Information Sciences 307 (2015) 18–38

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Resolving polysemy and pseudonymity in entity linking with comprehensive name and context modeling

Ming Zhao-Yan [a,*], Chua Tat Seng [b]

[a] Department of Computer Science, Digipen Institute of Technology, Singapore
[b] School of Computing, National University of Singapore, Singapore

ARTICLE INFO

Article history:
Received 29 November 2013
Received in revised form 6 February 2015
Accepted 14 February 2015
Available online 20 February 2015

Keywords:
Entity linking
Coreference
Polysemy
Pseudonymity
Name modeling
Context modeling

ABSTRACT

Names are important atomic information carriers in unstructured text. Matching names that refer to the same entities is an important issue in text analysis and a key component in many real world applications. Generally referred to as *entity linking*, it is defined as a task that aligns a name mentioned in free text to its corresponding entry in a Knowledge Base (KB). The difficulty of the task lies in the many-to-many correspondence between names and entities, causing the pseudonymity and polysemy issues. Existing work usually focuses on resolving polysemy by aggregating large numbers of loosely arranged features in supervised learning frameworks, with very few targeting the pseudonymity or both issues with the same depth. In this work, we tackle both issues by comprehensive modeling of an entity's name and context: we tackle the pseudonymity by modeling name variants on the query name and the KB title; and polysemy by modeling heterogeneous aspects of the query and KB context. Specially, we harness entity coreferences within query and KB documents together with the external alias resources for modeling name variants, and further use the name variants to identify focused context. Moreover, we propose a recall-boosted retrieval method for efficient candidate entity generation. Experimental results show that our proposed approach outperforms the state-of-the-art systems on the benchmark data.

1. Introduction

Names, such as person, organization and location names, are important atomic information carriers in unstructured text, such as newspaper articles. The referents that these names (or the rigid designators, as defined by Kripke [39]) stand for are called "Named Entities" in text processing research. On the other hand, the definitions and other detailed information about the entities are usually manually compiled and stored as entries in structured Knowledge Base (KB) such as dictionaries or encyclopedias. Entity linking is therefore a task that automatically aligns a name mentioned in unstructured text to its corresponding entry in a knowledge base.

Entity linking has been found useful in many real-world applications. A direct application is in an educational environment where entity linking can provide fast access to reference knowledge in study materials such as lecture notes and assignments. In encyclopedia itself, a new knowledge entry's content can be cross-referenced to existing entries, so as to build comprehensive knowledge interlinking. Wikify [51] is a successful system toward the goal of enriching Wikipedia articles with interlinks. Other systems include the Microsoft Smart Tags in later versions of Microsoft Word and the "Instant

* Corresponding author.
E-mail address: mingzhaoyan@gmail.com (Z.-Y. Ming).

http://dx.doi.org/10.1016/j.ins.2015.02.025
0020-0255/© 2015 Elsevier Inc. All rights reserved.

Lookup" feature of the Trillian[1] instant messaging client. Toward the building of the semantic web [50], entity linking can extend its application scope to providing instant references to any type of web document, such as microblogs, reviews, forum discussion threads, and more.

Entity linking is also an important topic in the text analysis community and the data management community [59]. Toward populating structured knowledge base, Text Analysis Conference 2009 introduced the *entity linking* task [47] that takes an entity mention and the document it appears in as the query and the Wikipedia as the KB. When no entity in the KB can be matched to the query, the query is predicted as NIL. This happens when the KB is not big enough, or the query is a newly emerged concept. Prior tasks such as Web People Search (WePS) [1] and Global Entity Detection and Recognition (GEDR) in Automatic Content Extraction focus on specific types of entities and less structured documents as knowledge bases. With entity linking enriched text, other text processing tasks such as summarization [56], entailment, and text categorization [53,70,65] can also benefit from the additional information attached to the original documents. In Fig. 1, we use an example consisting of a linking query and its KB entry to illustrate the TAC entity linking task [46,32].

However, entity linking is not a trivial task, due to the fact that names are often not the unique identifiers for entities. In other words, the relation between names and entities is not one-to-one but a many-to-many mapping. Specifically, this means that a name may stand for multiple entities (**polysemy**), such as *ABC* may refer to *American Broadcasting Company*, *Australian Broadcasting Corporation*, *ABC (newspaper)*; and an entity may also have multiple names (**pseudonymity**), such as full name, acronym, spelling variations, metaphorical names, and other aliases. For example, *American Broadcasting Company* (an American commercial broadcasting television network created in 1943) can be called as *Alphabet Network* (its alias), *ABC* (its call sign), and *American Broadcasting Company* (its official name). Therefore, the difficulty of entity linking lies in the many-to-many correspondence between names and entities, causing synonym and ambiguity, or the pseudonymity and polysemy issues.

Most existing works have successfully resolved the polysemy (ambiguity) issue [47,34,68], with very few works targeting the pseudonymity issue or both with the same depth. To resolving ambiguity, a typical approach in current work is to aggregate large numbers of loosely arranged features in a supervised learning framework. While the framework works well, it deserves more principled modeling of the problem in order to generate structured features. To tackle the pseudonymity issue, current work usually adopts an external alias list or equivalent. However, this method is subject to availability of such a list and the quality and quantity of items it covers. In view of the above, in this work, we propose to tackle both of the issues by comprehensive modeling of an entity's name and context.

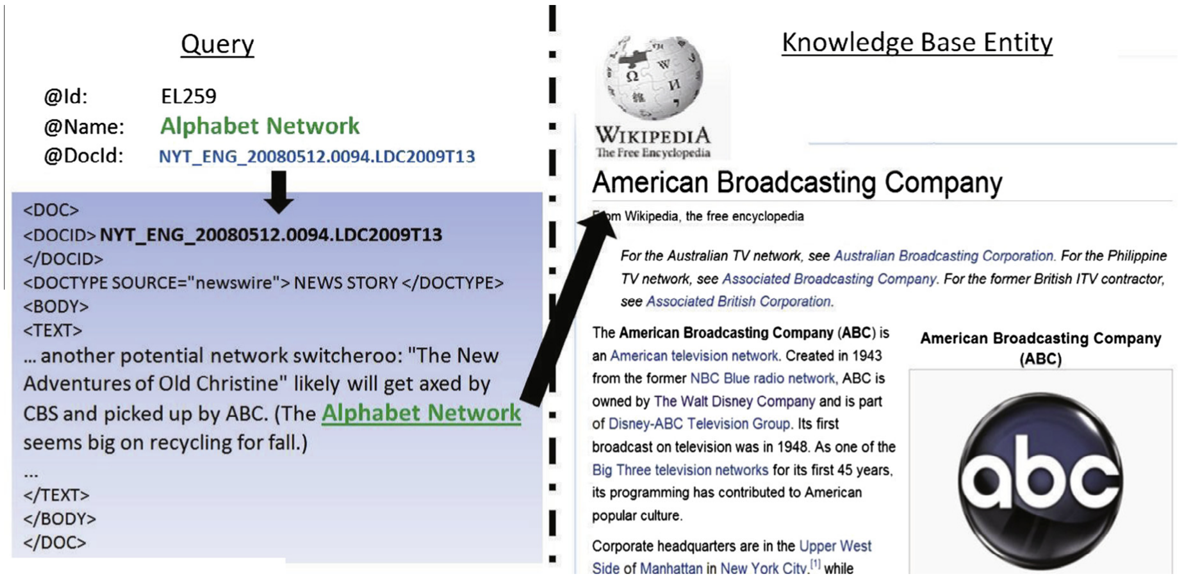
Modeling name variants. As the name is the primary identifier of an entity, the modeling of pseudonymity, or name variants, plays an important part in entity linking. Most existing work explored name variants at the knowledge base side [47,68,2], namely, acquiring name variants for entries in KB. We take this one step further. Besides the external name mapping resources for enriching knowledge base entries, we harness entity coreferences within both query and knowledge base documents for expanding the query mention and the KB entry title respectively. In other words, we solve the pseudonymity issue by modeling name variants of both the query name and the KB title. Specifically, we propose a rule-based entity coreference method based on the Stanford multi-pass sieve framework to find in-document variants of the names. For example, we can find *North Queensland Cowboys*, which appears at the beginning of a query document, as a referent to the query mention *Cowboys*, where the full name is exactly the title of the KB entry linked to the query. At the knowledge base side, we extract alias lists from sources such as "titles of entity pages", "disambiguation pages", "redirect pages", and "anchor texts", which is also widely adopted name variations mining methods in the literature [68,67].

Modeling context. Though names are key identifiers of entities, they are not the unique ones. The fact that many names can refer to multiple entities causes the polysemy issue. Tackling polysemy by appropriate disambiguation models is important for the entity linking task. While most systems have an explicit or implicit context modeling component for disambiguation purpose [6,10,3], we propose a novel method that uses coreferences to identify the more focused context within a document. Besides the surrounding text, some novel aspects of entities, such as the attributes, the popularity, the categories, are also modeled.

Overall, our entity linking system consists of two stages: candidate generation and entity disambiguation. For candidate generation, we propose to use a recall-oriented retrieval model. For candidate disambiguation, we cast the linking between a query mention and a candidate entity in a learning-to-rank framework. Our proposed models for name variants and contexts are embodied as features which characterize matching between query name and entity name, query name and entity context, query context and entity name, and query context and entity context.

We empirically evaluate our proposed method for entity linking with the official Knowledge Base Population track entity linking task data. The experimental results show that the proposed retrieval based entity candidate generation method greatly enhances the recall, which raises the upper bound and reduces the cost of the follow-on computationally intensive entity disambiguation process. For entity disambiguation, the proposed name variant expansion model and context model outperform the state-of-the-art learning-to-rank models with uniform features. We show that the coreference enhanced name matching and context matching models are effective in resolving the pseudonymity and the polysemy issues in entity linking. The contributions of this work are threefold:

[1] http://www.trillian.im/.



Query

@Id: EL259
@Name: Alphabet Network
@DocId: NYT_ENG_20080512.0094.LDC2009T13

```
<DOC>
<DOCID> NYT_ENG_20080512.0094.LDC2009T13
</DOCID>
<DOCTYPE SOURCE="newswire"> NEWS STORY </DOCTYPE>
<BODY>
<TEXT>
... another potential network switcheroo: "The New
Adventures of Old Christine" likely will get axed by
CBS and picked up by ABC. (The Alphabet Network
seems big on recycling for fall.)
...
</TEXT>
</BODY>
</DOC>
```

Knowledge Base Entity

American Broadcasting Company

m Wikipedia, the free encyclopedia

For the Australian TV network, see Australian Broadcasting Corporation. For the Philippine TV network, see Associated Broadcasting Company. For the former British ITV contractor, see Associated British Corporation.

The **American Broadcasting Company** (**ABC**) is an American television network. Created in 1943 from the former NBC Blue radio network, ABC is owned by The Walt Disney Company and is part of Disney-ABC Television Group. Its first broadcast on television was in 1948. As one of the Big Three television networks for its first 45 years, its programming has contributed to American popular culture.

Corporate headquarters are in the Upper West Side of Manhattan in New York City.[1] while

American Broadcasting Company (ABC)

Fig. 1. An example illustrating the entity linking task. On the left is a typical entity linking query consisting of an entity name and the free text article it is mentioned in. On the right presents a knowledge base (Wikipedia in this example) and a KB entry "American Broadcasting Company" that is the referent of the query name.

- First, we propose novel models for resolving pseudonymity and polysemy in entity linking: a customized coreference method for the entity linking task is designed to expand name variants on both the query name and the KB title, which is further applied in identifying the targeted context.
- Second, we propose a novel entity linking framework: a machine learning ranker scores the ambiguous candidate entities with "no entity found" prediction. The structured feature representations are from name modeling and context modeling which resolve the pseudonymity and polysemy issues.
- Third, we propose a recall-boosted retrieval method for efficient candidate entity generation, which is flexible enough to find name variants but sufficiently restrictive to produce a manageable candidate list despite a large-scale knowledge base.

The remaining sections are organized as follows. We first review related literatures in Section 2. Section 3 presents the ranking-based entity linking framework, including the proposed candidate generation method and the learning model. Sections 4 and 5 detail the name modeling and context modeling and their associated features. We conduct empirical evaluations in Section 6 and conclude the work in Section 7.

2. Related work

2.1. Cross-document entity coreference

Given a query consisting of an entity string and a contextual document it appears in, entity linking is to determine which entry (or none of the entries) in a Knowledge Base (KB) of entities is being referred to by the query [47]. Entity linking with knowledge base is closely related to cross-document entity coreference. Cross-document coreference analysis refers to the process of determining whether or not two mentions of entities in different document refer to the same entity [37]. While in the cross-document entity coreference task mentions of the same named entity across documents are to be grouped or chained together, entity linking can be seen as resolving cross-document entity coreference between the name mention's context document and the knowledge base documents. Bagga and Baldwin [4] tackled the problem of cross-document coreference by comparing, for any pair of entities in two documents, the word vectors built from all the sentences containing mentions of the targeted entities. However, these studies targeted the clustering of all mentions of an entity across a given document collection rather than the mapping of these mentions to a given reference list of entities.

The key challenge of resolving ambiguities in entity linking is also tackled in cross-document coreference. A typical disambiguation method is to represent the entities by their surrounding text and measure the degree of contextual similarity to determine the entities to be clustered in the same group [4]. For example, in the early work by [4], they presented a successful method of resolving people with the same name using the vector space model. Gooi and Allan [17] centered on each mention, creating entity models such as the Incremental Vector Space, KL-Divergence, Agglomerative Vector Space, and clustered the entities in order to create coreference chains (equivalence classes where every mention in the same class refers to the same entity). The textual context based disambiguation method is well utilized in entity linking too. However, the successful coreference resolution is insufficient for entity linking, as the coreference chain must still be correctly linked with the proper knowledge base entry. The line between cross-document coreference and entity linking is getting closer. In Mayfield et al.'s [45] cross-document coreference resolution work, they actually referred to the entity coreferencing between free texts and knowledge bases. Ng et al. [60] achieved great success by employing world knowledge to enhance coreference resolution. Hajishirzi et al. [23] proposed a novel algorithm to the two problems, coreference resolution and named-entity linking, jointly.

General noun phrase (NP) coreference resolution, including within-document and cross-document, is well studied in computational linguistics [64,57]. Machine learning approaches to the NP coreference resolution include work by Soon et al. [64] and Ng and Cardie [57] where the problem is cast as a classification task in which two NPs are classified as coreferences or not based on constraints learned from the annotated corpora. Soon et al. [64] applied a decision tree induction based coreference system to standard coreference data set, achieving comparable performance to the best-performing knowledge-based coreference engines. In [57], a follow-on clustering step identifies the possible contradictory pairwise classifications and constructs a partition on the set of NPs. These approaches can be used in the entity linking task, where the query mention's coreferents found in the query document can be used to expand the name variants.

2.2. Entity linking with knowledge bases

Bunescu and Pasca [6], Cucerzan [10], Mihalcea and Csomai [51], and Milne and Witten [52] presented pioneering work in the area of entity linking. Bunescu and Pasca [6] proposed approaches to link person names in Washington Post news articles with their referent Wikipedia article, while Cucerzan [10] proposed to link all entities mentioned in news articles. Mihalcea and Csomai [51] developed Wikify, a system which is concerned with the automatic link generation in Wikipedia by first detecting and then identifying the terms and phrases from which links should be made. Following them, a task on entity linking was organized under the Knowledge Base Population track at TAC since 2009 [47], where person, organization, and geo-political entities appear in news articles are to be linked to Wikipedia entries.

Recent entity linking systems usually consist of two major steps: candidate selection and entity disambiguation. Hachey et al. [22] analyzed systems by considering three steps: *extract*, *search*, and *disambiguate*. Here we see the candidate selection step as equivalent to the combination of extraction and search. Sil and Yates [63] sought to jointly solve the named-entity recognition and linking problems, while in this work we assume the entities are already recognized. The candidate selection step generates a manageable set of possible linking targets for the disambiguation step, thus reduce the time required for the whole process. Using known name variants as the candidate list is adopted by a number of systems. The NLPR_KBP system [27] first built a name-to-entity dictionary using the redirect links, disambiguation pages, anchor texts of Wikipedia; they then generated the name mention's candidate entities by looking up the name's corresponding entry in the dictionary. Similar approaches are adopted by Zhang et al. [68] where they first tried to find variations for every entity in the KB and then generated an entity candidate set for a given query. Recently, Guo et al. [20] proposed a novel candidate generation approach to generate high recall candidate set with small size. The main framework is based on the longest common string between the query and the candidate. In this work, we propose a more comprehensive matching scheme, which considers both word level and string level matches.

After candidate generation, entity disambiguation is followed. It is usually seen as the key challenge in entity linking systems and sometimes used interchangeably with "entity linking". Generally, two main categories of approaches, unsupervised and supervised, have been proposed for entity disambiguation. In the unsupervised case, Cucerzan [10] proposed a similarity based vector space model for linking an ambiguous mention in a document with an entity in Wikipedia. The approach ranks the candidate entities and chooses the entity with maximum agreement between the contextual information extracted from Wikipedia and from the document, as well as the agreement with the category tags associated with the candidate entities.

Hoffart et al. [30,31] represented a mention and candidates as vectors of their contextual noun phrase and co-occurring named entities, and use the similarity between the vectors to determine which candidate the mention refers to. Han et al. [27,24,25] proposed a generative probabilistic model to leverage heterogeneous entity knowledge (including the entity popularity, name, and context) for the entity linking task. Varma et al. [66] adopted an information retrieval approach where they indexed the knowledge base and handled queries that are acronym and not acronym differently. Gottipati and Jiang [18] proposed an approach to entity linking based on a statistical language model-based information retrieval model of KL divergence ranking function with query expansion, using both local contexts especially the name entities and global world knowledge to expand query language models. Graph-based approaches are also adopted by a number of systems [21,19] where global relationship between entities are considered collectively [61,58]. More recently, Dalton and Dietz [11] introduced the neighborhood relevance model which uses relevance feedback techniques to identify the salience of entity context using cross-document evidence.

In contrast to the unsupervised approaches, supervised methods have the advantage of combining the features without the need to tune the rules or weights for the specific data sets. Supervised methods to entity disambiguation mostly concern with two major issues: the selection of a learning model and the representation of query mentions and entities in feature spaces. For the selection of learning models, the entity disambiguation task of one query mention with multiple candidates is cast into either a classification model [6,52,68] or a ranking model [12,69]. Classification models learned from labeled data to determine whether or not a query refers to the candidate entity. Milne and Witten [52] considered the candidate entities as the different senses of the query mention. They used three features to train a classifier: the commonness of the sense (or the candidate entity), the relatedness of the query to the sense's surrounding text, and the context quality of the sense. The sense (or the candidate entity) that has the highest probability is chosen as the target if there is more than one candidate classified as answers. They experimented with classification algorithms like Naive Bayes, C4.5, and SVM, and found that C4.5 worked best in their setting.

In Zhang et al.'s [68] binary classification framework, the training or testing instances are formed by representing each (query, entity) pair as a feature vector. The (query, entity) pairs that are linked with each other are the positive examples, otherwise negative. They leveraged the hyperlink from the Wikipedia article containing the query mention to the entity to build the positive training set, and trained an SVM classifier for new cases. When there were more than one entities labeled as positive for a single query, a token-based vector space method was utilized to further rank the entities.

For supervised ranking model setting, a list of candidate entities is considered for a given query. Learning-to-rank models such as ranking SVM [36,8] and ListNet [7] are employed to rank the candidates so as to pick out the best one. Dredze et al. [12] defined ordered pair constraints for the incorrect and correct KB entries corresponding to a query mention in a supervised machine learning ranker, trained the model such that correct entry has a higher score than the rest, and used the library SVMrank to solve the optimization of the loss function. Finally, a single correct candidate is selected for a query. While Dredze et al.'s [12] ranking model is a pairwise approach, Zheng et al. [69] adopted the listwise methods where the constraints are based on a list of candidates for a query in the training set. They compared the pairwise learning-to-rank method and the listwise method, with two classification methods, SVM and Perceptron. They found that the learning-to-rank methods are significantly better than the classification methods. Recently, He et al. [28] adopted deep neural network to learn the document and the entity representation. This method has been shown to outperform the state-of-the-art method. He et al. [29] proposed a collective disambiguation method which a global predictor is learned based on a local predictor. However, these method mainly aim to tackle the polysemy issue, but not touch on the pseudonymity issue.

Supervised ranking models have some inherent advantages over the classification models. First, ranking models handle a list of candidates at a time therefore more suitable for the selection of the top one as the answer. Second, ranking models take into account the relative strength of the candidate entities with respect to a query while classification models fail to consider the relation between candidates. Third, ranking models tackle the NIL queries in a natural way by adding a pseudo NIL entity, resulting in a more uniformed entity linking framework. The results from the official entity linking task also confirm that ranking based methods outperform the others when the same features and resources are used [34,33]. Though by using a large number of features, these supervised methods are able to tackle the polysemy and pseudonymity issues in some way. However, they are less comprehensive and lack a structured explanation. Considering the above, in this work we choose the supervised ranking methods as our framework and model the name variants and the context information explicitly in a structured way for the entity linking task.

Li et al. [43] collect additional evidences scattered in internal and external corpus to augment the knowledge base and enhances its disambiguation power. They have shown that this method can greatly improve the named entity disambiguation performance on short texts. In particular, they used the hyperlinks in Wikipedia and page labels as the evidences from internal corpus, which is similar to our approach of modeling the semantic context. For external corpus, they used Google search result. Though we do not have such a component, the social context which takes information from Twitter can be seen as the external evidence we use in this work. Similarly, Sil et al. [62] proposed to explore external knowledge so that their system can handle any knowledge database. Cornolti et al. [9] proposed a general framework for comparing entity annotation systems on various dataset. In this work, we only focus on the entity linking part of the entity annotation task. More recently, Jin et al. [35] studied the entity linking problem for rare names, such as unknown entities and phrases, which further expanded the scope of the problem. Meij et al. [49] explored the use of entity linking and retrieval for semantic search, which started an application scenario for the methods we developed for the entity linking problem itself.

3. Entity linking framework

3.1. Problem formulation

Given a query $Q : (q_m, q_c)$, where q_m denotes the name mention in the query and q_c denotes the q_m's context document, and a knowledge base $\mathcal{K} = \{E_j : j = 1, \ldots, N\}$ consisting of N entities where each entity $E : (e_m, e_d)$ is represented by its name e_m and description e_d, the entity linking task can be formulated as a ranking problem that identifies from the whole list of $\mathcal{K}$ a knowledge base entry E^* that is equivalent to the entity expressed by Q. To instantiate the degree of matching, we use $h(Q, E_j)$ to denote a ranking function producing the matching score between the query Q and the candidate entity E_j. As the size of $\mathcal{K}$ is usually in millions, to reduce the computational cost, candidate generation is usually performed as a first step. A subset $\mathcal{K}'$ of $\mathcal{K}$ with a size of M $(M \ll N)$ entities are actually examined. The method for generating the candidate set is detailed in Section 3.2.

So far we assume that a query mention Q has a target entity E in $\mathcal{K}$. When Q has no referent entity in $\mathcal{K}$, it is called a NIL linking. To learn when to predict NIL in ranking models uniformly, we augment $\mathcal{K}'$ to further include a pseudo entity *NIL*, following the approach first proposed by [48]. In other words, $\mathcal{K}'$ includes the NIL entity plus all the entities that are generated as the candidates for the query.

To compute the matching score $h(Q,E)$ for ranking, we take a structured approach: the query name mention and its context and the knowledge base entry's title and description text are matched in a symmetric way. Specifically, four matches are conducted, namely: $f_{mm}(q_m,e_m)$ where the query name is matching with the knowledge base entry name (or title), $f_{cc}(q_c,e_c)$ where the query context is matched with knowledge base entry context (or description), and $f_{cm}(q_m,e_c)$ and $f_{cm}(e_m,q_c)$, where the names are searched against one another's contexts respectively. $h(Q,E)$ is thus defined as a ranking function that takes the four matching scores as input $h(Q,E) = h(f_{mm}(q_m,e_m), f_{cc}(q_c,e_c), f_{cm}(q_m,e_c), f_{cm}(e_m,q_c))$.

Within the above structured matching framework, pseudonymity is solved by modeling name variants on both the query name q_m and the KB title e_m, using entity's coreferents and external world knowledge, in order to expand them to be two sets for potential match; this part will be detailed in Section 4.

Polysemy of a name mention is disambiguated by modeling the query context q_c and the KB description e_c. Various approaches have been proposed to solve the ambiguity problem. In this work, we integrate all the effective features in existing work and propose new ones. In particular, we use the sentence set identified by the coreferent entities as more focused context. We further propose some novel aspects of the context such as semantic context and social context of the entity, which will be elaborated in Section 5.

3.2. Recall-boosted retrieval for candidate generation

A KB usually contains millions of entries (each of which represents one entity). Among them, some of the entities are closer to each other, which usually requires dedicated process to distinguish them. To reduce the computational cost by applying the sophisticated and more expensive process on a fraction rather than all the candidates, we propose to generate a smaller candidate list from the whole list [41]. Therefore, as a first step of the entity linking system, we need to reduce the candidate set while ensuring high recall.

Most of the existing systems address candidate generation mostly by the matching of names: the query mention and the KB entity titles, in terms of tokens and string similarity [12,48,68]. However, as some query mentions are orthographically different from the titles of their referents in the KB, it may cause failures in the name-based candidate generation. Therefore the context should be considered at this early stage in case the name matching fails.

To further improve the recall of the candidates, we propose to augment the name-based approaches with a number of recall-boosting features in an efficient retrieval model [44]. While existing systems try to find candidates by word-level matching, we further include character-level matching to account for possible spelling variations. Besides the document as a whole, we also consider the document parts such as title, first paragraph as indexing fields. This will give more weights to those document parts and potentially produce better retrieval accuracy. In other words, the whole KB is indexed with carefully designed fields at both word and character levels. At searching time, the entity linking query, including the name mention and the context, is represented in a structured search query against specific KB fields.

We implement the retrieval model based on Lucene,[2] the de facto standard for search libraries. We choose Lucene for its efficient and stable performance of indexing and search. Table 1 summarizes the fields of KB entries (3rd column) on which we built index and the fields of queries (2nd column) which we used to search the indices. In the table, 'word' as the indexing unit (1st column) means that we tokenize the corresponding texts word by word and then index the words. We use 'Character' as the indexing unit for titles and their acronyms to include different permutations of characters in names. For example, "Macao" will be indexed as five characters and get some similarity score when searched with "Macau". At the word level, the two names are different.

Following [44], the overall ranking score for one KB entry is a linear combination of the individual scores obtained by the field pairs. Specifically, for each KB entry we index these fields by words: *title*, *info box*, *first paragraph of the description*, *entity article* (the whole article of the entity description), and *info box*. Besides the indexing fields proposed by [44], to further account for the variations, we index the following fields of a KB entry by characters: *title*, *title acronym* (the title acronym is extracted by taking the first character of each non-stop title words; if the title is already an acronym, this field is the same as *title*). Note that we further augment the fields by *expanding the query name and the KB entry title with their variants*. The variants of query names are mostly from their coreferences in the source documents, and those of entity titles are from the KB. We will detail our method for modeling names in Section 4.

The above proposed retrieval based candidate generation aims to achieve high recall by employing the large number of attributes as the searchable fields. Moreover, pre-indexing the fields ensures high efficiency at the same time. Our experiments show that the proposed method uses an average of 800 ms per query on a computer with a 1.7 GHz CPU and 2 GB RAM.

3.3. ListNet learning-to-rank framework

Given the uniquely identified knowledge base, one query usually has only one (or no) correct matching entity. We thus formulate the task as a ranking problem where multiple candidates are ranked. We consider the best match entity ranks the first with a higher relevance score than the second ranked entry, and so on. Ranking models use the differences of feature

[2] http://lucene.apache.org/core/.

Table 1
Search query fields vs. (indexed) KB entry fields. Note that we further augment the fields by expanding the query name and the KB entry title with their variants.

Indexing unit	Search query field	KB entry field
Word	Name	Entity title
Character	Name	Entity title
Word	Name	Entity info box
Character	Acronym	Acronym of entity title
Word	Acronym	Entity article
Word	Context sentence of the mention	Entity info box
Word	Context document of the mention	Entity article

values, which capture the features better than classification models that use absolute feature values. In theory, all the ranking models are applicable. Here we choose the ListNet learning-to-rank framework for its proven effectiveness [69]. Below we reiterate the ListNet model in the entity linking scenario. Note that the contribution of this work is not the use of the ListNet framework, but the formulation of entity-query relations that reflect in the feature design.

Particularly, a training set is given as $S = \{(\mathbf{v}^i, \mathbf{y}^i) : i = 1, \ldots, m\}$, where $\mathbf{v}^i = (v^i_1, \ldots, v^i_{n^i})$ denotes a list of feature vectors for Q^i and its candidate entity lists $E^i_1, \ldots, E^i_{n^i}$, and $\mathbf{y}^i = (\mathbf{y}^i_1, \ldots, \mathbf{y}^i_{n^i})$ is the corresponding list of scores. For each v^i_j, the feature vector created for a query-entity pair (Q^i, E^i_j) the ranking function h outputs a score $h(v^i_j)$. For $\mathbf{v}^i$, we obtain a list of scores $z^i = (h(v^i_1), \ldots, h(v^i_{n^i}))$. The scores produced by the ranking function and the real scores associated with the training data are mapped into probability distributions, and the metric between the distributions are used as the loss function.

Following the formulation in [7], to make the calculation tractable, the top k probability of objects is considered rather than the full list. Denote the top k subgroup of permutations as $\Im_k(j_1, j_2, \ldots, j_k)$, the top k probability of $(j_1, j_2, \ldots, j_k)$ is calculated as:

$$P(\Im_k(j_1, j_2, \ldots, j_k)) = \prod_{t=1}^{k} \frac{h(v_{j_t})}{\sum_{l=t}^{n} h(v_{j_t})}. \tag{1}$$

With cross entropy as metric, the loss with respect to the training data for the ListNet model can be written as

$$L(\mathbf{y}^i, z^i) = -\sum_{\forall g \in \Im_k} P_{y^i}(g) \log(P_{z^i}(g)). \tag{2}$$

Denoting the ranking function as h_w based on the Neural Network model w, the gradient of $L(y^i, z^i)$ with respect to parameter w can be calculated as:

$$\Delta w = \frac{\partial L(\mathbf{y}^i, z^{i(h_w)})}{\partial w} = -\sum_{\forall g \in \Im_k} \frac{\partial P_{z^{i(h_w)}}(g)}{\partial w} \frac{P_{y^i}(g)}{P_{z^{i(h_w)}}(g)}. \tag{3}$$

In the learning process, w is updated with learning rate η by $w = w - \eta \times \Delta w$ in each iteration. More details on ListNet formulation can be found in [7].

At the training stage, for each instance $(\mathbf{v}^i, \mathbf{y}^i)$, only the target entity's corresponding score $\mathbf{y}^i_j$ is assigned to 1, the other scores in the list are set to 0. For queries that have no target entities in the knowledge base, the whole list of scores are 0. In testing, when a new query $Q^{i'}$ and its associated candidate entities $(E^{i'}_1, \ldots, E^{i'}_n)$ are given, we construct feature vector $\mathbf{v}^{i'}$ from them and use the trained ranking function to assign scores and rank the candidate entities in descending order of the score. The top ranked entity will be the predicted target to the query mention; if the pseudo entity NIL gains the highest score, we regard the query mention has no corresponding entity in the knowledge base. Note that as NIL is not a real entity, we aggregate all the candidates for a query to generate a pseudo feature vector to represent it (Section 5.5). It is in accordance with our intuition that a NIL decision can only be made after examining all the candidates.

In the followings, we will detail our proposed models for generating the features, which capture the relation between a query and an entity in terms of name-based (Section 4) and context-based matching (Section 5). Note that in the work by [44], some name-based and context-based features are also proposed in a learning framework. While we propose this approach independently, this is not surprising that the basic approach is agreed among entity linking researchers. In particular, in the following, we propose a more comprehensive feature design, where the customized co-referents name modeling, the social media based context features, and the name-context matching features, accentuate the novelty of this work.

4. Modeling name variants

We model name variants of both the query name q_m and the KB title e_m in order to tackle the pseudonymity issue. Two components are proposed: (a) for expanding q_m, we propose to harness entity co-referents of the query mentions in the query context document and (b) for expanding e_m, we utilize the available resources in Wikipedia to build an alias list for each KB entry as have been done in [10,5,12]. Existing work used co-referents to model name variants usually applied some off-the-shelf NLP tools without considering the specific requirements in the entity linking task. In this work, we pay special attention on customizing the co-reference components. Further, for the first time, we use the co-referents found in context matching. For utilizing external resources in Wikipedia, we exhaust all the available alias mapping lists we can find in existing literature.

4.1. Query name expansion using entity coreference

As writers often use name variants to avoid duplicate expressions or use short names such as acronyms to alleviate long typing, it is fairly common for one of the mentions of an entity in a document to be a long, typical surface form of that entity (such as, "North Queensland Cowboys"), while the other mentions are shorter surface forms (such as, "Cowboys"). We also observe that longer mentions are usually used for an entity's first appearance in a document. Motivated by the above, we expand the query name mention e_m with its coreference NEs in the query document using coreference resolution techniques.

To find entity mentions that have high-confidence of referring to the query mention in the query context document, Zhang et al. [67] paid special attention to finding expanded names for acronyms. They proposed a supervised learning algorithm to expand acronyms in order to reduce the ambiguity of the acronym mentions. In this work, instead of tackling acronyms only, we utilize the full-fledged coreference resolution system: the Stanford NLP group's coreference resolution system [40], and adapt it to the entity linking task. The Stanford system is a collection of deterministic coreference resolution models that incorporate lexical, syntactic, semantic, and discourse information. In the system, the models are applied one at a time from highest to lowest precision in a sieve architecture. Lee et al.'s [40] multi-pass sieve coreference resolution framework guarantees that stronger features are given precedence over weaker ones and that each decision is made using all of the information available at the time. Though it is one of the best off-the-shelf coreference systems, we still need to adjust and adapt it for our task. Firstly, we want to make sure the query mention is processed. Since it is not originally designed for the entity linking task, the original system may chunk the mentions in a way that the exact query mention is not covered for processing. Moreover, as the entity linking system usually deals intensively with acronyms and string variations, we propose to adapt the coreference module to solve them. Specifically, we add three new sieves into the original sets of sieves, as summarized in Table 2 where new sieves are highlighted in boldface. As the original framework is highly modular, new coreference modules can be plugged in without any change to the other modules. We describe the new modules below.

- **Mention Detection:** To make sure that the query mention is a candidate for the follow-on coreference steps, we add the following new condition for mention detection, in addition to e and highest recall sieve (consider all noun phrase, possessive pronoun, and entity mentions) with six exclusion rules as in [40]. This will further increase the recall of the original mention detection component as well. Specifically, we add all the exact appearances of the query mention in its context document as the must-have candidates, in order to eliminate cases when the mentions of concern are filtered out by the original coreference system's mention detection sieve.
- **Relaxed String Match:** This sieve considers two nominal mentions as coreferent if:

  String similarity. The two mentions are non-acronyms and have a longest common subsequence (LCS) similarity score above a threshold (we empirically set it to be 0.75).

  String containing. One mention contains the other mention and they share the same head words [13].
- **Relaxed Acronym Match:** When at least one of two mentions is an acronym, the acronym mention and the other acronym or the acronym version of the other non-acronym mention (we take the leading character of each non-stop word of a nominal mention as its acronym), have a LCS similarity score above a threshold (we empirically set it to be 0.75) and satisfy one of the following constraints:

Table 2
The sieves used in our system: the majority of them are adapted from the Stanford Coreference Resolution system; sieves new to this paper are in boldface.

	Ordered sieves
1.	**Mention Detection Sieve**
2.	Discourse Processing Sieve
3.	Exact String Match Sieve
4.	**Relaxed String Match Sieve**
5.	**Relaxed Acronym Match Sieve**
6.	Precise Constructs Sieve (e.g., appositives)
7–9.	Strict Head Matching Sieves A–C
10.	Proper Head Word Match Sieve
11.	Alias Sieve
12.	Relaxed Head Matching Sieve
13.	Lexical Chain Sieve
14.	Pronouns Sieve

Position Constraint 1. The expanded mention is located in the first paragraph of the context document.
Position Constraint 2. The expanded mention is located in front of the acronym mention within a window of three sentences.
Position Constraint 3. The expanded mention is in the same sentence of the acronym mention and immediately following the acronym or with only punctuation in between.

After applying the coreference sieves, the Stanford system will perform a post processing step to guarantee the output of the system matches the shared task requirement and the official annotation specification, namely by discarding singleton clusters and the mention the appears later in the text in appositive and copulative relations. In the entity linking task, however, we do not need to conform to the requirements, and thus we remove the post processing steps. When the query mentions turn out to be singletons, this indicates that no coreferent is found. When the query mentions appear later in appositive and copulative relations, we will still keep them to ensure coreferents found for the query mentions are kept. We denote the coreferent of the query mention q_m as $q_{m_1}, \ldots, q_{m_p}$ where p is the total number of coreferents found.

In this way, we expand the query name q_m to include the coreferents from the its context document as its variants. The name variants and the original query name form a vector $(q_m, q_{m_1}, \ldots, q_{m_p})^T$, which is denoted as $\mathbf{q}_m$ henceforth. Though not comprehensive, this vector contains the pseudonymity of the query we can find from its context.

4.2. Entity name expansion using knowledge base

On the knowledge base side, each entry represents an entity, with the entity article title as its name. In this subsection, we will explore the available resources in the knowledge base itself to enrich the name representation of the entity name e_m. In general, we will make use of the link structure from Wikipedia, specifically the Redirect, Disambiguation, and Hyperlink, to extract the name variations of an entity as have been done in [10,5,12]. In addition, we will also resolve the coreferents within the entity page as we have done on the query name expansion.

4.2.1. Within document coreference on entity KB page

Entity page in Wikipedia usually consists of a title and a descriptive body of text, as shown in the above example. The page is uniquely identified by its title, a sequence of words with the first word always capitalized. The title is denoted as e_m in this work. As in Section 4.1, we will use the adapted the Stanford Coreference Resolution system to find the coreferent entities of the title, thus enrich the representation of e_m.

Title: American Broadcasting Company
Description:

...
The formal name of the operation is **American Broadcasting Companies, Inc.**, and that name appears on copyright notices for its in-house network productions and on all official documents of the company, including paychecks and contracts. A separate entity named ABC Inc., formerly Capital Cities/ABC Inc., is that firm's direct parent company, and that company is owned in turn by Disney. The network is sometimes referred to as the "**Alphabet Network**", due to the letters "ABC" being the first three letters of the Roman-Latin alphabet, in order.
...

In the above example, applying coreference resolution, "American Broadcasting Companies, Inc." and "Alphabet Network" are found to be coreferents of the entity title "American Broadcasting Company", where the found alias "Alphabet Network" is especially helpful as it is a target mention that shares little string similarity with the entity title.

4.2.2. Redirect pages in KB

Redirect pages are one type of linking structure in the knowledge base which we can use to enrich an entity name. A Wikipedia redirect page typically contains only a reference to an entity page. It is a pseudo page with a title that is an alternate name or spelling for the entity (*e.g.*, *Beehive State* for *Utah*. When a user attempts to access a redirect page like *Beehive State*, Wikipedia server will return the canonical page *Utah* which contains the actual content for describing the entity. Redirect pages can cover a wide variety of name variants, such as acronyms, translations in other languages, common misspellings, and synonyms. By harvesting the redirect pages and their associated canonical pages, we can store these variant titles under the same entity names.

4.2.3. Disambiguation pages in KB

Disambiguation pages are another type of linking structure in knowledge base that we can use to extract the alternate surface forms for an entity name. A disambiguation page is a page created for ambiguous names, *e.g.*, names that denote two or more entities in Wikipedia. It is a specially marked page that has a title in the form of "Entity name (disambiguation)", and a text body that contains a list of references to pages for entities that are typically mentioned using the title. For example, the disambiguation page "Apple (disambiguation)" lists more than 40 associated entities, including "Plants and plant parts", "Companies", "Films", "Television", "Music", "People", "Places", "Technology", and others. By extracting surface forms to entity mappings from the disambiguation pages, additional aliases can be found. Usually, these aliases have some additional information added on to the query, *e.g*. "Apple Inc."

4.2.4. Hyperlink anchors on KB pages

The articles in Wikipedia contain abundant hyperlinks. The anchor text of the links provides another source of name variants, for example, "[[iron (metaphor)|iron metaphor]]". The references to other Wikipedia pages are within pairs of double square brackets. When the reference contains a vertical bar, it indicates that the text at the left of the bar is the name of the referred article (*e.g*., "iron (metaphor)") and the text on the right (*e.g*., "iron metaphor") is the mention surface (also referred to as the anchor of the hyperlink) that is displayed. In other words, the surface form "iron metaphor" is a name variant of "iron (metaphor)". To make full use of the hyperlink/anchor for expanding entity names, we extract all the surface forms to refer to a Wikipedia entity page. To filter out the noise, we keep those anchor text and hyperlink pairs that occur in at least two articles. In particular, to access Wikipedia information for the methods described above for entity name expansion, we use Java Wikipedia Library[3] and the Freebase data.[4]

In summary, we expand the entity name e_m to include the aliases from the knowledge base as its variants. The name variants and the original KB entry title form a vector $(e_m, e_{m_1}, \ldots, e_{m_p})^T$, which is denoted as $\mathbf{e}_m$ henceforth.

4.3. Name based features

As a direct and accurate way to link a mention with its KB entry, the name-based features that matches the query name q_m and the KB title e_m are widely adopted by top performing systems in the official KBP entity linking track [66]. However, these features may not be able to handle the pseudonymity issue where the surface names of the query and the target entity are orthogonally different. With the coreference enhanced name variants $\mathbf{e}_m$ and $\mathbf{q}_m$, we thus can have more name based features, which are summarized in Table 3. We elaborate them in the following.

We divide the name based features into two categories, namely, the name and the acronym. First, we have the "exact match" feature. This is a Boolean feature which is set to one when the query name and the entity title are exactly the same (**NN1**). With the expanded query name set and the entity name set, we have **NN2** which is set to one if one of the names from either set is the same. To account for minor differences such as spelling variation (such as "Air Macao" and "Air Macau"), we measure the string similarity between the query name and the entity title in terms of Longest Common Subsequence and Edit Distance (**NN3**). Similarly, as the names are expanded into sets, we take the highest string similarities between the two sets as additional features (**NN4**). Considering that some query names may be the short forms of the entity titles (such as "Obama" and "Barack Obama"), we have the "containing" feature: when the query/entity names contain each other, the feature **NN5** is set to one. **NN6** is the "expanded name" version of **NN5**.

Given that sometimes the query name or the entity title is an acronym, when one of them is not, we need to extract its acronym to make the matching reasonable. We thus have three acronym matching features here: **NA1** is a Boolean feature which is set to one when the acronym of the query/entity names are exactly the same; **NA2** is the "expanded name" version of **NA1**, which is set to one when it matches one of the expanded query/entity names' acronym; and **NA3**, a real valued feature which measures the string similarity between the capitalized characters in their original order in the two names. The method for extracting acronym from a normal name is the same as that we have used earlier: we take the leading characters of each non-stop words of the normal name as its acronym. While [44] proposed similar features (NA1 and NA3), they did not give the details of their implementation. Moreover, NA2 is a novel feature that is first introduced in this work.

5. Modeling heterogeneous contexts

We address the polysemy issue by modeling the query and the entity contexts. The rational is that entity contexts give information for distinguishing the entities when the names are ambiguous. Traditionally, contexts are referred to as the surrounding texts of a name mention. Here we consider multiple heterogeneous contexts. For the usual text context, we use the coreference information to find more focused context within a document. Besides the text context, we introduce the attribute context, *i.e*., the attributes of the entities; the semantic context, *i.e*., the category or the hyponym of an entity; the social context, *i.e*., the frequency an entity is mentioned in social media. In the following, we will detail each of the contexts one by one.

5.1. Coreference enhanced textual context

Textual context is the commonly recognized context, referring to the whole or part of an article which contains the query or describes the entity. In some work, the context of a name mention is its surrounding word window of a fixed size, for

[3] http://www.ukp.tu-darmstadt.de/software/JWPL.

[4] Freebase provided Wikipedia redirects and Wikipedia hyperlink anchor-entity mapping at: http://download.freebase.com/wex/.

Table 3
Features that instantiate $f_{mm}(q_m, e_m), f_{cc}(q_c, e_c), f_{cm}(q_m, e_c), f_{cm}(e_m, q_c)$ for the ranking based learning model. Features that are first proposed in this work are highlighted in bold font.

Name-*vs.*-***Name*** features for $f_{mm}(q_m, e_m)$	
Name	**NN1**: Exact match between the query name and the entity title **NN2**: **Exact match between the expanded query/entity names** **NN3**: String similarity (longest common subsequence, Dice, edit distance) between query/entity name **NN4**: Highest string similarity between the expanded query/entity names **NN5**: Query/entity name contained in the other **NN6**: **One of the expanded query/entity names contained in the other**
Acronym	NA1: Acronym exact match between the query/entity names **NA2**: **Acronym exact match between the expanded query/entity names** **NA3**: String similarity between the capitalized characters in the two names
Context-*vs.*-***Context*** features for $f_{cc}(q_c, e_c)$	
Textual	**CT1**: Cosine similarity between the TF-IDF vectors of the query/entity article bodies **CT2**: Cosine similarity between the TF-IDF vectors of the sentence containing the query mention and the first paragraph of the entity article **CT3**: **Cosine similarity between the TF-IDF vectors of the sentences containing the query mention and its co-referents and the sentences containing the entity and its co-referents**
Attribute	**CA1**: **Overlap between the sets of countries extracted from the query document and the entity article** **CA2**: Overlap between the sets of time symbols extracted from the query document and the entity article **CA3**: Overlap between the sets of person names extracted from the query document and the entity article **CA4**: Overlap between the sets of NEs extracted from the two text contexts
Semantic	**CS1**: **Similarity between the contexts' term vectors augmented by the category tags** **CS2**: The Wikipedia taxonomy, as used in [6] **CS3**: Type (ORG, PER, GPE, etc.) match
Social	**CC1**: The Wikipedia hyperlink graph in-degree and out-degree of the candidate entity **CC2**: The number of references to the candidate entity's Wikipedia page **CC3**: The rank of the entity's Wikipedia page in a search engine's result for the query **CC4**: **The number of times the entity mention appears in Twitter search results**
Name-*in*-***Context*** features	
	QE1: **The number of times the query name appears in the candidate entity article** **QE2**: **The number of times the expanded query names appear in the candidate article**
	EQ1: **The number of times the candidate entity name appears in the query document** **EQ2**: **The number of times the candidate's expanded names appear in the query document**

example, 50 words [27]. Bag-of-Words (BoWs) approaches such as vector space model, language model, and their variants are widely used in modeling the entity context for disambiguation [48]. Usually, punctuation and functional words are removed before the matching. The underlying principle is similar to the Lesk [42] algorithm for word sense disambiguation, which identifies the most likely sense of a word in a given context based on a measure of context overlap between the dictionary definitions of the ambiguous word. For entity linking, BoWs approaches approximate a fuzzy way of measuring the overlap between an entity mention's context words with candidate KB entity's descriptions.

One such BoWs model, the vector space model, has been widely used in measuring text similarity. We consider a popular variation of this model [71]: given a query context $\mathbf{q_c}$ and an entity context $\mathbf{e_c}$, the similarity score $S_{\mathbf{q_c},\mathbf{e_c}}$ is computed as follows:

$$\begin{aligned} S_{\mathbf{q_c},\mathbf{e_c}} &= \frac{\sum_{t \in \mathbf{q_c} \bigcap \mathbf{e_c}} w_{\mathbf{q_c},t} w_{\mathbf{e_c},t}}{\sqrt{\sum_t w_{\mathbf{q_c},t}^2}\sqrt{\sum_t w_{\mathbf{e_c},t}^2}}, \quad \text{where} \\ w_{\mathbf{q_c},t} &= \ln\left(1 + \frac{N}{f_t}\right), \quad w_{\mathbf{e_c},t} = 1 + \ln(tf_{t,\mathbf{e_c}}). \end{aligned} \tag{4}$$

As summarized in Table 3, we have three textual features. The first textual feature **CT1** measures the cosine similarity between the TF-IDF vectors of the query and the article body of the entity, where N is the number of query documents in the collection, and f_t is the number of query documents containing the term t, and $tf_{t,\mathbf{e_c}}$ is the frequency of term t in $\mathbf{e_c}$.

The second textual feature **CT2** measures the cosine similarity between the TF-IDF vectors of the 50 word windows of the query mention and the entity mention. Here N is the number of 50 word contexts of the query in the collection, f_t is the number of the query's contexts containing the term t. **CT2** is a relatively more focused context similarity measure than **CT1**. Other more sophisticated term weighting models [54,55] can also be applied here.

To improve the performance of the BoWs similarity, an effective approach is to identify the more focused and relevant text within the documents. As we have identified the co-referents of the entity name, a natural next step is thus to use the sentences that the co-referents appear in as the context. Intuitively, this approach will be more targeted than those that use a fixed size surrounding word window. It will also be more comprehensive as those co-referents bearing sentences may not be close to the entity name and thus may not be considered as contexts in the word window approach. Specifically, we harness entity coreferences within both query and knowledge base documents together with the available external name mapping resources, and further use co-referents to identify relevant sentences for context modeling.

Feature **CT3** is the cosine similarity between the TF-IDF vectors of the sentences containing the query mention and its co-referents and the sentences containing the entity and its co-referents. Here N is the number of the sentences containing the query name and its variants in the collection, and f_t is the number of such sentences containing the term t. **CT3** is a more focused and comprehensive context similarity measure than **CT1** and **CT2**.

5.2. Attribute context

We define attribute context as the real world attributes of a query/entity, such as the country a person was born in, the time an event happened, and the persons a person was related to. By matching the attributes of a query mention and a candidate entity, we can obtain more evidences of whether they can be linked or not. Therefore, we propose four attribute context matching features, as presented in Table 3.

As each entity may have various attributes, we consider the general ones, such as the places, the time, and the persons related to the entity. For places, we use only the country names, as more detailed addresses may be very sparse and hard to match. As a result, our first attribute context feature **CA1** measures the overlap between the sets of countries extracted from the query document and the entity article. **CA2** measures the overlap between the sets of time symbols extracted from the query document and the entity article, which are normalized into a predefined format for the matching purpose. **CA3** measures the overlap between the sets of person names extracted from the query document and the entity article. **CA1, CA2** and **CA3** are mainly designed for person (PER), organization (ORG), and geo-political entity (GPE) types of entities, as they are common in practice and cover most of our experimented entities. Other attribute features can be added for specific applications. Moreover, to account for other attributes not covered and the absence of the above attributes, we have **CA4**, a fuzzy way to match attribute context which measures the overlap between the sets of NEs extracted from the two text contexts, as has been done in [68].

For the query context document, we run an NE recognizer [15] to label the NEs present in the text. For the knowledge base articles, we extract all the linked entities, and look up their types in the knowledge base whenever available.

5.3. Semantic context

We introduce the third category of context matching feature, the semantic context, which we refer to as the features derived from the ontology used for the knowledge base schema, or from a related hierarchy or taxonomy. KB entries can be indexed with human or machine generated metadata consisting of keywords or categories in a domain-appropriate taxonomy. In fact, every article in Wikipedia is required to have at least one category. Category information enables the articles to be placed into one or more topics. These topics can be further categorized by associating them with one or more parent categories. We extracted the categories of each entity page and assigned them as tags to the corresponding entity. For the query mention, we use Wikitology [16] to assign a category tag. Specifically, Wikitology uses ontology terms obtained from the explicit category system in Wikipedia as well as the relationships induced from the hyperlink graph between related Wikipedia pages. Based on Wikitology, we compute top ranked categories for the query documents. As a result, we have the semantic context feature **CS1**, which measures the similarity between the two context articles' term vectors augmented by the category tags, and **CS2**, which measures the similarity between the category tags of the query mention and the entity. The category tags are first used in works by Bunescu and Pasca [6] and Cucerzan [10].

Entity type, PER, ORG, or GPE, is also the semantic information we can harness [26]. We thus have another semantic context feature **CS3**, a Boolean indicator of whether the query mention's entity type is the same with the candidate entity's type, *i.e.*, whether both are PER, ORG, GPE, or UKN (unknown). However, the types in the reference KB were incomplete. McNamee and Dang [47] reported that only 35% of KB nodes were assigned a tag of PER (person), ORG (organization), or GPE (geo-political entity); the remainder was typed UKN (unknown). Following their approach, we manually assigned types to 90% of nodes.

5.4. Social context

The last category is the set of **social context** features which is about the larger web context the entity is in. Here, we try to capture the popularity of a candidate entity from various aspects. Though for some document genres it may be unsafe to favor common entities, it seemed beneficial for this task to make use of the popularity degree [48]. We did this in three ways. First, two intrinsic properties of the KB nodes are captured, based on the fact that the nodes were derived from a part of Wikipedia. This results in: (a) **CC1**, which is based on the graph-theoretic properties, *i.e.*, the in-degree and the out-degree of the candidate entity in the Wikipedia graph; and (b) **CC2**, which measures the number of references to the candidate entity's Wikipedia page, which is an indicator of the external attentions that the entity receives. Second, we use the PageRank result from a web search engine, and obtain our third feature **CC3**, which is the rank of the entity's Wikipedia page in

Google's search result for the query name. Third, given the large amount of emerging user-generated-content in social media, we include a fourth social context feature **CC4**, which measures the number of times the entity mention appears in Twitter[5] search results. As a realtime updating social media service, Twitter result is expected to provide more up-to-date social statistics than a common web search engine result. In this work we assume that the entity linking query is issued at current time and the statistics are gathered based on recent tweets. In the case that the query context is about a distant past, we may restrict the Twitter search scope to be within similar period. Note that this group of features is query-independent.

5.5. Features

The choice of features can have an important effect on machine learning performance. As summarized in Table 3, we have **Name-*vs*.-Name** features by modeling query/entity names, and **Context-*vs*.-Context** features by modeling query/entity contexts, which instantiate $f_{mm}(q_m, e_m)$ and $f_{cc}(q_c, e_c)$ respectively.

While [44] proposed similar name-based and context based features (NN1, NN3, NA1, NA3 CT1, CT3, CC1, CC3), they did not give the details of their implementation. Moreover, all the new features are proposed by this work the first time. A new category, name-context match feature, is also introduced by this work.

The remaining new category is the set of **Name-*in*-Context** features, which instantiate $f_{cm}(q_m, e_c)$ and $f_{cm}(e_m, q_c)$. Given a query q and a candidate entity e, we first consider the presence of the query name q_m and its expanded set $\mathbf{q_m}$ in the entity's context e_c. This results in two features: **QE1**, the number of times the query name is found in the candidate entity article; and **QE2**, the number of times the expanded query names appear in the candidate entity article. In the same way, we further consider the presence of the candidate entity name e_m and its expanded set $\mathbf{e_m}$ in the query's context q_c, which result in another two features: **EQ1**, the number of times the candidate entity name is present in the query context document; and **EQ2**, the number of times the candidate entity's expanded names appear in the query context document.

Finally, for modeling the NIL queries, we add some features similar to that done in [48]. Specifically, we include the mean, max, min and deviation for four atomic features for all candidate entities considered. The four atomic features include **NN2** with LCS as similarity, **CT3**, **CA4**, and **QE2** in Table 3. To better use the candidate generation results, we also include in the feature set the raw score of the entity passed from the candidate generation step.

6. Experiments

6.1. Experimental settings

TAC data sets: to evaluate the proposed entity linking performance, we adopt a benchmark set from the official TAC-KBP entity linking track as summarized in Table 4. Each query consists of a mention name and its context document, and is manually linked to an entity from a knowledge base, which consists of 818,741 entities from Wikipedia. We use 3904 queries from TAC2009 [47] as the development set, which are marked in Newswire articles. We use TAC2010 [34] data for training and testing, which consists of 1500 queries for training and 2250 queries for testing. The TAC2010 query mentions are marked in newswire articles and blog posts.

Evaluation metric: we adopt the measure used in the Entity Linking task of TAC-KBP to evaluate the performance of entity linking. This measure is the micro-averaged accuracy, which is defined as the ratio of the number of correctly linked queries by the total number of test queries.

6.2. Evaluating the adapted Stanford Coreference Resolution method

Before we evaluate the impact of within document coreference on entity linking performance, we first evaluate the adapted coreference resolution method proposed in Section 4.1. We conduct a manual examination on a set of 97 documents, which are randomly selected from the development queries' source data set. For each document, we run the original coreference resolution method with the original sieves, and our adapted coreference resolution method with the three new sieves added incrementally (see Table 2). We adopt the MUC measure, the most widely used coreference evaluation metric [40]. It focuses on the links (or, pairs of mentions) in the data. The number of common links between a set of key entities (K) and a set of response entities (R) divided by the number of links in K represents the recall, while, precision is the number of common links between entities in K and R divided by the number of links in R.

We manually check the results from the two methods, and present in Table 5.[6] We can see that by adding new sieves, the coreference performance on the query mentions generally increase in terms of F_1. Note that though our performance is better than the original system on finding coreferents for entity linking queries, it does not show how the new sieves work on general coreference task. Next, we look into the recall and precision to see how the new sieves bring in improvement. When "Mention Detection" sieve is added, the recall increases by 7.4%. This is what we expect to see as the query mention is the "must consider"

[5] www.twitter.com.

[6] Note that we only consider the links that are in the coreference chain of the query mention, which is different from the official coreference evaluation where all the chains are considered.

Table 4
Summary of the TAC dataset used. The development set is the TAC2009 test data, and the training and test data is from TAC2010.

	Development		Training		Test	
Queries	3904		1500		2250	
inKB	1675	43%	1074	(72%)	1020	(45%)
NIL	2229	57%	426	(28%)	1230	(55%)
PER	627	16%	500	(33%)	751	(33%)
ORG	2710	69%	500	(33%)	750	(33%)
GPE	567	15%	500	(33%)	749	(33%)
News	3904	100%	783	(52%)	1500	(67%)
Blog	0	0%	717	(48%)	750	(33%)

Table 5
Performance comparison of the original Stanford Coreference Resolution method and our adapted method by the MUC measure.

System	Precision	Recall	F_1
Original sieves by Lee et al. [40]	0.613	0.593	0.596
+Mention Detection	0.602	0.637	0.619
+Relaxed String Match	0.635	0.634	0.634
+Relaxed Acronym Match	0.656	0.634	0.644

with this new sieve, which rules out the cases when the original system fails to identify the query mention. By adding the "Relaxed String Match" and "Relaxed Acronym Match" sieves, the precision is boosted. This can be explained by the fact that a considerable number of query mentions' coreferents are the short form (or expanded form) of the mentions. Therefore, by emphasizing string similarity and acronym-full name match, the coreference precision on the entity linking type of mentions is improved. From this small scale study, we can see that the proposed adapted coreference resolution method is better suitable for entity linking than the original method. We will further evaluate the influence of the adapted coreference resolution component on the whole entity linking system.

6.3. Evaluating candidate generation

We evaluate our candidate generation method on our development data set. First, we compare our recall-boosted retrieval method with three representative systems as presented in Table 6. We can see that our method outperforms the other three methods, in terms of recall and the number of returned candidates. Our method with only one candidate per query outperforms the best of the three systems that have about three candidates by about 11% from recall of 0.589 to 0.697. Note that we achieve a recall of 0.697 at top 1 (equivalent to the micro-averaged accuracy), which is quite high considering that the candidate generation is only a preprocessing step for entity linking. One contributor of our approach is the character-level search, which can be illustrated by this example: our system successfully finds *Yulia Tymoshenko* (former Prime Minister of Ukraine) as the top candidate for query *Yuliya Tymoshenko*, though the two names are not the same at word level. This shows that our retrieval method with comprehensively designed matching fields is effective in boosting the recall of candidate generation. Another contributor is the coreference-enhanced query name expansion, which we will discuss in the next subsection.

Moreover, by considering the same number of candidates per query, our approach can control the response time for one query easily with little variance across queries when put into practice. As our method can easily control the number of candidates, we can get an even higher recall by slightly increasing the candidate number (recall of 0.761 with three candidates, and 0.987 with 20). Finally the top 20 candidates (per query) are passed to the disambiguation step in the next stage.

6.3.1. Effect of coreference based name expansion on candidate generation

Adding name variants in search queries is a key strategy that we use to enhance recall in candidate generation. When modeling name variants, we mainly use adapted within document coreference to expand query names and knowledge base relation to expand entity titles. In Fig. 2, we compare the performance of the candidate generation with and without name variants from the adapted coreference module at different query limits. We can see that both settings start with a recall of around 65% and reach their maximum at the query limit of 200. A knee seems to appear at 20, which suggests the possibility of certain efficiency gain. However, a further reduction from 20 to 10 results in a substantial drop in the recall, despite possibly higher efficiency.

Comparing the plots in Fig. 2, we can see that incorporating coreference based name variants consistently improves the recall of the candidate generation at all query limits. At lower query limits, the effect is more significant. Though the performance of the two methods gets closer as the limit increases, we do see considerable margin when the query limit is below

Table 6
Performance comparison of candidate generation on TAC2009 data.

System	Recall	# of candidate per query
Bunescu and Pasca [6]	0.563	3.6
Cucerzan [10]	0.586	3.2
Varma et al. [66]	0.594	3.0
Ours	0.697	1.0

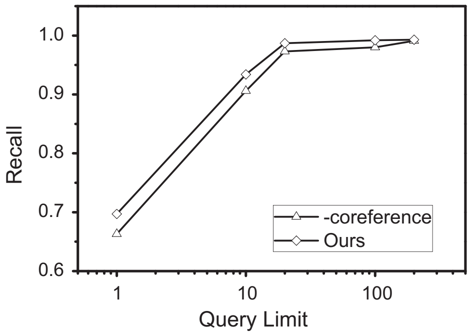


Fig. 2. Effect of query limit and the query name modeling on candidate generation.

20. The maximum gain is at the query limit of 1, which is about 5%. These show that the name variants do play an important part in candidate generation, thus need to be considered at this early stage rather than later during the disambiguation stage. An example to illustrate this is query *EL1689: Iron Lady*. The top four candidates by our full fledged systems are *The Lady Iron Chef*, *Lady Bird Johnson*, *Iron Will*, and *Ellen Johnson-Sirleaf*. The fourth is the correct one but it is hard to be found by general candidate generation methods. Our adapted coreference module found *Ellen Johnson-Sirleaf* as a coreferent to *The Lady Iron Chef* in the query's source document and thus successfully expanded the query name.

6.4. Evaluating coreference enhanced name/context modeling

In this work we aim to solve the pseudonymity issue in entity linking by modeling the name variants, and the polysemy issue through heterogeneous context modeling. As we instantiate the name and context modeling as the features in our learning framework, we do subtractive analysis on these features to evaluate the effectiveness of the modeling. In the following, we will first analyze the three groups of features, namely, name-vs.-name, context-vs.-context, and name-in-context by removing one group at a time; and then analyze the influence of within document coreference by removing those affected features one at a time.

6.4.1. Effect of name, context, and name-context features

In Fig. 3(a), we present ablation study results on removing the features presented in Table 3 one group at a time. Each bar corresponds to the entity linking accuracy after removing one group of feature. The first bar shows the result with all features. The bars shortened most from the all-feature bar represent the most influential features. From Fig. 3(a), we observe that:

(1) Overall, removing any group of features degrades performances, showing that all features contribute to the whole system. Name matching features are the most important group of features, followed by the context matching features, and the name-in-context features are the least influential. This indicates that entities are generally identifiable by their names, and a relatively smaller set of entities require context matching for disambiguation. We conjecture that name-in-context features are dependent on the length of the query source documents and the entity articles, thus cannot provide constant information about query-entity relevance. This also suggests that: when efficiency is concerned, name-in-context features can be excluded; however, if high precision is demanded, all features should be considered.
(2) Name matching contributes more than acronym matching. According to the proposed name modeling, our name matching features actually include name variants, of which acronym is a type. Therefore, name matching subsumes acronym matching to some extent, except for those names whose acronym is not in the same source document.
(3) Text is the most influential context among the proposed heterogeneous contexts. We can see that removing textual context matching features leads to the most performance degradation among the four types of contextual features. This is consistent with findings in [22] that text context similarity is a competitive disambiguation method given a proper candidate generation step. Attribute is the second important context, followed by social context and semantic

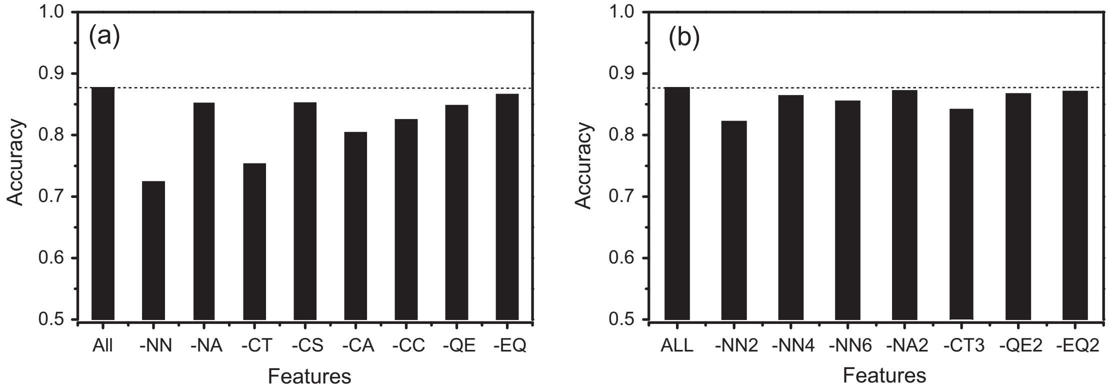


Fig. 3. (a) Ablation study on the name-vs.-name, context-vs.-context, name-in-context features as presented in Table 3. (b) Ablation study on coreference enhanced features. ALL indicates all features, "-" means subtract.

context. By examining the individual queries, we find that attribute context contributes most in ORG and GPE types of queries and social context contributes most in PER queries. We conjecture that this is because that, for ORG and GPE, the location and time attributes could be the key identifiers when the names are ambiguous, while for PER, more popular and influential people's names usually appear more often on the web thus getting higher scores in the social context features. An interesting example is *EL725: Chicago*. Though there is an exact name matched entry *Chicago* in the KB, our system correctly links the query to *Chicago Fire* mainly based on the context. This shows the importance of context modeling in solving the polysemy issue.

6.4.2. Effect of coreference-enhanced features

We mainly use coreference to expand query names and to find the local context of query mentions. According to Table 3, the features enhanced by coreference include NN2, NN4, NN6, and NA2 from name-vs.-name group, CT3 from context-vs.-context group, and QE2 and EQ2 from name-in-context group. To see the effect of the adapted within document coreference, for NN2, NN4, NN6, NA2, QE2, and EQ2, we remove all name variants obtained from within document coreference and keep only those variants from the knowledge base (such as redirects). Similarly, the textual context feature CT3 considers only sentences that contain the original query mention.

In Fig. 3(b), we present the ablation study on coreference-enhanced features by changing them one at a time on our development data. From the figure, we can see that all coreference-enhanced features benefit from the enhancement as evidenced by the fact that all the de-enhanced features produce worse results than the enhanced one. We also note that removing coreference affects the name-vs.-name features (NNs and NA) more than context-vs.-context and name-in-context features. The two most affected features are the exact name match feature (NN2) and sentence-based context feature (CT3). One example is query *EL1870: Little Damascus* from the development data. Coreference resolution finds its pseudonym *Nablus* from the query context, which is exactly the title of the correct KB entry. This suggests that coreference does introduce name variants for exact name match that contributes to resolving the pseudonymity issue. Moreover, it also helps to identify the relevant local context (the sentences contain *Nablus* in the example) in the query source document, resulting in more focused textual context representation and better disambiguation for resolving polysemy.

Note that from Fig. 3(b), the small margin by adding some coreference-enhanced features suggests that these features may be removed when efficiency is of concern. However, considering that the margin is gained over an already high result, the features may be kept when high precision is desirable.

Further we collect some statistics about the sources of correct links. In the results of our method, we find that 76.4% of the correctly linked queries are exact matches to the corresponding entity names and 12.7% of them are found through coreferences. Adding the alias list increases the exact matches to 81.5%, which has some overlap with the coreference found names. This may suggest that coreference is helpful in the "difficult" queries. Though these percentages seem quite high, it does not mean exact match can achieve higher performance than some systems. The results here are based on the correctly linked queries but not all queries. In the later case, there can be some false positives caused by the ambiguity of the queries.

6.5. Overall entity linking results

Our proposed approach (**Ours**) is implemented as described in Section 3.2. For each query the top 20 candidates are sent to the ranking based disambiguation model (Section 3) with name modeling and context modeling generated features (Sections 4 and 5). The query is linked the top one ranked entity, NIL prediction will be given if the top entity is NIL. The ranking model is trained on KBP2010 training data.

6.5.1. Comparison with the baselines

We first include two no-algorithm baselines: **NIL baseline** which predicts all the queries to be NIL, and **Title baseline** which predicts the queries to be linked to entities of the exact title (the redirects of entity titles are considered too). We also compare the proposed learning-to-rank based entity linking model with the two widely adopted machine learning models as the supervised baselines, SVM and SVM_Rank. Note that NIL is added as a pseudo entity for NIL prediction. **SVM** performs a binary prediction for each query-candidate pair. If there are multiple positive prediction per query, the one with the highest probability will be the final output. **SVM_Rank** is a learning-to-rank algorithm which we train and tune on the KBP2010 training data with the same set of features as our ListNet framework. We also compare our results with the KBP2010 systems.

Table 7 presents the end-to-end comparison between our system, baselines, and the benchmark systems on the TAC test data. The first four lines are our baselines. The NIL baseline accuracy is actually the percentage of queries with NIL as the gold answer. The Title baseline is a strong baseline for this task, achieving a score of 11 points above the median and 7.4 points below the maximum score achieved by submissions to the TAC2010 shared task as given in the fifth and sixth rows. As our supervised learning baselines, SVM_Rank works better than SVM, suggesting that ranking is more suitable than classification for the entity linking task. This is in line with our earlier discussion that entity linking is a ranking problem given that one query is associated with multiple candidates. Of all the comparing systems, our approach achieves the highest accuracy for all queries and inKB queries with significant advantage over both supervised baselines.

Among the supervised models, we can see that our method with the ListNet model outperforms the SVM_Rank model. Though both are learning-to-rank models, ListNet works on lists of training instances and SVM_Rank works on pairs of instances. We conjecture that the listwise model performs better because it learns from the differences among all the candidates for a query: the resulting model captures the relations between all the candidates, while the pairwise model learns from two candidates at a time.

6.5.2. Results by genre and entity type

In Table 8, we present the entity linking results by dividing them into entity types and query context genres. Firstly, we can see that our system is consistently better than the baselines across genres and entity types. Next, we notice that the percentage of NIL queries (as reflected in the NIL baseline scores) varies greatly across genre and entity types. In particular, the NIL percentage in news articles is much higher than in blog posts for ORG and PER, but much lower for GPE. The high percentage of NIL for PER (0.91) shows that many people names in news articles are not referred in Wikipedia. The better performance on NIL queries may be attributed to the approaches we take. As introduced in Section 5.5, we use the mean, max, min and deviation of four atomic features as the NIL feature. While SVM_Rank adopts the same features as ours, we conjecture that the ListNet framework can better model the listwise relationship between candidates, which may lead to better prediction of NIL queries.

An interesting observation is that all systems achieve relatively high scores for PER, and the scores vary little across systems, especially in Newswire text. The title baseline that conducts exact word match performs near perfectly on PER (0.97) in news. Given that PER entities have a high NIL percentage, this suggests that those people names can be found in Wikipedia are well canonicalized, which may be attributed in part to the editorial standard associated with the news.

Overall, our system works better on news articles than on blog posts. This holds for the SVM baseline and the SVM_Rank baseline. We conjecture that this may be attributable to the performance of coreference, which works better on news articles that follows the journalistic conventions for introducing new entities into discourse fairly unambiguously. Similarly, the higher writing quality of news articles will also have more accurate name entity recognition results, leading to better context modeling, especially so for attribute context which relies heavily on name entity recognition.

6.5.3. Comparison with the state-of-the-arts

To compare our system with the state-of-the-art methods, we explore the literature and identify a few renowed systems. Further, we make the comparison on additional publicly available dataset besides the TAC data. This is also to measure the flexibility and adaptability of our method.

In particular, we experimented on IITB and MSNBC dataset in addition to TAC data set. IITB is a detailed data set that includes annotations about all the mentions [10]. It contains more than a hundred manually annotated Web pages. MSNBC on the other hand only annotates important entities and their referring mentions [38].

We choose to compare with the following state-of-the-art systems that have public available source codes. **AIDA** [9] annotates mentions using the Stanford NER Tagger and adopts the YAGO 2 knowledge base. We use the local disambiguation configuration where each mention is disambiguated independently from others. **Illinois Wikifier** annotates mentions using the Illinois NER system and links them to Wikipedia. has been designed to deal with English documents of arbitrary length, its software can be downloaded.[7] **Wikipedia Miner** is an implementation of Wikification algorithm in [52]. It is one of the first machine learning approaches to entity linking. Finally, Zhang et al. [67] is a state-of-the-art system that adopts a supervised acronym expansion module. Other systems like TagMe [14] and DBpedia Spotlight [50] provide WEB API, on which we cannot specify the type of entities to be linked thus not compare here. Moreover, since the focus of this research is on entity linking

[7] http://cogcomp.cs.illinois.edu/page/download_view/Wikifier.

Table 7
The overall comparison with the baselines on TAC data.

System	All Queries	inKB	NIL
NIL baseline	0.547	0.000	1.000
Title baseline	0.794	0.606	0.950
SVM	0.804	0.741	0.856
SVM_Rank	0.840	0.774	0.894
TAC 2010 Median	0.684	–	–
TAC 2010 Maximum	0.868	0.806	0.920
Ours	0.873[a,b]	0.836[a,b]	0.903[a]

[a] Statistical significance over the baselines SVM at 0.95 confidence interval using paired *t*-test.
[b] Statistical significance over the baselines SVM Rank at 0.95 confidence interval using paired *t*-test.

Table 8
The entity linking results by entity types and query context genres. ORG is short for organization, GPE for geo-political entity, and PER for person.

System	Newswire			Blog		
	ORG	GPE	PER	ORG	GPE	PER
NIL baseline	0.726	0.210	0.910	0.332	0.566	0.331
Title baseline	0.748	0.656	0.970	0.804	0.767	0.829
SVM	0.765	0.638	0.974	0.867	0.749	0.871
SVM_Rank	0.792	0.757	0.974	0.871	0.763	0.880
Ours	0.829	0.830	0.982	0.900	0.782	0.896

rather than named entity recognition, we only evaluate on the mentions that are recognized by at least one of the systems. For our system and Zhang et al. that do not have an NER tagger, we use the combined NER results of the other systems.

From Fig. 4, we can see that our system performs the best on TAC and MSNBC data sets and slightly worse than Wikipedia Miner on IITB data set. All systems perform best on TAC data and worst on IITB data. We find that IITB annotations are more diversified while the other two are mainly on salient entities, which are inherently more discriminative and entail more context information for disambiguation.

Zhang et al. and our system are in the first tier among all systems. We conjecture that the modeling of name variants is the major reason. In a way, acronym expansion as adopted by Zhang et al. can be seen as part of the within document coreference resolution in our name modeling component. As our adapted Stanford coreference method finds a more comprehensive set of name variants within the query document, we conjecture that our method is better than [67] at solving the pseudonymity issue. Other factors contribute to the difference of the overall performance may include the contextual modeling. Compared to [67] where the context is modeled by combining LDA based semantic similarity with ordinary term matching, our context modeling is more heterogeneous. To handle polysemy, we devise four groups of contextual features for disambiguation, namely, the textual context, semantic context, attribute context, and social context, which are more comprehensive than [67]'s approach.

Among the lowest three systems, Wikipedia Miner performs better than the other two. AIDA does not perform well on IITB data while maintains reasonable performance on the other two data sets. We conjecture that this is caused by the low performance of Stanford NER tagger which is adopted by the AIDA system.

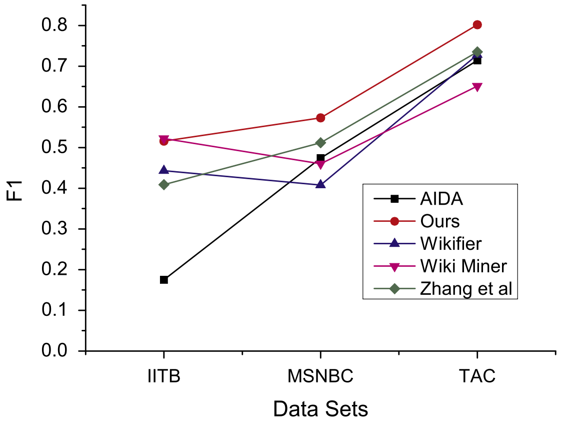


Fig. 4. Entity linking performance comparison with the state-of-the-art systems on three standard data sets.

7. Conclusion

Entity linking that aligns name mentions in free text to their corresponding entry in a knowledge base is an important task in text processing related research problems and a key component in many real-world applications. Pseudonymity and polysemy issues make entity linking a challenging task that most existing systems fail to tackle it comprehensively.

In this paper we proposed a comprehensive framework that models the two constituents of an entity, the name and the context, in order to tackle the pseudonymity and polysemy issues respectively. Particularly, for name modeling, we adapted an open source coreference tool to identify the name and nominal mentions that were deemed coreferents as name variants, in addition to the available external name mapping resources from the knowledge base. For context modeling, we further utilized the coreferent results and extracted heterogeneous aspects of a query and an entity's context for enhanced disambiguation. Moreover, we proposed a novel retrieval-based recall boosting model for efficient candidate entity generation. Experiments on TAC benchmark data showed that our approach outperforms two supervised learning baselines and the state-of-the-art models of entity linking. Detailed analysis showed that coreferent enhanced name expansion greatly improves the name modeling performance on tackling the pseudonymity issue. We also find in our ablation study that the name modeling and context modeling features captured different aspects of the problem. The proposed recall-boosted retrieval method for candidate generation was also proved to be effective by comparing with a few baselines. Overall, our results and findings are in line with Hachey et al.'s [22] recent work on evaluating three canonical entity linking systems. They concluded that "coreference and acronym handling lead to substantial improvement, and search strategies account for much of the variation between systems". We thus conclude that our success can be attributed in part to the fact that our coreference enhanced name modeling goes beyond coreference and acronym handling, and our system has a well-designed search module and comprehensive context modeling.

It remains an interesting future work to apply the proposed method to various problems that require an entity linking component, including some timely issues such as the alignment of users with different aliases in different social networks.

References

[1] J. Artiles, J. Gonzalo, S. Sekine, The semeval-2007 weps evaluation: establishing a benchmark for the web people search task, in: Semeval, 2007, pp. 64–69.

[2] J. Atkinson, G. Salas, A. Figueroa, Improving opinion retrieval in social media by combining features-based coreferencing and memory-based learning, Inform. Sci. 299 (0) (2015) 20–31.

[3] N. Ayat, R. Akbarinia, H. Afsarmanesh, P. Valduriez, Entity resolution for probabilistic data, Inform. Sci. 277 (0) (2014) 492–511.

[4] A. Bagga, B. Baldwin, Entity-based cross-document coreferencing using the vector space model, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, 1998, pp. 79–85.

[5] D. Bikel, V. Castelli, R. Florian, D. Han, Entity linking and slot filling through statistical processing and inference rules, in: Proceedings of Text Analysis Conference, 2009.

[6] R. Bunescu, M. Pasca, Using encyclopedic knowledge for named entity disambiguation, in: The European Chapter of the ACL, Trento, Italy, 2006, pp. 9–16.

[7] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, H. Li, Learning to rank: from pairwise approach to listwise approach, in: ICML, ACM, Corvalis, Oregon, 2007, pp. 129–136.

[8] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, H.-W. Hon, Adapting ranking svm to document retrieval, in: Proceedings of the ACM International Conference on Research and development in Information Retrieval, ACM, Seattle, Washington, USA, 2006, pp. 186–193.

[9] M. Cornolti, P. Ferragina, M. Ciaramita, A framework for benchmarking entity-annotation systems, in: Proceedings of the International Conference on World Wide Web, 2013, pp. 249–260.

[10] S. Cucerzan, Large-scale named entity disambiguation based on Wikipedia data, in: Proceedings of the Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning, vol. 2007, 2007, pp. 708–716.

[11] J. Dalton, L. Dietz, A neighborhood relevance model for entity linking, in: Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, 2013, pp. 149–156.

[12] M. Dredze, P. McNamee, D. Rao, A. Gerber, T. Finin, Entity disambiguation for knowledge base population, in: Proceedings of the International Conference on Computational Linguistics, Beijing, China, 2010, pp. 76–82.

[13] M. Elsner, E. Charniak, The Same-head Heuristic for Coreference, Association for Computational Linguistics, 2010.

[14] P. Ferragina, U. Scaiella, Tagme: on-the-fly annotation of short text fragments (by Wikipedia entities), in: Proceedings of the ACM Conference on Information and Knowledge Management, ACM, 2010, pp. 1625–1628.

[15] J.R. Finkel, T. Grenager, C. Manning, Incorporating non-local information into information extraction systems by gibbs sampling, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, 2005, pp. 363–370.

[16] T. Finin, Z. Syed, J. Mayfield, P. McNamee, C. Piatko, Using wikitology for cross-document entity coreference resolution, in: AAAI Spring Symposium on Learning by Reading and Learning to Read, AAAI Press, 2009, pp. 29–35.

[17] C. Gooi, J. Allan, Cross-document coreference on a large scale corpus, in: HLT-NAACL, vol. 4, Boston, Massachusetts, USA, 2004, pp. 9–16.

[18] S. Gottipati, J. Jiang, Linking entities to a knowledge base with query expansion, in: Proceedings of the Annual Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 804–813.

[19] Y. Guo, W. Che, T. Liu, S. Li, A graph-based method for entity linking, in: Proceeding of the International Joint Conference on Natural Language Processing, Citeseer, 2011, pp. 1010–1018.

[20] Y. Guo, B. Qin, Y. Li, T. Liu, S. Li, Improving candidate generation for entity linking, in: Natural Language Processing and Information Systems, Springer, 2013, pp. 225–236.

[21] B. Hachey, W. Radford, J.R. Curran, Graph-based named entity linking with Wikipedia, in: Web Information System Engineering, Springer, 2011, pp. 213–226.

[22] B. Hachey, W. Radford, J. Nothman, M. Honnibal, J.R. Curran, Evaluating entity linking with Wikipedia, Artif. Intell. (2012) 130–150.

[23] H. Hajishirzi, L. Zilles, D.S. Weld, L.S. Zettlemoyer, Joint coreference resolution and named-entity linking with multi-pass sieves, in: Proceedings of the Annual Conference on Empirical Methods in Natural Language Processing, 2013, pp. 289–299.

[24] X. Han, L. Sun, A generative entity-mention model for linking entities with knowledge base, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2011, pp. 945–954.

[25] X. Han, L. Sun, An entity-topic model for entity linking, in: Proceedings of the Annual Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2012, pp. 105–115.

[26] X. Han, J. Zhao, Named entity disambiguation by leveraging Wikipedia semantic knowledge, in: Proceedings of the ACM Conference on Information and Knowledge Management, ACM, 2009, pp. 215–224.

[27] X. Han, J. Zhao, Nlpr_kbp in tac 2009 kbp track: a two-stage method to entity linking, in: Proceedings of Text Analysis Conference, 2009.

[28] Z. He, S. Liu, M. Li, M. Zhou, L. Zhang, H. Wang, Learning entity representation for entity disambiguation, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2013, pp. 30–34.

[29] Z. He, S. Liu, Y. Song, M. Li, M. Zhou, H. Wang, Efficient collective entity linking with stacking, in: Proceedings of the Annual Conference on Empirical Methods in Natural Language Processing, 2013, pp. 426–435.

[30] J. Hoffart, M.A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, G. Weikum, Robust disambiguation of named entities in text, in: Proceedings of the Annual Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2011, pp. 782–792.

[31] J. Hoffart, S. Seufert, D.B. Nguyen, M. Theobald, G. Weikum, Kore: keyphrase overlap relatedness for entity disambiguation, in: Proceedings of the ACM Conference on Information and Knowledge Management, ACM, 2012, pp. 545–554.

[32] H. Ji, H. Dang, J. Nothman, B. Hachey, Overview of tac-kbp2014 entity discovery and linking tasks, in: Proceedings of Text Analysis Conference (TAC2014), 2014.

[33] H. Ji, R. Grishman, H.T. Dang, Overview of the tac 2011 knowledge base population track, in: Proceedings of Text Analysis Conference, 2011.

[34] H. Ji, R. Grishman, H. Dang, K. Griffitt, J. Ellis, Overview of the tac 2010 knowledge base population track, in: Proceedings of Text Analysis Conference, 2010.

[35] Y. Jin, E. Kıcıman, K. Wang, R. Loynd, Entity linking at the tail: sparse signals, unknown entities, and phrase models, in: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, ACM, 2014, pp. 453–462.

[36] T. Joachims, Optimizing search engines using clickthrough data, in: Proceedings of the Annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 133–142.

[37] R. Kibble, R. Kibble, K. van Deemter, K.V. Deemter, Coreference annotation: Whither? in: International Conference on Language Resources and Evaluation, 2000, pp. 1281–1286.

[38] S. Kulkarni, A. Singh, G. Ramakrishnan, S. Chakrabarti, Collective annotation of Wikipedia entities in web text, in: Proceedings of the Annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Paris, France, 2009, pp. 457–466.

[39] S.A. Kripke, Wittgenstein on Rules and Private Language: An Elementary Exposition, Harvard University Press, 1982.

[40] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, D. Jurafsky, Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task, in: CONLL Shared Task, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 28–34.

[41] S. Lee, J. Lee, S. won Hwang, Efficient entity matching using materialized lists, Inform. Sci. 261 (0) (2014) 170–184.

[42] M. Lesk, Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone, in: Proceedings of the 5th Annual International Conference on Systems Documentation, ACM, Toronto, Ontario, Canada, 1986, pp. 24–26.

[43] Y. Li, C. Wang, F. Han, J. Han, D. Roth, X. Yan, Mining evidences for named entity disambiguation, in: Proceedings of the Annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 1070–1078.

[44] C.-Y. Lin, G. Zheng, Msra at tac 2011: entity linking, in: Proceedings of Text Analysis Conference, 2011.

[45] J. Mayfield, D. Alexander, B. Dorr, J. Eisner, T. Elsayed, T. Finin, C. Fink, M. Freedman, N. Garera, P. McNamee, Cross-document coreference resolution: a key technology for learning by reading, in: AAAI Symposium on Learning by Reading and Learning to Read, AAAI Press, 2009.

[46] K. Mazaitis, R.C. Wang, F. Lin, B. Dalvi, J. Bauer, W.W. Cohen, A tale of two entity linking and discovery systems, in: Proceedings of Text Analysis Conference (TAC2014), 2014.

[47] P. McNamee, H.T. Dang, Overview of the tac 2009 knowledge base population track, in: Proceedings of Text Analysis Conference, 2009.

[48] P. McNamee, M. Dredze, A. Gerber, N. Garera, T. Finin, J. Mayfield, C. Piatko, D. Rao, D. Yarowsky, M. Dreyer, Hltcoe approaches to knowledge base population at tac 2009, in: Proceedings of Text Analysis Conference, 2009.

[49] E. Meij, K. Balog, D. Odijk, Entity linking and retrieval for semantic search, in: International Conference on Web Search and Data Mining, 2014, pp. 683–684.

[50] P.N. Mendes, M. Jakob, A. García-Silva, C. Bizer, Dbpedia spotlight: shedding light on the web of documents, in: Proceedings of the 7th International Conference on Semantic Systems, ACM, 2011, pp. 1–8.

[51] R. Mihalcea, A. Csomai, Wikify!: linking documents to encyclopedic knowledge, in: Proceedings of the ACM Conference on Information and Knowledge Management, ACM, Lisbon, Portugal, 2007, pp. 233–242.

[52] D. Milne, I.H. Witten, Learning to link with Wikipedia, in: Proceedings of the ACM Conference on Information and Knowledge Management, ACM, Napa Valley, California, USA, 2008, pp. 509–518.

[53] Z.-Y. Ming, K. Wang, T.-S. Chua, Prototype hierarchy based clustering for the categorization and navigation of web collections, in: SIGIR, ACM, New York, NY, USA, 2010, pp. 2–9.

[54] Z.-Y. Ming, T.-S. Chua, G. Cong, Exploring domain-specific term weight in archived question search, in: CIKM, ACM, 2010, pp. 1605–1608.

[55] Z.-Y. Ming, K. Wang, T.-S. Chua, Vocabulary filtering for term weighting in archived question search, in: Advances in Knowledge Discovery and Data Mining, Springer, 2010, pp. 383–390.

[56] Z.Y. Ming, J. Ye, T.S. Chua, A dynamic reconstruction approach to topic summarization of user-generated-content, in: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, ACM, 2014, pp. 311–320.

[57] V. Ng, C. Cardie, Improving machine learning approaches to coreference resolution, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, pp. 104–111.

[58] C. Wang, K. Chakrabarti, T. Cheng, S. Chaudhuri, Targeted disambiguation of ad-hoc, homogeneous sets of named entities, in: Proceedings of the International Conference on World Wide Web, ACM, 2012, pp. 719–728.

[59] G. Weikum, M. Theobald, From information to knowledge: harvesting entities and relationships from web sources, in: SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, ACM, 2010, pp. 65–76.

[60] A. Rahman, V. Ng, Coreference resolution with world knowledge, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2011, pp. 814–824.

[61] L. Ratinov, D. Roth, D. Downey, M. Anderson, Local and global algorithms for disambiguation to Wikipedia, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2011, pp. 1375–1384.

[62] A. Sil, E. Cronin, P. Nie, Y. Yang, A.-M. Popescu, A. Yates, Linking named entities to any database, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, 2012, pp. 116–127.

[63] A. Sil, A. Yates, Re-ranking for joint named-entity recognition and linking, in: Proceedings of the ACM Conference on Information and Knowledge Management, ACM, 2013, pp. 2369–2374.

[64] W.M. Soon, H.T. Ng, D.C.Y. Lim, A machine learning approach to coreference resolution of noun phrases, Comput. Linguist. 27 (4) (2001) 521–544.

[65] H. Toba, Z.-Y. Ming, M. Adriani, T.-S. Chua, Discovering high quality answers in community question answering archives using a hierarchy of classifiers, Inform. Sci. 261 (2014) 101–115.

[66] V. Varma, V. Bharat, S. Kovelamudi, P. Bysani, G. Santosh, K. Kumar, N. Maganti, Iiit hyderabad at tac 2009, in: Proceedings of Text Analysis Conference, 2009.

[67] W. Zhang, Y.C. Sim, J. Su, C.L. Tan, Entity linking with effective acronym expansion, instance selection and topic modeling, in: International Joint Conferences on Artificial Intelligence, AAAI Press, 2011, pp. 1909–1914.

[68] W. Zhang, J. Su, C. Tan, W. Wang, Entity linking leveraging automatically generated annotation, in: Proceedings of the International Conference on Computational Linguistics, Beijing, China, 2010, pp. 1290–1298.

[69] Z. Zheng, F. Li, M. Huang, X. Zhu, Learning to link entities with knowledge base, in: The Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Los Angeles, California, 2010, pp. 483–491.

[70] X. Zhu, Z.-Y. Ming, X. Zhu, T.-S. Chua, Topic hierarchy construction for the organization of multi-source user generated contents, in: SIGIR, ACM, New York, NY, USA, 2013, pp. 233–242.

[71] J. Zobel, A. Moffat, Inverted files for text search engines, ACM Comput. Surv. 38 (2) (2006) 1–56.