Entity Markup for Knowledge Base Population

Lili Jiang(✉)

Department of Computing Science, Umeå University, Umeå, Sweden
lili.jiang@cs.umu.se

Abstract. Entities (e.g. people, places, products) exist in various heterogeneous sources, such as Wikipedia, web page, and social media. Entity markup, like entity extraction, coreference resolution, and entity disambiguation, is the essential means for adding semantic value to unstructured web contents and this way enabling the linkage between unstructured and structured data and knowledge collections. A major challenge in this endeavor lies in the ambiguity of the digital contents, with context-dependent semantic and dynamic. In this paper, I introduce the main challenges of coreference resolution and named entity disambiguation. Especially, I propose practical strategies to improve entity markup. Furthermore, experimental studies are conducted to fulfill named entity disambiguation in combination with the optimized entity extraction and coreference resolution. The main goal of this paper is to analyze the significant challenges of entity markup and present insights on the proposed entity markup framework for knowledge base population. The preliminary experimental results prove the significance of improving entity markup.

1 Introduction

Entity markup, like entity extraction, coreference resolution, and named entity disambiguation, is the essential means to deliver semantic value to unstructured web contents and enable the linkage between unstructured and structured data and knowledge bases. Named entity disambiguation (NED) is a task of linking mention in given text to a unique entity in existing knowledge base (i.e. Wikipedia). NED is one of many importation operations for data management, information retrieval, semantic mining. Further research in entity disambiguation is necessary to help promote information quality and improve data reporting in multidisciplinary fields requiring accurate data representation.

Despite many advances in the last few years, fully automatic NED is inherently difficult and may also be computationally expensive [6,16,17,20,26,36]. NED methods have been shown to perform very well for prominent entities mentioned in high-quality texts like news articles, but they degrade in terms of both precision and recall when dealing with lesser known long-tail entities. Since advanced methods utilize machine learning or extensive statistics for semantic relatedness measures among entities, the availability of labeled training data is usually a big bottleneck.

However, even if we had perfect NED methods for aligning ambiguous names in text documents with canonicalized entities registered in a knowledge base, the

P. K. Reddy et al. (Eds.): BDA 2017, LNCS 10721, pp. 71–89, 2017.
https://doi.org/10.1007/978-3-319-72413-3_5

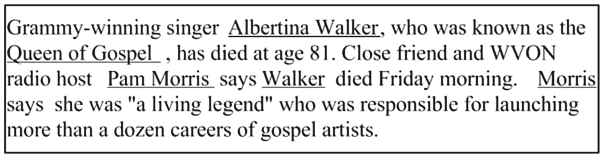
Grammy-winning singer Albertina Walker, who was known as the Queen of Gospel , has died at age 81. Close friend and WVON radio host Pam Morris says Walker died Friday morning. Morris says she was "a living legend" who was responsible for launching more than a dozen careers of gospel artists.

Fig. 1. A text illustrating mentions in NED

envisioned cross-linkage between unstructured web contents and semantic data collections would still have big gaps. For example, considering the text snippet in Fig. 1, both "Queen of Gospel" and "Walker" refer to "Alberta Walker", but automated algorithms typically lack the background knowledge is challenging. Another example in the text of Fig. 1 is that "Pam Morris" is relatively unambiguous but the isolated mention "Morris" in NED is more challenging. The reason is that finding the correct link in NED requires disambiguating based on the mention string and often non-local contextual features. However, we can nevertheless capture their mentions under different names and try to gather equivalence classes of text phrases that refer to the same entity. This is known as the problem of coreference resolution(CR) [13,34,35,37]. The other reason of the failed NED is the dynamic world: new entities come into existence. When facing such emerging entities, CR methods are also helpful. In addition to dealing with the recognized and emerging entities, CR methods can also help to increase the recall of NED for known entities, simply by capturing more surface phrases (e.g. [23,28]). For example, we should discover mentions such as "Donald Trump" and "the USA president" to infer that they denote the same entity. We can map more text mentions onto entities, thus improving NED recall at high precision. Systematically gathering different mentions names for entities is the problem of Dictionary Building. It has been studied in the literature, harnessing href anchor texts, click logs, and other assets [18,39]. However, doing this for emerging entities that are not yet registered in a knowledge base is a largely unexplored task.

This paper presents a framework for entity markup, where I combine entity extraction, coreference resolution, and named entity disambiguation in a joint manner. During this process, practical strategies are proposed to get optimum results.

2 Terminology, Problem, and Framework

2.1 Terminologies

- **Entity:** Any object existing in the real world can be entity, such as person, organization, location, and product.
- **Mention:** Mention is the surface name, which an entity is referred in text. In other words, mention is the instance of entity. For example,

"Albertina Walker" can be the mention of entity "Albertina Walker" (en.wikipedia.org/wiki/Albertina_Walker).

- **Entity Extraction** (EE)**:** The input text (e.g., web pages, news articles, etc.) is processed to discover *mentions* of named entities, that is, surface phrases that are likely to denote individual entities (as opposed to common noun phrases). Our implementation currently uses the Stanford NER Tagger [10] (a trained CRF) and Illinois Mention Detector [2] for this purpose.
- **Name Entity Disambiguation** (NED) is the process of linking the named mentions in text to entities registered in the existing knowledge bases (e.g., Wikipedia). Mention "Albertina Walker" could be easily linked to the American gospel singer Albertina Walker in Wikipedia. However, the following "Walker" may refer to numerous distinct candidates: "Alice Walker", "Derek Walker", or "Kara Walker". Mention "Pam Morris" should be linked to *Null* as it has no corresponding RDF triples in Knowledge base.
- **Entity Candidate:** Possible entities (with unique canonical names) from Knowledge base, a mention may denote. We harness existing knowledge bases like DBpedia or YAGO.
- **Coreference Resolution** (CR) is the process of finding all the mentions (i.e. named mention, nominal mention, and pronoun mention) in documents that refer to the same entity. Taking the given example text, mentions "singer Albertina Walker", "Albertina Walker", "Queen of Gospel" as well as "Walker" refer to the same entity[1].
- **Coreference Equivalence Class:** Coreference equivalence class (aka., coreference chain) is the set of all the mentions, which refer to the same entity in a given text. For example, here we can have two coreference equivalence classes {singer Albertina Walker, Albertina Walker, Queen of Gospel, Walker, she} and {Pam Morris, Morris}.

2.2 Problem Formulation

Given a document d, we extract all mentions and formulate as $M = \{m_1, m_2, \ldots, m_{n_m}\}$, where all mentions are linearly ordered by positions. We define entity candidate list as $E = \{E_1, E_2, \ldots E_{n_m}\}$, where $E_i = \{e_{i1}, e_{i2}, \ldots\}(0 < i \leq n_m)$ denotes the entity candidate list of mention m_i.

After that, we propose a coreference resolution classifier to generate coreference equivalence classes $C = \{C_1, C_2, \ldots C_{n_c}\}$, where C_i denotes a single coreference equivalence class and $C_i \cap C_j = \emptyset$ $(1 \leq i, j \leq n_c)$. $coreferent(m_i, m_j)$ is true if mentions m_i and m_j are within the same coreference equivalence class. The core task of our work is as follows: given the mentions $M = \{m_1, m_2, \ldots, m_{n_m}\}$ extracted from document d, we first generate coreference equivalence classes $C = \{C_1, C_2, \ldots, C_{n_c}\}$ and entity candidates $E = \{E_1, E_2, \ldots E_{n_m}\}$. Based on M, C, and E, we built a mention-entity graph as shown in Fig. 2. Let $\psi(m_i, e_{ij})$ be a score function reflecting the likelihood that candidate entity e_{ij} is the correct disambiguation linking entity for $m_i \in M$. Let $\phi(m_i, m_j)$ be a score function

[1] http://en.wikipedia.org/wiki/Albertina_Walker.

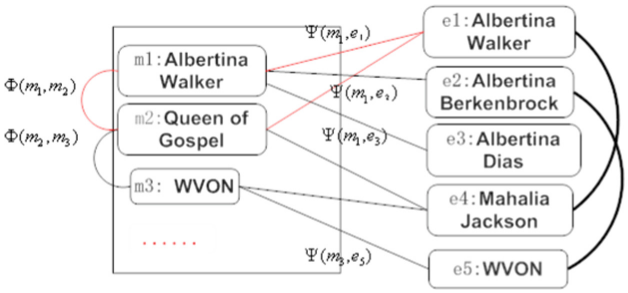

Fig. 2. A mention-entity graph example in NED

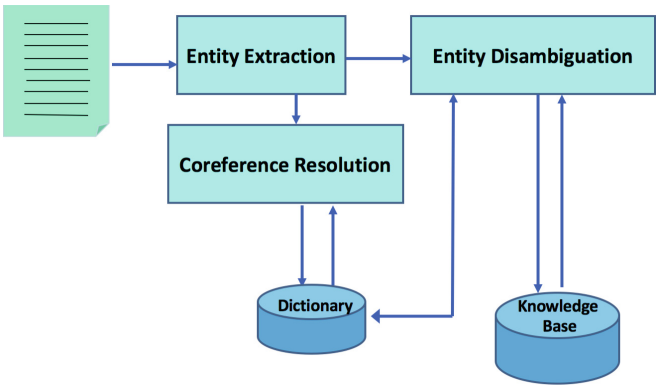

Fig. 3. An overall framework for entity markup

reflecting the likelihood that mention m_i and mention m_j are *coreferent*. Let $\kappa(e_i, e_j)$ be a score function reflecting the coherence of entity e_i and entity e_j. Scanning from m_1 to m_{n_m} using random walk with restart, ψ, ϕ, and κ work together to link each mention on the left side to a unique entity on the right side, and we finally generate the entity candidate list as $E' = \{E'_1, E'_2, \ldots E'_{n_e}\}$, where each element is a ranked list of entities $\{e_{i1}, e_{i2}, \ldots\}$ registered in Knowledge base $\mathcal{KB}$ for each named mention $m_i \in M$. Meanwhile, we output updated results for coreference equivalence classes $C' = \{C'_1, C'_2, \ldots C'_{n_{c'}}\}$.

2.3 Entity Markup Framework Overview

Figure 3 gives a pictorial overview of the proposed framework of entity markup. It consists of three main functional components: entity extraction (EE), followed by coreference resolution (CR) and entity disambiguation (NED). Details will be explained in the following sections.

3 Entity Extraction and Sieve

Entity extraction is also called entity recognition or mention extraction. We recognize all the mentions using the state-of-the-art NER models from Stanford [10] and Illinois [2].

After recognition, we first filter out some nonsense or incorrectly extracted entity mentions. Secondly, we remove or correct some nominal mentions by exploring the position relations (i.e. overlapping and embedding) between named mentions and nominal mentions.

- For any mention, we will filter it out if it meets any condition as follows: (1) it is consist of stop words; (2) it contains too many punctuation; (3) if it is started with conjunction or ended with a conjunction (e.g. as) or pronoun (e.g. his, her); (4) it contains incomplete punctuation (i.e. half bracket or quotation mark); (5) a mention is one word and the word prior to or following this mention is a noun phrase.
- For embedding mentions, we keep both of them (e.g., "Dutch" embedded in "Dutch Soccer Captain"). For overlapping mentions, four false positive cases will be considered: (1) only one of them is recognized correctly, such as "Sen. Bill" and "Bill Frist"; (2) both of them are recognized correctly, such as "Irishman Patrick Butler" and "16-year-old Irishman"; (3) neither of them are recognized correctly but an integrated them is a correct noun phrase. For example, "President Ali" and "Ali Abdullah Saleh" could be integrated as President Ali Abdullah Saleh; (4) both mentions are recognized and the only difference is definite article, such as "The Justice" and "Justice". In the following, we identify all the overlapping mentions, and then filter out them or modify them to correct form. For an embedding pair of named mention m_i and nominal mention m_j, if m_j contains certain stop words/determiners (e.g. 'a', 'an', 'the', 'something', 'anything', 'nothing', 'there', or 'here') as prefixes or suffixes, and m_i equals m_j after removing these articles, we only keep m_i. If the predicted mention type of m_i is *PERSON*, *ORGANIZATION*, or *LOCATION*, and both m_i and m_j are consist of nouns, we will merge m_i and m_j into a single nominal mention. For any other embedding pairs, we will drop m_j from mention list M. After the sieve, an updated mention list M is obtained for further use in the following sections.

4 Coreference Resolution

An effective coreference resolution system is an important component in any NLP pipeline that deals with language understanding tasks, such as question answering and information extraction. In this paper, we regard it critical for named entity disambiguation task. We first provide detailed error analysis with examples regarding the different kinds of errors that show up in the basic coreference resolution system (Sect. 4.1). According to these error analysis, we propose additional features and constrains to obtain coreference equivalence classes for a given document (Sect. 4.2).

4.1 Error and Challenge Analysis

We test the basic coreference system (i.e. Illinois CR system) on the news corpus and analyze the results. Errors decreasing precision and recall are categorized with examples according to their causes.

Lack of Alias Detection. Missing aliases (e.g. nickname, acronyms) reduces the performance of coreference resolution. The following three examples show three pairs of missing aliases in the state-of-the-art CR systems.

> Richard N. Gottfried is a leading policy-maker nationally... Dick Gottfried is also a member of the Steering Committee...

> But now some residents are worried that the Gansevoort Peninsula, also known as Pier 52, will continue to be used ...

> The Delaware Department of Transportation (DelDOT) is an agency of the U.S....

Inaccuracy of Appositive. Three main appositive errors are observed.

Geographical location mismatch. Some false positives are caused due to the mismatch between cities and countries. Taking the following as example, as a borough in New Jersey, "Fair Haven" is incorrectly resolved to be coreferent with NJ.

> POWER-Thomas C., of Fair Haven, NJ, died at home on October 8, 1997.

Mismatch of preposition and head word. In the adverbial modifier with a preposition, the head word is usually incorrectly resolved to be *coreferent* with the subject after it (e.g. "South Florida" and "a weary public" as follows).

> Around South Florida, a weary public was trying to cope with fears...

Entity mismatch within an entity set. Entities belonging to the same entity set are sometimes resolved as appositive falsely, when they are displayed in a row with comma as separators.

> Aluvial slopes are inhabited by Pedunculate Oak, linden, European hornbeam, and European Turkey oak.

Overall, according to our experimental results, the appositive for people formed as (*proper noun, common noun*) are returned with high precision and low recall. The appositive formed as (*proper noun, proper noun*) is usually determined incorrectly.

Side-effects from string similarity. String similarity is essential and yet risky in coreference resolution. As shown in the example as follows, "Mr. Clinton" and "Hillary Rodham Clinton" are incorrectly resolved to be *coreferent*.

Mr. Clinton was accompanied by his wife, Hillary R. Clinton.

4.2 Coreference Resolution Learning and Inference

In previous section, we analyzed some errors from the state of the arts in coreference resolution. As coreference resolution data is not totally linearly separable, in this case, learning with further inference outperforms either local classifier or global classifier when the number of training examples is not sufficiently large [31]. In this section, we proposed a two-stage method for coreference resolution. Firstly in the learning stage, we train a local classifier for each pair of mentions, and generate a score indicating pairwise probability of coreference resolution. Herein, each mention pair suffices symmetry. Secondly in the inference stage, we employ deterministic constrains to aggregate the scores generated by the classifier, and then link mention pairs into coreference equivalence classes. Herein, we fix the transitivity volition. Finally, the coreference equivalence classes formed by only one mention will be deleted.

Learning. We train a logistic regression classifier with a probability $\phi(m_i, m_j)$ as output for each pair of mentions m_i and m_j. It refers to the probability that m_i and m_j is *coreferent*. After learning, coreference resolution score of pairwise mention is a function $M \times M \rightarrow [0, 1]$, where 0 and 1 are the minimum and maximum coreference score.

$$\phi(m_i, m_j) = \frac{1}{1 + \sum_k \exp(w_k * f_k(m_i, m_j))} \tag{1}$$

Where w_k is the weight vector learned from training data, f is the feature vector and $f_k(m_i, m_j)$ is the value of the kth feature.

We create training samples according to the widely used method from [38]. Given a mention m_j from training data, this method generates positive samples with m_j and its closest preceding coreferent mention m_i, and negative samples with m_j and every intervening mention $m_{i+1}\ m_{i+2}\ \ldots\ m_{j-1}$.

Learning Features. Table 1 provides a concise view for all the features we used in the learning phase. Details about these features are explained in the following.

Co-occurring distribution probability. We use a model based on knowledge base $\mathcal{KB}$ (YAGO) to get a prior probability that two mentions linking to the same entity. In $\mathcal{KB}$, we have anchor link for each mention to a Wikipedia entry page (i.e. entity). Thus, the occurrence frequency of each mention and the frequency of its linking to an entity are obtained. The probability $p(m_i, e)$ $(e \in E(m_i)$, $m_i \in M)$ is defined as the fraction between the number of occurrences of m_i

Table 1. Learning features

Feature Type	Features	Description
Popularity	Popularity(m_i,m_j)	The probability that mentions denote the same entity
People coreference	Person(m_i,m_j)	The probability that mentions denote the same person
	isSameGender(m_i, m_j)	True if the person mentions has the same gender, and False otherwise
Alias	PatternAlias(m_i,m_j)	True if the mentions is a pattern-based alias of the other, and False otherwise
	KBAlias(m_i,m_j)	True if the mentions is a knowledge-based alias of the other, and False otherwise
	Acronym(m_i,m_j)	The probability that a mention is an acronym of the other
	Abbreviation(m_i,m_j)	The probability that a mention is an abbreviation of the other
Relation	isInRelation(m_i,m_j)	True if there is relation word (e.g. wife, husband) between these two mentions, and False otherwise
	isNounInPreposition(m_i,m_j)	True if one mention is in the a preposition phase, followed by the other mention, and False otherwise
	isLocationHerachy(m_i,m_j)	True if these two mentions are in the different level of a location hierarchy tree (e.g. m_i is a state, while m_j is a country)
String match	SubString(m_i,m_j)	True if one of the two mentions is the substring of the other, False otherwise
	Head(m_i,m_j)	True if these two mentions has the same head word, False otherwise
	Jaccard(m_i,m_j)	Jaccard measure
	DF(m_i,m_j)	TFIDF measure
Distance	CharacterDistance(m_i,m_j)	Normalized distance between two mentions according to characters
	WordDistance(m_i,m_j)	Normalized distance between two mentions according to words
	SentenceDistance(m_i,m_j)	Normalized distance between two mentions according to sentences
Entity type	IsPerson(m_i)	True if m_i is predicted as PERSON, and False otherwise
	IsORGANIZATION(m_i)	True if m_i is predicted as ORGANIZATION, and False otherwise
	IsMISC(m_i)	True if m_i is predicted as unknown, and False otherwise
	TypeMatch1(m_i, m_j)	True if predicted entity types are identical but not unknown, and False otherwise
Mention type	IsNominalMention(m_i)	True if m_i is a nominal mention, and False otherwise

in $\mathcal{KB}$ actually referring to e, and the total number of occurrences of m_i in $\mathcal{KB}$ as mention. Assume the probability for each mention is independent, the probability that two mentions m_i, m_j denote the same entity can be calculated as $p(m_i, m_j) = p(m_i, e)(m_j, e)$.

People-oriented resolution. Two definitions are given firstly: (1) half name: person name with only one token or an appellation plus a single token (e.g. Jack, Mary, Mr. Smith); (2) full name: person name (exclusive appellation words) with token size larger than one (e.g., John Smith, George W. Bush).

It is found that a rather high percentage of documents contain person names, so we specially propose an algorithm as feature for person name coreference resolution, based on the following observations: (1) if the full name of a person is mentioned explicitly at least once in a document, the corresponding half name is usually used to refer to the same person in its context; (2) for full name, it may be referred by different half names, for example, "Richard Abruzzo" was mentioned as "Richard" as well as "Abruzzo" in the same document; (3) for half name, the literally same half name may denote different full names, for example "Bob" denotes both "Bob Behn" as well as "Robert D. Behn" in the same document; (4) for a half name, its full name is usually found right ahead of it at least once, especially when the half name is mentioned the first time.

Given a document, we extract all person names, and divide them into a full name list and a half name list. For each pair of (*full name, half name*), we compute a score about how likely they denote the same person as output. This score is computed based on string similarity p_{ssim}, lexicon-based nickname similarity p_{lsim}, and positional similarity p_{psim}. For example, "Richard" and "Richard Stallman" has a string similarity as 0.5. In our lexicon, "Bob" and "Robert" has a probability of 0.9 to denote the same person. For each pair of half name and full name, its positional similarity depends on the number of full names between them and ahead them. We compute the score using the following formula:

$$\begin{aligned} p(h,f) &= (p_{ssim}(h,f) + \rho p_{lsim}(h,f)) \\ &\quad \times (1 + \theta p_{psim}(h,f)) \\ \rho &= \begin{cases} 1 & (p_{ssim}(h,f) = 0) \\ 0 & (p_{ssim}(h,f) > 0) \end{cases} \end{aligned} \tag{2}$$

Where h is a half name, and f is a full name. $p_{ssim}(h,f)$ is their string-based similarity in terms of Jaccard ratio. $p_{lsim}(h,f) = p_{ssim}(h_n,f) * p_{occur}(h_n,h)$ is the lexicon-based similarity according to person nickname lexicon. h_n is h's nickname extracted from lexicon, for example, "Dick" and "Richard" are nickname with each other. $p_{occur}(h_n,h)$ is the probability that h_n is likely to be used to represent h. $p_{psim}(h,f)$ is the positional similarity between h and f, where $p_{psim}(h,f) = (N(f_h) - N(f_b) - N(f_a))/N(f_h)$, and $N(f_b)$ is the number of full names between h and f, $N(f_a)$ is the number of full names ahead of both h and f, while $N(f_h)$ is the number of full names containing h. Note that only full name with string similarity or lexical similarity will be considered in positional similarity. There are two factors in Eq. 2: ρ is used to active lexical similarity when string similarity equals zero. θ is 1 if f appears ahead of h, otherwise, θ is 0.5. A running example is described as follows, h ("Dick") and f ("Richard Stallman") appear in the same document. $p_{ssim}(h,f)$ equals 0, ρ equals with 1. For h_n ("Richard"), their $p_{lsim}(h,f)$ is computed as $0.5 * 0.85 = 0.425$, $p_{psim}(h,f)$ is assumed to be 0.6, thus the final $p(h,f)$ is $0.425 * (1 + 0.6) = 0.68$.

After that, each pair of half name and full name in given document is assigned a score bounded in [0,1]. Note that the proposed person coreference resolution does not consider the match between full names. For example, if

Table 2. Nickname patterns

Pattern1	Pattern2	Pattern3
aka	*aka*	*known as*
better known as	*nee*	*nickname of*
alias	*whose real name is*	*nickname for*
also known as	*was born*	
nickname	*is/was/once called*	
is/was/once called		

"George W. Bush", "George W. H. Bush" and "Bush" appear in the same document. It will find "Bush" to match one of them, and never explore whether these two full names represent the same person, which will be handled in the named entity disambiguation component in Sect. 5.

Alias detection. We detect aliases based on pattern and knowledge base respectively. (1) pattern: Table 2 shows three types of alias patterns through extending the patterns in [3], "`mention` *pattern1* `alias`", "`alias` *pattern2* `mention`", and "*pattern3* `mention alias`". (2) knowledge base: we query mentions against Freebase to get the alias attributes (e.g. common.topic.alias of Freebase). For example, "The Big Apple" and "The Melting Pot" are obtained by querying "New York City".

Acronym detection. Acronym is a special case for coreference resolution. For example, "Delaware Department of Transportation" may be mentioned by using its acronym "DelDOT". Regarding the special characteristics for detecting acronym, the naive patterns of "*expanded form (acronym)*" and "*acronym (expanded form)*" are very useful. In combination with these two naive patterns above and other two functions (i.e. $AcronymOnline(m_i)$ and $AcronymRules(m_i)$), we propose an algorithm to identify acronyms in coreference resolution as shown in Algorithm 1.

Algorithm 1. Acronym Detection

Input: M
Output: $AcronymMap$

```
for m ∈ M do
    IsA = IsAcronymGuo(m_i);
    if IsA=true then
        AcronymMap ← AcronymOnline(m_i)
    else
        AcronymMap ← AcronymRules(m_i)
```

$AcronymMap$ is a hashmap with mention as key and an acronym list as value. For each mention $m_i \in M$, we first judge whether m_i is an acronym of

some other mention using an effective function $IsAcronymGuo$ [12] (line 2). This function recognizes mention m_i as an acronym, if and only if mention m_i satisfies the following conditions: (1) it contains no more than 4 letters with no less than 2 upper case letters; (2) it must not contain more than 2 lower case letters. If m_i is acronym, we further search all its acronym expansion using online acronym detector[2] $AcronymOnline(m_i)$ given m_i as query. After that we only keep mentions of $AcronymOnline(m_i)$, which exist in the given document (line 3–4). And then we add all pairs (m_i, m_j) in $AcronymOnline(m_i)$ to $AcronymMap$. If m_i is not an acronym, we generate acronym for m_i based on hand-crafted rules, including constructing its acronym by getting the initial capital letters of m_i, extracting the patterns of "*expanded form (acronym)*" or "*acronym(expanded form)*" (line 5–6). Then we extract all other mentions $m_j \in M$ meeting the patterns with m_i to form $AcronymRules(m_i)$. After that, we add all detected pairs (m_i, m_j) in $AcronymRules(m_i)$ to $AcronymMap$.

Relation detection (boolean). For any two mentions m_i and m_j, we detect the following three boolean features: (1) relation detection: if relation cues exist between mentions m_i and m_j (i.e. wife, husband, aunt, uncle, nephew and etc.). (2) preposition detection: if m_i is the head mention in adverbial modifier following a preposition, and m_j is the subject in the modified sentence. (3) location mismatch detection: if both m_i and m_j are locations and belong to different levels in a location hierarchy. For example, in the previous mentioned example, "Galway" is city, while "Ireland" is a country. These features are motivated by the observation that some mentions are most unlikely to be *coreferent*, if some special relation (e.g. above three relations) exists between them.

String match (boolean, double). For any two mentions m_i and m_j, we get the following three features according to their surface string. (1) if m_i is the substring of m_j. (2) if m_i and m_j have the same head word, such as "Grammy-winning singer Albertina Walker" and "Albertina Walker". (3) string similarity: following the stop-words removal (e.g. a, an, the, of, and), we use two basic state-of-the-art measures, Jaccard and TFIDF to obtain a string similarity between m_i and m_j.

Distance measure. We use three types of distance features, which respectively count how many characters, words, and sentences apart the two given mentions are. These features are motivated from our observations that for different types of *coreferent* mentions, their distance features are not always the same. For example, abbreviation mentions are usually laid closely, while appositive and pattern-based aliases are often in the same sentence. Acronym coreference mentions may be laid closely or apart from each other by sentences.

Entity type and mention type (boolean). We use existing natural language processing tool (i.e. Stanford NER) to predict the entity type of each mention, and also check whether a pair of mentions are identical in terms of PERSON, ORGANIZATION, or LOCATION. Moreover, mention type of nominal mention

[2] http://acronyms.silmaril.ie/.

is considered as a feature. These features are motivated from the fact that entities of different types (i.e. PERSON, ORGANIZATION, and LOCATION) have different characteristics, some further processing can be used to handle each of them specially.

Inference. Pairwise classifier is simple and flexible with successful achievements in previous research studies. However, it has disadvantage that it is possible that these independent decision will not be consistent with each other (i.e. transitivity violation). For example, mention m_i and m_j are deemed *coreferent*, m_h and m_j as *coreferent*, there is no guarantee that the classifier will deem m_i and m_h as *coreferent*. After pairwise classifier, we have to do inference to ensure the transitivity consistence: when mentions *coreferent*(m_i, m_j) and *coreferent*(m_j, m_k) are true, *coreferent*(m_i, m_k) must be true.

We propose the following constrains in inference phase based on error analysis introduced in Sect. 4.1. Constraints are used to enforce accurate coreference resolution at testing time [31]. For any mentions m_i and m_j, they should not be *coreferent* if they meet any one of the following four constrains: (1) *gender disagreement*: we detect person gender through extracting appellation words (e.g. 'Mr.', 'Mrs.', and 'Miss'). For all full names and last names, we also use the US census to further predict the gender of the person name. (2) *number disagreement*. If either mention has numbers (e.g. product model number) and they are distinguishing digitals. (3) *category disagreement*. If both mentions are recognized in different categories with the same entity type (e.g. city and province). (4) *relation agreement*. If the coreference mentions are close with each other in position, and there is also a relation word between them.

Some rules above are overlapped with training features. However, there is no conflicts as some of them may be weakened in the training model and we strengthen them here. We built coreference equivalence classes through merging mention pairs in a consistent way, which meets the transitivity and constrains above.

5 Named Entity Disambiguation

There existed some works on named entity disambiguation [16,26,36], we first provide detailed error analysis with examples regarding the different kinds of errors that show up in the basic NED methods (Sect. 5.1). According to these error analysis, we propose a general random walk based solution for NED (Sect. 5.2).

5.1 Error and Challenge Analysis

According to the observations on results from the state-of-the-art methods of NED, some errors decreasing performance of named entity disambiguation are presented as follows.

Obsession over Prominent Entity. The state-of-the-art methods do well when the mentions are linked to prominent entities, this biases to a poor performance when they are working on long tail entities or more ambiguous mention. Taking the following text as example, "Albertina Walker" can be easily disambiguated as the American gospel singer, "Pam Morris" is disambiguated as $Null$. However, the following "Morris" is linked to the popular baseball player "Matt Morris" incorrectly. "Chicago" could be correctly linked to the US city, however it will be much more challenging if it denotes other non-prominent entities (e.g. basketball team, bank name).

> Close friend and WVON radio host Pam Morris says Albertina Walker died Friday morning in Chicago .... Morris says Walker was "a living legend" ....

Ambivalence on String Similarity. Undoubtedly, exact string match is effective in NED. The state-of-the-art methods mentioned above links "Don Evans" in the following example to "Don Evans" (`./wiki/Don_Evans`) or *Null* instead of the correct person "Donald Evans"(`./wiki/Donald_Evans`). The problem is that the correct one is sometimes exclusive from the high-ranking entity candidates in the prior stage based on the initial string similarity filtering.

> George W. Bush also named Don Evans as Secretary of Commerce.

Haste on Emerging Entities. All the emerging entities, which are not registered in existing knowledge bases are always linked to *Null* individually by most of the entity disambiguation methods. However, quite few of these methods explore the relevance between these emerging entities. For instance, all the underlined mentions as follows should be linked to *Null*, among which "Golden Managers Acceptance Corporation" and "Golden MAC" denote the same organization.

> Duff Co. downgraded the program of Golden Managers Acceptance Corporation from Duff 1+ to Duff 1. The assets in the Golden MAC program continue to ..

Classifying them into a coreference equivalence class will be beneficial to the knowledge base population for further use. In this example, a new entry page or disambiguation page could be created for them in Wikipedia or YAGO Knowledge base, and these two mentions in text should be redirected to the same entry page.

5.2 Random Walk Based Named Entity Disambiguation

Some errors in entity disambiguation are caused due to lack of coreference information (e.g. "Pam Morris" and "Morris"), which leads to low ranking or exclusive of the correct entity in the entity candidate list, while the biggest challenge in

coreference resolution is lack of background semantic knowledge (e.g. "Albertina Walker" and "Queen of Gospel"). These two tasks should not be treated individually and it is ideal to correct prior errors from both tasks as more information is obtained in the following steps. Thus, we propose a robust graph based framework for NED.

With the entity extraction (EE), coreference resolution (CR), and named entity disambiguation (NED) from the previous steps, we build a mention-entity graph G as shown in Fig. 2. The left column contains the mentions $M = \{m_1, m_2, \ldots m_{n_m}\}$ extracted from given document, and we get an initial coreference resolution score $\phi(m_i, m_j)$ as edge weight for each pair of mentions using Eq. 1. Mentions within the same equivalence class are linked by solid edge, and other edges between mentions are marked using dashed lines. The right column contains the entity candidates $E = \{E_1, E_2, \ldots\}$ from Yago Knowledge base. We harness the existing knowledge bases (i.e. YAGO), which provides a catalog of entities and their surface names. AIDA [16] presents a disambiguation framework combining local context measurement and global coherence. (1) Local context measurement ψ_l. On the mention side, it collects all the tokens in given text as context. On the entity side, it considers the keyphrases or salient words, precomputed from Wikipedia articles. In addition, it uses WordNet to do syntactic contextualization to obtain phrases typically used with the same verb that appears with the mention in the input text. (2) Global coherence ψ_g. It qualifies the coherence between two entities by the number of incoming links in Wikipedia articles, This motivates from the fact that most texts deal with a single or a few semantically related topics such as rock music or internet technology. We use the similarity values ψ_l and ψ_g for a mention m and entity e respectively.

In Fig. 2, the initial $\phi(m_i, m_j)$ and $\psi(m_i, e)$ have been assigned by our coreference resolution classifier and AIDA disambiguation framework. We update the disambiguation edge weight $\psi(m_i, e)$ through combining the following functions of mention m in given document.

We used the random walk with restart probability α, i.e., the probability with which the random walk jumps back to seed node s, and thus "restarts". Random walk models the distribution of rank, given that the distance random walkers can travel from their source (i.e., mention) is determined by alpha. At each step of random walk with a restart probability α, it jumps to a random node, and with probability $1 - \alpha$ follows a random outgoing edge from the current node. In fact the expected walk-length is $1/\alpha$. The formula now becomes $x' = (1-\alpha)Ax + \alpha E$. Here, alpha is the restart probability, which is a constant between 0 and 1, and E is the vector containing the source of information - i.e. in our case it is all zero, except for the red vertex where our information starts to spread. Ax is the node weight of mention m in the previous iteration, here E is obtained using the following formula, where α is fixed to 0.5 to keep the random walkers not to travel too far.

$$nodeweight(v)_{t+1} = (1-\alpha) \times nodeweight(v)_t + \alpha \sum edgeweight(v, w) * nodeweight(w) \quad (3)$$

5.3 Dictionary Building and Knowledge Population

We handle two cases in dictionary building $dict(M, E, C)$ for mention m as follows: (1) Add linkable mention m to M; (2) The non-linkable mention are supposed to be the new emerging entities. Regarding these newly discovered entity, if there are several mentions within an equivalence class, a representative mention will be created and initial popularity value will be created. With the growing number of discovered coreferent mentions, its popularity value will be updated. When the popularity value is sufficient large, the newly discovered entity could be added to knowledge base. This part is worthwhile further exploration in future. Machine learning and Crowdsourcing techniques could be involved for screening and evaluating newly entities.

6 Experimental Study

6.1 Dataset

We used the following two public datasets for evaluation: (1) APW: 150 Associated Press news articles published on October 1st and 150 published on November 1st, 2010, taken from the GigaWord 5 corpus [32]. Mentions were extracted and matched to entities in Wikipedia as ground truth. (2) CONLL [27]: CoNLL 2003 data, which consists of proper noun annotations for 1393 Reuters newswire articles. All these proper nouns were hand-annotated with corresponding entities in YAGO2.

6.2 Evaluation and Discussion

As shown in Table 3, we use document precision, precision and MAP as evaluation measures for named entity disambiguation [16]. To quantify how the various aspects of our proposed strategies affect the performance of named entity disambiguation, we studied two variations. (1) baseline named entity disambiguation algorithm with random walk; (2) baseline with coreference resolution. (3) baseline with coreference resolution and optimized entity extraction. Experimental results shows the effectiveness of the optimized coreference resolution and entity extraction for entity disambiguation. In this paper, we aim to run through the whole entity markup framework, even with preliminary experimental results. More experimental studies could be conducted, and more advanced methods should be designed for a holistically optimized solution in entity markup.

According to the experimental results, future direction on entity extraction is still promising although it has been studied for many years. Quite a number of experimental errors are raised due to the *Geography ambiguity* especially in the United States. It is common for two cities(towns) sharing the same name in different states, such as “Burlington, New Jersey” and “Burlington, Vermont”. It is also common for the same name denoting both state and city, such as “New York” or “Washington”. State abbreviation is popular such as “Connecticut” and “Conn”. A gazetteer (toponymical dictionary), which is a geospatial dictionary of place names, must be beneficial here.

Table 3. Entity disambiguation evaluation

Data set	APW2010			CONLL-Test		
	Doc. Precision	Precision	MAP	Doc. Precision	Precision	MAP
Baseline	0.8163	0.8093	0.8076	0.7923	0.7424	0.7871
Baseline + CR	0.8168	0.81	0.809	0.8102	0.7587	0.7469
Baseline + CR + EE	0.8187	0.8144	0.834	0.8189	0.7707	0.8016

7 Related Work

Coreference resolution (CR) finds the mentions in text that refer to the same entity [13,19,35,37]. Entity coreference resolution is a well studied problem with many methods and tools [1,2,5,8,9,21,22,33,38,41]. CoNLL (the Conference on Natural Language Learning) 2011 [30] and 2012 [29] included a shared task of coreference resolution in which training and test data is provided by the organizers which allows participating systems to be evaluated and compared in a systematic way. Recently, more work showed that joint models resolve mentions across multiple entities result in better performance than simply resolving mentions in a pair-wise comparison. [22] introduces a joint coreference resolution model which combines events and entities by incorporating verbs from event as context features. [14] focuses on enhancing coreference resolution with named entity disambiguation in natural language processing tasks.

Named entity disambiguation (NED) [11] links the mentions in document to entities registered in the existing knowledge bases (e.g., Wikipedia). Earlier work [4,24] on entity disambiguation exploits local features (e.g., bag of words, n-grams), and compares the lexical context around the ambiguous mention to the content of the candidate disambiguation's Wikipedia text. Later on, extended resources are used to explore semantic features, and the most widely used resources includes WordNet [25], Freebase (www.freebase.com), and Yago [40]. Wikipedia also offers some helpful features, like redirection page, disambiguation page, infoboxe, category hierarchy, and hyperlink. Based on these, work on entity disambiguation has stressed on global features exploration [7,15,16,26], these approaches give high confidence to entity candidates, which are strongly related to each other within one document. Entity disambiguation systems with only local features are strong baseline hard to beat, and the systems combining both local and global features could get marginal improvements. However, the biggest challenge is to find tradeoff between local and global features as they have significant strengths and weaknesses of each [36]. Recent years, some work explored other natural language processing tasks to boost entity disambiguation, such as word sense disambiguation, relation extraction, and coreference resolution.

8 Conclusion

This paper introduces the importance and challenges in entity markup (i.e., entity extraction, coreference resolution, and named entity disambiguation).

A practical entity markup framework is proposed to enhance named disambiguation in combination with optimized entity extraction and coreference resolution. The running examples and preliminary experimental studies prove the proposed strategies to enhance entity markup, enriching knowledge base population.

Acknowledgment. Many thanks Johannes Hoffart and Gerhard Weikum for some discussions relevant with this paper. Thanks Stephan Seufert for his original version of random walk codes.

References

1. Aktolga, E., Cartright, M.A., Allan, J.: Cross-document cross-lingual coreference retrieval. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 1359–1360. CIKM (2008)
2. Bengtson, E., Roth, D.: Understanding the value of features for coreference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, pp. 294–303 (2008)
3. Bollegala, D., Honma, T., Matsuo, Y., Ishizuka, M.: Mining for personal name aliases on the web. In: Proceedings of the 17th international conference on World Wide Web, pp. 1107–1108, WWW 2008 (2008)
4. Bunescu, R.: Using encyclopedic knowledge for named entity disambiguation. In: EACL, pp. 9–16 (2006)
5. Chang, K.W., Samdani, R., Rozovskaya, A., Rizzolo, N., Sammons, M., Roth, D.: Illinois-coref: the UI system in the CONLL-2012 shared task. In: CoNLL Shared Task (2012)
6. Cornolti, M., Ferragina, P., Ciaramita, M.: A framework for benchmarking entity-annotation systems. In: Proceedings of the International Conference on World Wide Web (WWW), pp. 249–260 (2013)
7. Cucerzan, S.: Large-scale named entity disambiguation based on Wikipedia data. In: Proceedings 2007 Joint Conference on EMNLP and CNLL, pp. 708–716 (2007)
8. Durrett, G., Klein, D.: Easy victories and uphill battles in coreference resolution. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (2013)
9. Finin, T., Syed, Z., Mayfield, J., McNamee, P., Piatko, C.: Using wikitology for cross-document entity coreference resolution. In: Proceedings of the AAAI Spring Symposium on Learning by Reading and Learning to Read. AAAI Press (2009)
10. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the Association for Computational Linguistics, ACL 2005 (2005). http://nlp.stanford.edu/software/CRF-NER.shtml
11. Getoor, L., Machanavajjhala, A.: Entity resolution: theory, practice & open challenges. Proc. VLDB Endow. **5**(12), 2018–2019 (2012)
12. Guo, Y., Qin, B., Li, Y., Liu, T., Li, S.: Improving candidate generation for entity linking. In: Natural Language Processing and Information Systems, pp. 225–236 (2013)
13. Haghighi, A., Klein, D.: Simple coreference resolution with rich syntactic and semantic features. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 3, pp. 1152–1161. EMNLP (2009)

14. Hajishirzi, H., Zilles, L., Weld, D.S., Zettlemoyer, L.S.: Joint coreference resolution and named-entity linking with multi-pass sieves, pp. 289–299. ACL (2013)
15. Han, X., Zhao, J.: Named entity disambiguation by leveraging Wikipedia semantic knowledge. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 215–224, CIKM 2009 (2009)
16. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: Proceedings of EMNLP, EMNLP 2011, pp. 782–792 (2011)
17. Isele, R., Bizer, C.: Learning expressive linkage rules using genetic programming. PVLDB **5**(11), 1638–1649 (2012)
18. Jiang, L., Wang, J., Luo, P., An, N., Wang, M.: Towards alias detection without string similarity: an active learning based approach. In: SIGIR, pp. 1155–1156 (2012)
19. Kobdani, H.: Linked open government data: lessons from. Institut *für* Maschinelle Sprachverarbeitung (2012)
20. Kulkarni, S., Singh, A., Ramakrishnan, G., Chakrabarti, S.: Collective annotation of Wikipedia entities in web text. In: KDD, pp. 457–466 (2009)
21. Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., Jurafsky, D.: Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, CONLL Shared Task 2011, pp. 28–34 (2011)
22. Lee, H., Recasens, M., Chang, A., Surdeanu, M., Jurafsky, D.: Joint entity and event coreference resolution across documents. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, pp. 489–500 (2012)
23. Lin, T., Mausam, E.O.: No noun phrase left behind: detecting and typing unlinkable entities. In: EMNLP-CoNLL, pp. 893–903 (2012)
24. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM 2007, pp. 233–242 (2007)
25. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
26. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proceedings of Conference on Information and Knowledge Management, CIKM 2009, pp. 509–518 (2008)
27. Technical report. http://www.mpi-inf.mpg.de/yago-naga/aida/
28. Nakashole, N., Tylenda, T., Weikum, G.: Fine-grained semantic typing of emerging entities. In: ACL (2013, to appear)
29. Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., Zhang, Y.: CoNLL-2012 shared task: modeling multilingual unrestricted coreference in ontonotes. In: Joint Conference on EMNLP and CoNLL - Shared Task, pp. 1–40. Association for Computational Linguistics (2012)
30. Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., Xue, N.: CoNLL-2011 shared task: modeling unrestricted coreference in ontonotes. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, CONLL Shared Task 2011, pp. 1–27 (2011)
31. Punyakanok, V., Roth, D., Yih, W., Zimak, D.: Learning and inference over constrained output. In: IJCAI, pp. 1124–1129 (2005). http://cogcomp.cs.illinois.edu/papers/PRYZ05.pdf
32. Parker, R., Graff, D., Kong, J., Chen, K., Maeda, K.: English Gigaword Fifth Edition. Technical reports HPL-2009-155 (2013)

33. Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., Manning, C.: A multi-pass sieve for coreference resolution. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, pp. 492–501 (2010)
34. Rahman, A., Ng, V.: Coreference resolution with world knowledge. In: ACL, pp. 814–824 (2011)
35. Ratinov, L.A., Roth, D.: Learning-based multi-sieve co-reference resolution with knowledge. In: EMNLP-CoNLL, pp. 1234–1244 (2012)
36. Ratinov, L.A., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to wikipedia. In: ACL, pp. 1375–1384 (2011)
37. Singh, S., Subramanya, A., Pereira, F.C.N., McCallum, A.: Large-scale cross-document coreference using distributed inference and hierarchical models. In: ACL, pp. 793–803 (2011)
38. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to coreference resolution of noun phrases. Comput. Linguist. **27**(4), 521–544 (2001)
39. Spitkovsky, V.I., Chang, A.X.: A cross-lingual dictionary for English Wikipedia concepts. In: LREC, pp. 3168–3175 (2012)
40. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: WWW, pp. 697–706 (2007)
41. Wick, C.M., Culotta, A., Rohanimanesh, K., Mccallum, A.: An entity based model for coreference resolution (2009)