Exploring Multiple Feature Spaces for Novel Entity Discovery

Zhaohui Wu[†*], **Yang Song**[⋆], **C. Lee Giles**[‡†]
[†]Computer Science and Engineering, [‡]Information Sciences and Technology
Pennsylvania State University, University Park, PA 16802, USA
[⋆]Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
zzw109@psu.edu, yangsong@microsoft.com, giles@ist.psu.edu

Abstract

Continuously discovering novel entities in news and Web data is important for Knowledge Base (KB) maintenance. One of the key challenges is to decide whether an entity mention refers to an in-KB or out-of-KB entity. We propose a principled approach that learns a novel entity classifier by modeling mention and entity representation into multiple feature spaces, including contextual, topical, lexical, neural embedding and query spaces. Different from most previous studies that address novel entity discovery as a submodule of entity linking systems, our model is more a generalized approach and can be applied as a pre-filtering step of novel entities for any entity linking systems. Experiments on three real-world datasets show that our method significantly outperforms existing methods on identifying novel entities.

Introduction

Comprehensive knowledge bases have been an elusive goal of AI for decades (Hoffart et al. 2013), where a key challenge is to discover new knowledge that emerges continually from various sources such as news streams, social feeds, academic publications, etc. It is reported that Wikipedia and its sister projects grow at a speed of more than 10 edits per second[1]; while the English version of Wikipedia increases by more than 800 articles per day[2]. However, manual approaches to discover new entities is for most cases not scalable and cannot guarantee the coverage or timeliness[3]. Thus, algorithms that automatically discover and ingest novel entities in to a KB are critical for its freshness and completeness. Typically, entity discovery and linking contains 3 major steps. The first step is entity recognition, which extracts the surface forms or mentions of entities from a data stream. Then, entity linking or disambiguation module decides which entity entry in the KB a mention refers to. Finally, if none of the existing entries matches the mention, it will be treated as a novel (or NIL, unlinkable, out-of-KB) entity.

While there exists plenty of related work on entity recognition and disambiguation (Shen, Li, and Doan 2005; Ji et al. 2014; Shen, Wang, and Han 2015), novel entity discovery has not been fully studied. Most entity linking (EL) systems either simply ignore the novel entities (Cucerzan 2007; Han, Sun, and Zhao 2011), or treat them as NIL entities without carefully modeling their representation. Existing work (Ratinov et al. 2011; Hoffart, Altun, and Weikum 2014) have shown that identifying NIL entities is challenging, especially for those entities with ambiguous names.

Most EL systems address NIL entities by predefining a threshold on the confidence score of linking a mention to the top ranked candidate entity. If the score is below the threshold, the mention will be marked as a NIL entity (Bunescu and Pasca 2006; Gottipati and Jiang 2011). However, this approach may not work for real-world applications because tuning a robust threshold for all possible novel entities is impractical, especially when NIL entity mentions are underrepresented due to the lack of informative context. Hoffart et al. (2014) addressed the problem by remodeling NIL entities in a KB and reestimating the threshold based on the output confidence score of their EL system. However, their keyphrase based context modeling approach may lack capability on capturing the semantics of novel entity mentions, as the mentions may have very few effective contextual words that overlap with the predefined keyphrase set, especially at the early emerging time of novel entities.

Other EL systems such as Ratinov et al. (2011) train a binary classifier to determine whether the top ranked candidate is a NIL or a correct matching entity, using the same features as the entity candidate ranking plus additional features that are indicative of NIL. The mentions for which the top-ranked candidate did not match the gold entity are treated as negative (NIL) examples, while the mentions that got correct matching serve as positive (in-KB) examples. However, this would result in incorrect negative labels on in-KB mentions (if their top-ranked candidate is incorrect) thus decrease the quality of the trained classifier. For example, if the entity candidate ranker mistakenly ranks the entity "Apple Corp." at the top for the mention "Apple" in a news about Apple Inc. while the gold entity "Apple Inc." is in the second or third position, then their system will incorrectly label the

[*] This work was primarily done when the first author was on an internship at Microsoft Research.

[1] http://tools.wmflabs.org/wmcounter/

[2] http://tools.wmflabs.org/wmcharts/wmchart0002.php

[3] Our estimation on 1381 tail entities show that the average latency of their Wikipedia article creation behind news appearance is 133 days.

mention “Apple” as a NIL example.

Previous methods focus mostly on addressing NIL entities as a submodule of entity linking, without crafting well-designed features or general techniques to model and identify novel entities in a principled way. A critical question to ask is: can a novel entity discovery module be designed to be seamlessly integrated into an entity discovery and linking system? If so, what techniques would be mostly effective to model the semantic representation of novel entities?

Here, we address the novel entity discovery issue as a binary classification problem with the emphasis on high accuracy. Our model leverages the top K labeled candidates for training to determine whether a mention refers to a novel or in-KB entity. Since novel entity mentions usually occur with limited contextual information, using features such as contextual words only may not be sufficient enough to characterize the novel entities. We thus explore multiple semantic spaces for modeling novel entities including contextual, neural embedding, topical, query, and lexical spaces. Our contributions can be summarized as follows:

- We study novel entity discovery by modeling entity representation in multiple spaces and analyze the strength and the weakness of each representation.
- We present a high precision novel entity classifier and demonstrate its effectiveness on three real-world datasets by comparing to other state-of-the-art methods.
- We further show that our model can be generalized as a preprocessing step to greatly improve the performance of EL systems.

Problem Definition and Approach

We formalize the novel entity discovery task as follows. Given a list of mentions $M = \{m_i | i = 1, ...|M|\}$ in a document d from a document stream DS and a knowledge base KB (Wikipedia), our goal is to discover all mentions in M that are novel entities, i.e., mentions that cannot be mapped to any existing entities in the KB.

The entity set of a KB is given as $E = \{e_1, ..., e_N\}$. Each entity e_i is encoded into a semantic space S and $k_S(e_i, e_j)$ is a kernel function that measures the semantic similarity between e_i and e_j in S. Suppose the gold entity that m_i refers to is e_{m_i}, then our goal is to learn a binary decision function $f(m_i) = \max_{e \in E} k_S(e_{m_i}, e)$ such that if $e_{m_i} \in E$ $f(m_i) = 1$; else $f(m_i) = 0$.

The key challenge then becomes how to define the semantic space S and the function k_S. In this work, we assume that S can be decomposed into multiple subspaces S_1 ,..., S_p where for each subspace S_i there is a kernel function k_i. For simplicity, we denote $e_{m^{top}} = \max_{e \in E} k_S(e_m, e)$. Thus we can rewrite k_S to be some function g:

$$k_S = g(k_1(e_m, e_{m^{top}}), ..., k_p(e_m, e_{m^{top}})) \qquad (1)$$

so that we can treat $k_i(e_m, e_{m^{top}})$ as an individual feature to learn the function g from the training data.

How can we correctly identify $e_{m^{top}}$ during the prediction? If e_m is novel, any entity being identified as $e_{m^{top}}$ should not affect the prediction. However, if $e_m \in E$, then mistakenly identify $e_{m^{top}}$ might result in a false positive prediction. To address this problem, we need a strong entity candidate ranker that can find the correct $e_{m^{top}}$. On the other hand, we also need our novel entity classifier to make correct decisions even when the ranker makes mistakes. Based on our empirical study, we found that the missing (correct) $e_{m^{top}}$ usually appears on top ranked positions. We thus use the top k ranked candidates of each mention for training. Our experiments shall indicate that this is a more effective approach than using only the top-ranked candidate. We will also discuss our strategies for entity candidate ranking and the selection of K.

Feature Spaces

We consider modeling entities in five different spaces, namely, contextual, neural embedding, topical, query, and lexical spaces. The intuition is that multiple spaces can improve both the representation of mentions and entities thus could give more accurate estimation of their semantic relatedness. The features thus do not need to be tailored to model NIL entities and can also be applied to general entity linking.

Contextual Space

Building a good entity representation using contextual information has shown as an effective way for entity linking (Cucerzan 2007) and there exists various context modeling methods based on bag-of-words, named entities, and Wikipedia elements (titles, anchor texts, and categories). We consider contextual information that could be more descriptive and informative. Specifically, we model the contextual space of an entity e or a mention m into three parts: supportive entities, salient entities, and dependent words.

Supportive Entities are entities used to define or describe e. We use all hyperlinked entities in the Wikipedia article of e. The weight, or the importance of a supportive entity of e, is calculated by using TFIDF measurement, where TF is the number of occurrences of the supportive entity in the article of e and DF is the number of Wikipedia articles containing (or linking to) the supportive entity.

Salient Entities are entities co-occurring with the entity in a context (defined as a sentence for salient entities and dependent words). They can be found from profiles of other entities in the KB. For Wikipedia, we use all hyperlinked entities appearing in the context of e. Similarly, we use TFIDF to measure the importance of a salient entity, where TF is the number of total co-occurrences of the salient entity and e and DF is the number of Wikipedia articles containing (or linking to) the salient entity.

Dependent Words are words appearing as dependency of the entity in a context. We use Stanford Dependencies[4] that represent the grammatical relations between words using triplets: name of relation, governor and dependent, generated by the Stanford Parser 3.4 (Socher et al. 2013). We count the words being the immediate governors or dependents of the entity. Again, TFIDF is used to measure the importance of the words to the entity.

For a supportive or salient entity e_t, $p(e_t|t) \cdot$ TFIDF(t) is used to measure the weight of e_t to m. t denotes a possible

[4] http://nlp.stanford.edu/software/stanford-dependencies.shtml

mention of e_t that appears in the context of m. TFIDF(t) measures the importance of t to the context of m. $p(e_t|t)$ measures the probability that t refers to e_t, estimated by the fraction of times that t links to e_t in Wikipedia. The dependent words of a mention are extracted in the same way from dependencies generated by Stanford Parser and weighted using TFIDF. When modeling m, we consider also including documents that contain m in temporal proximity, by keeping a small temporal window such as one or two days before or after the publication date of d. Finally, the cosine similarity between m and e in the contextual space is computed to represent their semantic relatedness.

Neural Embedding Space

While contextual space can serve as a strong representation of entities, it may fail in cases where contextual information of a mention is very limited, e.g., a mention with only one short sentence. This applies to many real world scenarios such as user input queries, product reviews, or tweets. Therefore, the challenge is how to effectively model mentions and entities that do not have enough contextual entities and dependent words to form their contextual representation.

Previous work has shown the promise of using neural network word/entity representation for entity linking (He et al. 2013; Sun et al. 2015). Therefore, we precompute semantic embedding for all words in the vocabulary and all entities in a KB, so that we can compute the semantic relatedness of the mention m given its contextual words between the entity e. We leverage the pre-trained word, phrase and entity embedding based on the word2vec model (Mikolov et al. 2013). There are 3 million words and phrases vectors and 1.4 million freebase entity vectors trained on Google News dataset containing about 100 billion words. Specifically, let vec(x) denote the vector of x. Given a mention m with its contextual space $c = [w_1, ..., w_{|c|}]$ and an entity e, we consider the following similarity features in the embedding space : 1) cosine similarity between vec(m) and vec(e), and 2) cosine similarity between vec(c) + vec(m) and vec(e), where vec$(c) = \sum$ vec(w_i).

Topical Space

While the contextual and embedding spaces provide explicit and latent semantic representation respectively, they do not consider the global topical coherence between mentions and entities. An intuitive assumption is that an entity would tend to occur in documents of a particular topical distribution. For example, Swift as a person would be more likely to appear in entertainment news while as a programming language would be more likely to appear in technology news.

To model the topical space, we choose to use the Open Directory Project's (ODP) categories due to its broad, general purpose topic coverage and the availability of high-quality labeled data (Collins-Thompson and Bennett 2010).[5] The ODP hierarchical classifier we used was originally trained on 1.2M ODP Web documents from 62,767 categories based on the Refined Experts model (Bennett and Nguyen 2009). Using a dump of ODP from early 2008, we identified the topic categories (some categories like regional are not topical and were discarded) that had at least 1K documents, which results in a total of 219 categories. We then use these 219 categories as the topical space. For any given document, the classifier outputs a 219 dimension vector that represents its category distribution. In our experiments, we choose the top-5 predicted categories for each document since we observed that the probability scores of categories after top-5 results are very small(≤ 0.1).

The similarity feature is then defined as the cosine similarity between the two category vectors of the mention m's document d and the Wikipedia article of candidate entity e.

Query Space

Another possible feature space that has not been well explored for novel entity classification is the search engine's user query history. Two research questions we ask here are: how helpful are the contextual words of entities in user queries? And does there exist any temporal patterns for novel entities after they first emerge in user queries?

To answer the questions, we randomly sampled 100 entities in our WebNews dataset that appeared in Wikipedia after March 1st 2014. We then mined all the user queries that contain these entity words from the query logs of a commercial search engine in the first half year of 2014. For each entity, we plot its daily query counts and rank them by the average query count per day to indicate their popularity. We found that the top popular emerging entities (around 10%) have significant temporal spikiness before their creation date in Wikipedia, as shown by examples of "true detective" and "htc one" in Figure 1 while most of the tail entities have fewer occurrences in user queries.

Besides, we found that the contextual words appearing in queries that contain the entities are indeed very informative for disambiguation (Blanco, Ottaviano, and Meij 2015). For example, in the query log of Feb. 2014, we found that the top contextual words for entity "true detective" are {"hbo", "episode", "alexandra daddario"}, and those for "bridgegate" being {"Chris Christie", "allegation", "scandal"}. These results show the promise of contextual words inside queries as well as the spikiness of queries that contain novel entities.

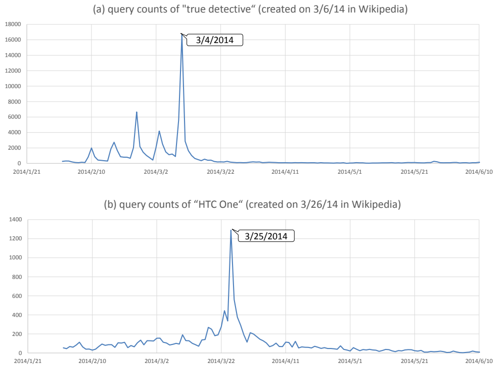
For each in-KB entity e, we maintain a contextual word vector and a query count sequence. For a mention m that appears at time t, its contextual word vector and query count sequence are built based on the queries at or around t. The cosine similarity between the two contextual word vectors is then computed as one feature to measure relatedness between m and e in query space. The short time series distance (Möller-Levet et al. 2003) between the two query count sequences are calculated as another query feature[6].

Lexical Space

We also incorporate the lexical features of entities names. We apply the normalized Levenshtein distance (Navarro

[5] While topic modeling techniques such as LDA (Blei, Ng, and Jordan 2003) can be an effective way to construct the latent topical space, empirically we found ODP gives more discriminative topics for Wikipedia articles and news documents.

[6] We uniformly sample the same number of data points from the two sequences


(a) query counts of "true detective" (created on 3/6/14 in Wikipedia)

(b) query counts of "HTC One" (created on 3/26/14 in Wikipedia)

Figure 1: Query sequences of entities from query history

2001) between a mention m and an entity name e, which is defined as:

$$nld(m,e) = \frac{levenshtein(m,e)}{\max\{len(m), len(e)\}} \quad (2)$$

Data Sets and Evaluation Methodology

We use three datasets for our evaluation: 1) AIDA-EE dataset from Hoffart et al. (2014), 2) WebNews dataset from a Web news portal, and 3) Wikievents dataset from Wikipedia event news.[7] All three datasets use Wikipedia as the knowledge base for evaluation.

AIDA-EE dataset contains 300 documents with 9,976 entity names linked to Wikipedia dump of 2010-08-17. There are 150 documents from 2010-10-01 in the training dataset and 150 documents from 2010-11-01 in the test dataset, with 187 and 162 out-of-KB entity mentions in the training and test dataset respectively.

WebNews dataset was constructed using news stream from a news portal in order to perform a large-scale evaluation for novel entity discovery. We first collected sampled news articles in Science, Technology and Business from September 2013 to July 2014, resulting in a total of 2,240,695 articles. Meanwhile, we queried all Wikipedia titles for their article creation time based on the Wikipedia API[8] and collected all the titles created between Jan. 1 2014 and Jun. 1 2014. After removing 712 redirected titles (could link to articles created before 2014), we got 311,628 titles in total. For each title, we then searched for news articles containing the title by performing exact string matching of the title on news article content with three additional constraints: 1) the news articles must be published near the creation date of the title in Wikipedia ($<$ 15 days before and $<$ 5 days later), which aims at increasing the chance that the mentions of a Wikipedia title in the news articles truly refer to the Wikipedia title; 2) the creation date of the Wikipedia article is after the pivot date D_0, which guarantees that the mention refers to a novel entity; 3) The Wikipedia (dump on D_0) API returns more than 10 candidates for querying the title, which ensures the ambiguity of the mention.

By choosing D_0 to be Dec. 31 2013, it resulted in 893 novel entities with 7451 news articles from Jan. 1 2014 to May 29 2014. We then identified other entity mentions using Cucerzan's EL system that is built using Wikipedia dump of Oct. 7 2013 (Cucerzan 2014; 2012), which generates entity mentions and their linked entities with the confidence score in [0,1]. We chose those results with confidence score greater than 0.8 as in-KB entities. Finally a manual verification was made by two annotators individually with an agreement of 91% on novel entity labels. The total number of entity mentions is 168,290 while 18,924 of them are novel entity mentions. 781 novel entities in [1/1/14, 4/30/14] with their mentions and articles were used for training while the other 112 novel entities in [5/1/14, 5/29/14] with their mentions and articles were used for testing.

Table 1: Statistics of datasets

Statistics	AIDA-EE	WebNews	WikiEvents
documents	300	7451	5946
mentions	9,976	168,290	15,773
+ mentions	561	18,924	670
words/doc	538	550	35
mentions/doc	33	23	3

WikiEvents dataset was crawled from Wikipedia current events[9] to evaluate the performance of our model on short texts. Each document contains a short description on a news event, where the anchor texts are used as entity mentions linked to Wikipedia articles. We crawled all the news event descriptions from Jan. 1 2013 to Jan. 31 2015, in total of 9004 documents with 18,959 anchor links. Note that some background texts (not entity mentions) are also linked. For example, in "A gunman kills eight people in a house-to-house rampage in Kawit, the Philippines", "kills eight people" is linked to "Kawit_shooting". We filtered those uncapitalized anchor texts and kept the documents with at least one mention-entity pair. This gave us 15,773 mention-entity pairs in 5946 documents. We use 4162 documents from Jan. 1 2013 to Nov. 13 2013 for training and the other 1784 from Nov. 14 2013 to Jan. 31 2015 for testing. In our experiments, we set the pivot date as Jan. 1 2010, which gives us 259 novel entities (414 mentions) in the training set and 142 novel entities (256 mentions) for testing.

To well handle the heterogeneous similarity features, we use the gradient boosting tree (Friedman 2001) to learn classifiers from the training datasets by joining the K-best candidates to form the feature space, setting NumTrees=100, NumLeaves=20, MinInstancesInLeaf=10, and LearningRate=0.2. For AIDA-EE dataset, to compare with Hoffart et al. (2014), we report the average precision, recall and F1 over all documents on the test dataset. For the other two datasets, documents may have very few novel entity mentions (some in WikiEvents have none), we report the micro precision, recall and F1 on the test datasets. Besides, following Ratinov et al. (2011), we use *ranking accuracy* (the fraction of in-KB entity mentions that have its correct reference in the top K candidates) to measure the performance of different methods on finding the correct top ranked candidate; and use *linking accuracy* (the fraction of all mentions that have been correctly linked to in-KB entities or identified as novel entities) to show performance gain the novel entity classifier brings for an EL system.

[7] http://www.cse.psu.edu/ zzw109/data.html

[8] http://en.wikipedia.org/w/api.php?action=query&prop=revisions&rvprop=timestamp|| ids&rvlimit=1&rvdir=newer&format=xml&titles=

[9] http://en.wikipedia.org/wiki/Portal:Current_events

Experimental Results

We first compare our model for novel entity identification with two state-of-the-art baselines.

D2W refers to the method by Ratinov et al. (2011) that trains a novel entity classifier using a linear SVM with local context, global Wikipedia coherence, and additional linker features. We exclude the global features since their results showed that those features are not consistently helpful for novel entity classification.

EE represents the method by Hoffart et al. (2014) that remodels novel entities using keyphrase vectors. We reimplement it on WebNews using Wikipedia dump of Dec. 31 2013 and on WikiEvents datasets using Wikipedia dump of Jan. 1 2010.

Table 2 shows that our model clearly outperforms the two baselines in precision and F1 on all the three datasets.[10] Specifically, our method achieves a high precision (85+%) with a recall of 70+%, which improves the precision by 44+% and F1 by 12+% over D2W, due to a stronger representation of novel entities. For example, only our model can correctly identify "Bill Oates"[11] as a novel entity, while the other two mis-linked it to in-KB entity "William Oates". Despite that the D2W model shows the best recall among all methods, it has very poor precision comparing to our model and EE model. By doing deeper analysis, we discovered that D2W cannot model entities with limited context well enough, the model is therefore more likely to mis-classify those in-KB entities mentions to be novel entities. Hence, we see high recall but low precision from that model.

We evaluate the effectiveness of feature spaces via feature ablation. Figure 2 shows the precision, recall, and F1 scores of the novel entity classifiers that exclude each feature space on WebNews dataset[12]. Overall, we see significant performance decrease when a feature space gets ablated, showing the usefulness of different feature spaces. Comparatively, the contextual and topical features show higher importance than lexical, neural embedding, and query spaces.

While our main focus is not entity linking in this paper, our method naturally includes a step that can perform entity linking for in-KB entities. Hence, we also evaluate different strategies for ranking entity candidates. Table 3 compares the ranking performance of features on different spaces. $p(e_t|t)$ is the fraction of times the entity e_t is the target link for the anchor text t, which has shown as a strong baseline for entity ranking in Ratinov et al. (2011). "Context", "Embedding", "Topical", and "Lexical" represents ranking methods based on the corresponding similarity features. The "Average" and "Maximum" use the average and maximum scores over the five features listed in Table 3 respectively. Here the ranking accuracy is calculated based on the top $K = 3$ candidates. The similarity feature in contextual space has incorporated the $p(e_t|t)$ when computing weight and thus outperforms $p(e_t|t)$. In WikiEvents dataset, the "context" does not perform the best probably because the short texts affect the quality of context modeling. Usually, training a ranker using all the features could achieve the best accuracy. However, since our goal is not to achieve the best possible ranking performance on each dataset, but to build a general and efficient novel entity classifier, we use only the context based similarity to select the top K entity candidates for further novel entity classification. The results of choosing different K for training novel entity classifier are shown in Figure 3. The other two datasets have similar results.

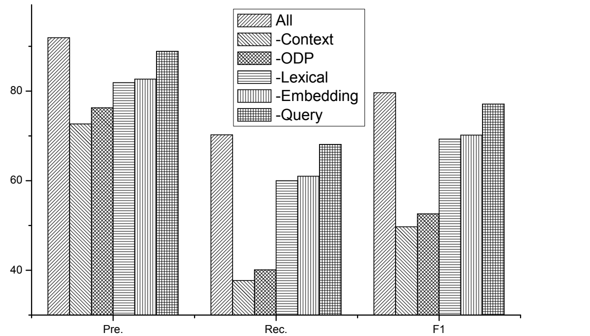


Figure 2: Study on the importance of different feature spaces. "-Context" refers to the feature space by removing the contextual space; so do others.

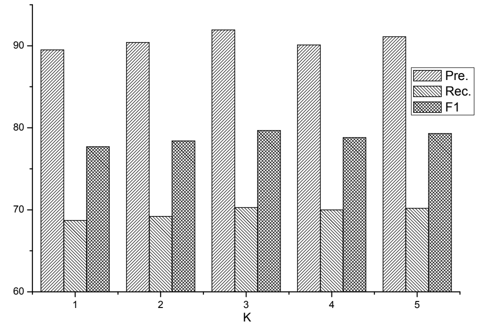


Figure 3: Study on the impact of K on WebNews dataset. $K = 3$ achieves the best performance and the results are not sensitive to the choice of K.

Finally, we evaluate how much improvement can be achieved by incorporating our classifier to EL systems. Therefore, we use our novel entity classifier as a preprocessing step to filter out the identified novel entities and then run a subsequent entity linking using various systems. "$p(e_t|t)$" and "Context" are the same methods we used for entity candidate ranking in Table 3. We treat them as methods for entity linking and thus only look at the top 1 candidate. "Cucerzan's" refers to the EL system (Cucerzan 2014) we used to construct the WebNews dataset. "NEC+X" indicates using our novel entity classifier as a preprocessing step for X. As shown in Table 4, our model can consistently give additional improvement over various EL systems in terms of linking accuracy.

Related Work

There is a rich literature on entity linking/disambiguation, which usually consists of two main modules: candidate entity ranking and unlinkable/NIL mention prediction (Shen, Wang, and Han 2015). Many existing work such as (Mihalcea and Csomai 2007; Cucerzan 2007; Hoffart et al. 2011;

[10] For the AIDA-EE dataset, the reason that the average F1 being lower than both the average precision and recall is that some documents may have precision or recall being 0.

[11] Google blog Bostons chief information officer.

[12] The trends are similar in the other two datasets

Table 2: Performance comparison on novel entity identification. Bold values indicate the best performance in each dataset. * and † indicate the improvements over D2W and EE, respectively, are statistically significant ($p<0.05$) using Student's t-test.

	AIDA-EE			**WebNews**			**WikiEvents**		
Methods	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
D2W	66.59	**90.88**	63.89	61.35	**80.53**	69.64	59.17	**79.26**	67.76
EE	97.97	70.69	68.92	83.34	58.71	68.89	79.25	54.62	64.67
Novel Entity Model	**98.31***†	73.27†	**71.92***†	**91.94***†	70.27†	**79.66***†	**85.63***†	70.57†	**77.37***†
↑ over D2W	47.6%	-19.4%	12.6%	49.9%	-12.7%	14.4%	44.7%	-11.0%	14.2%
↑ over EE	0.35%	3.6%	4.4%	10.3%	19.7%	15.6%	8.1%	29.2%	19.6%

Table 3: Ranking accuracy of different entity candidate ranking strategies.

Features	**AIDA-EE**	**WebNews**	**WikiEvents**
$p(e_t\|t)$	92.05	94.50	95.80
Context	**93.75**	**95.68**	95.63
Embedding	81.78	83.24	85.86
Topical	86.70	92.93	86.73
Lexical	90.50	91.26	94.35
Average	93.43	94.23	**96.13**
Maximum	92.09	92.18	93.76

Table 4: Linking accuracy. Adding our novel entity classifier as a preprocessing step consistently improves the accuracy of different entity linking systems on all three datasets. * indicates the improvements are statistically significant ($p<0.05$) using Student's t-test.

System	**AIDA-EE**	**WebNews**	**WikiEvents**
$p(e_t\|t)$	73.05	74.06	75.06
NEC+$p(e_t\|t)$	**74.10***	**77.33***	**75.98***
↑ over $p(e_t\|t)$	1.44%	4.42%	1.23%
Context	76.27	78.38	74.79
NEC+Context	**77.20***	**81.17***	**75.93***
↑ over Context	1.22%	3.56%	1.52%
Cucerzan's	76.58	78.85	79.36
NEC+Cucerz.	**77.82***	**81.93***	**80.75***
↑ over Cucerz.	1.62%	3.91%	1.75%

Han, Sun, and Zhao 2011; Chisholm and Hachey 2015; Blanco, Ottaviano, and Meij 2015) simply assume all mentions could be linked to in-KB entities thus ignore the unlinkable problem. Related work that handle NIL entities can be divided into several groups.

Some systems from TAC2010 (Ji et al. 2010) use a simple rule: if the candidate entity set of a mention is empty, then the mention is NIL. Clearly, this approach has an extremely low recall on NIL entity prediction because most NIL entity mentions (especially those with ambiguous names) will have entity candidates.

Many approaches use NIL thresholding, which predict a mention as NIL if the confidence score of its top ranked entity is below a predefined NIL threshold (Bunescu and Pasca 2006; Kulkarni et al. 2009; Ferragina and Scaiella 2010; Gottipati and Jiang 2011; Shen et al. 2012; Li et al. 2013). However, tuning a robust NIL threshold is hard because it is usually data and model dependent.

Instead of finding a threshold, a number of systems directly train a binary classifier (mostly using SVM) on $(m, e_{m^{top}})$ pairs to predict if the top ranked candidate is a correct reference or a NIL entity (Zheng et al. 2010; Ji et al. 2010; Ratinov et al. 2011; Zhang et al. 2011). A positive/negative example is a mention whose $e_{m^{top}}$ matches/does not match the gold entity. However, a critical problem is that an incorrect top ranked candidate does not necessarily indicate a NIL, because the candidate entity ranking may make some mistakes. Those false examples will decrease the quality of the classifier. Our model uses top K candidates for training and relies on labels that truly reflect if a mention refers to a novel or in-KB entity.

Others incorporate the NIL prediction into the candidate entity ranking by adding a NIL entry into the candidate set or KB (Dredze et al. 2010; Han and Sun 2011; Rao, McNamee, and Dredze 2013; Hoffart, Altun, and Weikum 2014). However, more efforts are needed in modeling the NIL entry in the same feature space of other in-KB entities.

Most previous work do not consider using NIL entity predictor as a preprocessing step for entity linking systems, with an exception of Hoffart et al. (2014). However, their keyphrase based method might not be able to capture novel tail entities without distinguishable contextual keyphrases. Comparing to them, our model uses more comprehensive features and achieves better performance on both identifying novel entities and improving entity linking accuracy.

We also note that there are other works addressing novel entity discovery by detecting new names and assigning fine-grained semantic types (Ling and Weld 2012; Lin, Mausam, and Etzioni 2012; Nakashole, Tylenda, and Weikum 2013).

Conclusion and Future Work

We empirically studied novel entity discovery by modeling mentions and entities in multiple feature spaces, including context, neural embedding, topical, query, and lexical spaces; and demonstrated that they are effective, although of different contributions, for novel entity classification and EL. This approach differs from existing work that address NIL mention prediction as a submodule of EL by developing a general novel entity classifier that can be applied to novel entity filter for general EL systems. We showed its effectiveness on different types of documents including regular news articles and short text documents.

The entity type features were not explored since a more accurate and fine-grained novel entity typing tool would be a challenging future work. Another direction is to further improve the recall of the model by disambiguating novel entities that have certain relatedness to some in-KB entities. Typical cases include new entities that derive from old ones, such as "Surface Pro 3" v.s. "Microsoft Surface", or "Microsoft Cortana" v.s. "Cortana (Halo)". Disambiguating those entities would need more effort on understanding of their fine-grained semantic types and relations.

Acknowledgments

We acknowledge helpful discussions with Ming-Wei Chang, Hao Ma, Kuansan Wang and useful comments from referees.

References

Bennett, P. N., and Nguyen, N. 2009. Refined experts: improving classification in large taxonomies. In *Proceedings of SIGIR*, 11–18.

Blanco, R.; Ottaviano, G.; and Meij, E. 2015. Fast and space-efficient entity linking for queries. In *Proceedings of WSDM*, 179–188.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *JMLR* 3:993–1022.

Bunescu, R. C., and Pasca, M. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*, volume 6, 9–16.

Chisholm, A., and Hachey, B. 2015. Entity disambiguation with web links. *TACL* 3:145–156.

Collins-Thompson, K., and Bennett, P. N. 2010. Predicting query performance via classification. In *Advances in Information Retrieval*. 140–152.

Cucerzan, S. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL*, volume 7, 708–716.

Cucerzan, S. 2012. The msr system for entity linking at tac 2012. In *Proceedings of TAC2012*.

Cucerzan, S. 2014. Name entities made obvious: the participation in the erd 2014 evaluation. In *Proceedings of ERD2014*, 95–100.

Dredze, M.; McNamee, P.; Rao, D.; Gerber, A.; and Finin, T. 2010. Entity disambiguation for knowledge base population. In *Proceedings of COLING*, 277–285.

Ferragina, P., and Scaiella, U. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of CIKM*, 1625–1628.

Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* 1189–1232.

Gottipati, S., and Jiang, J. 2011. Linking entities to a knowledge base with query expansion. In *Proceedings of EMNLP*, 804–813.

Han, X., and Sun, L. 2011. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of ACL*, 945–954.

Han, X.; Sun, L.; and Zhao, J. 2011. Collective entity linking in web text: a graph-based method. In *Proceedings of SIGIR*, 765–774.

He, Z.; Liu, S.; Li, M.; Zhou, M.; Zhang, L.; and Wang, H. 2013. Learning entity representation for entity disambiguation. In *Proceedings of ACL*, 30–34.

Hoffart, J.; Altun, Y.; and Weikum, G. 2014. Discovering emerging entities with ambiguous names. In *Proceedings of WWW*, 385–396.

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

Hoffart, J.; Suchanek, F. M.; Berberich, K.; and Weikum, G. 2013. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. In *Proceedings of IJCAI*, 3161–3165.

Ji, H.; Grishman, R.; Dang, H. T.; Griffitt, K.; and Ellis, J. 2010. Overview of the tac 2010 knowledge base population track. In *Proceedings of TAC2010*.

Ji, H.; Dang, H.; Nothman, J.; and Hachey, B. 2014. Overview of tac-kbp2014 entity discovery and linking tasks. In *Proceedings of TAC2014*.

Kulkarni, S.; Singh, A.; Ramakrishnan, G.; and Chakrabarti, S. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of KDD*, 457–466.

Li, Y.; Wang, C.; Han, F.; Han, J.; Roth, D.; and Yan, X. 2013. Mining evidences for named entity disambiguation. In *Proceedings of KDD*, 1070–1078.

Lin, T.; Mausam; and Etzioni, O. 2012. No noun phrase left behind: Detecting and typing unlinkable entities. In *Proceedings of EMNLP-CoNLL*, 893–903.

Ling, X., and Weld, D. S. 2012. Fine-grained entity recognition. In *Proceedings of AAAI*, 94–100.

Mihalcea, R., and Csomai, A. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of CIKM*, 233–242.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, 3111–3119.

Möller-Levet, C. S.; Klawonn, F.; Cho, K.-H.; and Wolkenhauer, O. 2003. Fuzzy clustering of short time-series and unevenly distributed sampling points. In *Advances in Intelligent Data Analysis V*. 330–340.

Nakashole, N.; Tylenda, T.; and Weikum, G. 2013. Fine-grained semantic typing of emerging entities. In *Proceedings of ACL*, 1488–1497.

Navarro, G. 2001. A guided tour to approximate string matching. *ACM computing surveys* 33(1):31–88.

Rao, D.; McNamee, P.; and Dredze, M. 2013. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, Multilingual Information Extraction and Summarization*. 93–115.

Ratinov, L.; Roth, D.; Downey, D.; and Anderson, M. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of ACL*, 1375–1384.

Shen, W.; Wang, J.; Luo, P.; and Wang, M. 2012. Linden: linking named entities with knowledge base via semantic knowledge. In *Proceedings of WWW*, 449–458.

Shen, W.; Li, X.; and Doan, A. 2005. Constraint-based entity matching. In *Proceedings of AAAI*, 862–867.

Shen, W.; Wang, J.; and Han, J. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *TKDE* 27(2):443–460.

Socher, R.; Bauer, J.; Manning, C. D.; and Ng, A. Y. 2013. Parsing with compositional vector grammars. In *Proceedings of the ACL*, 455–465.

Sun, Y.; Lin, L.; Tang, D.; Yang, N.; Ji, Z.; and Wang, X. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *Proceedings of IJCAI*, 1333–1339.

Zhang, W.; Sim, Y. C.; Su, J.; and Tan, C. L. 2011. Entity linking with effective acronym expansion, instance selection, and topic modeling. In *Proceedings of IJCAI*, volume 2011, 1909–1914.

Zheng, Z.; Li, F.; Huang, M.; and Zhu, X. 2010. Learning to link entities with knowledge base. In *Proceedings of ACL*, 483–491.