE mention just using its neighborhood and usually based on supervised models [7, 34, 57, 58]. However, due to the lack of enough informative context, a global process of the target document using semantic analysis of multiple NE mentions is more strong. Some recent works later applied this approach by disambiguating a set of relevant mentions simultaneously which is called *collective approaches*, usually through supervised or graph-based reranking models [11, 12, 17, 18, 20, 22–24, 30, 31, 34–36, 42, 43, 47, 58, 74, 78, 85]. Several measures to choose the "collaborators" include *collaborative learning* [12], *ensemble ranking* [42, 74], *co-occurred concept mentions* [53, 70], *topic modeling* [11, 93], *relation extraction* [13],

coreference [70], *semantic relatedness* [13, 38, 70], *Neighborhood Expansion with Pseudo-Relevance Feedback* [21], *meta-paths* [38] and *social networks* [11, 38]. In our work, we have also took advantage of the collaborators and have proposed our methods to generate the networks of similar NE mentions in each target document.

2.2 EL Architecture and Approaches

The EL approaches provided by the researchers usually follow a common architecture in several major steps. The differences are in proposing diverse techniques for each step of this architecture. Figure 2.1 shows a general architecture for EL systems including three major steps. They are described in the following sections.

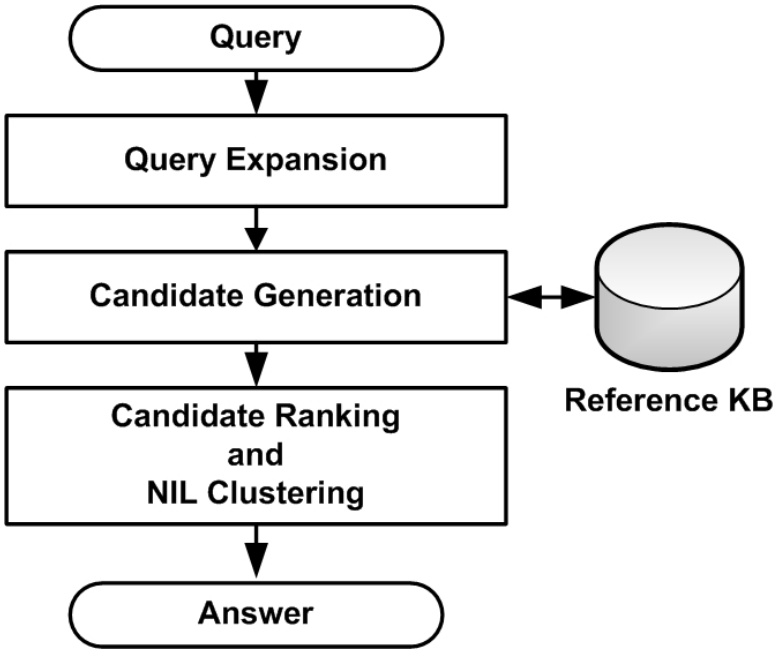


Fig. 2.1 General EL System Architecture

2.2.1 Query Expansion

Expanding the query from its context can effectively reduce the ambiguities of the query name, under the assumption that two name variants in the same document refer to the same entity. For example, without the query expansion, the query name *Roth* is linked to seventy-six entities in Wikipedia, but its expansion *John D. Roth* is only linked to two entities [97]. Thus, query expansion is performed as the first step for EL. This step often includes a classification of the query into the possible entity types PER (e.g., George Washington), ORG (e.g., Microsoft), GPE (e.g., Heidelberg city). The GPE is abbreviation of *GeoPolitical Entity*, a geographical area which is associated with some sort of political structure (different from natural toponyms as rivers, mountains, seas, etc.). Following, some popular techniques for the query expansion are described.

- *Wikipedia Hyperlink Mining.* A hyperlink is a structural component that connects the web page to a different location. The Wikipedia pages contain many hyperlinks having useful information for the query expansion. The method extracts the name variants of an entity in KB by leveraging the knowledge sources in Wikipedia: “titles of entity pages", “disambiguation pages"[1], “redirect pages"[2] and “anchor texts". With the acquired name variants for entities in KB, the possible KB candidates for a given query name can be retrieved by string matching. If the query name is an acronym, it can be expanded from the target document. [10, 98] employed the Wikipedia hyperlink mining for the query expansion. For specific types of NE more focuses approaches can be used as person name grammars for PER, acronym expansion/compression or suffix removing for ORG and geo-disambiguation techniques for GPE.

- *Coreference resolution.* Several queries can be expanded based on source document coreference resolution. The goal is to explore NE mentions that have relations, shared

[1]http://en.wikipedia.org/wiki/Wikipedia:Disambiguation
[2]http://en.wikipedia.org/wiki/Wikipedia:Redirect

events, common attributes or co-occurrences with the query name in the same target document. When two NE mentions are coreferential, the one is usually a full form (the antecedent) and the other is an abbreviated form or an acronym (a proform or anaphor). When a query name is acronym, the technique helps to expand the full form of this mention. [10] used a Chinese name coreference resolution system to get the possible name variants for each query name.

- *Statistical Models.* With a set of expansions extracted from the target document for a query name, the method applies a supervised learning algorithm to infer which expansions are valid. Each $\langle queryname, expansion \rangle$ pair, in the form of feature vector, is presented to a classifier (e.g., SVM). A NIL response is returned if there are no positively classified expansions. Otherwise, the candidate with highest confidence score is selected. [98] employed a statistical model based on SVM to expand queries. [96] presented a query expansion feature set to be used by the classifier.

2.2.2 Candidate Generation

A KB always contains a huge number of entities. It is impractical a brute-force searching to consider whole entities in the KB for linking NE mentions. Thus, candidate generation step is a solution to retrieve the most important entities in the KB that can potentially be candidates for the query name. This module selects the KB entities that might correspond to the query name, typically with basis on string similarity. The common methods to generate candidate set is described following.

- *Fuzzy Title Matching.* The fuzzy title matching (aka, approximate title matching) is the technique of finding the KB entities that match a string pattern in their title approximately (rather than exactly). For each query, the EL system generates a set of candidates using fuzzy title matching (e.g., Dice similarity). In this technique, the

system explores those candidates with a Dice similarity between the query name and the candidate title higher than a predefined threshold. [10, 59] used the fuzzy title matching similarity to generate candidates from the KB.

- *Information Retrieval (IR).* IR obtains information resources relevant to query from a collection of information resources. Searches can be based on metadata or on full-text indexing. [67] used the IR technique to generate the candidates. In their system, the KB was loaded into an IR engine. It retrieved top 100 candidates for each query. In three separate experiments they load the KB information to the IR engine, first just KB titles, second title and infoboxes, and finally, the entire KB documents. In their evaluation, the best recall was obtained by the last experiment.

2.2.3 Methods for Candidate Ranking

This step sorts the retrieved candidates according to the likelihood of being the correct referent. The ranking methods in the state of the art can be classified into supervised methods, unsupervised methods and knowledge-based methods [66]. These methods are described following.

- *Supervised Disambiguation (SD).* The first category applies Machine Learning (ML) techniques for inferring a classifier from training (manually annotated) data sets to classify new examples. Researcher proposed different methods for SD. A *Decision List* [82] is a SD method containing a set of rules (if-then-else) to classify the samples. [41] used learning decision lists for *Attribute Efficient Learning*. [49] introduced another SD method, *Decision Tree*, that has a tree-like structure of decisions and their possible consequences. *C4.5* [77], a widely used algorithm of learning decision trees was outperformed by other supervised methods [60]. [40] studied on the *Naive Bayes* classifier. This classifier is a supervised method based on the Bayes' theorem and is

a member of simple probabilistic classifiers. The model is based on computing the conditional probability of each class membership depending on a set of features under the hypothesis of conditional independence. Although the model seems, at first glance, too heavy in practice it works well in this task. [60] demonstrated good performance of this classifier compared with other supervised methods. [51] introduced *Neural Networks*. The model is presented as a system of interconnected neurons usually organized into an input layer, an output layer and a set of intermediate (hidden) layers. Although [87] showed an appropriate performance by this model, the experiments were performed with a small size of data. However, the dependency to large amount of training data is a major drawback, [66]. Recently, different combination of supervised approaches have been proposed. The combination methods are highly interesting since they could cover the weakness of each stand-alone SD methods [66]. SD systems obviously rely on the set of supervised (annotated) training sets, and, so on a highly costly human work. This is why researchers have moved to unsupervised or semi-supervised methods. As whatever supervised approach, SD can suffer the problem of lacking enough supervised data for training. For facing this problem unsupervised methods have been proposed.

- *Unsupervised Disambiguation (UD).* The underlying hypothesis of UD is the distributional hypothesis, i.e., a word can be defined by the company it has. Indeed, each word is correlated with its neighboring context. Co-located words generate a cluster tending to the same sense or topic. No labeled training data set or any machine-readable resources (e.g. dictionary, ontology, thesauri) are applied for this approach [66]. *Context Clustering* [84] is a UD method by which each occurrence of a target word in a corpus is represented as a context vector. The vectors are then gathered in clusters, each indicating a sense of target word. A drawback of this method is that, a large

amount of un-labeled training data is required. [46] studied on *Word Clustering* a UD method based on clustering the words which are semantically similar. Later on, [72] proposed a word clustering approach called *clustering by committee* (CBC). [91] described another UD method *Co-occurrence Graphs* assuming that co-occurrence words and their relations generate a co-occurrence graph. In this graph, the vertices are co-occurrences and the edges are the relations between co-occurrences.

- *Knowledge-based Disambiguation (KD).* The goal of this approach is to apply knowledge resources (such as dictionaries, thesauri, ontologies, collocations, etc.) for disambiguation [4, 8, 27, 45, 56]. Although these methods have lower accuracy compared with supervised techniques, they use to have a wider coverage [66].

2.2.4 NIL Clustering

Queries are tagged as NIL (Not-In-KB) when no entity in the KB corresponds to them (or we are not able to select the appropriate one). So NIL implies that a new entry could be included into the KB. In a set of documents, several NE mentions may refer to a same NIL entity. In the KBP contest, all NE mentions related to the same NIL entity should be grouped by the same id (e.g. NIL001, NIL002, ...). So each NIL cluster could correspond to a new entry in the reference KB. Several techniques have been applied to this task (provided that the query has been classified as NIL):

- *Name String Matching.* This technique consists in grouping queries by term matching. For instance, two NIL queries with the name “ABC" would be clustered together and, so, would be given the same NIL ID. A query name may contain fragments of other query names. To this end, the fuzzy string matching is applied. It clusters all NIL queries with the Dice similarity between the query names higher than a predefined threshold. [9, 10, 37, 53, 79, 81] used the name string matching to cluster NIL queries.

- *Hierarchical Agglomerative Clustering (HAC).* In this approach, those query names referring to the same Not-in-KB entity are clustered using HAC algorithm. Hierarchical clustering can be approached by top-down and bottom-up algorithms. The Bottom-up algorithm (aka., HAC) treats each query names as a singleton cluster at the outset and then sequentially merge (or agglomerate) pairs of clusters as long as two clusters exceeded a similarity threshold. [59, 75, 98] presented their works using this approach. In contrary, top-down approach works by starting with a root cluster, where all the candidates are placed and then recursively splitting the clusters based on the most likely partition in each stop.

- *Graph based clustering.* Using this approach, a graph structure is generated for the clustering. [97] employed the graph-based approach to cluster the NIL queries. They used *Spectral Graph Partitioning* (aka., Spectral clustering) [68] to generate the globally optimized entity clusters. The results obtained by the spectral graph partitioning usually outperform the traditional clustering algorithms such as k-means or minimum-cut [97].

- *Topic Modeling.* Topic modeling is a statistical model to explore the topics underlying the documents. The topic modeling approaches are widely used in different NLP applications [97]. *Latent Dirichlet Allocation* (LDA) is a common topic model approach that was first presented as graphical model for topic discovery by [6]. The topics are probability distributions over words. [92, 97] used the topic modeling for the NIL clustering.

- *Linking to Larger KB and mapping down.* This approach clusters NIL queries using a larger KB. [79] explored the NIL query names in full dump of Wikipedia. From 2,250 queries evaluated, their approach tagged 1,263 to NIL. 56% of the NIL queries were

linked to the full dump of Wikipedia and 44% of NIL queries had no reference in the KB.

2.3 EL Evaluation Frameworks

The EL problem is currently receiving substantial attention in the IR community, given its recent inclusion as a specific task in the NIST-sponsored[3] *Automated Content Extraction*[4] (ACE) evaluations (i.e., the ACE-2008 cross-document co-reference resolution task), and in the *Text Analysis Conference*[5] (i.e., the Knowledge Base Population task, referred to as TAC-KBP). In addition, the *Entity Recognition and Disambiguation* (ERD) challenge[6] is recently organized to focus on this task. These frameworks are described following:

- *ACE Evaluations.* The Automatic Content Extraction 2008 ACE cross-document co-reference resolution task is pioneer organized evaluation for defining the recent EL task. The objective of the NIST-sponsored ACE series of evaluations was to develop human language technologies that provide automatic detection and recognition of key information about real-world entities, relations, and events in source language text and to convert that information into a structured form, which can be used by follow-on processes, such as classification, filtering and selection, database update, relationship display, and many others. An ACE system produces information about objects discussed in the source language text. The strings of text are not the objects, but are merely mentions of the real-world objects about which information should be extracted. These objects have included, over the course of the evaluations, various types of entities, relations, events, values, and temporal expressions. The emphasis has

[3]http://www.nist.gov
[4]http://www.itl.nist.gov/iad/mig/tests/ace/
[5]http://www.nist.gov/tac/
[6]http://web-ngram.research.microsoft.com/ERD2014/

been on object coreference resolution, such that all data pertaining to the same unique ACE object are collected into a single XML-formatted “record" on a *per document* basis.

- *TAC-KBP Evaluations.* As the most important challenging competition, EL evaluation a task within the Knowledge Base Population (KBP) track at Text Analysis Conference (TAC) has been the subject of significant study over the past seven years. Since the first KBP track held in 2008, the research in the area of EL has greatly developed. TAC is organized and sponsored by the U.S. National Institute of Standards and Technology (NIST) and the U.S. Department of Defense. TAC has commenced its activity since 2008 and developed out of NIST's *Text REtrieval Conference* (TREC) and *Document Understanding Conference* (DUC). The main goal of the KBP track at TAC is to gather information about a specific entity that is scattered among the documents of a large collection, and then use the extracted information to populate an existing KB.

- *ERD Evaluations.* A recent notable contribution to research in the field of EL was made by the participants of the ERD. As the most structured challenging competition[7], ERD has commenced its activity since 2014 in the content of SIGIR conference[8] whereby the organizers intended to improve the results of search engines based on the recognized entities in the searched queries. The objective of an ERD system is to recognize mentions of entities in a given text, disambiguate them, and map them to the entities in a given entity collection or KB. The Challenge is composed of two parallel tracks. In the “long text" track, the challenge targets are pages crawled from the Web; these contain documents that are meant to be easily understandable by humans. The “short text" track, on the other hand, consists of web search queries that are intended

[7]The ERD organized and sponsored by Google and Microsoft.
[8]http://sigir.org/sigir2014/

for a machine. As a result, the text is typically short and often lacks proper punctuation and capitalization.

- *GERBIL: General Entity Annotator Benchmarking Framework.* The GERBIL [88] is an evaluation framework for NED task. The idea behind this framework is to provide researchers with easy-to-use interfaces that allow evaluation of annotation tools on multiple datasets. It aims to ensure that researchers can derive meaningful insights pertaining to the extension, integration and use of annotation applications. GERBIL provides the results to allow them to easily compare the strengths and weaknesses of their implementations with respect to the state of the art. With the permanent experiment URIs provided by this framework, the reproducibility and archiving of evaluation results can be ensured. Besides, the framework generates data in machine-processable format, allowing for the efficient querying and post-processing of evaluation results.

2.4 Wikipedia, a valuable KB in EL task

The Wikipedia is the most important KB which is widely used for different tasks, especially within the TAC-KBP tracks. Several unique characteristics of the Wikipedia are broadly used in the EL task such as infobox property of each Wikipedia page. The infobox property can be used to disambiguate the query names. In addition, each entity is assigned to one or more than one categories. These types of characteristics have made Wikipedia very popular for the researchers specially those acting in the EL task. The Wikipedia structure contains several features including *redirection pages*, *disambiguation pages*, *infoboxes, categories*, and *hyperlinks*, which can be used to disambiguate query names:

- *Categories*: To each entity in the Wikipedia, one or more categories can be allocated. These categories indicate the topic associated to each Wikipedia entity.

- *Hyperlinks*: Each entity occurring in the context of a Wikipedia page of other entities is referred to by a hyperlink. This Characteristic generates a large network of semantic knowledge over the Wikipedia KB.

- *Redirect Pages*: For each possible AN for each Wikipedia entity, there is a redirect page. For instance, "D.C." is redirected to the Wikipedia page "Washington, D.C.". There are many samples from this type existing in the Wikipedia.

- *Disambiguation Pages*: Several entities in the Wikipedia use the same title. For instance, "ABC" refers to more than one hundred entities. For such names there is a disambiguation page which is considered as a characteristics of Wikipedia. In each page, a set of possible entities, each of which with a short description and corresponding to that name are suggested.

- *Infoboxes*: An infobox is a structured table located in the right sides of Wikipedia pages. It summarizes the main information existing in the context of that entity page. The information inside the infoboxes usually use as facts in different NL task, especially, the EL task. The infobox format follows existing templates, that are suggested to be followed by Wikipedia editors. Currently for English Wikipedia more than 10,000 template exist.

Chapter 3

Methodology

This section describes the methodology we used during development of the proposed EL system. The EL system developed during this research follows the typical architecture in the state of the art (Figure 3.1). In general, the system links entities in a pipeline including three main steps: a) document preprocessing, b) candidate generation and filtering, and c) candidate ranking and NIL clustering. Details of each step are provided next.

3.1 Document Preprocessing

Input to our EL system consists of:

1. a reference KB,

2. a target document,

3. a query (a NE denomination together with an offset in the target document, i.e. an entity mention in the target document.). The requested entity mention is also called a query name.

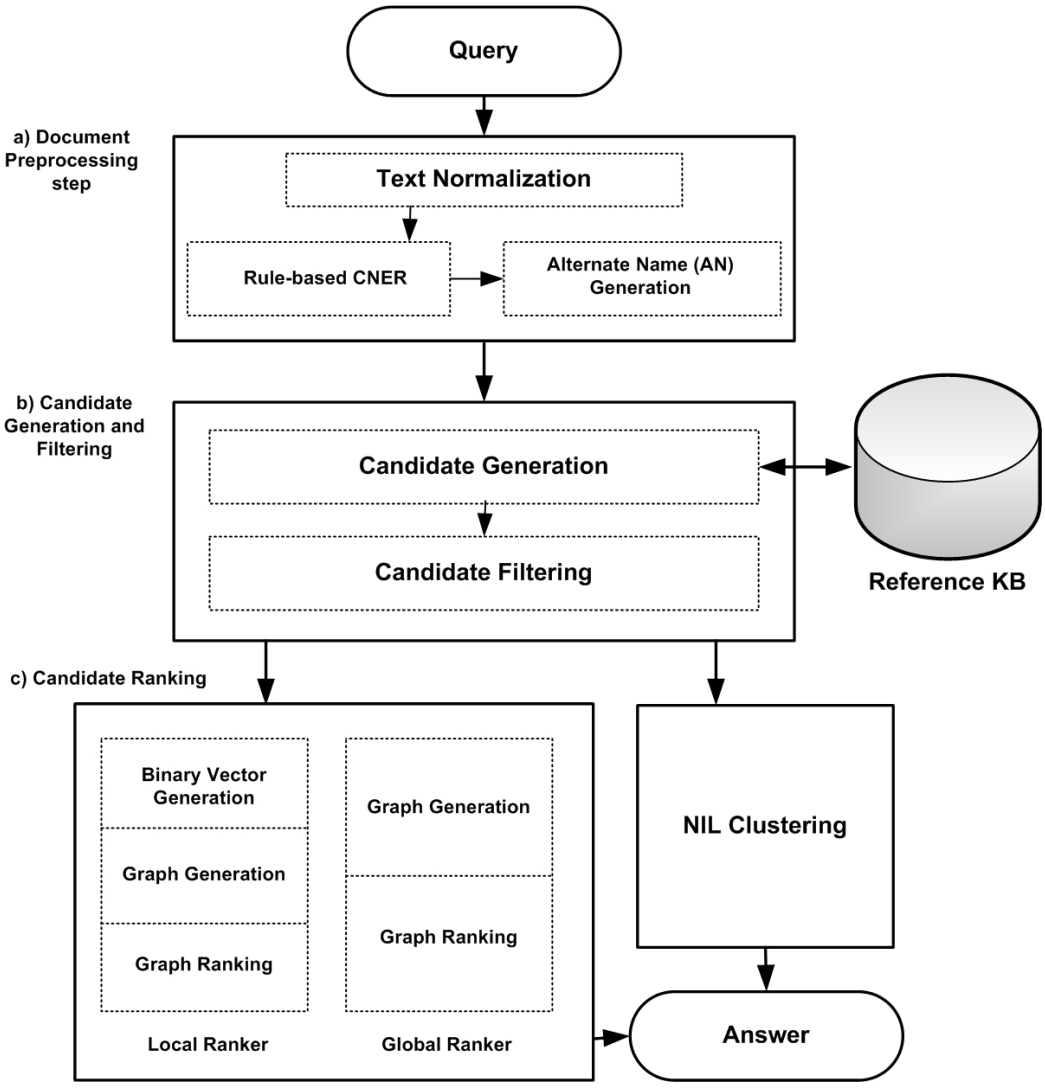


Fig. 3.1 The architecture of our EL system.

Highly ambiguous query names often occur in the target documents: they may refer to several entities in KB. In such cases, document preprocessing is the first step of the EL task in the system and can reduce the ambiguity and enrich the documents through finding variants of each query name, query type annotation and integrating more discriminative information. In order to preprocess the target document, the system applies the following techniques:

Query Information Extraction. In order to proceed each query, the system extracts queries' information consisting of a query Id (`queryId`), a name string (`queryName`), target document Id (`docId`), and the start and end offsets where the name string occurs (`begOfst`, `endOfst`). This information is used in different parts of system according to the needs. The most important one is the query name which is the same as NE mention occurring in the specific offset of the target document.

Text Normalization. To goal of this step consists of removing noise from documents and providing basic structure to them. The target document has to be normalized in order to be used in further steps. A normalized document can increase the accuracy of the system in linking candidates. Non-textual part of documents and HTML tags (e.g., in Web documents) are considered noise. Besides, textual part of documents are splitted into sentences using a statistical sentence boundary detection tools called Splitta [28]. It includes proper tokenization and models for sentence boundary detection. Its models are trained from Wall Street Journal news and the Brown Corpus.

Rule-based Combination NERC Approach (RCNERC). The NE Recognition and Classification (NERC)[1] is one of the initial steps in the linking task which can enrich the target document by annotating and classifying the query names to different types. It can help to filter-out those candidates with type different to each query type. Our system classifies queries into three entity types:

- PER: to indicate person type entities (e.g., "`George Washington`", the first president of the United States). Usually, a huge amount of query names occurring in the documents categorized as PER.

[1] n.b. The query type is not known a priori and has to be guessed.

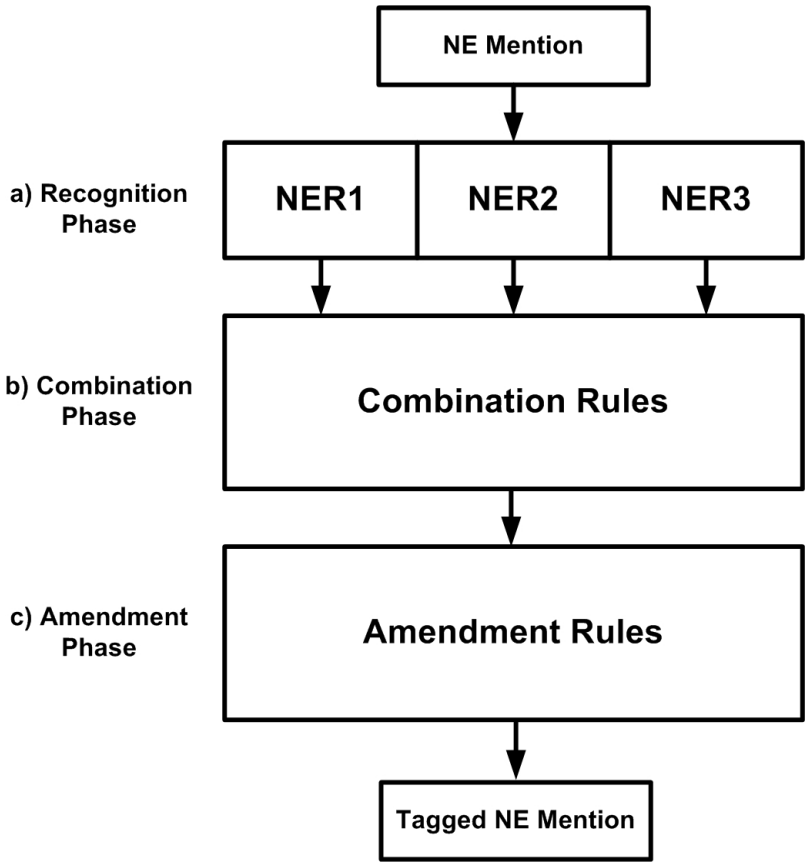


Fig. 3.2 The architecture of the RCNERC system.

- ORG: to represent organization type entities (e.g., "`Microsoft`", an American multinational technology corporation). A considerable occurrences of query names with the ORG type are in the form of acronym.

- GPE: to represent geo-political entities (e.g., "`Heidelberg city`", a city situated on the River Neckar in south-west Germany). There are some cases that select the correct query type between ORG and GPE would be highly challenging. We will discuss it later on.

An accurate NERC system has high impact on the final result of the system. We particularly focused on this step to provide a NERC system as accurate as possible. For this reason, the first step was to reduce the error rate of the NERC system. Each NERC system (even those

in state of the art) has an error rate. To minimize this ratio, we proposed a three-phase NERC approach as indicated in Figure 3.2.

1. *Recognition and Classification Phase.* In this phase we have used three major state-of-the-art NERC systems including Stanford [25][2], Illinois [80][3] and Senna [16][15][4].

   - *Stanford*: Stanford NER is a Java implementation of a Named Entity Recognizer. It provides a general implementation of (arbitrary order) linear chain Conditional Random Field (CRF) sequence models.
   - *Illinois*: It uses several features to achieve new state of the art performance on the NER task using four fundamental design decisions: text chunks representation, inference algorithm, using non-local features and external knowledge.
   - *Senna*: It outputs a host of NLP predictions: part-of-speech (POS) tags, chunking (CHK), NER, semantic role labeling (SRL) and syntactic parsing (PSG).

   Each NERC system provides their results that could be PER (person), ORG (organization), GPE (geo-political entity), MISC (miscellaneous), and N/A (not available). Some documents such as discussion fora (blogs, forums) do not often follow a standard structure and query names are usually lowercase. Thus, we run NERC system two times, one without any change on the document and next by uppercasing all the words (excluding stopwords). By this technique, many query names annotated by N/A type can be recognized.

2. *Combination Phase.* In the second phase, the system incorporates the results of all NERC systems (Stanford, Illinois, and Senna) using a combination technique and based on the majority voting. The idea behind that is that query type having the

[2]http://nlp.stanford.edu/software/CRF-NER.shtml
[3]http://cogcomp.cs.illinois.edu/page/software_view/NETagger
[4]http://ml.nec-labs.com/senna/

X	X	X	$\longrightarrow$	X
X	X	Y	$\longrightarrow$	X
X	Y	Z	$\longrightarrow$	3rd Phase
X	X	–	$\longrightarrow$	X
X	–	–	$\longrightarrow$	X
–	–	–	$\longrightarrow$	3rd Phase

Table 3.1 The rules used in the Combination phase. X, Y, or Z illustrates PER, ORG, and GPE and "−" represents "N/A" or "MISC" query types.

most agreement (at least two NERC systems) is likely the correct type for that query name. As shown in Table 3.1 this module recognizes the query type using the combination rules. For instance, consider the result of first (recognition and Classification) phase as follows: "`[NERC1:PER; NERC2:PER; NERC3:ORG]`", the result of combination phase would be "`[CNERC:PER]`" given that the PER type has the major agreement. As another example, considering "`[NERC1:ORG; NERC2:N/A; NERC3:MISC]`" the result of combination phase would be "`[CNERC:ORG]`". In the case that all NERC systems have three different answers or all answers are N/A or MISC, the system will classify the query name in the third phase.

3. *Amendment Phase.* The third phase modifies the results of the second (Combination) phase using predefined rules such as *pattern extraction and matching* rules. For instance, we have provided a set of patterns containing general organization terms (such as "`group`", "`team`", "`Inc.`", "`Institute`", "`School`", "`Center`", "`Foundation`") (Table 3.2-1). Those query names that contain such terms are modi-

#	**Pattern**	**System Act**	**Example**
1	X_1 or X_2 or $X_3 \in$ {ORG terms}	RCNERC (Amendment phase)	"`Apple Inc.`"
2	[X] ⟨ORG relation⟩ [Y]	RCNERC (Amendment phase)	"`Spain vs England`"
3	[GPE X], [GPE Y]	Candidate Filtering	"`London, England`"
4	[GPE X]{...}[GPE Y]	Candidate Filtering	"`Barcelona ... Spain`"

Table 3.2 The patterns used for type amendment and candidate filtering.

fied and tagged as "ORG". As an example, the query name "`Apple Inc.`" is tagged as ORG, given that it contains a general organization term ("`Inc.`"). In addition, there is a challenging case for most existing NERC systems. This problem occurs when some entities are referred to by their simple form. For instance, in the string "`Spain vs England`", the query names "`Spain`" and "`England`" should refer to organizations, especially, sport teams like "`Spain national football team`". But the existing NERC systems often detects "`Spain`" or "`England`" as a geo-political entity. We have used pattern matching techniques for facing this problem (Table 3.2-#2). We consider a text window of size ± 30 offsets around each query name. In this window if the system detects the organization patterns like "`[X] vs [Y]`", or "`[X] won [Y]`", the query names X or Y are re-annotated as ORG. Besides, many query names, especially in discussion fora (forum, blogs) refer to unknown persons, e.g., "`hawk2005`" or "`sunboy_US`". This query names should be tagged by PER type. The NERC systems often tag them as MISC or N/A. Thus at the end of amendment phase if the system could not categorize a query name using the predefined sets of organizations and geo-political names, it is annotated as PER type.

Alternate Name Generation During the study we inferred that many query names have more than one informative form in the same target document. For instance, consider the query name “Barack". Its expansion “Barack Obama" occurring in the same target document can be used as an Alternate Name (AN) for this query name. Thus, instead of searching for all “Barack" in the KB, we only search for the candidates with the title name “Barack Obama". This technique helps to filter-out many noisy candidates can cause occurring a mistake in detecting true answer. We added the AN generation module to reduce our searching domain in the KB and to make the process more accurate. ANs can considerably reduce the ambiguities of the query name and improve accuracy, under the assumption that two name variants in a same document can refer to the same entity. Besides, AN generation effectively contributes on an improvement of the recall allowing the identification of KB candidates whose names are distant from the original query name (especially, acronyms and nicknames). This module generates a set of name variants for each query name following the techniques below:

1. *Document-based Acronym Expansion.* Acronyms form a major part of query names and can be highly ambiguous. For instance, “ABC” is referred to around 100 entities. As another example, “JT” can refer to either “Justin Timberlake" an American singer and actor, “James Taylor" lead singer of Kool and the Gang, or “Jersey Telecom“ the Jersey telephone company. The purpose of acronym expansion is to reduce its ambiguity. We consider two expansion mechanisms. The first one reformulates acronyms according to textual patterns, e.g., finding expressions like “Congolese National Police (PNC)”, or “PNC (Congolese National Police)”. In some cases, the distance between the acronym and its full names is long. Thus, we apply the second strategy; the system explores inside the target document to gather all subsequent words with the first capital letters in order and matched with the acronym letters. For instance, “American Broadcasting

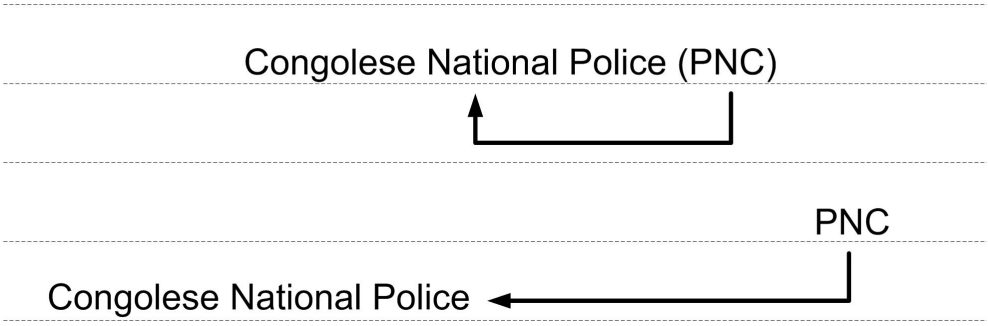


Fig. 3.3 Two techniques for acronym expansion.

Company" can be mapped to "ABC", both occurring in the same target document. In this case, supposing q as query name, the expansion has the following characteristics:

- the number of capitalized words is $len(q)$.
- the first letter of the words inside the expansion form occurs in q.
- the first letters of the words inside the expansion form are in the same order as they are in q.
- the lowercase word can only be one of the "for", "and", "in", "of" and "at".

2. *Gazetteer-based Acronym Expansion.* In some occasions, the query names can be expanded using external mapping gazetteers in the form of $\langle$*abbreviation, expansion*$\rangle$ such as:

   - the US states mappings such as the pair <"CA", "California"> or <"MD", "Maryland">.
   - Country abbreviation mappings such as the pairs <"UK", "United Kingdom">, <"US", "United States">, or <"UAE", "United Arab Emirates">.

3. *Redirect and Nickname Mapping.* We have provided a dictionary of mappings in order to redirect the query names to their more informative form. As an exam-

ple, “president Obama" can be mapped to the “Barack Obama" the US president. Further, a query name may indirectly refer to an entity. For instance, the query names “the Big Kiwi" and “Greek Freak" refer to the American basketball players “Steven Adams" and “Giannis Antetokounmpo", respectively. Given that many existing nicknames are in the Wikipedia, we have developed a module to automatically extract them from the Wikipedia documents. This module detects the nicknames using pattern extraction technique. For instance, in the sentence “Kenny Satterfield nicknamed Andersen ‘Birdman’ for his arm span", the “Andersen ‘Birdman’" is considered as the nickname for “Kenny Satterfield".

4. *Google Crosswiki Dictionary.* A query name may contain orthographic irregularity (e.g., “Equador" is the wrong form of “Ecuador", a country in South America) or have partial form of its entity name (e.g., “Barca" the abbreviated form of “FC Barcelona"). These query names usually exist in less-structured target documents such as discussion fora. To get more discriminating form of a partial query name, we use Google Crosswiki dictionary. It keeps a huge amount of IR data related to the Google search queries. For each query, there are a sort of suggested named entities, each of which associated with a confidence score. As an example, supposing the query “Man U", the first suggest in the dictionary is “Manchester United F.C." with a confidence score higher than other entities.

3.2 Candidate Generation and Filtering

After preprocessing step, the system explores a set candidates for each query name in two steps. At the outset, a set of initial candidates is generated more focusing on recall. Next, we

Algorithm 1: Candidate Generation
Input:
q : *query name* $V_q = \{v_q^1, \ldots, v_q^1\}$: *set of ANs containg q iteself.* K : *reference KB* $C = \{\}$: *set of candidates* *thr* : *Dice threshold*
Process:
1: *for* $v_q^k \in V_q$: 2: *for* $e \in K$: 3: *if* $Dice(e, v_q^k) \geq thr$: 4: $C.append(e)$ 5: *return* C

(a)

Algorithm 2: Candidate Filtering
Input:
$C = \{c_1, \ldots, c_n\}$: *set of candidates* *Boolean* F_f : *Filtering function*
Process:
1: *for* $c_i \in C$: 2: *if* $F_f(c_i)$: 3: $C.remove(c_i)$ 4: *return* C

(b)

Table 3.3 The Algorithms used for the Candidate Generation step (a) and for the candidate filtering step (b).

filter-out some candidates by applying several matching techniques. Details of each step are provided next.

Candidate Generation. A KB always contains a huge number of entities. It is impractical a brute-force searching to consider whole entities in the KB to discover the best match to each query names. Thus, candidate generation step is a solution to retrieve the most important entities in the KB that can potentially be candidates for the query name. The set of candidates generated in this step likely contains the correct entity, if it exists. Table 3.3a shows the algorithm used for the candidate generation step. Given a query name, a set of candidates is found by fuzzy name matching similarity. The system retrieves those entities in KB whose names are similar enough, using Dice measure, to one of the name variants of the query found by the AN generation step. In our experiments we used a similarity threshold of 0.9, 0.8 and 1 for PER, ORG and GPE, respectively.

Candidate Filtering. During the candidate generation step, our priority was retrieving the candidates as much as possible to ensure the existence of the correct candidate in the set of whole candidates. Indeed, that step is less focused on the precision and more on boosting the recall. In the candidate filtering step, we filter-out those candidates which cannot be a true answer as shown in Table 3.3b. This technique helps the EL system to select the best candidate. The system faces this step using three filtering techniques:

1. *Title Matching.* This technique removes the unmatched candidates by defining a set of noisy terms for each type. We define the noisy set containing terms like "`group`", "`team`" for GPE type. If the system infers that the query type is GPE, then, all candidates which their titles contain the noisy (organization) terms are removed from the set of candidates. Likewise, we can use this set to remove redundant candidates for a query name recognized as PER. In general, each general organization terms (e.g.,

"`Inc.`", "`team`", "`company`") can be considered as a noisy term in the title of the candidates when the query type (recognized by the NERC system) is PER or GPE. Thus, the system eliminates those entities from the set of candidates. For instance, suppose the query name "`Liverpool`" with the GPE type inferred by the NERC system. The system removes the candidate "`Liverpool railway station`" from the set of candidates given that the "`railway`" and "`station`" are considered as noisy terms for a query with the GPE type.

2. *Type Matching.* In our experiments, the entities existing in the reference KB are usually associated with their types. The types are PER, ORG, GPE, and UNK[5] . The system extracts each candidate type from its corresponding KB document[6] and compares each candidate type with the query type recognized and classified by our NERC system. It thereupon eliminates those candidates having different types. As an example for this step, consider "`England`" as a query name in the target documents. The system retrieves two candidates in the candidate generation step : "`England`" as a country and next as a football team. If the system recognizes its type as ORG, then, the candidate with the GPE type ("`England`" as the country) will be removed from the set of candidates.

3. *Pattern Extraction and Matching.* Our pattern extraction technique is useful to discriminate query names and improve the accuracy of the system in discovering the correct entities. This technique has a high impact, especially, for query names with GPE types. Consider the query name X occurring in the pattern "`[GPE X], [GPE Y]`" (Table 3.2-#3). We have previously provided a gazetteers of cities, states, and countries. If X exists in the gazetteer containing the city names and Y exists in the

[5] *UNK* indicates unknown query types.

[6] The reference KB (Wikipedia) contains for each entry an associated document that we name its content as *wikitext*.

gazetteers containing the state or country names, and also the pattern "`[X], [Y]`" exists as a candidate in the set of candidates, then the candidate filtering step is applied. For instance, assume the query name X as "`London`" and Y as "`England`". If "`London, England`" exists in the set of candidates, other candidates referring to "`London, Ontario, Canada`", "`London, California`", or "`London, Ohio`" are removed form the set of candidates. As another example, if the system discovers the NE mention "`Spain`" in the same document where the query name "`Barcelona`" exists, then, the system infers that the correct candidate for this query name is "`Barcelona, Spain`" (Table 3.2). Other entities such as "`Barcelona, Arkansas`" and "`Barcelona, Cornwall`" will be removed from the set of candidates (Table 3.2-4). Consequently, the ranking step will be ignored since the pattern would be considered as strong evidence for that query name.

3.3 Candidate Ranking

After generating the set of candidates the system selects the candidate that is a correct reference for that query name. This step ranks candidates based on the degree of similarity between the set of candidates and each query name (Table 3.4). To this purpose, we have proposed an unsupervised disambiguation approach that combines two graph-based methods. The first one uses the sentence in which the query name occurs (local ranker) and the second (global ranker) exploits the information in the text window in size of ± 1 sentence of the query sentence[7] (totally three sentences). The reason to select three sentences around the query name is that we considered the semantic relatedness of this text window highly coherent. The details of each method are provided next.

[7]The query sentence is a sentence where the query name occurs.

Algorithm 3: Candidate Ranking
Input:
q : *query name* $C = \{c_1, \ldots, c_n\}$: *set of candidates.* $Link_q$: *the correct candidate for query name* q.
Process:
1: *if* $C == \emptyset$: 2: $link_q = nil$ 3: *else* : 4: *for* c_k *in* C : 5: *if* $c_k = ArcMax\ Sim(q, C)$: 6: $link_q = c_k$ 7: *return* $link_q$

Table 3.4 The Algorithm used for Candidate Ranking step.

3.3.1 Candidate Ranking using Local Information

This section describes an unsupervised graph-based approach using local information occurring in the target document (*local ranker*). In our experiments, the *local information* refers to the data existing in a sentence where the query name occurs. The hypothesis behind it, is based on this idea that a semantic relation exists between query name and each NE mention (the pair ⟨query name, NE mention⟩) in the same sentence. The system uses these semantic relations to rank candidates. To this end, it extracts the context between each pair in the same sentence. A binary vector (a row matrix) is then assigned to the context elements (bag of lemmas) between each pair. In order to rank the candidates, the system generates a star graph for the query name and for each candidate. The system computes the similarity between

query graph and each candidate graph. The goal is to select the most similar candidate to the query name. Central vertex of query graph is labeled with the query name and central vertex of each candidate graph is labeled with the candidate name. Other vertices in the graphs are labeled with those NE mentions existing in the set of pairs. Each edge is labeled with the semantic relation existing between the linked entities. This is represented by a binary vector corresponding to each pair. The system ranks each candidate based on the degree of similarity between query graph and each candidate graph. Following detail of each step is explained.

Binary Vector Generation. In this step, we exploit the context between components of all the pairs ⟨query name, NE mention⟩ occurring in the query sentence of the target document, and all sentences of each candidate document (Wikitext). Our hypothesis is based on the fact that the context between the components of each pair contains discriminating information and can be used to rank candidates. The system only considers the query sentence instead of all sentences in the target document. Consider a query name q along with its target document d_q in which the query name occurs, the query sentence s_o and a set of NE mentions occurring in the query sentence $M_{s_o} = \{m_q^1, \ldots, m_q^r\}$. The system extracts each pair λ_i composed by the query name q and each NE mention $m_q^i \in M_{s_o}$:

$$\lambda_i = \langle q, m_q^i \rangle \tag{3.1}$$

where, $i \in \{1, \ldots, |M_{s_o}|\}$. Consider the set of candidates $C = \{c_1, \ldots, c_n\}$. Let the set of all sentences in each candidate's document $S_c = \{s_c^1, \ldots, s_c^t\}$, and a set of NE mentions in each candidate's document $M_c = \{m_c^1, \ldots, m_c^u\}$, the system extracts each pair λ consisting of query

name and a mention both occurring in the same sentence:

$$\lambda_{i,j,k} = \langle q_{i,j}, m_{i,j,k} \rangle \tag{3.2}$$

where, $i \in \{1,\ldots,|C|\}$, $j \in \{1,\ldots,|S_c|\}$, and $k \in \{1,\ldots,|M_j|\}$, i.e., $m_{i,j,k}$ is the k-th mention in j-th sentence of i-th candidate. As an example to show how the binary vectors are made, consider the following sentences:

> "Toyota was the largest automobile manufacturer in 2012 (by production) ahead of the Volkswagen group and General Motors.",
>
> "Toyota was started in 1933 as a division of Toyoda Automatic Loom Works devoted to the production of automobiles under the direction of the founder's son, Kiichiro Toyoda.",

Assuming the query name as "Toyota", the pairs from the first sentence would be:

$$\lambda_1 = \langle \text{“Toyota”}, \text{“Volkswagen} \rangle$$
$$\lambda_2 = \langle \text{“Toyota”}, \text{“General Motors} \rangle,$$

and from the second sentence are:

$$\lambda_3 = \langle \text{“Toyota”}, \text{“Toyoda Automatic Loom Works} \rangle$$
$$\lambda_4 = \langle \text{“Toyota”}, \text{“Kiichiro Toyoda} \rangle,$$

The context between occurrences q and m defined as follows:

$$W_\lambda = w_1 w_2 \ldots w_n, \tag{3.3}$$

for the example above the context between the elements of the pairs are defined as:

W_{λ_1}="`was the largest automobile manufacturer in 2012 (by production) ahead of the`",

W_{λ_2}="`was the largest automobile manufacturer in 2012 (by production) ahead of the Volkswagen Group and`",

W_{λ_3}="`was started in 1933 as a division of`",

W_{λ_4}="`was started in 1933 as a division of Toyoda Automatic Loom Works devoted to the production of automobiles under the direction of the founder's son,`",

the word sequence W_λ is then lemmatized, and all stopwords are removed. The system gathers all lemmas of all pairs together to create a bag of lemmas.

$$L_\lambda = \{l_1, l_2, \ldots, l_y\} \tag{3.4}$$

$$L_T = \bigcup_{\lambda \in \Lambda} L_\lambda \tag{3.5}$$

where L_λ is a bag of lemmas of each pair λ, Λ is a set of all existing pairs, L_T is a bag of lemmas of all pairs Λ. For our example, L_λ and L_λ are as following:

L_{λ_1}={"`be`", "`large`", "`automobile`", "`manufacturer`", "`2012`", "`production`", "`ahead`" }

L_{λ_2}={"`be`", "`large`", "`automobile`", "`manufacturer`", "`2012`", "`production`", "`ahead`", "`volkswagen`", "`group`"}

L_{λ_3}={"`be`", "`start`", "`1933`", "`division`"}

L_{λ_4}={"be", "start", "1933", "division", "toyoda", "automatic", "loom", "works", "devote", "production", "automobile", "direction", "founder", "son"}

$$\Lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4 \ldots\}$$

L_T={"be", "large", "automobile", "manufacturer", "2012", "production", "ahead", "volkswagen", "group", "start", "1933", "division", "toyoda", "automatic", "loom", "works", "devote", "production", "automobile", "direction", "founder", "son", ... }, Next, the system generates for each pair λ_i a vector of features using the bag of lemmas (binary vectors). For doing so, the system generates a binary vector (a row matrix) φ_i assigned to pair λ_i (Equation 3.6). Following the distributional hypothesis our claim is that φ_i represents the semantics relation between q and m. The value of each element of the vectors is initially set to zero. The number of vectors is equal to the number of pairs ($|\Lambda|$) and the number of elements of each vector, i.e. the dimension of the semantic space, is equal to the number of lemmas in L_T ($|L_T|$). For each vector, if the system finds same lemma in both bag of lemmas (L_T) and the corresponding L_λ, the element of that vector is set to one (Equation 3.6).

$$\varphi_i = \begin{matrix} l_1 & l_2 & \ldots & l_d \end{matrix} \\ \begin{bmatrix} b_i^1 & b_i^2 & \ldots & b_i^d \end{bmatrix} \tag{3.6}$$

Each element in vectors is equal to:

$$b_i^j = \begin{cases} 0 & \text{if } l_j \text{ not in } L_{\lambda_i} \\ 1 & \text{if } l_j \text{ in } L_{\lambda_i} \end{cases} \tag{3.7}$$

In the example we can consider the following binary vectors:

$$\varphi = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ \vdots \end{array} \begin{bmatrix} \begin{array}{c}\text{large}\\\text{automobile}\end{array} & \text{manufacturer} & 2012 & \text{production} & \text{ahead} & \begin{array}{c}\text{volkswagen}\\\text{group}\end{array} & \text{start} & 1933 & \text{division} & \text{toyoda} & \begin{array}{c}\text{automatic}\end{array} & \text{loom} & \text{works} & \text{devote} & \text{direction} & \text{founder} & \text{son} & & \dots \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix}$$

The Interpretation of the matrix below is, whether the *i*th lemma from the bag of total lemmas exists in the set of lemmas belongs to each pair. If yes, the corresponding value of the matrix is equal to 1, otherwise, 0.

Graph Generation. In order to rank the candidates, the system generates a star graph for the query $G_q = (V_q, E_q)$, and for each candidate, $G_c = (V_c, E_c)$, in which V and E are the sets of vertices and edges respectively (Figure 3.4). This step aims to generate a graph structure to measure the similarity between G_q and each G_c in order to select the most similar candidate to the query. Central vertex of G_q is labeled with the query name and central vertex of each G_c is labeled with the candidate name. Other vertices in G_q and G_c are labeled with those NE mentions existing in the set of pairs $\lambda \in \Lambda$. Each edge is labeled with the semantic relation existing between the linked entities which is represented by a binary vector φ corresponding to each pair λ. Given that we are dealing with star graphs, each graph G_i with central vertex i can be represented as the list of pairs (e_j, v_j) for all outcoming edges of

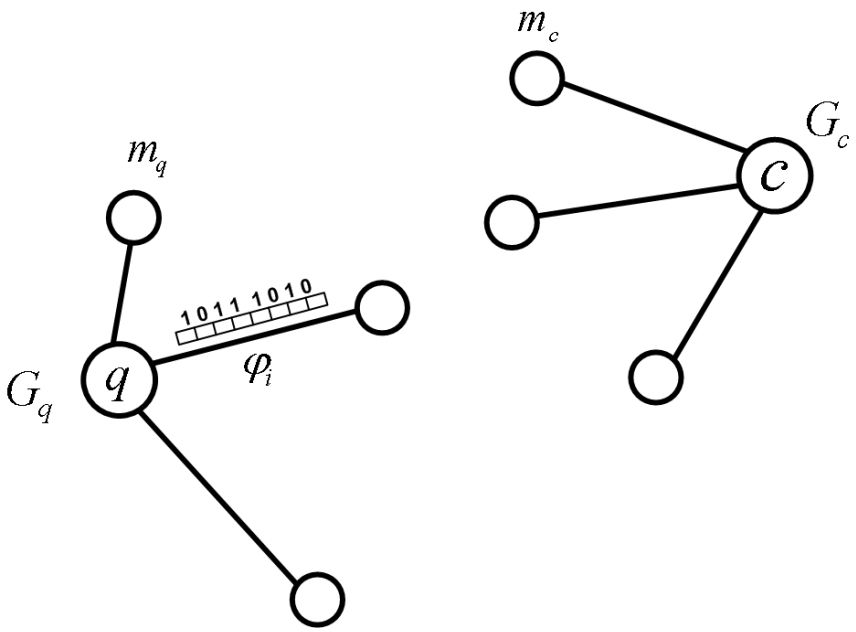


Fig. 3.4 A sample graph structure.

vertex i (Equation 3.8).

$$G_i := \{\langle e_j, v_j \rangle\}_i = \{\langle \varphi_j, m_j \rangle\}_i \tag{3.8}$$

Graph Ranking. For ranking candidates, each $G_{c \in C}$ is scored based on the similarity between G_q and G_c which is equal to the degree of similarity between outcoming edges of both graphs G_q and G_c (Equation 3.9).

$$Sim(G_q, G_c) = Sim(\{\langle \varphi_i, m_i \rangle\}_q, \{\langle \varphi_j, m_j \rangle\}_c) \tag{3.9}$$

In order to calculate $Sim(G_q, G_c)$, the system first compares the similarity β between each vertex $m_i \in G_q$ and each vertex $m_j \in G_c$ (except q and c_i) using *Levenshtein distance* metric

(Equation 3.10).

$$\beta_{m_i,m_j} = 1 - \frac{lev_{m_i,m_j}}{|m_i| + |m_j|} \tag{3.10}$$

where β_{m_i,m_j} is the degree of similarity between $m_i \in G_q$ and $m_j \in G_c$, and lev_{m_i,m_j} is *Leveshtein* metric for measuring the difference between two strings m_i and m_j, and $|m_i|$ and $|m_j|$ are lengths (number of characters) of m_i and m_j, respectively. For instance, if m_i="`Barcelona`" and m_j="`F.C. Barcelona`", then, $lev_{m_i,m_j} = 5$ and $|m_i| + |m_j| = 23$, therefore, $\beta_{m_i,m_j} = 1 - \frac{5}{23} = 0.78$. In addition, the system compares the similarity β between each edge $\varphi_i \in G_q$ and each edge $\varphi_j \in G_c$ using *Dice* metric (Equation 3.11).

$$\beta_{\varphi_i,\varphi_j} = dice_{\varphi_i,\varphi_j} = \frac{2T_{i,j}}{T_i + T_j} \tag{3.11}$$

where $\beta_{\varphi_i,\varphi_j}$ is the degree of similarity between $\varphi_i \in G_q$ and $\varphi_j \in G_c$, and $dice_{\varphi_i,\varphi_j}$ is the function to calculate *dice* coefficient between φ_i and φ_j, $T_{i,j}$ is the number of positive matches between vectors φ_i and φ_j, and T_i and T_j are the total number of positive presences in the vectors φ_i and φ_j respectively. For instance in Equation 3.11, suppose that φ_i=[1110001010][8] and φ_j=[0010001011], then, $dice_{\varphi_i,\varphi_j} = \frac{2\times3}{9} = 0.66$. Furthermore, for each G_c the system generates a set of links $H_{q,c} = \{h_1, \ldots, h_f\}$, each link h between vertices m_q and m_c. As shown in Figure 3.5, each link h has attached weight α. To calculate the value of each α, the system combines the similarities β_{m_i,m_j} and $\beta_{\varphi_i,\varphi_j}$ (Equation 3.12).

$$\alpha_{h \in H_{q,c}} = \beta_{m_i,m_j} + (1 - \beta_{m_i,m_j})\beta_{\varphi_i,\varphi_j} \tag{3.12}$$

[8] In this example, we assumed $|\varphi| = 10$, however in the real samples, $|\varphi|$ is much more than this amount (usually, $|\varphi| > 100$).

Subsequently, to score each candidate, the average of all α values for each G_c is obtained:

$$Sim(G_q, G_c) = X_{c \in C} = \frac{\sum_{h \in H} \alpha_h}{|H|} \tag{3.13}$$

where $X_{c \in C}$ indicates the score obtained by the candidate $c \in C$. The system then selects that candidate having the highest score as the correct reference of the query (Equation 3.14).

$$answer_q := \{z \in C_q | \forall c \in C_q : X_c \leq X_z\} \tag{3.14}$$

where $answer_q$ indicates the entity in the KB to which the query refers.

3.3.2 Candidate Ranking using Global Information

In previous approaches, we took advantage of semantic relation between the query name and NE mentions in the query sentence. However, in many cases the query name is the only NE mention existing in the query sentence. In other words, in these cases the query sentence does not contain enough evidence to disambiguate the query name. For these cases we have proposed a graph-based approach based on global information in the target document (global ranker). In this approach, we consider the fact that NE mentions existing in a document are usually coherent. They form an inter-related semantic network and each group of mentions can be clustered by one or more topics. Furthermore, in a document with different and distinct subjects, the mentions are usually more correlated whenever their offsets in the document get closer. Thus, to disambiguate a query name we extract this network between the NE mentions existing in the target document. To this objective, we present an unsupervised approach to disambiguate NE query names. Our system generates a network of relations using a graph-based method and based on semantic similarity between the NE mentions.

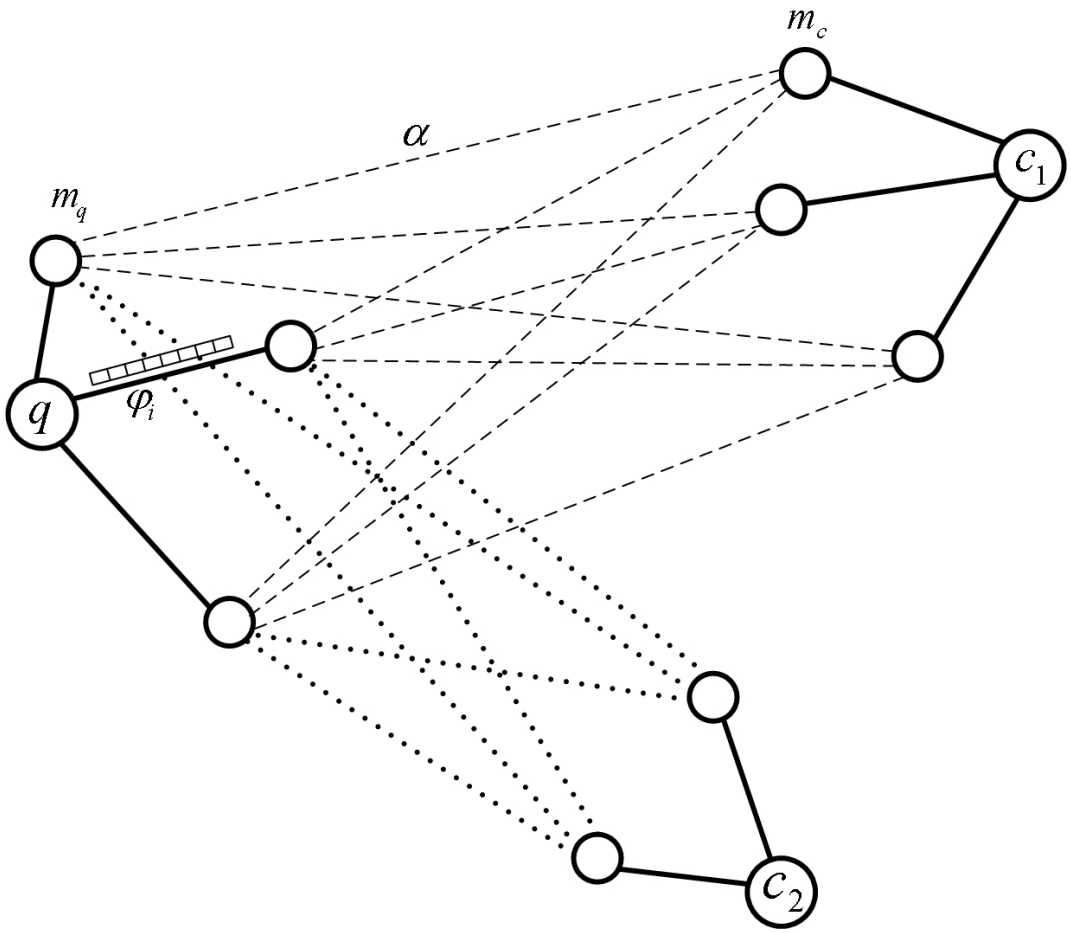


Fig. 3.5 A sample graph structure with α relation.

Graph Generation. In the graph generation step, the system generates a set of graphs for the query name q and each candidate c_i. The vertices are NE mentions (except the query name) extracted from target document and each candidate's document. Each edge is a semantic relation between each two vertices. We measure the relation degree between each two vertices using the semantic similarity between them. To measure the semantic similarity, we apply *DISCO* (extracting DIstributionally related words using CO-occurrences). The similarities are based on the statistical analysis of very large text collections. The detail is provided next.

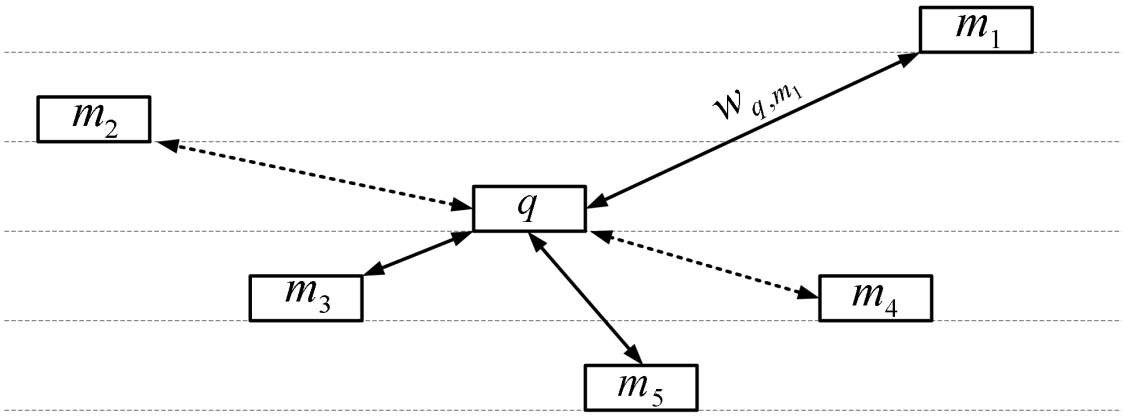


Fig. 3.6 Extracting those NE mentions having significant semantic relation with query name q. The dotted lines represent weak semantic relations less than the predefined threshold (in our experiments, set to 0.01).

(a) *Query Graph Generation.* Consider query name q along with its target document d_q in which the query name occurs and with the start and end offsets of the query name. We consider a text window (± 1 sentence around the sentence containing q). We consider the text window to filter out those NE mentions that are not relevant to q. We extract all possible NE mentions $M_q = \{{m_q}^1, \ldots, {m_q}^n\}$ from the text window. As shown in Figure 3.6, the system computes the semantic similarity between the query name and each mention $\langle q, {m_q}^i \rangle$. To compute the semantic similarity we apply DISCO. For instance, in our experiment the similarity between the pair $\langle Barcelona, Spain \rangle$ measured by DISCO is 0.061. The system then selects those NE mentions having a degree of similarity more than a threshold (in our experiments, set to 0.01) with the query name. It helps to eliminate those NE mentions without enough semantic similarity from the set of NE mentions. Next, we generate the query graph $G_q = (V_q, E_q)$ where the V_q is the set of NE mentions ($V_q = M_q$) and E_q is the set of semantic relations (labeled by weight w), each of which between two vertices in G_q. Furthermore, all edges without semantic relation (aka., $w = 0$) are disjoined and all single vertices (the vertices without any incoming edge) are eliminated.

(b) *Candidate Graph Generation.* Consider each candidate c associated with its document d_c. The system extracts the set of all NE mentions $M_c = \{{m_c}^1, \ldots, {m_c}^k\}$ existing in the first 10 sentences of d_c. Similar to the the query graph generation step, we compute the semantic similarity between the query name q and each NE mention $\langle q, {m_c}^j \rangle$. The system next removes those mentions with a similarity less than threshold (0.01). Each candidate's graph $G_c = (V_c, E_c)$ is then generated where the V_c is set of NE mentions in each candidate's document ($V_c = M_c$) and E_c is set of semantic relations each of which between two vertices in G_c. All edges without semantic relation are disjoined and all single vertices are eliminated. Figure 3.7a shows a set of graphs generated for the query name and each candidates (in this sample, two candidates).

Graph Ranking. Ranking the candidates is the most crucial task in an EL system. In this step, the system detects the most relevant candidate for each query based on the semantic similarities between the topics of the query graph and each candidate's graph.

(a) *Topic Selection.* In each graph, we compute the input degree centrality for each vertex. It recognizes the most important vertices as topics for the query name. To this end, we compute the degree centrality for each vertex v as follows:

$$C_D(v) = deg(v) = \frac{\sum_{e \in E^*} w_e}{|V| - 1} \tag{3.15}$$

where $|V|$ is total number of vertices in each graph and E^* is the set of incoming edges to this vertex v and w_e is the weight of each incoming edge. In each graph, the set of n-top vertices (having the highest degree centrality) is considered as the set of topics relevant to the query (i.e., a topic is simply a node having a high input degree

centrality.):

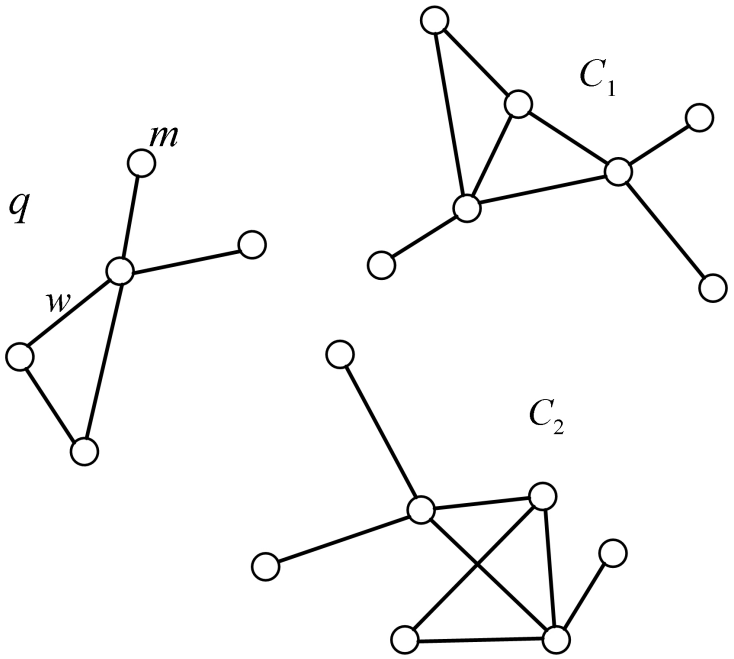
$$T_G = \{V'_{|n|} \subseteq V_G \mid \forall v \notin V',\ deg(v'_{i-1}) \geqslant deg(v'_i) \geqslant deg(v)\} \tag{3.16}$$

where T_G is the set of topics and n is the number of topics which is the same in all graphs. In other words, $T_G = \{t_1, \ldots, t_n\}$ is the subset of n vertices for each graph that t_1 and t_n are the vertices having the highest and lowest degree centrality in this set, respectively. This step helps to semantically determining the most relevant NE mentions as the topics for each query. The system iterates the process to generate the set of n-topics for each graph. In Figure 3.7b, the topics are indicated as filled vertices.

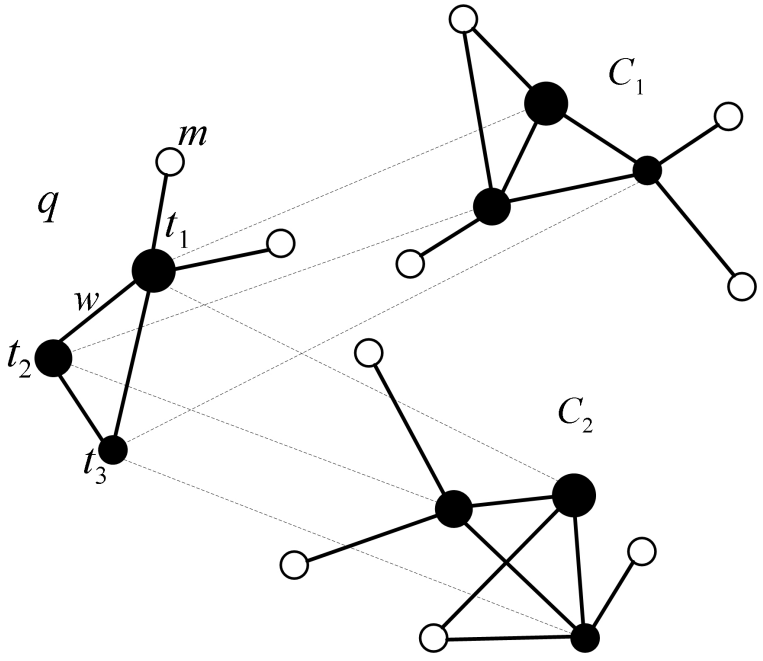
(b) *Topic Comparison.* To select the best candidate for the query, it should be inferred which candidate shares the most similar topics with the query. To this objective, we compute the semantic relations (shown as dotted lines in Figure 3.7b) between the topics of the query name and each candidate in a top-down order. It implies that the topic with the highest degree centrality in the query graph is compared with the topic having the highest degree centrality in each candidate's graph. As shown by Eq. 3.17, the total score of each candidate is the average of the semantic similarity obtained between each pair $\langle t_q, t_c \rangle$:

$$S_c = \frac{\sum_{k=1}^{n} Sim(t_k^q, t_k^c)}{n} \tag{3.17}$$

where, S_c is the score of candidate c, and t_k^q and t_k^c are the k-th topic for the query name q and candidate c, respectively. The *Sim* function computes the semantic similarity between t_k^q and t_k^c and n is the number of topics in the graphs. Finally, the system ranks the candidates based on the scores and selects a candidate having the highest score as the correct reference of that query name in the reference KB.



(a) Set of semantic graphs for the query and candidates.



(b) Topic comparison between the semantic graphs.

Fig. 3.7 An example indicating a set of graphs and also the semantic relations between the topics in the graphs.
†: n.b. the topics and the relation between them are indicated as filled vertices and dotted lines, respectively. In the Figure 3.7b, the biggest vertex indicates the first topic and the smallest one shows the last topic.

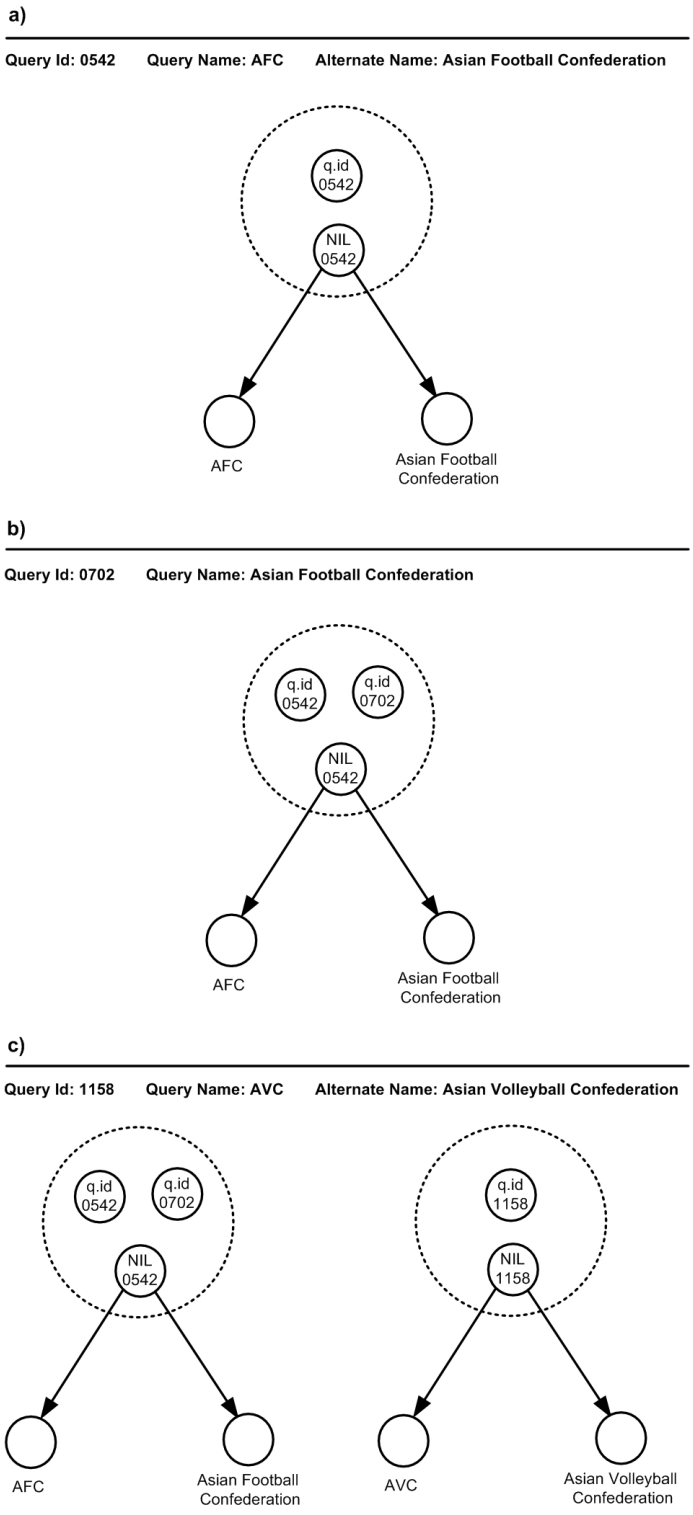


Fig. 3.8 An Example for the NIL Clustering approach.

Algorithm 4: NIL Clustering
Input:
q : *query name* $CLR = \{clr_1, \ldots, clr_m\}$: *set of nil clusters.* q_{nil} : *nil query.* thr_{nil} : *nil threshold.*
Process:
1: *for* clr_j *in* CLR : 2: *if* $Sim(q_{nil}, clr_j) \geq thr_{nil}$: 3: $clr_j.join(q_{nil})$ 4: *else* : 5: $create(clr_{new})$ 6: $id_{clr_{new}} = id_{q_{nil}}$ 7: *return* CLR

Table 3.5 The Algorithm used for NIL Clustering step.

3.3.3 NIL Clustering

Many query names refer to the entities that are not present in the reference KB (NIL queries–NIL). For those queries, the system should cluster them into groups, each referring to a same Not-In-KB entity (NIL Clustering). To this objective, a term clustering method is applied to cluster such queries (Table 3.5). At the outset, each initial NIL query forms a cluster assigning a NIL id. The system afterwards applies a fuzzy matching technique to compare the next NIL query with each existing NIL cluster using a Dice similarity. The comparison is between the properties of the new NIL query and each cluster. The properties (of the cluster or NIL query) are the query name and set of ANs corresponding to that query name. If the similarity

is higher than a predefined NIL threshold (0.8), the new NIL query obtains the identifier of this cluster, otherwise, it forms a new NIL cluster obtaining a new NIL id. In our experiments we manually selected 0.8 as NIL threshold (Equation 3.18). We applied this approach since it is simple and has a performance near to other NIL clustering approaches. Figure 3.8 shows an example for our NIL clustering approach. Firstly, consider the query name "`AFC`" with id=0542 associated with its AN "`Asian Football Confederation`" is selected as a NIL query and is referred to NIL clustering step for assigning a NIL id. Suppose that this query has no corresponding NIL cluster. It creates a new NIL cluster assigning the query id as the cluster id. In this example, both query id and NIL id is 0542 (Figure 3.8-a). The system thereupon explores the corresponding cluster for the next NIL query name "`Asian Football Confederation`" with id=0702. It computes the Dice similarity between the query name ("`Asian Football Confederation`") and each properties of all NIL clusters. Upon the first comparison using Dice metric matches, this query is associated to the cluster. In the example, the appropriate cluster for the NIL query is one with the NIL id 0542 (Figure 3.8-b). Finally, the system explores the suitable cluster for next NIL query "`AVC`" with id 1158 associated with its AN "`Asian Volleyball Confederation`". All Dice similarities between the new NIL query ("`AVC`" and its expansion "`Asian Volleyball Confederation`" and properties in the NIL cluster are less than the threshold. Therefore, the new NIL query is considered as a new cluster (Figure 3.8-c). The system iterates the process until all NIL queries are grouped to the clusters.

$$id_{nil} = \begin{cases} id_{clr} & \text{if } dice_{q,clr} \geqslant 0.8 \\ id_q & otherwise \end{cases} \tag{3.18}$$

where, id_{nil} is the Id of NIL query, id_{clr} is the Id of an existing cluster, $dice_{nil,clr}$ is Dice function applied to NIL query and existing cluster, and id_{nclr} is Id of a new cluster.

Chapter 4

Evaluation and Result Analysis

In order to evaluate the performance of the system, we have participated in an evaluation framework (TAC-KBP) which provides joint test-bed to compare the results. In this section we explain our evaluation framework by which our EL system was examined (Section 4.1) and subsequently we describe the improvements (Section 4.2.1) and analyze the results in different aspects (Section 4.2.2).

4.1 Evaluation Framework

We evaluated our system in the framework of the TAC-KBP 2014 Mono-Lingual (English) EL evaluation track[1]. With previous versions of our system we also participated in TAC-KBP 2012 [29] and TAC-KBP 2013 [1]. As the most important challenging competition, TAC-KBP EL track has been the subject of significant study over the past seven years. Since the first KBP track held in 2008[2], the research in the area of EL has greatly developed[3]. The

[1] `http://www.nist.gov/tac/`

[2] It was initiated in 2008 and developed out of NIST's Text REtrieval Conference (TREC) and Document Understanding Conference (DUC).

[3] The Text Analysis Conference (TAC) is organized and sponsored by the U.S. National Institute of Standards and Technology (NIST) and the U.S. Department of Defense.

<DOCID> eng-NG-31-108519-8977045 </DOCID>
<DOCTYPE SOURCE="usenet"> USENET TEXT </
DOCTYPE>
<DATETIME> 2007-10-10T22:25:00 </DATETIME>
<BODY>
<HEADLINE>
Dollars for Death
</HEADLINE>
<TEXT>
<POST>
<POSTER> Anybody <anybod...@canada.com> </
POSTER>
<POSTDATE> 2007-10-10T22:25:00 </POSTDATE>
Dieticians

The **American Dietetic Association (ADA)** has 67,000 members. Their motto is "Everything in moderation." That includes McDonald's, other fast food restaurants, dairy products, NutraPoison, and sugar-rich soda. Of course, the one concept that they do not limit is donations by various industry groups who delight in seeing the ADA's continuing ……. i4crob(at)earthlink.net
</POST></TEXT></BODY></DOC>

Fig. 4.1 A sample target document for the query name “ADA" from the TAC 2013 data set.

main goal of TAC-KBP track is to gather information about a specific entity that is scattered among the documents of a large collection, and then use the extracted information to populate an existing reference KB.

4.1.1 Evaluation Task Definition

Given a set of queries, each of which consisting of a query name and a target document (Figure 4.1) in which the query name occurred, and the start and end offsets of the query

name, the system should provide the identifier of the KB entity to which the query name refers if existing, or a NIL Id if there is no such KB entity. In fact many queries use the same target document. From 5234 queries in the evaluation data only 118 documents were used. This increases the difficulty of the task because in each document many mentions and their corresponding offsets are associated to different queries corresponding or not to the same entities in KB. The EL system is also required to cluster together queries referring to the same Not-in-KB (NIL) entities and to provide a unique ID for each cluster.

Each query entry will consist of the following five fields:

- **<query id>** - A query ID, unique for each entity name mention.
- **<name>** - The full name string of the query entity mention.
- **<docid>** - An ID for a document in the source corpus from which the name string was extracted.
- **<beg>** - The starting offset for the name string.
- **<end>** - The ending offset for the name string.

A sample query from the KBP2014 EL evaluation is the following one:

```
<query id="EDL14_ENG_0049">
   <name>Valerie<name>
   <docid>bolt-eng-DF-170-181103-8893099</docid>
   <beg>4361</beg>
   <end>4367</end>
</query>
```

The term "<name>" mentioned in the query sample above is equivalent to the query name (target NE mention).

4.1.2 Evaluation Metrics

In the evaluation, several queries may refer to the same entity in the KB (in-KB entities). All NIL queries referring to the same Not-in-KB entity should also be grouped in the same cluster. Thus, in both cases (in-KB and Not-in-KB entities) an EL system should cluster the queries. A modified B-cubed [3][4] metric (called B-cubed+) is applied to evaluate these clusters[5]. Consider the following equation:

$$G(e,e') = \begin{cases} 1 & \textit{iff } L(e) = L(e') \wedge C(e) = C(e') \wedge GI(e) = SI(e) = GI(e') = SI(e') \\ 0 & \textit{otherwise} \end{cases}$$

where $L(e)$ and $C(e)$ are respectively the category and the cluster of a NE mention e, SI(e) and GI(e) are the system and gold-standard KB identifier, and $G(e,e')$ is the correctness of the relation between two NE mentions e and e' in the distribution. B-cubed+ precision of a NE mention is the proportion of correctly related NE mentions in its cluster (including itself). The overall B-Cubed+ precision is the averaged precision of all NE mentions in the distribution. B-Cubed+ recall is similar to B-Cubed+ precision, replacing cluster with category. Formally:

$$\textit{Precision B-Cubed+} = Avg_e[Avg_{e'.C(e)=C(e')}[G(e,e')]]$$

$$\textit{Recall B-Cubed+} = Avg_e[Avg_{e'.L(e)=L(e')}[G(e,e')]]$$

$$\textit{F\_Measure B-Cubed+} = 2 \times \textit{Precision} \times \textit{Recall} / (\textit{Precision} + \textit{Recall})$$

4.1.3 Evaluation Data

Reference Knowledge Base. The reference KB includes hundreds of thousands of entities based on articles from an October 2008 dump of English Wikipedia, which includes 818,741

[4]The idea behind the B-cubed metric considers the EL task as a cross-document coreference task, in which the set of tuples is grouped by both in-KB and Not-in-KB entity ids.

[5]The scorer is available at: `http://www.nist.gov/tac/2012/KBP/tools/`

entries. Wikipedia has some features that can be very helpful for the task, like the descriptions associated with each entry, that can be used to help in the disambiguation process by comparing the context in which an entity appears against the context of the Wikipedia entry description. The articles also have a title that formally names the entity, which sometimes is followed by a string that discriminates entities that share the same name (e.g., Python (programming language) and Python (mythology) correspond to two different entities in Wikipedia). As shown in Table 4.1, each entry in the KB includes the following:

- a name string (like, "`Parker, Florida`")
- an assigned entity type of `PER`, `ORG`, `GPE`, or `UKN` (unknown)
- a KB entity id (a unique identifier, like "`E0000012`")
- a set of 'raw' slot names and values (facts) which is extracted from Wikipedia infoboxes.
- some disambiguating text (i.e., text from each Wikipedia document–wikitext)

KBP reference KB has been created only from those Wikipedia entries containing infoboxes. In addition, a small percentage of the Wikipedia infoboxes had abnormalities in their structure that made their infoboxes tough to parse. These entries were also eliminated from the reference KB by the KBP organizers.

Training and Evaluation Corpus. The training data in TAC-KBP 2014 is the evaluation data from the its past years. Table 4.2 shows the sources and sizes. We evaluated our system over TAC-KBP 2014 Mono-Lingual (English) EL evaluation data set. The evaluation data set includes 5,234 queries, each query consisting of a query name (target NE mention), with a target document in which query name occurs and start and end offsets of the query name inside the target document. A target document may be used for several queries often

```
<entity wiki_title="Parker,_Florida" type="GPE" id="E0000012"
name="Parker, Florida">
<facts class="Infobox Settlement">
<fact name="official_name">Parker, Florida</fact>
<fact name="subdivision_name"><link entity_id="E0679687">United
States</link></fact>
<fact name="subdivision_name1"><link entity_id="E0373950">Florida
</link></fact>
 ⋮
</facts>
<wiki_text><!  [CDATA[Parker, Florida
Parker is a city in Bay County, Florida, United States.  As of the
2010 census it had a population of 4,317.  It is part of the Panama
City-Lynn Haven-Panama City Beach Metropolitan Statistical Area.
According to the United States Census Bureau, the city has a total area
of 6.3 km² (2.4 mi²).  1.9 square miles (4.9 km²) of it is land and
0.5 square miles (1.3 km²) of it (20.16%) is water.  [ .............
]  In the city the population was spread out with 21.2% under the age
of 18, 9.1% from 18 to 24, 24.3% from 25 to 44, 28.0% from 45 to 64,
and 17.4% who were 65 years of age or older.  The median age was 40.9
years.  For every 100 females there were 94.8 males.  For every 100
females age 18 and over, there were 91.6 males.  As of the 2000 census,
the median income for a household in the city was $35,813, and the
median income for a family was $43,929.  Males had a median income of
$28,455 versus $21,205 for females.  The per capita income for the city
was $18,660.  About 10.1% of families and 12.2% of the population were
below the poverty line, including 21.3% of those under age 18 and 4.6%
of those age 65 or over.
]]></wiki_text>
</entity>
```

Table 4.1 An entry sample in the reference KB. The entry represents the geo-political entity “Parker, Florida" associated with its facts and document (wikitext).

corresponding to different offsets of the same query name. In addition, the distribution of queries per type is not uniform in the evaluation data.

4.2 Evaluation Results and Analysis

In this section, we describe the results obtained by our EL system and analyze them in different aspects. Table 4.3 illustrates our results measured by accuracy, B-cubed, and B-

Genre/Source	Size (entity mentions)		
	Person	Organization	GPE
2009 Eval	627	2710	567
2010 Training Web data	500	500	500
2010 Eval Newswire	500	500	500
2010 Eval Web data	250	250	250
2011 Eval Newswire	500	491	500
2011 Eval Web data	250	259	250
2012 Eval Newswire	702	388	381
2012 Eval Web data	216	318	221
2013 Eval Newswire	333	333	333
2013 Eval Web/Discussion Fora data	333	333	333

Table 4.2 Training data for TAC-KBP 2014 EL task.

cubed+ metrics (The metrics are explained in Section 4.1.2). We have computed precision, recall, and F1 for both B-cubed and B-cubed+ metrics. We evaluated two systems: first, our baseline system [65] by which we participated in TAC-KBP 2014 (mentioned by *BL_SYS*) and second, the results obtained by our final system (mentioned by `F_SYS`) in which we applied several improvements over BL_SYS. The table also splits the results by those query answers existing in reference KB (In-KB) and those not in the KB (NIL) also by three query types: person (PER), organization (ORG), and geo-political entity (GPE). As shown in the table, we evaluated the systems over three evaluation genres including News Wires (NW), Web Documents (WB), and Discussion Fora (DF). Both WB and DF (e.g., fora, blogs) are highly challenging given that they contain many orthographic irregularities. Below in Section 4.2.1, we explain the improvements we applied within F_SYS. Consequently in

System Results	Metrics						
	Accuracy	B^3 P.	B^3 R.	B^3 F1	B^3+ P.	B^3+ R.	B^3+ F1
All	**0.840**	**0.963**	**0.813**	**0.882**	**0.820**	**0.702**	**0.757**
In-KB	0.800	0.964	0.876	0.918	0.779	0.751	0.765
NIL	0.888	0.963	0.739	0.836	0.868	0.645	0.740
PER	0.864	0.978	0.804	0.883	0.851	0.713	0.776
ORG	0.744	0.944	0.798	0.865	0.716	0.614	0.661
GPE	0.862	0.938	0.850	0.892	0.826	0.754	0.788
NW	0.793	0.949	0.856	0.900	0.768	0.703	0.734
WB	0.846	0.959	0.792	0.867	0.822	0.689	0.749
DF	0.875	0.980	0.796	0.878	0.862	0.714	0.781

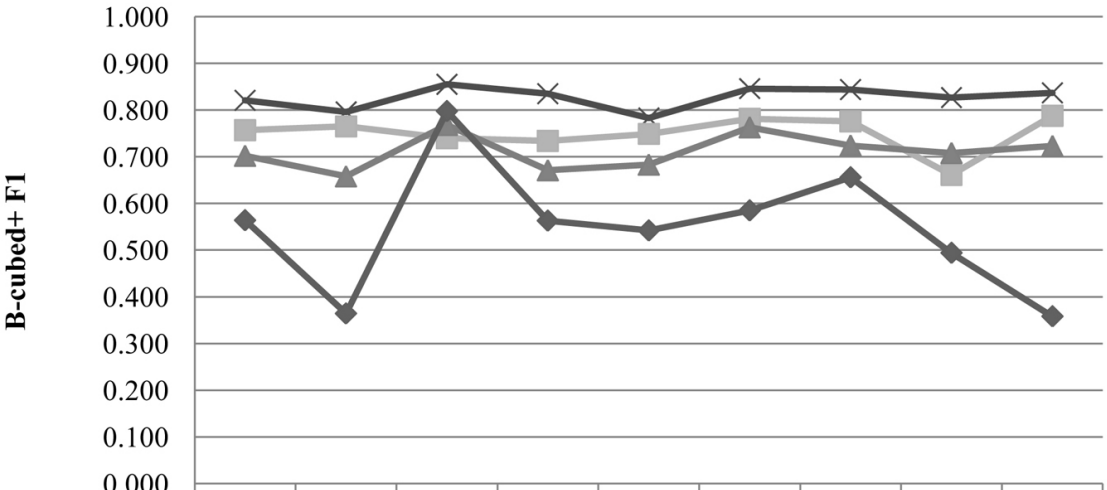
Table 4.3 The F_SYS results measured by the accuracy, B-cubed, and B-cubed+ metrics over TAC-KBP 2014 Mono-Lingual (English) EL evaluation data set.

Section 4.2.2, we analyze the results of each system in different aspects as well as the impact of each improvement on F_SYS performance.

4.2.1 Improvements

Compared with the baseline system (BL_SYS), we improved our final system (F_SYS) in several ways including:

- We applied a global ranker (candidate ranking using global information) in the cases that query sentence in the target document is not enough informative as is the case where no NE but the query name occurs in this content. The global ranker generates a query graph in which the vertices are the NE mentions extracted from a text window of 3 sentences (including query sentence)‘. It also generates a set of graph, each of



	All	in-KB	NIL	NW	WB	DF	PER	ORG	GPE
Highest	0.821	0.796	0.855	0.835	0.783	0.846	0.844	0.827	0.837
F_SYS	0.757	0.765	0.740	0.734	0.749	0.781	0.776	0.661	0.788
Median	0.702	0.658	0.768	0.671	0.683	0.763	0.724	0.708	0.723
BL_SYS	0.564	0.364	0.798	0.563	0.542	0.585	0.656	0.494	0.358

Fig. 4.2 The results of BL_SYS and F_SYS measured by B^3+ F1 and the accuracy over the TAC-KBP 2014 Mono-Lingual (English) EL evaluation data set.

which related to a candidate. The vertices in each candidate graph are the NE mentions extracted from the first 10 sentences of the candidate document. Each edge in the set of graphs is weighted with the semantic similarity between each two vertices. Although, the vertices can also consider all unigrams (such as verbs, adj.), in our experiences we only consider the NE mentions occurring in the target documents.

- We applied a dictionary of nicknames extracted from the Wikipedia. Many entities such as persons, organizations, and geo-political entities are known by their nicknames. For instance, “`Dubya`", “`The Big Apple`", and “`the Country Music Capital`" refer to “`George H. W. Bush`", “`New York City`", and “`Nashville, Tennessee`", respectively. The dictionary of nicknames helps to infer the correct reference of such query names. To provide the dictionary of nicknames,

we have previously developed a system to extract the nicknames from the content of Wikipedia documents.

- Many query names existing in the target document contain orthographic irregularities. For instance, in sentence "`Man utd vs Liverpool`", the query name "`Man utd`" is referred to "`Manchester United F.C.`" or in sentence "`Equador is country in South America`", the correct form of "`Equador`" is "`Ecuador`". To tackle this problem, we applied Google CrossWiki dictionary containing a huge amount of mapping based on the search results obtained by Google search engine.

- We applied pattern extraction and matching to recognize geo-political entities (Table 3.2). Consider the query name X occurring in the pattern "`[GPE X], [GPE Y]`". We have previously provided gazetteers of cities, states, and countries. If X exists in the gazetteer containing the city names and Y exists in the gazetteers containing the state or country names, the candidate filtering step is then applied. For instance, assuming X (query name) as "`Barcelona`" and Y as "`Spain`", other entities such as "`Barcelona, Arkansas`" and "`Barcelona, Cornwall`" will be removed from the set of candidates. In addition, we use the evidences to select correct geo-political entities. For instance, in sentence:

> "`Texas is an unincorporated community located along the border of Monroe and Old Bridge townships in Middlesex County, New Jersey, United States.`"

Consider "`Texas, New York`", "`Texas, West Virginia`", and "`Texas, New Jersey`" as the candidates for the query name "`Texas`". The system selects "`Texas, New Jersey`" as the correct reference of this query name. The NE

mention “`New Jersey`" is considered as an evidence for choosing the candidate “`Texas, New Jersey`".

- A difficult challenge in the EL task is the case that a query name can simultaneously refer to either organization or geo-political entities. For instance, in the string *“Spain vs England*", the NERC systems often detect “`Spain`" or “`England`" as geo-political entities. However, they are organizations and usually refer to sport teams. To tackle, we consider a text window of size ± 30 offsets around the query name. The system extracts the organization patterns in the text window, e.g., “`[X] vs [Y]`", or “`[X] won [Y]`" (Table 3.2). The query names X or Y are recognized as ORG and all geo-political entities are eliminated from the set of candidates.

- NERC is an important subtask in EL. In BL_SYS, we applied only one NERC system (Illinois). However, we realized that relying on just one NERC system causes reduction in the accuracy of the system. Thus, we applied a hybrid approach–RCNERC (details in Section 3.1) in F_SYS by combining three NERC systems: Stanford, Illinois, and Senna.

4.2.2 Result Analysis

In this section, we analyze the results obtained by BL_SYS and F_SYS. We evaluated both systems over TAC-KBP 2014 EL evaluation data. Figure 4.2 represents the results by BL_SYS and F_SYS compared with the median of all participants in TAC-KBP 2014 EL evaluation track and also with the team obtained the highest result. Our final system (F_SYS) could achieve a result better than the median and BL_SYS and less than the highest result. As shown in the figure, BL_SYS better detects and clusters Not-In-KB entities (NIL) than In-KB entities. By applying several improvements (described in Section 4.2.1), the accuracy of F_SYS in linking in-KB entities increased (0.364 to 0.765). The lowest results

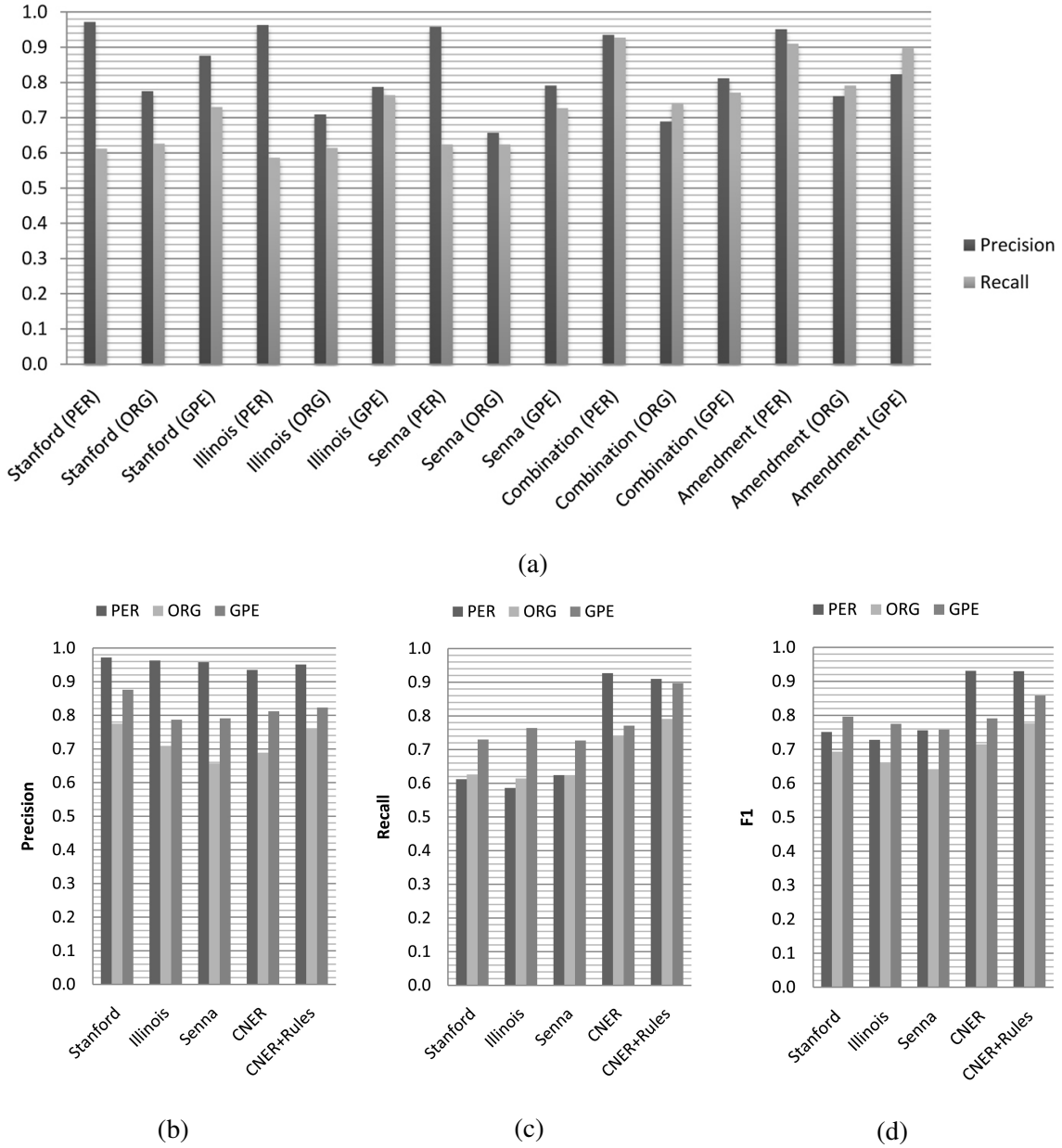


Fig. 4.3 The recall, precision and F1 of each three phases of RCNERC system.

of BL_SYS belongs to GPE queries (0.358), given that the number of candidates generated for the GPE type is more than PER and ORG and therefore is more ambiguous. In addition, the lowest result in F_SYS belongs to ORG (0.661) which is less than the median (0.708). This reduction was considerably compensated applying the pattern extraction and matching techniques (Table 3.2). F_SYS has a score higher than BL_SYS in linking In-KB queries. Since the number of GPE-In-KB queries is more than the GPE-NIL queries, it caused a better result for Overall-GPE queries in F_SYS. Besides, the highest and lowest improvements in our results (compared with B_SYS) belong to GPE (+0.430) and PER (+0.120) queries, respectively. In addition, the nearest and farthest results to the participant with the highest score belong to GPE (-0.049) and ORG (-0.166), respectively.

We also analyzed the result of RCNERC system in each phase. Figure 4.3a indicates the precision and recall of the NERC systems in the recognition phase (Stanford, Illinois, Senna), in the combination phase, and also in the amendment phase. The precision in detecting query types in the recognition phase is better than its recall. The reason is because of orthographic irregularities existing in target documents such as discussion fora. The NERC systems in recognition phase recognize them as MISC or N/A. We have solved this issue by inferring the correct query types in the combination and amendment phases. Also in this phase, the PER type has the highest difference between recall and precision and GPE has the lowest in all Stanford, Illinois and Senna NERC systems. We can consider this difference as wide diversity and highly ambiguous nature of person entities (compared with organization and geo-political entities) in the target documents, especially in discussion fora. We improved the recall and reduce the difference between them in the last phase. As depicted in the Figure 4.3b, the precision obtained for the PER type is the highest in all phases. It demonstrates that if the system could detect the person query names in the target documents, most of the time, it annotates them correctly. On the contrary, the precision for ORG type is the lowest one. Because the existing NERC systems usually have a lower precision in annotating organization

query names and often recognize them as a geo-political entity. In the recognition phase, the highest recall belongs to the GPE type, but in combination and amendment phases the PER type has the highest one. It demonstrates that PER type took advantage of the combination phase more than other types (Figure 4.3c). We also measured F1 in each phase. Figure 4.3d illustrates F1 for PER, ORG and GPE types in different phases. The F1 in last two phases for all types is higher than the first phase. It demonstrates the positive impact of our proposed three-phase RCNERC system in detecting mention types.

Figure 4.4 represents the distribution of candidates for each query in the candidate generation step (initial candidates) and candidate filtering step (filtered candidates[6]). The initial and filtered candidates are depicted by the black and gray spots, respectively. As shown in this figure, the system generates less than three candidates for most queries. We also illustrated the frequency of the queries by the number of candidates in Figure 4.5. The Figure 4.5a shows the frequencies after the candidate generating step and Figure 4.5b shows the frequencies after applying the candidate filtering step. The number of candidates was successfully reduced to just one candidate by applying our pattern extraction and matching techniques. It helps to boost the accuracy of system in detecting the correct candidate. For instance, if the query name is "`London`" and the system detects mention "`England`" in the same target document, it realizes the semantic relation. Consequently, it eliminates all other entities such as "`London, Ontario`", "`London, Arkansas`", "`London, California`", "`London, Kentucky`" and "`London, Minnesota`" from the set of candidates. In the latest version of Wikipedia (2015 dump of Wikipedia), 19 entities (considering just GPE entities) are referred by query name "`London`". This makes the process highly ambiguous. By our filtering method (using pattern matching) we eliminate other candidates and achieve just one candidate.

[6]The candidates that remain after filtering step

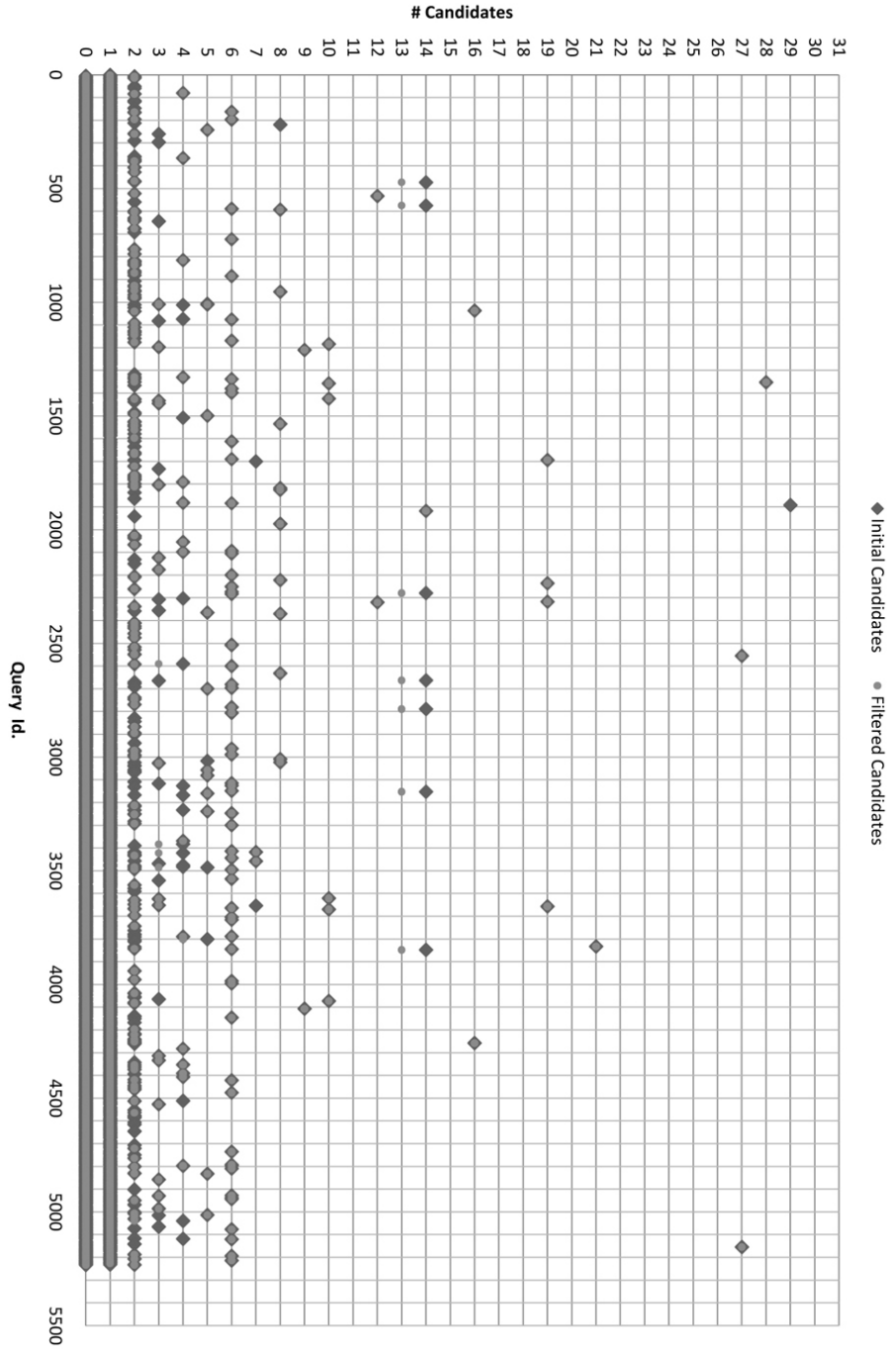
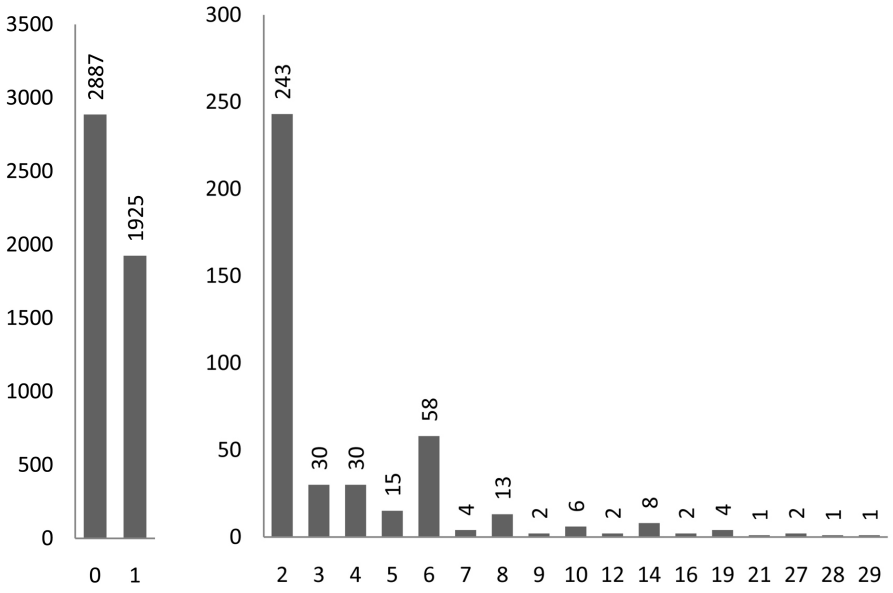
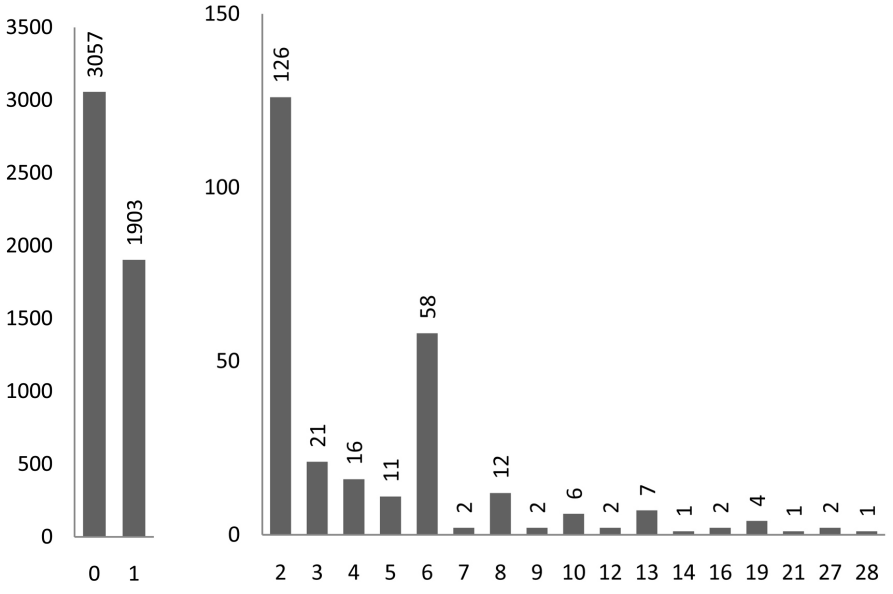


Fig. 4.4 The distribution of candidates for each query in the candidate generation step (initial candidates) and in the candidate filtering step (filtered candidates).



(a)



(b)

Fig. 4.5 Frequency of the the queries by the number of candidates.

Table 4.4 shows the accuracy error rate in both candidate generation and candidate filtering steps. This error rate indicates whether the correct answer of the EL system is among the set of candidates in both candidate generation and candidate filtering steps. We have

Accuracy Error Rate	**In-KB**	**PER-In-KB**	**ORG-In-KB**	**GPE-In-KB**
Candidate Generation	0.168	0.169	0.274	0.100
Candidate Filtering	0.008	0.002	0.027	0.007

Table 4.4 The Accuracy error rate in candidate generation and candidate filtering steps.

computed the error rate only for in-KB queries. The table separately shows the error rates for those in-KB queries which are PER, ORG, and GPE. As shown in the table, the error rate in the candidate filtering step (0.008) is much less than the error rate in candidate generation step (0.168). In the candidate generation step, the highest error rate is below ORG type (0.274) and the lowest one is below GPE type (0.100). Similarly, the highest error rate in the candidate filtering step is below ORG type (0.027) and the lowest one is below PER type (0.002). It should be mentioned that the errors in the candidate generation and filtering steps affects on the results of the candidate ranking step. Therefore, a reason to get a low score for the ORG type in Figure 4.2 is within the candidate generation step. This step generates a set of candidates using the Dice similarity. On the contrary with the PER and GPE types, those entities with the ORG type can be referred by very short mentions. We have used a trade-off between the number of candidates and the dice similarity threshold (0.8 for ORG type) in the candidate generation step. Thus, in some ORG queries the correct answer is eliminated from the set of candidates.

Further, we explain the result of candidate ranking step. This step was applied to the queries with two or more candidates, 274 over 5,234 queries. The queries with no candidates are considered as NIL (3,057 queries). In addition, for those queries with one candidate (1,903 queries), that candidate is selected as the answer of the EL task. We also evaluated the accuracy of F_SYS by two ranking approaches, first, using local ranker F_SYS Local (Section 3.3.1) and, second, using global ranker F_SYS Global (Section 3.3.2).

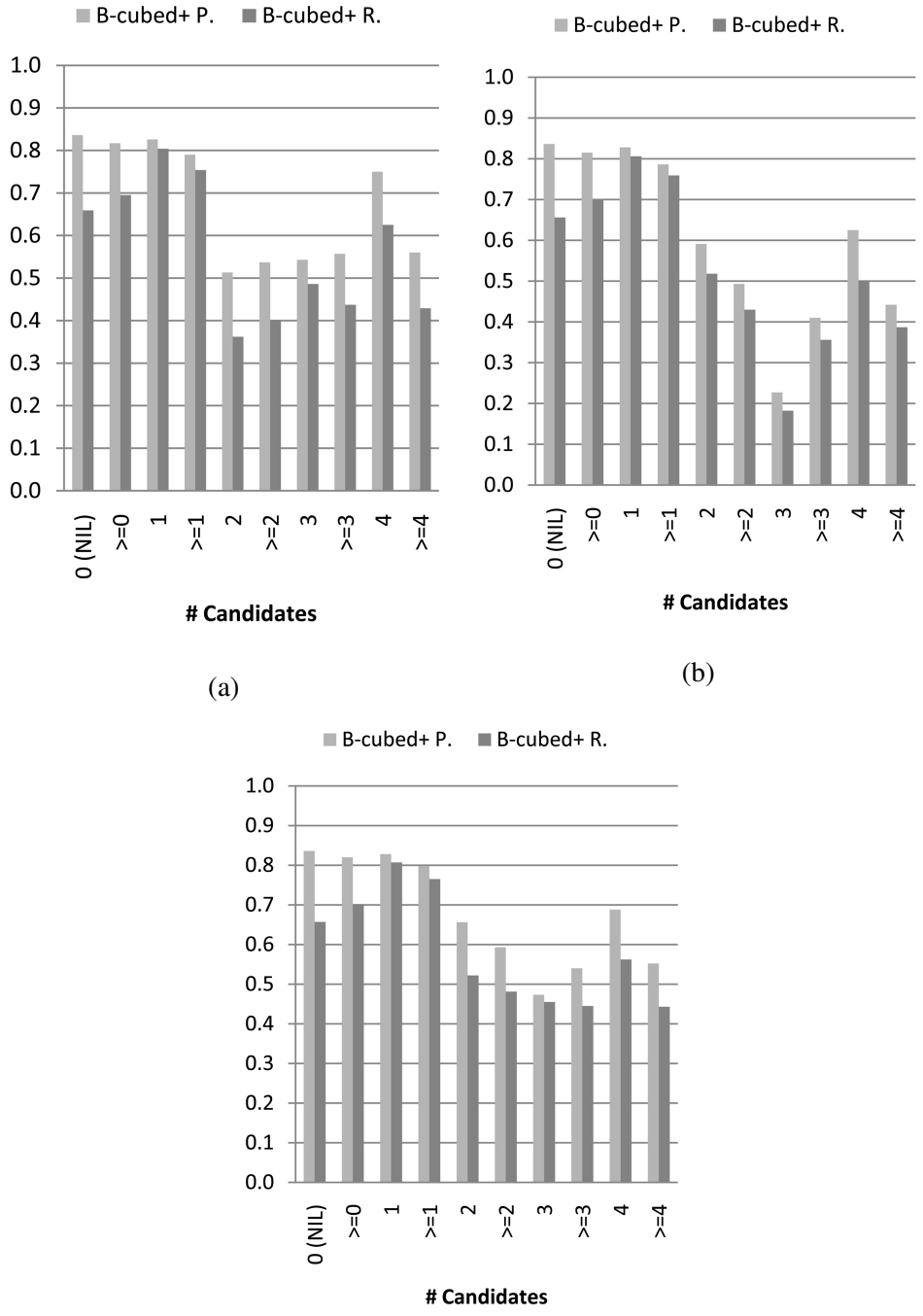


Fig. 4.6 The B^3+ precision and recall of Local (a), Global (b) and Local+Global (c) rankers by the number of candidates.

Impact on F_SYS	B^3+ F1								
	All	in-KB	NIL	NW	WB	DF	PER	ORG	GPE
Redirects	+0.033	+0.058	+0.002	+0.029	+0.061	+0.01	0	+0.028	+0.135
Nicknames	+0.030	+0.045	+0.012	+0.010	+0.014	+0.062	+0.045	+0.012	+0.003
Pattern Extraction	+0.005	+0.007	0	+0.004	+0.009	0	-0.001	+0.003	+0.022
RCNERC	+0.078	+0.138	+0.007	+0.141	+0.079	+0.024	+0.112	+0.056	0

Table 4.5 The impact of improvement modules on the F_SYS results (measured by the B^3+ F1 metric).

Figures 4.6a, 4.6b and 4.6c respectively depict B^3+ precision and recall values of Local, Global and Local+Global rankers with respect to the number of candidates. The local ranker (Figure 4.6a) has a better precision and recall for queries with 3 or more candidates while the global ranker (Figure 4.6b) has better results for the queries with 2 candidates. Using combination approach for the rankers, we improved the results in most parts. As shown in Figure 4.6c the combination of both rankers boosts precision and recall.

In addition, we separately measured the impact of each module by which we improved the results of F_SYS (compared with BL_SYS). These modules are *Redirects*, *Nicknames*, *Pattern Extraction and Matching*, and *RCNERC*. Table 4.5 depicts the result of each module measured by B^3+ metric. This table shows the module impacts in different perspectives: for in-KB and NIL queries, over three genres (NW, WB, and DF), and finally for different query types. Among different modules, the highest and lowest impacts belong to the RCNERC system (+0.078) and Pattern Extraction (+0.005). It demonstrates that our three-phase NERC system has a high impact on system's overall result. The Redirect and Nickname modules almost had the same impact on the F_SYS (+0.03). Besides, the modules have higher impacts on in-KB queries compared with NIL queries. Of these, the RCNERC again has the highest

impact on the in-KB queries (+0.138). In general, the modules have low impact on the NIL queries. Among different genres, NW (+0.141) and WB (+0.079) have the highest impacts from the RCNERC system, respectively. In case of DF, the highest impact belong to the Nickname module (+0.062) since the nicknames occur in DF more than two other genres. Among query types, the PER type has the highest influence from the RCNERC system (+0.112) and the lowest from Pattern Extraction module (-0.001). In case of ORG and GPE type, the highest impacts are from RCNERC system (+0.056) and Redirects (+0.135), respectively. The RCNERC has a high impact on PER type since the three-phase NERC system highly improved its recall. While the most positive impact of Redirects occurs for GPE type (+0.135), the results of PER type improved more by the Nicknames mapping (+0.045). Meanwhile, the pattern extraction module outcomes a little negative impact on PER type (-0.001). It has a positive influence on ORG and GPE types. The table demonstrates that the improving modules have the positive impacts in most parts (except one with a little negative impact–Pattern Extraction/PER).

Chapter 5

Conclusions and Future Work

This document described the works towards developing an Entity Linking (EL) system aiming to disambiguate NE mentions existing in a target document. The EL task is highly challenging since each entity can usually be referred to by several NE mentions (synonymy). In addition, a NE mention may be used to indicate distinct entities (polysemy). During this research we found that the EL task is even more challenging due to the wide range of difficulties faced to the task. Thus, much effort is needed to overcome these challenges. To overcome, it is so necessary and crucial to address this this hardness with the help of semantic knowledge under the context of documents. There are the cases that disambiguation task is even tough for a human annotator and obviously is more challenging for a machine. Thus, the future perspective of the task and its success depends on how much we can tackle the difficulties with the semantic process of the existing resources.

In this research, we evaluated our EL system in TAC-KBP working framework in which the system input is a set of queries, each containing a query name, target document name, and start and end offsets of query name existing in the target document. The output is either a NE entity id in a reference KB or a NIL id in the case a system could not find any appropriate entity for that query. Our results show that we have had overall results higher than median of

all participants in TAC-KBP 2014 EL evaluation track. The main contributions of the thesis have been presented in Section 1.4.

Even if the writing of their PhD thesis is a major undertaking for any graduate student, it is also true that any work of research, even if it closes pending questions, always leaves new ones open. This thesis is no exception, and a number of ideas have not been thoroughly explored–including some which have been scratched at the surface. This section tries to collect such possible future lines of research, grouping them by the chapter in which the work related to them is exposed.

- The EL systems usually answer correctly in the case of well-known and trivial query names, however, they are generally faced to crucial challenges when either query names are highly ambiguous or the document in which the query exist, lacks enough discriminative information related to that query. In such situations, semantic analysis of the target document would be highly essential. Although in this research we proposed the methods to exploit the semantic knowledge lied in the document, however, there still exist the cases that the disambiguation task is even challenging for a human annotator. This urges not only deep semantic analysis of the target document but also the use of different knowledge resources. Thus, more effort by the researcher focused on this topic is necessary to tackle this type of challenges.

- As a future work, the approach can be developed over a multi-lingual EL systems to disambiguate named entity mentions existing in cross-lingual documents. In the first stage, the system can be prepared to work over Spanish and Chinese-language documents and then over the Right-to-Left languages such as Persian and Arabic. The idea behind is that a large amount of web information is provided by Right-to-Left languages, however, there still exist no considerable tools for linking entities in such languages.

- Although we could improve the recall and precision of NERC system in detecting different types, but there are still challenges that should be solved during NE recognition and classification. Since the accuracy of the NERC system has a high impact on the system's final answer, each efforts in this step would improve the whole performance of the system.

- The performance of EL systems tightly relied on the resources used in disambiguation task. Out of date resources will directly affect of the system. To this end, it is necessary to keep them updated in short span of time. The Nickname mapping dictionary is an example in this case. Nowadays, the use of Nicknames is increasing which makes the linking task highly ambiguous. Providing the dictionary of Nickname mappings is the best way to resolve this issue. However, manually elicitation of Nicknames dictionary would be highly tough and time consuming. In our study, we developed a module to automatically extract nicknames from the source documents using pattern matching technique. As a future work, this module should be developed to encompass more patterns. It helps accurately extracting more nicknames.

- Try to experiment on the collaborative native of some queries as in the case of several queries referring to the same reference document.

References

[1] Alicia Ageno, Pere R. Comas, Ali Naderi, Horacio Rodrıguez, and J. Turmo. The talp participation at tac-kbp 2013. In *In the Sixth Text Analysis Conference (TAC 2013), Gaithersburg, MD USA*, 2014.

[2] Masayuki Asahara and Yuji Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics*, 2003.

[3] Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *1st international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, 1998.

[4] Satanjeev Banerjee and Ted Pedersen. Extended gloss overlaps as a measure of semantic relatedness. *IJCAI*, 3, 2003.

[5] Michele Banko, Oren Etzioni, and Turing Center. The tradeoffs between open and traditional relation extraction. *ACL*, 8, 2008.

[6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[7] Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 9–16, 2006.

[8] Horst Bunke and eds Alberto Sanfeliu. Syntactic and structural pattern recognition: theory and applications. *World Scientific*, 7, 1990.

[9] Amev Burman, Arun Jayapal, Sathish Kannan, Madhu Kavilikatta, Ayman Alhelbawya, Leon Derczynski, and Robert Gaizauskas. Usfd at kbp 2011: Entity linking, slot filling and temporal bounding. In *Proceedings of Text Analysis Conference*, 2011.

[10] Taylor Cassidy, Zheng Chen, Javier Artiles, Heng Ji, Hongbo Deng, Lev-Arie Ratinov, Jing Zheng, Jiawei Han, and Dan Roth. Cuny-uiuc-sri tac-kbp2011 entity linking system description. In *Proceedings of Text Analysis Conference*, 2011.

[11] Taylor Cassidy, Heng Ji, Lev-Arie Ratinov, Arkaitz Zubiaga, and Hongzhao Huang. Analysis and enhancement of wikification for microblogs with context expansion. In *COLING*, 2012.

[12] Zheng Chen and Heng Ji. Collaborative ranking: A case study on entity linking. In *Proceedings of EMNLP*, 2011.

[13] Xiao Cheng and Dan Roth. Relational inference for wikification. *Urbana*, 51, 2013.

[14] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, 1999.

[15] Ronan Collobert. Deep learning for efficient discriminative parsing. In *In International Conference on Artificial Intelligence and Statistics*, 2011.

[16] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[17] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL*, 2007.

[18] Silviu Cucerzan. Tac entity linking by performing full-document entity extraction and disambiguation. In *Proceedings of Text Analysis Conference*, 2011.

[19] Silviu Cucerzan and David Yarowsky. Language independent ner using a unified model of internal and contextual evidence. In *proceedings of the 6th conference on Natural language learning-Volume 20. Association for Computational Linguistics*, 2002.

[20] Jeffrey Dalton and Laura Dietz. Umass ciir at tac kbp 2013 entity linking: query expansion using urban dictionary. In *Text Analysis Conference*, 2013.

[21] Laura Dietz and Jeffrey Dalton. Acrossdocument neighborhood expansion: Umass at tac kbp 2012 entity linking. In *In Text Analysis Conference (TAC)*, 2012.

[22] Angela Fahrni, Thierry Göckel, and Michael Strube. Hits'monolingual and cross-lingual entity linking system at tac 2012: A joint approach. In *TAC Workshop*, 2012.

[23] Norberto Fernandez, Jesus A. Fisteus, Luis Sanchez, and Eduardo Martin. Webtlab: A cooccurencebased approach to kbp 2010 entity-linking task. In *Proc. TAC 2010 Workshop*, 2010.

[24] Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *In Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628, 2010.

[25] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, 2005.

[26] Angela Fogarolli. Word sense disambiguation based on wikipedia link structure. In *International Conference on Semantic Computing*, 2009.

[27] King Sun Fu. Syntactic pattern recognition and applications. *Prentice-Hall*, 1982.

[28] Dan Gillick. Sentence boundary detection and the problem with the us. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers. Association for Computational Linguistics*, 2009.

[29] Edgar Gonzalez, Horacio Rodriguez, Jordi Turmo, Pere R. Comas, Ali. Naderi, Alicia Ageno, Emili Sapena, Marta Vila, and M. Antonia Marti. The talp participation at tac-kbp 2012. In *In Text Analysis Conference, USA*, 2013.

[30] Swapna Gottipati and Jing Jiang. Smu-sis at tac 2010-kbp track entity linking. In *Proc. TAC 2010 Workshop*, 2010.

[31] Yuhang Guo, Guohua Tang, Wanxiang Che, Ting Liu, , and Sheng Li. Hit approaches to entity linking at tac 2011. In *Proceedings of Text Analysis Conference*, 2011.

[32] Ben Hachey, Will Radford, and James R. Curran. Graph-based named entity linking with wikipedia. In *Proceedings of the 12th International Conference on Web Information System Engineering*, pages 213–226, 2011.

[33] Hui Han, Hongyuan Zha, and C. Lee Giles. Name disambiguation in author citations using a k-way spectral clustering method. In *In Digital Libraries, 2005. JCDL'05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on*, pages 334–343, 2005.

[34] Xianpei Han and Le Sun. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of ACL*, 2011.

[35] Xianpei Han and Jun Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of CIKM*, 2009.

[36] Xianpei Han, Le Sun, and Jun Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. ACM*, 2011.

[37] Zhengyan He and Houfeng Wang. Collective entity linking and a simple slot filling method for tac-kbp 2011. In *Proceedings of Text Analysis Conference*, 2011.

[38] Hongzhao Huang, Yunbo Cao, Xiaojiang Huang, Heng Ji, and Chin-Yew Lin. Collective tweet wikification based on semi-supervised graph regularization. *Proceedings of the ACL, Baltimore, Maryland*, 2014.

[39] Kristy Hughes, Joel Nothman, and James R. Curran. Trading accuracy for faster entity linking. In *Australasian Language Technology Association Workshop*, page 32, 2014.

[40] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *the Eleventh conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1995.

[41] Adam R. Klivans and Rocco A. Servedio. Toward attribute efficient learning of decision lists and parities. *The Journal of Machine Learning Research*, 7:587–602, 2006.

[42] Zornitsa Kozareva, Konstantin Voevodski, and Shang-Hua Teng. Class label enhancement via related instances. In *Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics*, 2011.

[43] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM*, 2009.

[44] John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi. Lcc approaches to knowledge base population at tac 2010. In *Proceedings of the Text Analysis Conference*, 2010.

[45] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *the 5th annual international conference on Systems documentation*. ACM, 1986.

[46] Dekang Lin. Automatic retrieval and clustering of similar words. In *the 17th international conference on Computational Linguistics*, volume 2. Association for Computational Linguistics, 1998.

[47] Xiaohua Liu, Yitong Li, Haocheng Wu, Ming Zhou, Furu Wei, , and Yi Lu. Entity linking for tweets. *ACL*, 1:1304–1311, 2013.

[48] Ian MacKinnon and Olga Vechtomova. Improving complex interactive question answering with wikipedia anchor text. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval, ECIR'08*, 2008.

[49] John F. Magee. *Decision trees for decision making*. Graduate School of Business Administration, Harvard University, 1964.

[50] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. Association for Computational Linguistics*, 2003.

[51] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[52] Paul McNamee, Hoa Trang Dang, Heather Simpson, Patrick Schone, and Stephanie Strassel. An evaluation of technologies for knowledge base population. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 369–372, 2010.

[53] Paul McNamee, James Mayfield, Douglas W. Oard, Tan Xu, Veselin Stoyanov Ke Wu, and David Doermann. Cross-language entity linking in maryland during a hurricane. In *Proceedings of Text Analysis Conference*, 2011.

[54] Olena Medelyan, David Milne, Catherine Legg, and Ian H. Witten. Mining meaning from wikipedia. *International Journal of Human-Computer Studies 67*, 9:716–754, 2009.

[55] Laurent Mertens, Thomas Demeester, Johannes Deleu, and Chris Develder. Ugent participation in the tac 2013 entity-linking task. In *Text Analysis Conference*, pages 1–12, 2013.

[56] Rada Mihalcea. Co-training and self-training for word sense disambiguation. In *the Conference on Natural Language Learning*, 2004.

[57] Rada Mihalcea and Andras Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the 16th Conference on Information and Knowledge Management*, pages 233–242, 2007.

[58] D. Milne and I.H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th Conference on Information and Knowledge Management*, pages 509–518, 2008.

[59] Sean Monahan, John Lehmann, Timothy Nyberg, Jesse Plymale, and Arnold Jung. Cross-lingual cross-document coreference with entity linking. In *Proceedings of Text Analysis Conference*, 2011.

[60] Raymond J. Mooney. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 82–91, 1996.

[61] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. In *Lingvisticae Investigationes 30.1*, 2007.

[62] Ali Naderi, Horacio Rodriguez, and Jordi Turmo. The talp participation at erd 2014. In *In Proceedings of the first international workshop on Entity recognition and disambiguation, pp. 89-94. ACM*, 2014.

[63] Ali Naderi, Horacio Rodriguez, and Jordi Turmo. Topic modeling for entity linking using keyphrase. In *In Proceedings of Natural Language Processing and Cognitive Science workshop, Venice, Italy*, 2014.

[64] Ali Naderi, Horacio Rodriguez, and Jordi Turmo. Binary vector approach to entity linking: Talp in tac-kbp 2014. In *In the Seventh Text Analysis Conference, Gaithersburg, MD USA*, 2014.

[65] Ali Naderi, Horacio Rodrıguez, and Jordi Turmo. Binary vector approach to entity linking: Talp in tac-kbp 2014. In *Text Analysis Conference*, 2014.

[66] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2), 1990.

[67] Dávid Márk Nemeskey, Gábor András Recski, Attlia Zséder, and Andras Kornai. Budapestacad at tac 2010. In *Text Analysis Conference*, pages 1–3, 2010.

[68] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems 2*, pages 849–856, 2002.

[69] Vincent Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics*, 2010.

[70] Hien T. Nguyen, Tru H. Cao, and Trong T. Nguyen. Jvn-tdt entity linking systems at tac-kbp2012. In *Proc. of Text Analysis Conference*, 2012.

[71] Alex Olieman, Hosein Azarbonyad, Mostafa Dehghani, Jaap Kamps, and Maarten Marx. Entity linking by focusing dbpedia candidate entities. In *the first international workshop on Entity recognition and disambiguation*, pages 13–24. ACM, 2014.

[72] Patrick Pantel and Dekang Lin. Discovering word senses from text. In *the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002.

[73] Marius Pasca. Outclassing wikipedia in opendomain information extraction: Weakly-supervised acquisition of attributes over conceptual hierarchies. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 2009.

[74] Marco Pennacchiotti and Patrick Pantel. Entity extraction via ensemble semantics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 2009.

[75] Danuta Ploch, Leonhard Hennig, Ernesto William De Luca, Sahin Albayrak, and T. U. DAI-Labor. Dai approaches to the tac-kbp 2011 entity linking task. In *Proceedings of Text Analysis Conference*, 2011.

[76] Simone Paolo Ponzetto and Michael Strube. Knowledge derived from wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research*, 30: 181–212, 2007.

[77] John Ross Quinlan. *C4. 5: programs for machine learning*, volume 1. Morgan Kaufmann, 1993.

[78] Will Radford, Ben Hachey, Joel Nothman, Matthew Honnibal, and James R. Curran. Cmcrc at tac10: Document-level entity linking with graphbased re-ranking. In *Proc. TAC 2010 Workshop*, 2010.

[79] Will Radford, Ben Hachey, Matthew Honnibal, Joel Nothman, and James R. Curran. Naive but effective nil clustering baselines – cmcrc at tac 2011. In *Proceedings of Text Analysis Conference*, 2011.

[80] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 2009.

[81] Lev Ratinov and Dan Roth. Glow tac-kbp2011 entity linking system. In *Proceedings of Text Analysis Conference*, 2011.

[82] Ronald L. Rivest. Learning decision lists. *Machine learning*, 2(3):229–246, 1987.

[83] Tjong Kim Sang, Erik F., and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL*, 2003.

[84] Hinrich Schutze. Dimensions of meaning. In *Supercomputing '92: ACM/IEEE Conference on Supercomputing*, pages 787–796. IEEE Computer Society Press, 1992.

[85] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM*, 2013.

[86] Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, pages 93–128, 2006.

[87] Geoffrey Towell and Ellen M. Voorhees. Disambiguating highly ambiguous words. *Computational Linguistics*, 24(1):125–145, 1998.

[88] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo Michael Röder, Ciro Baron, Andreas Both, Martin Brümmer, and Diego Ceccarelli et al. Gerbil: General entity annotator benchmarking framework. In *In Proceedings of the 24th International Conference on World Wide Web, pp. 1133-1143. International World Wide Web Conferences Steering Committee*, 2015.

[89] Kees Van Deemter and Rodger Kibble. On coreferring: Coreference in muc and related annotation schemes. *Computational linguistics*, 26(4):629–637, 2000.

[90] Vasudeva Varma, Praveen Bysani, Vijay Bharat Kranthi Reddy, Karuna Kumar Santosh GSK, Sudheer Kovelamudi, N. Kiran Kumar, and Nitin Maganti. Iiit hyderabad at tac 2009. In *Proceedings of the Text Analysis Conference*, 2009.

[91] Dominic Widdows and Beate Dorow. A graph model for unsupervised lexical acquisition. In *the 19th international conference on Computational linguistics*, volume 1. Association for Computational Linguistics, 2002.

[92] Jian Xu, Zhengzhong Liu, Qin Lu, Yu-Lan Liu, and Chenchen Wang. Polyucomp in tac 2011 entity linking and slot-filling. In *Proceedings of Text Analysis Conference*, 2011.

[93] Jian Xu, Qin Lu, Jie Liu, and Ruifeng Xu. Nlp-comp in tac 2012 entity linking and slot-filling. In *In Proceedings of the Fourth Text Analysis Conference*, 2012.

[94] et al. Yang, Xiaofeng. Yang, xiaofeng and guodong zhou and jian su and chew lim tan. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1. Association for Computational Linguistics*, 2003.

[95] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods.

[96] Wei Zhang, Yan Chuan Sim, Jian Su, and Chew Lim Tan. Entity linking with effective acronym expansion, instance selection and topic modeling. In *Proceedings of IJCAI*, 2011.

[97] Wei Zhang, Jian Su, Bin Chen, Wenting Wang, Zhiqiang Toh, Yanchuan Sim, Yunbo Cao, Chin Yew Lin, and Chew Lim Tan. I2r-nus-msra at tac 2011: Entity linking. In *In Text Analysis Conference*, 2011.

[98] Wei Zhang, Jian Su, and Chew Lim Tan. A wikipedia-lda model for entity linking with batch size changing instance selection. In *Proceedings of IJCNLP*, 2011.

Appendix A

Evaluation Results

Detailed Evaluation Results obtained by BL_SYS and F_SYS

System: BL_SYS	Measurement			
	Accuracy	$B+$ Precision	$B+$ Recall	$B+$ F1
All Docs-Overall-All Entities	0.646	0.622	0.517	0.564
All Docs-Overall-PER	0.737	0.723	0.601	0.656
All Docs-Overall-ORG	0.569	0.528	0.463	0.494
All Docs-Overall-GPE	0.448	0.411	0.317	0.358
All Docs-InKB-All Entities	0.403	0.391	0.340	0.364
All Docs-InKB-PER	0.470	0.464	0.403	0.432
All Docs-InKB-ORG	0.267	0.248	0.227	0.237
All Docs-InKB-GPE	0.380	0.364	0.307	0.333
All Docs-NotInKB-All Entities	0.928	0.890	0.723	0.798
All Docs-NotInKB-PER	0.953	0.931	0.761	0.838
All Docs-NotInKB-ORG	0.938	0.871	0.752	0.807
All Docs-NotInKB-GPE	0.713	0.594	0.354	0.444
NW-Overall-All Entities	0.627	0.598	0.532	0.563
NW-Overall-PER	0.779	0.764	0.694	0.727
NW-Overall-ORG	0.607	0.550	0.513	0.531
NW-Overall-GPE	0.423	0.396	0.312	0.349
NW-InKB-All Entities	0.495	0.480	0.435	0.457
NW-InKB-PER	0.700	0.691	0.631	0.659
NW-InKB-ORG	0.302	0.267	0.271	0.269
NW-InKB-GPE	0.355	0.344	0.292	0.316
NW-NotInKB-All Entities	0.856	0.801	0.700	0.747
NW-NotInKB-PER	0.909	0.883	0.799	0.839
NW-NotInKB-ORG	0.881	0.805	0.731	0.766
NW-NotInKB-GPE	0.670	0.582	0.384	0.463
WB-Overall-All Entities	0.648	0.616	0.484	0.542
WB-Overall-PER	0.789	0.762	0.585	0.662
WB-Overall-ORG	0.614	0.587	0.480	0.528
WB-Overall-GPE	0.419	0.372	0.297	0.330

Table A.1 The results obtained by BL_sys over TAK-KBP 2014 Mono-Lingual (English) EL evaluation data set.

System: BL_SYS: continued	Measurement			
	Accuracy	$B+$ Precision	$B+$ Recall	$B+$ F1
WB-InKB-All Entities	0.381	0.366	0.323	0.343
WB-InKB-PER	0.468	0.461	0.429	0.445
WB-InKB-ORG	0.315	0.301	0.257	0.277
WB-InKB-GPE	0.362	0.339	0.290	0.313
WB-NotInKB-All Entities	0.937	0.887	0.658	0.755
WB-NotInKB-PER	0.954	0.917	0.666	0.771
WB-NotInKB-ORG	0.986	0.943	0.757	0.840
WB-NotInKB-GPE	0.679	0.517	0.328	0.402
DF-Overall-All Entities	0.658	0.646	0.534	0.585
DF-Overall-PER	0.692	0.685	0.570	0.622
DF-Overall-ORG	0.264	0.218	0.230	0.224
DF-Overall-GPE	0.606	0.565	0.389	0.461
DF-InKB-All Entities	0.325	0.320	0.252	0.282
DF-InKB-PER	0.327	0.325	0.252	0.284
DF-InKB-ORG	0.063	0.058	0.053	0.056
DF-InKB-GPE	0.517	0.497	0.408	0.448
DF-NotInKB-All Entities	0.963	0.944	0.791	0.86
DF-NotInKB-PER	0.964	0.953	0.806	0.874
DF-NotInKB-ORG	1.000	0.804	0.876	0.839
DF-NotInKB-GPE	0.914	0.799	0.325	0.462

Table A.2 continued.

System: F_SYS	Measurement			
	Accuracy	$B+$ Precision	$B+$ Recall	$B+$ F1
All Docs-Overall-All Entities	0.840	0.820	0.702	0.757
All Docs-Overall-PER	0.864	0.851	0.713	0.776
All Docs-Overall-ORG	0.744	0.716	0.614	0.661
All Docs-Overall-GPE	0.862	0.826	0.754	0.788
All Docs-InKB-All Entities	0.800	0.779	0.751	0.765
All Docs-InKB-PER	0.802	0.791	0.754	0.772
All Docs-InKB-ORG	0.684	0.661	0.620	0.640
All Docs-InKB-GPE	0.872	0.836	0.833	0.834
All Docs-NotInKB-All Entities	0.888	0.868	0.645	0.740
All Docs-NotInKB-PER	0.914	0.900	0.680	0.775
All Docs-NotInKB-ORG	0.817	0.783	0.607	0.684
All Docs-NotInKB-GPE	0.824	0.787	0.443	0.567
NW-Overall-All Entities	0.793	0.768	0.703	0.734
NW-Overall-PER	0.821	0.805	0.747	0.775
NW-Overall-ORG	0.704	0.667	0.587	0.624
NW-Overall-GPE	0.826	0.798	0.737	0.766
NW-InKB-All Entities	0.801	0.777	0.753	0.765
NW-InKB-PER	0.816	0.803	0.756	0.778
NW-InKB-ORG	0.656	0.610	0.616	0.613
NW-InKB-GPE	0.856	0.832	0.820	0.826
NW-NotInKB-All Entities	0.780	0.752	0.617	0.678
NW-NotInKB-PER	0.830	0.809	0.733	0.769
NW-NotInKB-ORG	0.748	0.717	0.561	0.630
NW-NotInKB-GPE	0.718	0.678	0.433	0.528
WB-Overall-All Entities	0.846	0.822	0.689	0.749
WB-Overall-PER	0.845	0.830	0.659	0.735
WB-Overall-ORG	0.825	0.803	0.679	0.736
WB-Overall-GPE	0.870	0.826	0.757	0.790

Table A.3 The results obtained by F_sys over TAK-KBP 2014 Mono-Lingual (English) EL evaluation data set.

System: F_SYS: continued	Measurement			
	Accuracy	$B+$ Precision	$B+$ Recall	$B+$ F1
WB-InKB-All Entities	0.790	0.766	0.744	0.755
WB-InKB-PER	0.711	0.703	0.682	0.692
WB-InKB-ORG	0.774	0.765	0.704	0.733
WB-InKB-GPE	0.864	0.818	0.822	0.820
WB-NotInKB-All Entities	0.906	0.881	0.630	0.735
WB-NotInKB-PER	0.914	0.896	0.647	0.751
WB-NotInKB-ORG	0.889	0.851	0.647	0.735
WB-NotInKB-GPE	0.897	0.864	0.461	0.601
DF-Overall-All Entities	0.875	0.862	0.714	0.781
DF-Overall-PER	0.892	0.882	0.726	0.796
DF-Overall-ORG	0.545	0.530	0.442	0.482
DF-Overall-GPE	0.948	0.910	0.795	0.849
DF-InKB-All Entities	0.809	0.794	0.757	0.775
DF-InKB-PER	0.830	0.819	0.782	0.800
DF-InKB-ORG	0.484	0.468	0.386	0.423
DF-InKB-GPE	0.942	0.904	0.901	0.902
DF-NotInKB-All Entities	0.935	0.924	0.674	0.780
DF-NotInKB-PER	0.938	0.929	0.684	0.788
DF-NotInKB-ORG	0.769	0.756	0.647	0.698
DF-NotInKB-GPE	0.971	0.933	0.431	0.590

Table A.4 continued.

Appendix B

List of Publications

- A. Naderi, H. Rodríguez, and J. Turmo. “Unsupervised Entity Linking using Graph-based Semantic Similarity", *ACM Transactions on Information Systems (TOIS)*. (Submitted)

  Abstract: This article presents the works towards developing an unsupervised Entity Linking (EL) system using graph-based semantic similarity aiming to disambiguate Named Entity (NE) mentions occurring in target documents.

- A. Naderi, H. Rodríguez, and J. Turmo. “Binary Vector Approach to Entity Linking: TALP at TAC-KBP 2014." *Text Analysis Conference*, Gaithersburg, ML, USA, 2015. (Awaiting to publish) [64]

  Abstract: This document describes the work performed by the Universitat Politecnica de Catalunya (UPC) in its third participation at TAC-KBP 2014 in Mono-Lingual (English) Entity Linking task.

- A. Naderi, H. Rodríguez, and J. Turmo. “Topic Modeling for Entity Linking using Keyphrase," *11th Int. Workshop on Natural Language Processing and Cognitive Science (NLPCS)*, Venice, Italy, 2014. (Published) [63]

Abstract: This paper proposes an Entity Linking system that applies a topic modeling ranking. We apply a novel approach in order to provide new relevant elements to the model. These elements are keyphrases related to the queries and gathered from a huge Wikipedia-based knowledge resource.

- A. Naderi, H. Rodríguez, and J. Turmo. "The TALP participation at ERD 2014," *Int. Workshop on Entity Recognition and Disambiguation (ERD)*, Gold Coast, Queensland, Australia, 2014. (Published) [62]

  Abstract: This document describes the work performed by the TALP Research Center, UPC in its first participation at ERD 2014 short text evaluation track. The objective of this evaluation track is to recognize mentions of entities in a given short text, disambiguate them and map them to the entities in a given collection of knowledge base. To this end, we presented our system taking advantage of a topic modeling approach to rank candidates of each entity mentions occurring in the query text.

- A. Ageno, P.R. Comas, A. Naderi, H. Rodríguez, and J. Turmo. "The TALP participation at TAC-KBP 2013," *Text Analysis Conference*, Gaithersburg, ML, USA, 2014. (Published) [1]

  Abstract: This document describes the work performed by the Universitat Politecnica de Catalunya ' (UPC) in its second participation at TAC-KBP 2013 in both the Entity Linking and the Slot Filling tasks. I was in charge of the EL task.

- E. Gonzàlez, H. Rodríguez, J. Turmo, P.R. Comas, A. Naderi, A. Ageno, E. Sapena, M. Vila and M.A. Martí. "The TALP participation at TAC-KBP 2012," *Text Analysis Conference*, Gaithersburg, ML, USA, 2013. (Published) [29]

  Abstract: This document describes the work performed by the Universitat Politecnica de Catalunya (UPC) in its first participation at TAC-KBP 2012 in both the Entity Linking and the Slot Filling tasks. I was in charge of the EL task.