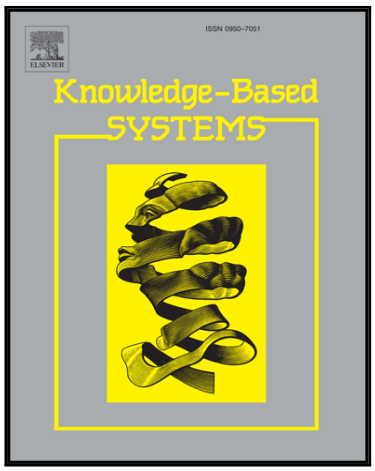
Accepted Manuscript



Named entity disambiguation for questions in community question answering

Fang Wang, Wei Wu, Zhoujun Li, Ming Zhou

PII: S0950-7051(17)30144-2
DOI: 10.1016/j.knosys.2017.03.017
Reference: KNOSYS 3866

To appear in: *Knowledge-Based Systems*

Received date: 28 July 2016
Revised date: 22 March 2017
Accepted date: 23 March 2017

Please cite this article as: Fang Wang, Wei Wu, Zhoujun Li, Ming Zhou, Named entity disambiguation for questions in community question answering, *Knowledge-Based Systems* (2017), doi: 10.1016/j.knosys.2017.03.017

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Named Entity Disambiguation for Questions in Community Question Answering☆

Fang Wang[a], Wei Wu[b], Zhoujun Li[a,*], Ming Zhou[b]

[a]*State Key Laboratory of Software Development Environment, Beihang University, China*
[b]*Microsoft Research, Beijing, China*

Abstract

Named entity disambiguation (NED) refers to the task of mapping entity mentions in running texts to the correct entries in a specific knowledge base (e.g., Wikipedia). Although there has been a lot of work on NED for long and formal texts like Wikipedia and news, the task is not well studied for questions in community question answering (CQA). The challenges of the task include little context for mentions in questions, lack of ground truth for learning, and language gaps between CQA and knowledge bases. To overcome these problems, we propose a topic modelling approach to NED for questions. Our model performs learning in an unsupervised manner, but can take advantage of weak supervision signals estimated from the metadata of CQA and knowledge bases. The signals can enrich the context of mentions in questions, and bridge the language gaps between CQA and knowledge bases. Besides these advantages, our model simulates people's behavior in CQA and thus is intuitively interpretable. We conduct experiments on both Chinese and English CQA data. The experimental results show that our method can significantly outperform state-of-the-art methods when we apply them to questions in CQA.

Keywords: named entity disambiguation, topic model, unsupervised learning, community question answering

☆This work was done when the first author was an intern in Microsoft Research Asia.

*Corresponding author

Email addresses: fangwang@buaa.edu.cn (Fang Wang), wuwei@microsoft.com (Wei Wu), lizj@buaa.edu.cn (Zhoujun Li), mingzhou@microsoft.com (Ming Zhou)

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

1. Introduction

Named entity disambiguation (NED) is a key step in connecting the structured world and the unstructured world, wherein we map entity mentions appearing in running texts to the correct referent entities in a specific knowledge base (e.g., Wikipedia). In practice, entity mentions in plain texts are often ambiguous, which brings big challenges for us to link the data in the two world: one mention may refer to many different entities in a knowledge base. For example, "Michael Jordan" may refer to the legendary American basketball player, or a famous researcher in machine learning and artificial intelligence. Conversely, one entity in a knowledge base can be expressed in many different ways in plain texts. For example, we have observed "Obama", "President Obama", and "B. H. Obama" in plain texts, and all of them refer to "Barack Obama", the US president in Wikipedia. Entity identification and disambiguation are challenging but important and even prerequisite for many tasks such as knowledge mining, text classification, search, and question-answering.

In this paper, we consider the problem of NED for questions in community question answering (CQA). In recent years, CQA portals like Yahoo! Answers have become hot platforms for people to share their knowledge. The knowledge in CQA is hidden in noisy running texts of question-answer pairs, including facts, common opinions, subjective cognition, and personal experiences etc. A crucial step in CQA knowledge mining is named entity disambiguation (NED) for questions, as questions are the keys to accessing the knowledge in CQA and entities are key elements in questions. In a randomly sampled 1 million questions from Yahoo! Answers, we have observed that 55.2% questions contain entries in Wikipedia, and 25.1% of them are ambiguous. NED for questions can facilitate many tasks in CQA such as question retrieval [1], question recommendation [2], and question routing [3]. As Khalid et al. [4] found, even a simple NED method could lead to improvements of precision on information retrieval for question answering.

Although there has been a lot of work on NED for long and formal texts like Wikipedia and news [5, 6, 7, 8, 9, 10], the task for questions in CQA has not been well studied. The challenges of NED for questions lie in three facts. First, questions

ACCEPTED MANUSCRIPT

are short (9.64 words per question on average), and therefore there is limited contextual words alongside entity mentions in questions. This makes existing work [7, 6] that relies on local context around the mentions fail on this task. Second, there is no ground truth for NED in CQA. It is not easy for us to build a large scale training set and conduct supervised learning like some existing work does [5, 11, 8]. Finally, the language of CQA is quite different from the language of knowledge bases like Wikipedia. When talking about the same entity, people in CQA and people in knowledge bases often care about different aspects of the entity. For example, regarding to the apple company, Wikipedia reports the company history, products, and corporation affairs, etc., while in CQA, people have more interests in prices of apple products and available softwares, which are more related to their lives. Moreover, in CQA both askers and answerers are used to using casual and informal words. Due to the language difference, existing work [9] that relies on word-entity associations estimated from knowledge bases does not work well on NED for questions in CQA , as will be seen in our experiments.

To tackle the above problems, we propose a topic modelling approach to NED for questions. To the best of our knowledge, this is the first work on NED for questions in CQA. Our model naturally incorporates priori knowledge learned from the metadata of CQA and Wikipedia into learning, and performs learning in an unsupervised manner. It simulates the way people generate questions and answers in CQA and formulates the relationship of topics, entities, and words in a unified framework. Specifically, we first estimate prior distributions of topics of question-answer pairs, prior entity distributions under topics, and prior word distributions under entities using the metadata of Wikipedia and CQA, and then incorporate these distributions into our topic model as hyperparameters of Dirichlet priors. We can prove that the prior distributions can weakly supervise the learning of the model by acting as the expectations of the distributions we aim to learn. By this means, the priori knowledge can enrich the context of mentions and bridge the language gaps between CQA and Wikipedia.

The novelty of our model lies in its generative process which describes how real world CQA data is created by people, and the way it leverages extra information in both CQA and Wikipedia to enhance learning. Besides, the learning process of our model does not need any human annotations. It can sufficiently leverage the power of

ACCEPTED MANUSCRIPT

big data in both Wikipedia and CQA to disambiguate entity mentions in questions. We conduct experiments on data crawled from Yahoo! Answers and Baidu Knows, and compare different models on large, human annotated English and Chinese question sets published at `https://github.com/NEDstudy/NEDforCQA`. The experimental results show that our model can significantly outperform state-of-the-art NED methods when we apply them to questions in CQA.

Our contributions in this paper are three folds: 1) proposal of the problem of named entity disambiguation for questions in CQA; 2) proposal of a novel weakly supervised topic model for NED of questions in CQA and theoretical analysis of the model; 3) empirical verification of the effectiveness of the model on large evaluation data sets.

The rest of the paper is organized as follows: Section 2 summarizes related work. In Section 3, we elaborate our method, including problem formalization, our model, and the learning algorithm. Section 4 reports experimental results, and finally Section 5 concludes the whole paper.

2. Related Work

Named entity disambiguation (NED) has been studied for many years. Early attempts use local and surface level information. Bunescu and Pasca [5] first defined a similarity measure to compute the cosine similarity between the text around the entity mention and the referent entity candidate's Wikipedia page. Later research [11, 12, 8, 13, 14, 15] extends this line by exploring richer feature sets, such as coherence features between entities. These methods often involve optimizing an objective function that contains both local and global terms, and thus require training on an annotated dataset. By modeling coherence using the page rank algorithm, Alhelbawy et al. [10] performed collective NED that did not require supervision. Pershina et al. [16] improved their method by using a personalized page rank algorithm. Most recent methods [17, 18, 19, 20] explore deep learning techniques to learn the representations of entities for NED. For example, He et al. [17] used deep neural networks to compute representations of entities and contexts of mentions from the knowledge base. However, most existing work focuses on long and formal texts such as news and Wikipedia,

ACCEPTED MANUSCRIPT

while little work exist on NED methods for questions in CQA to date.

Recently, NED for short texts [21, 22, 23] is attracting more and more attention. For example, Meji et al. [21] created a bunch of features using texts of tweets, entity knowledge in Wikipedia, and metadata in twitter such as hashtags and URLs, and trained several classifiers such as SVM [24, 25] and Naive Bayes with these features. Blanco et al. [23] proposed a probabilistic model for NED on search queries, which leveraged user-generated information on the web to link search queries to entities in a knowledge base. Similar to NED for tweets and search queries, in NED for questions we also consider leveraging extra information such as metadata in CQA to deal with the problems due to the shortness of questions. However, the metadata in CQA is quite different from the metadata in twitter and search engines. It is difficult for us to directly apply existing models to NED for questions. Therefore, we propose a new model which can naturally represent the generative process of CQA data and leverage the unique metadata in CQA for NED.

Our proposed model is inherited from the Latent Dirichlet Allocation model. It extends the traditional topic models [26, 27] by replacing the heuristically designed hyperparameters with prior distributions estimated from the metadata of CQA and Wikipedia. Before us, some researchers have already considered using topic models for NED [28, 9, 29, 30]. The key idea is building topic-entity associations. Based on this idea, different methods incorporate different types of information into the generative process. For examples, Han et al. [28] proposed a generative model to include information from entity popularity, mention-entity association and context similarity in a holistic way. By leveraging the cross-document hyperlinks in Wikipedia, Sen [9] adopted a latent topic model to learn the context-entity association to help disambiguation. To bridge the gap between the keywords in a query and the reference knowledge base, Li et al. [29] proposed a generative model to mine useful evidences from external corpus. Li et al. [30] also proposed a generative model to tackle the NED with knowledge bases without human labeled hyperlinks. We estimate prior entity-word distributions from Wikipedia using the model in [9] and bias the learning of entity-word distributions in CQA. Our model is unique in that it formulates users' behavior in CQA and incorporates the metadata in CQA into learning.

ACCEPTED MANUSCRIPT

Table 1: Important notations used in describing our model.

Variable	Description
θ	The topic distribution $\theta \sim Dirichlet(\alpha)$.
z	A topic in CQA data.
e	An entity in the knowledge base.
d	A concatenation of the question, its description, and its answer document.
w	A word in a document d, either a contextual word or a mention.
Λ	A $M \times K$ matrix, representing the entity-topic distributions.
Φ	An $N \times M$ matrix, representing the word-entity distributions.
A^0	The prior document-topic (question-category) distributions.
Λ^0	The prior entity-topic (entity-category) distributions.
B^0	The prior word-entity distributions.
T	The total number of documents in the corpus.
K	The total number of topics in CQA Data.
M	The total number of entities in the knowledge base.
D	The collection of all documents.
E	The union set of all the candidate entities.
N	The total number of words in D.

3. Our Approach

To disambiguate entity mentions in questions, our method has to deal with three problems: 1) how to leverage other signals to enrich context of mentions; 2) how to perform mention-entity matching without human annotations; 3) how to bridge the language gaps between CQA and Wikipedia. Our idea is that we exploit a generative model and take priori knowledge estimated from the metadata of CQA and Wikipedia as priors of the generative process. Without human annotations, the priori knowledge can weakly supervise the learning of the model, and connect CQA and Wikipedia. The advantages of our method include its interpretability, its flexibility on leveraging different types of extra signals, and its theoretical soundness on performing learning.

We first formalize the problem of NED for questions, then we describe details of

ACCEPTED MANUSCRIPT

Table 2: an example from Yahoo! Answers

Question	What **font** does **Apple** use?
Description	I was wondering what font Apple use for their website, engravings, software - both for headlines and body. Is it Arial?
Categories	Computers & Internet, Programming & Design
Answers	Adobe Myriad is used for all their Headlines, slogans & body copy. They have kept this branding since the replacement ... ...

our model, discuss the estimation of the prior signals, and finally give an inference algorithm. Important notations are listed in Table 1.

3.1. Problem Formalization

Given a question q, we recognize several mentions $M_q = \{m_1, m_2, \ldots, m_s\}$. Each mention m_i is a phrase in question q and corresponds to an entity candidate set $(e_{i1}, \ldots, e_{ik})$ in Wikipedia. Our goal is to predict the correct referent entity $e_i^\star$ from $(e_{i1}, \ldots, e_{ik})$ for mention m_i. In CQA, a question is associated with several categories selected by askers from a pre-defined category system (`https://answers.yahoo.com/dir/index`), and is often attached with a description (a.k.a, question body) which reveals more details about the asker's request. Besides the description and the categories, there is an answer list under the question. Table 2 shows an example from Yahoo! Answers, in which mentions in the question are in bold. Note that the answer is truncated due to the space limitation.

Fig. 1 gives the framework of the proposed NED method for questions. We first build a mention-candidate set by leveraging the metadata in Wikipeda for mention detection. Details are given in Section 3.2. We consider leveraging the metadata in CQA and the metadata in Wikipedia for NED. To learn the mapping of mentions in CQA and entities in Wikipedia, we propose to mine priori knowledge. Our intuition is as follows. We human beings understand questions by leveraging the knowledge in our mind which provides helpful signals for entity disambiguation. For example, given the question "What font does Apple use" and its category *Computers & Internet*, we know that candidate entity "Apple Inc." is more likely to belong to *Computers & In-*

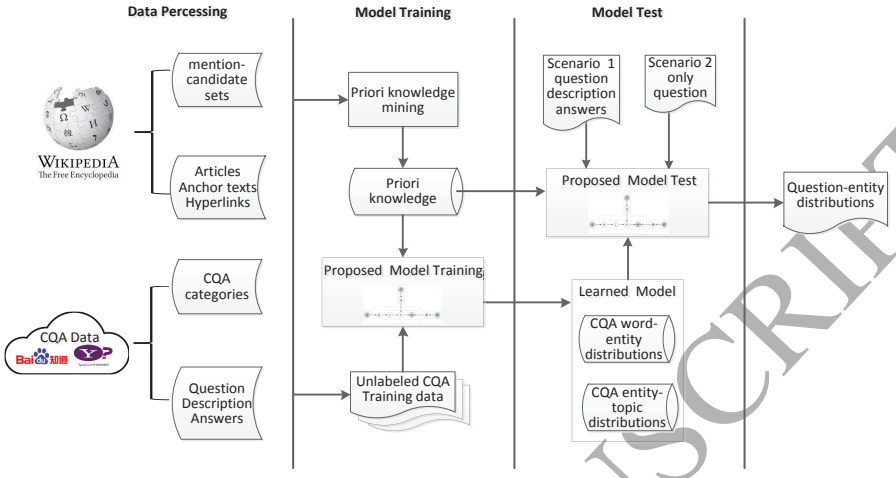


Figure 1: The framework of the proposed NED method for questions

ternet than the fruit "Apple", and the contextual words (i.e. "font" and "use") have stronger associations with "Apple Inc." than the fruit "Apple". Based on the above knowledge, we infer that the mention "Apple" in the question refers to "Apple Inc." in Wikipedia. In this case, we estimate priori question-category distributions, priori entity-category (topic) distributions, and priori word-entity distributions using questions with categories in CQA and articles with anchor texts in Wikipedia. Section 3.4 describes our priori knowledge mining approaches.

To this end, in training, in order to fully leverage the metadata in CQA, we concatenate question q, its description, and all its answers to form a document d. d is associated with the categories of q. Besides the mentions in q, we also recognize mentions in the description and the answers of q, and associate these mentions with their entity candidates in Wikipedia. As we have mentioned before, there is no ground truth for disambiguating the mentions in document d, and we try to learn a model using the categories of the questions, the mention-candidate set, and the contextual words in the training documents to disambiguate entity mentions appearing in questions in test sets. In addition, we leverage the mined priori knowledge to weakly supervise the learning of the model. Section 3.3 describes the model in detail.

We consider the NED for questions in two scenarios. The first scenario is knowl-

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

edge mining in CQA in which both questions and the metadata is available. For this scenario, we assume that the test set is homogeneous with the training set where each question is associated with its categories, its description, and its answers. The second scenario is question retrieval, question recommendation, or question routing in which only questions (i.e., question titles) are available. Therefore, for this scenario, we assume that the test set only contains questions with mentions and their entity candidates in Wikipedia. Note that for both of the two scenarios, our goal is to predict correct entities for mentions in questions. The difference is that in scenario one, we can leverage the metadata of questions to do the inference, while in scenario two, we only have questions. Section 3.5 introduces the Gibbs sampling algorithms for model training and test.

3.2. Mention-Candidate Construction

The first step for NED of questions is mention recognition and mention-candidate association. Following the previous work [6], we first construct a static dictionary with each entry as $\{m, (e_1, \ldots, e_k)\}$ using Wikipedia. Specifically, for an entity e in Wikipedia, we extract mentions using four resources: the title of e, the titles of pages redirecting to e, the titles of disambiguation pages containing e, and anchor texts with hyperlinks to e. To filter noise, we remove anchor texts composed by one character or numbers only, and anchor texts which appear less than 5 times. For Chinese data, we leverage Baidu Baike (`http://baike.baidu.com`) which is known as the Chinese Wikipedia to construct a dictionary. The only difference is that there is no redirect pages in Baidu Baike.

With the dictionaries, we identify mentions in questions, descriptions, and answers using the longest prefix match algorithm `http://en.wikipedia.org/wiki/Longest_prefix_match`), and associate the mentions with the entity candidates in the entries of the dictionaries.

3.3. Model

Before introducing our model, let us first imagine how users in CQA create questions and answers. In a common case, askers first have some topics in their mind.

ACCEPTED MANUSCRIPT

Following the topics, they come up with several relevant entities. Based on the topics and the entities, they select both mentions and contextual words (words alongside mentions) and write their questions. Answerers see the questions and understand both the topics and the entities in the questions. Following the same topics, they create answers with mentions and contextual words under entities relevant to those in the questions.

In this process, both askers and answerers have some priori knowledge. First, question categories may bias people's choice of topics, as before generating the data people have known the category system in CQA and have had several category candidates. Second, question categories also indicate the entities people choose to talk about in questions and answers. For example, if someone has selected a category "computers & internet", when he/she mentions "apple", most probably he/she refers to the apple company. Similarly, if he/she has selected "dinning out", he/she is more likely to talk about the fruit apple. Finally, both askers and answerers have basic knowledge such as basic characteristics of entities before they generate any data in CQA. The basic knowledge guides their selection of words/mentions in the generative process.

We try to simulate users' behavior through a generative model, and particularly we consider modeling users' priori knowledge and leveraging the knowledge in learning. Our idea is that we represent the priori knowledge as prior distributions of topics, entities and words, and take these prior distributions as hyperparameters of the Dirichlet priors in the generative model. We can prove that the prior distributions can weakly supervise the learning of the topic model by acting as the expectations of the distributions we aim to learn, as will be seen later.

Formally, suppose that there are T documents $\{d_i\}_{i=1}^{T}$. $\forall i$, document d_i is a concatenation of a question q_i, the description of q_i, and the answers of q_i. Document d_i can be represented as a union of a contextual word vector $(w_{d_i,1}, w_{d_i,2}, \ldots, w_{d_i,l_{d_i}})$ and a mention vector $(m_{d_i,1}, m_{d_i,2}, \ldots, m_{d_i,s_{d_i}})$, where l_{d_i} and s_{d_i} are the total number of contextual words and the total number of mentions in d_i, respectively. For example, the document in Table 2 is represented as $d = \{$what, **font**, do, **apple**, use, wonder, what, **font**, **apple**, use, **website**, ...$\}$, where words in bold are mentions and others are contextual words. Each document is related to K topics $\{z_i\}_{i=1}^{K}$. We assume that topics of documents correspond to categories of questions, and use a $T \times K$

ACCEPTED MANUSCRIPT

matrix A^0 to represent the priori knowledge about the question category. Given question q_i, $A^0(q_i)$ is the i-th row of A^0 with each element the prior probability of q_i belonging to a category. Taking the question in Table 2 for example, we may have $A^0(q) = \{Compter\ \&\ Internet^{0.15}, Programming\ \&\ Design^{0.04}, Software^{0.01}, ...\}$. For each topic z_j, there is an entity distribution Λ_j which reflects how likely people will talk about the entities under the topic (category). We use an $K \times M$ matrix Λ^0 to represent the priori knowledge about the entity topic, where M is the total number of entities. Given topic z_j, $\Lambda^0(j)$ is the j-th row of Λ^0 and is a prior entity distribution under the topic z_j. For example, the entity distribution of *Programming & Design* might be $\Lambda^0(z) = \{Microsoft^{0.10}, Linux^{0.09}, ..., Apple\ Inc.^{0.001}, ...\}$. Each entity corresponds to a word distribution. Here, the collection of words contains both contextual words and mentions. The word distributions measure the probabilities of words being used in documents related to the entities. Suppose that the total number of words is N, then we denote the word distribution corresponding to the v-th entity as Φ_v. We use an $M \times N$ matrix B^0 to represent the priori entity knowledge. $B^0(v)$ is the v-th row of B^0 and is a prior word distribution conditioned on the v-th entity. For example, we may have $B^0(Apple\ Inc.) = \{apple^{0.10}, software^{0.06}, ..., use^{0.007}, ..., website^{0.002}, ...\}$, indicating that the word "use" appearing in the context of the entity *Apple Inc.* with probability 0.007, etc.

The generative process can be formalized as

1. Draw a topic distribution $\theta_i \sim Dirichlet\left(A^0(q_i)\right)$ for each $i \in \{1, 2, \ldots, T\}$.
2. Draw an entity distribution $\Lambda_j \sim Dirichlet\left(\Lambda^0(j)\right)$ for each $j \in \{1, 2, \ldots, K\}$.
3. Draw a word distribution $\Phi_v \sim Dirichlet\left(B^0(v)\right)$ for each $v \in \{1, 2, \ldots, M\}$.
4. For each of the word (a contextual word/a mention) positions $(u, i), i \in \{1, \ldots, T\}$, $u \in \{1, \ldots, (l_{d_i} + s_{d_i})\}$,
   - Draw a topic $z_{i,u} \sim Multinomial(\theta_i)$.
   - Draw an entity $e_{i,u} \sim Multinomial(\Lambda_{z_{i,u}})$.
   - Draw a word (contextual word/mention) $w_{i,u} \sim Multinomial(\Phi_{e_{i,u}})$.

ACCEPTED MANUSCRIPT

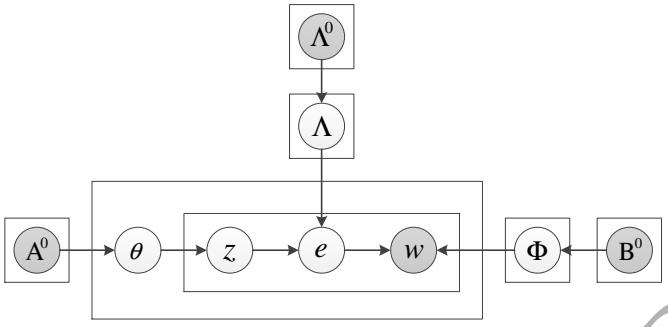


Figure 2: a generative model with priori knowledge

Here, $Dirichlet(A^0(q_i))$, $Dirichlet(\Lambda^0(j))$, and $Dirichlet(B^0(v))$ are Dirichlet prior distributions of topics, entities, and words respectively, where A^0, Λ^0, and B^0 are used as hyperparameters of the Dirichlet priors in the model. Fig. 2 gives the graphical model for the generative process.

We estimate A^0, Λ^0 and B^0 using question-answer pairs with categories in CQA and articles with anchor texts in Wikipedia, as will be seen later. Therefore, the proposed model leverages both the metadata in CQA and the metadata in Wikipedia in learning. It extends the traditional LDA [26] by replacing the heuristically designed hyperparameters with prior distributions estimated from the metadata. One advantage of doing so is that we can prove that the prior distributions are expectations of the topic distributions, the entity distributions, and the word distributions in the generative process. Formally, we have the following theorem:

Theorem 1. *In Model 2,* $\mathbb{E}(\theta_i) = A^0(q_i)$, $\mathbb{E}(\Lambda_j) = \Lambda^0(j)$, *and* $\mathbb{E}(\Phi_v) = B^0(v)$, $\forall i \in \{1, \ldots, T\}$, $j \in \{1, \ldots, M\}$ *and* $v \in \{1, \ldots, N\}$.

Proof. $\forall i \in \{1, \ldots, T\}$, let us suppose that $\theta_i = (\theta_i(1), \ldots, \theta_i(K))$. Since $\theta_i \sim Dirichlet(A^0(q_i))$, $\forall u \in \{1, \ldots, K\}$, we have

$$\begin{aligned}
\mathbb{E}(\theta_i(u)) &= \frac{\Gamma(\sum_{t=1}^{K} A^0(q_i)_t)}{\prod_{t=1}^{K} \Gamma(A^0(q_i)_t)} \int_0^1 \ldots \int_0^1 \ldots x_u^{A^0(q_i)_u} \ldots d(x_1) \ldots d(x_K) \\
&= \frac{\Gamma(\sum_{t=1}^{K} A^0(q_i)_t)}{\prod_{t=1}^{K} \Gamma(A^0(q_i)_t)} \cdot \frac{\prod_{t=1, t\neq u}^{K} \Gamma(A^0(q_i)_t) \cdot \Gamma(A^0(q_i)_u + 1)}{\Gamma(\sum_{t=1}^{K} A^0(q_i)_t + 1)} \\
&= \frac{A^0(q_i)_u}{\sum_{t=1}^{K} A^0(q_i)_t} = A^0(q_i)_u.
\end{aligned}$$

ACCEPTED MANUSCRIPT

Here, $\Gamma(\cdot)$ is the gamma function. $A^0(q_i)_t$ means the probability of category t in distribution $A^0(q_i)$, and we use the property of probabilistic distribution $\sum_{t=1}^{K} A^0(q_i)_t = 1$. With similar calculations, we have $\mathbb{E}(\Lambda_j) = \Lambda^0(j)$ and $\mathbb{E}(\Phi_v) = B^0(v)$, $\forall j \in \{1, \ldots, M\}, v \in \{1, \ldots, N\}$. □

Theorem 1 tells us that in the generative process of our model, topics, entities, and words are sampled from distributions under the guide of the prior distributions A^0, Λ^0, and B^0. When we estimate Λ and Φ using CQA data, the result will take the priors as basis and mix it with useful information mined from CQA. By this means, the priori knowledge learned from the metadata of CQA and Wikipedia actually supervises the learning of the model. More interestingly, since we estimate B^0 from Wikipeida and apply it to CQA through the proposed topic model, the language of CQA and the language of Wikipedia are connected. The result is that we learn a complete word-entity association that comprehensively reflects how people describe an entity in CQA and Wikipedia, as will be seen in our experiments.

We estimate Λ, and Φ in the model from data. With the posterior estimations, in test, given a question q with a mention m and its entity candidates $(e_{m,1}, \ldots, e_{m,k})$, we predict the referent entity for m by

$$e_m^\star = \text{argmax}_{1 \leqslant i \leqslant k} \, P\left(e_{m,i}|q\right). \tag{1}$$

3.4. Priori Distribution Estimation

We propose estimating the prior question-category distributions A^0, the prior entity-category distributions Λ^0, and the prior word-entity distributions B^0 using the metadata of CQA and the metadata of Wikipedia.

Estimation of A^0: A^0 represents prior question-category distributions, and we leverage questions with categories in CQA to estimate it. In many CQA web sites such as Yahoo! Answers and Baidu Knows, categories are organized in a tree structure. For simplicity, we only keep leaf categories. For each leaf category z, we gather all questions with z and train a naive Bayes classifier. Given a question $q = \{w_1, \ldots, w_n\}$ (w_i is the i-th word in q), we first calculate the posterior probability given by the naive

ACCEPTED MANUSCRIPT

Bayes classifier $p_{nb}(z|q) = \frac{p(z)p(q|z)}{p(q)} \propto p(z)\prod_{i=1}^{n} p(w_i|z)$, where $p(z)$ is assumed a uniform distribution. After that, we define the probability of q belonging to z as

$$p(z|q) = \eta \times p_{nb}(z|q) + (1-\eta) \times f(z), \tag{2}$$

where $f(z) = 1$ when z is selected by the asker of q, otherwise $f(z) = 0$. $\eta \in [0, 1]$ acts as a trade-off between the predicted categories and the categories selected by people. $p(z|q)$ balances the wisdom of data and the wisdom of people. The naive Bayes classifier can calibrate noise in human annotated categories and provide possible categories for new questions ($f(z)$ is always zero in this case). With $p(z|q)$, we rank all leaf categories for question q, and denote the top 10 ranked leaf categories as C_q. Finally, we define $A^0(q)$ as $\{p(z|q)/C \mid z \in C_q\}$, where $C = \sum_{z \in C_q} p(z|q)$.

Estimation of Λ^0: Λ^0 is prior entity-topic distributions. Since the topics in our model correspond to question categories, we leverage both the categories in CQA and the articles in Wikipedia to estimate Λ^0. Intuitively, a high probability of an entity under a category means either the entity is very relevant to the category or it is very common. We aim to let Λ^0 cover both of the two cases. For the former case, the key is to establish relationship between entities in Wikipedia and categories in CQA. Our idea is that we represent an entity as a bag of words using articles in Wikipedia and predict the probability of the entity under a category using the word-category distribution $p(w|z)$ in the naive Bayes classifier for A^0. Specifically, for each entity, we select 50 keywords using χ^2 independence test [7] from the article of Wikipedia. Suppose that for entity e, the set of keywords is $\{w_1, \ldots, w_{50}\}$, we calculate the likelihood of e relevant to a category z by $p_{rel}(e|z) = \prod_{i=1}^{50} p(w_i|z)$. For the latter case, we follow the existing work [28] and introduce entity popularity estimated from intra links of articles in Wikipedia. The popularity of e is defined as $pop(e) = \frac{n_{l_{in}}(e)}{\sum_{i=1}^{N_e} n_{l_{in}}(e_i)}$, where $n_{l_{in}}(e)$ is the number of incoming hyperlinks of e in Wikipedia, and N_e is the total number of entities in Wikipedia. $pop(e)$ measures the likelihood of e appearing in a document and is independent with categories. With $p_{rel}(e|z)$ and $pop(e)$, we define $\Lambda^0(z)$ as $\{p(e|z)/C_e\}$, where $p(e|z)$ is defined as

$$p(e|z) = \mu \times p_{rel}(e|z) + (1-\mu) \times pop(e), \tag{3}$$

ACCEPTED MANUSCRIPT

and $C_e = \sum_{i=1}^{M} p(e|z)$ is a normalizer. $\mu \in [0, 1]$ is a trade-off between the relevance and the commonness.

Estimation of B^0: we employ the collective context-aware topic model proposed by Sen [9] to estimate word-entity distributions B^0. Specifically, we implemented the combination of the context-aware topic model (CA) and the group learning model which is the best performing model in the paper. The model learns word distributions under entities using articles with anchor texts in Wikipedia. It takes context of mentions and entity groups estimated from Wikipedia into account, and represents the state-of-the-art topic model based method for NED in Wikipedia. Besides taking the model as priori knowledge, we also compare our model with it in our experiments.

3.5. Algorithm

We derive a collapsed Gibbs sampling algorithm to estimate Φ and Λ. Suppose that $z_{i,u}$ and $e_{i,u}$ represent the topic and the entity for the u-th word in document d_i respectively, $\vec{z}_{\neg(i,u)}$ and $\vec{e}_{\neg(i,u)}$ represent the vector of topics and the vector of entities in the corpus with $z_{i,u}$ and $e_{i,u}$ excluded respectively, and $\vec{w}$ is the vector of words in the corpus. We define $r = (i, u)$, and use $n^{(\cdot)}_{\cdot,\neg r}$ to represent counts with position r excluded. Then, we have

$$
\begin{aligned}
& p(z_r = k, e_r = e|\vec{z}_{\neg r}, \vec{e}_{\neg r}, \vec{w}) \\
&= \frac{p(\vec{e}, \vec{z}, \vec{w})}{p(\vec{z}_{\neg r}, \vec{e}_{\neg r}, \vec{w})} = \frac{p(\vec{w}|\vec{e})}{p(\vec{w}_{\neg r}|\vec{e}_{\neg r})p(w_r)} \cdot \frac{p(\vec{e}|\vec{z})}{p(\vec{e}_{\neg r}|\vec{z}_{\neg r})} \cdot \frac{p(\vec{z})}{p(\vec{z}_{\neg r})} \\
&\propto \frac{\Gamma\left(n_e^{(w_r)} + B^0(e)_{w_r}\right)\Gamma\left(\sum_{v=1}^{N} n_{e,\neg r}^{(v)} + B^0(e)_v\right)}{\Gamma\left(n_{e,\neg r}^{(w_r)} + B^0(e)_{w_r}\right)\Gamma\left(\sum_{v=1}^{N} n_e^{(v)} + B^0(e)_v\right)} \times \\
&\quad \frac{\Gamma\left(n_k^{(e)} + \Lambda^0(k)_e\right)\Gamma\left(\sum_{j=1}^{M} n_{k,\neg r}^{(j)} + \Lambda^0(k)_j\right)}{\Gamma\left(n_{k,\neg r}^{(e)} + \Lambda^0(k)_e\right)\Gamma\left(\sum_{j=1}^{M} n_k^{(j)} + \Lambda^0(k)_j\right)} \times \\
&\quad \frac{\Gamma\left(n_i^{(k)} + A^0(q_i)_k\right)\Gamma\left(\sum_{l=1}^{K} n_{i,\neg r}^{(l)} + A^0(q_i)_l\right)}{\Gamma\left(n_{i,\neg r}^{(k)} + A^0(q_i)_k\right)\Gamma\left(\sum_{l=1}^{K} n_i^{(l)} + A^0(q_i)_l\right)} \\
&= \frac{n_{e,\neg r}^{(w_r)} + B^0(e)_{w_r}}{\sum_{v=1}^{N} n_{e,\neg r}^{(v)} + B^0(e)_v} \cdot \frac{n_{k,\neg r}^{(e)} + \Lambda^0(k)_e}{\sum_{j=1}^{M} n_{k,\neg r}^{(j)} + \Lambda^0(k)_j} \cdot \frac{n_{i,\neg r}^{(k)} + A^0(q_i)_k}{\sum_{l=1}^{K} n_{i,\neg r}^{(l)} + A^0(q_i)_l}. \qquad (4)
\end{aligned}
$$

ACCEPTED MANUSCRIPT

where $n_e^{(v)}$ is the number of times that word v has been observed with entity e, $n_k^{(e)}$ is the number of times that entity e has been observed with topic k, and $n_i^{(k)}$ is the number of times that topic k has been observed in document d_i. $B^0(e)_v$, $\Lambda^0(k)_e$, and $A^0(q_i)_k$ are prior probabilities of word v under entity e, entity e under topic k, and topic k under question q_i, respectively.

With Equation (4), we can estimate Λ and Φ from training data by iteratively sampling and updating. In practice, however, we find that A^0, Λ^0, and B^0 are sometimes too small compared with the counts to impact the estimation of the parameters. Therefore, we introduce smoothing parameters $\{\delta_a, \delta_\lambda, \delta_b\}$ to scale A^0, Λ^0, and B^0. Thus, $p(z_r = k, e_r = e|\vec{z}_{\neg r}, \vec{e}_{\neg r}, \vec{w})$ becomes

$$\begin{aligned} & p(z_r = k, e_r = e|\vec{z}_{\neg r}, \vec{e}_{\neg r}, \vec{w}) \\ & \propto \frac{n_{e,\neg r}^{(w_r)} + B^0(e)_{w_r} \cdot \delta_b}{\left(\sum_{v=1}^{N} n_{e,\neg r}^{(v)}\right) + \delta_b} \cdot \frac{n_{k,\neg r}^{(e)} + \Lambda^0(k)_e \cdot \delta_\lambda}{\left(\sum_{j=1}^{M} n_{k,\neg r}^{(j)}\right) + \delta_\lambda} \cdot \frac{n_{i,\neg r}^{(k)} + A^0(q_i)_k \cdot \delta_a}{\left(\sum_{l=1}^{K} n_{i,\neg r}^{(l)}\right) + \delta_a}. \end{aligned} \tag{5}$$

Algorithm 1 gives the Gibbs sampling algorithm for training our model, where D represents the training set containing T documents, E_{d_i} represents the collection of entity candidates of mentions in document d_i, $E_{d_i}(w_r)$ represents the entity candidates of mention w_r, and C_{q_i} represents the top 10 leaf categories of question q_i in d_i. The complexity of Algorithm 1 is $O(N_{total}\overline{E_d}K)$, where N_{total} is the total number of words in training corpus, $\overline{E_d}$ is the average number of entity candidates per document, and K is the number of topics. After Algorithm 1 converges or the number of iterations reaches an upper bound, we estimate word distributions Φ and entity distributions Λ by

$$\Phi_{e,w} = \frac{n_e^{(w)} + B^0(e)_w \cdot \delta_b}{\sum_{v=1}^{N} n_e^{(v)} + \delta_b}, \quad \Lambda_{k,e} = \frac{n_k^{(e)} + \Lambda^0(k)_e \cdot \delta_\lambda}{\sum_{j=1}^{M} n_k^{(j)} + \delta_\lambda}, \tag{6}$$

where $\Phi_{e,w}$ measures the likelihood of word w appearing under entity e, and $\Lambda_{k,e}$ measures the possibility of entity e belonging to topic (category) k.

After we have the estimations of Φ and Λ, we infer referent entities for mentions in questions from test sets by a sampling algorithm similar to Algorithm 1. The only difference is that we replace $\frac{n_{e,\neg r}^{(w_r)} + B^0(e)_{w_r}\delta_b}{\left(\sum_{v=1}^{N} n_{e,\neg r}^{(v)}\right) + \delta_b}$ and $\frac{n_{k,\neg r}^{(e)} + \Lambda^0(k)_e\delta_\lambda}{\left(\sum_{j=1}^{M} n_{k,\neg r}^{(j)}\right) + \delta_\lambda}$ in Equation (5) with the estimated Φ_{e,w_r} and $\Lambda_{k,e}$, respectively. When the iterations of inference finish, we calculate $P(e|q)$ in Equation (1) as $p(e|q) = \frac{n_q^{(e)}}{N_q}$, where $n_q^{(e)}$ denotes the number of

ACCEPTED MANUSCRIPT

Algorithm 1: Gibbs sampling for training

Input: training corpus D, entity collection $E = \cup_{i=1:T}(E_{d_i})$, A^0, B^0, Λ^0, $\{C_{q_i}\}_{i=1}^{T}$, parameters $\{\mu, \eta, \delta_a, \delta_b, \delta_\lambda\}$

Output: multinomial parameters Φ and Λ

//Initialization

for *each* w_r, $r = \{i, u\}$, $i \in \{1, \ldots, T\}$, $u \in \{1, \ldots, l_{d_i} + s_{d_i}\}$ **do**

  if w_r *is a mention* **then**

    sample $(z_r, e_r) \sim A^0(q_i)_z \cdot \Lambda^0(z)_e \cdot B^0(e)_{w_r}$, where $e \in E_{d_i}(w_r)$, $z \in C_{q_i}$.

  else

    sample $(z_r, e_r) \sim A^0(q_i)_z \cdot \Lambda^0(z)_e \cdot B^0(e)_{w_r}$, where $e \in E_{d_i}$, $z \in C_{q_i}$.

//Gibbs sampling over burn-in period and sampling period

while *not finished* **do**

  for *each* w_r, $r = \{i, u\}$, $i \in \{1, \ldots, T\}$, $u \in \{1, \ldots, l_{d_i} + s_{d_i}\}$ **do**

    if w_r *is a mention* **then**

      sample $(z_r, e_r) \sim$ Equation (5), $e \in E_{d_i}(w_r)$, $k \in C_{q_i}$.

    else

      sample $(z_r, e_r) \sim$ Equation (5), $e \in E_{d_i}$, $k \in C_{q_i}$.

  //check convergence and read out parameters

  if *converged and L sampling iterations since last read out* **then**

    read out Φ and Λ according to Equation (6).

times that entity e is observed in question q, and N_q is the total number of contextual words and mentions in q.

4. Experiments

We conduct experiments to test the performance of the proposed method on NED for questions in CQA.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

4.1. Experiment Setup

4.1.1. Data Sets

We crawled 128,851,369 English questions (i.e. question titles) from Yahoo! Answers and 66,550,604 Chinese questions from Baidu Knows. Descriptions, categories, and answers associated with these questions were also crawled. Numbers of leaf categories are 984 in Yahoo data and 1,658 in Baidu data. We used the two data sets to train naive Bayes classifiers for estimating A^0 and Λ^0, as described in Section 3.4.

To recognize mentions and entity candidates, we built an English dictionary from a Wikipedia dump on May 3, 2013, and a Chinese dictionary from 5 million Baidu Baike pages. There is no dump data for Baidu Baike, and we thus crawled these pages from its website. Totally, we have 30 million English entities and 4,716,249 Chinese entities. The two encyclopedia data was also used to estimate entity popularity.

We recognized mentions and their entity candidates in the crawled data, and filtered the data by keeping questions with at least one mention that has at least 5 entity candidates. From these data, we randomly sampled examples to form training sets and data sets for human labeling. Table 3 gives an overview of the two training data sets. On average, each English document contains 17.7 mentions, and each mention corresponds to 5.1 entity candidates. Each Chinese document contains 37.3 mentions, and each mention corresponds to 2.5 entity candidates.

Table 3: Overview of the two training data sets

Statistics	**Yahoo! Answer**	**Baidu Knows**
Question #	120,000	120,000
Description #	94,624	61,131
Answer #	599,810	436,643
Vocabulary size	149,307	162,487
Total word #	16,138,148	10,673,456

For evaluation, we randomly sampled 6,000 Chinese questions and 2,000 English questions. These questions are not contained by the training sets. We recruited native speaker labelers to select the correct entity for each mention from a given candidate set. The guideline is that the labelers first browse Wikipedia or Baidu Baike. If they

ACCEPTED MANUSCRIPT

cannot make decisions, they have to investigate using search engines. Mentions that do not correspond to any candidate were labeled as "NIL" and removed. After removing questions without labeled mentions, we obtained 1,686 English questions and 5,101 Chinese questions. On average, each English question has 1.55 labeled mentions, and each Chinese question has 1.63 labeled mentions. These data was published as `https://github.com/NEDstudy/NEDforCQA`. We randomly split each of the two labeled sets into a test set and a validation set with a ratio 3:1.

To implement our model, we also have to estimate prior word distributions B^0. We implemented the best performing model in [9] to estimate B^0 from the encyclopedia data. To balance efficiency and efficacy, we first collected all entities that correspond to a mention in the training data and the data for labeling. Then we constructed a subset of Wikipedia and a subset of Baidu Baike with pages of these entities. Finally, we learned the collective context-aware topic models on the two subsets. Table 4 gives details of the two subsets.

Table 4: Details of the two subsets

Statistics	**Wikipedia**	**Baidu Baike**
Entity #	32,100	36,636
Hyperlink #	542,875	676,348
Vocabulary size	384,191	334,198
Total word #	23,274,551	46,252,998

4.1.2. Baselines

Cosine of the entity vectors and mention vectors was calculated as a straightforward baseline method. An entity vector was constructed by representing the title and the first paragraph of the entity page in Wikipedia or Baid Baike as bag of words and weighting the words with $tf \times idf$ scores. Similarly, a mention vector was created by using the whole question where the mention appears. Besides cosine, we compared our method with **AIDA** [13] and **Wikifier2.0** [8] which are representative state-of-the-art NED systems. We downloaded their source codes from `https://github.com/yago-naga/aida` and `http://cogcomp.cs.illinois.edu/page/download_view/Wikifier`.

ACCEPTED MANUSCRIPT

In addition, we also compared our method with three topic model based methods. The traditional **LDA** model [26] was considered as a baseline, since we used a topic modeling approach for NED in this work. We also compared our model with the entity-topic model (**ET model**) proposed by Han and Sun [31], which can jointly model context compatibility, topical coherence and their correlations. For consistency, we used our mention detection method during the model implementation. Since we estimated B^0 using the best performing model in [9], the model naturally became a baseline. We directly applied the model to NED for questions in CQA, and denoted the model as **CA+20G** following the notation in [9].

4.1.3. Evaluation Metrics

A mention-entity pair $\langle m, e\rangle$ is judged as correct if and only if e is the correct referent entity for m. We used micro-averaged and macro-averaged accuracy as evaluation metrics. Micro-averaged accuracy means the proportion of mentions correctly disambiguated, and macro-averaged accuracy is defined as the proportion of correct referent entities predicted for the same mention, averaged over all distinct mentions. We denote micro-averaged accuracy as A_{micro}, and macro-averaged accuracy as A_{macro}.

4.2. Parameter Tuning

There are several parameters we have to determine. In LDA and CA+20G, we fixed the hyperparameters of all the Dirichlet priors as 0.01. In ET Model, we set the parameters according to [31]. We selected the number of latent topics in LDA from $\{10, 20, 30, 40, 50\}$. In our model, we selected μ in Equation (3) and η in Equation (2) from $\{0.1, 0.2, ..., 0.9\}$ and the three scaling parameters $\{\delta_a, \delta_\lambda, \delta_b\}$ from 10, 100, 1000, 10000, 100000, and 1000000. We set 500 as the maximum iteration number for all the Gibbs sampling algorithms. On both of the two validation sets, we found that the optimal μ and η are 0.1 and 0.4 respectively. On Yahoo data, we finally set $\delta_a = \delta_\lambda = \delta_b = 100$, and on Baidu data, the best $(\delta_a, \delta_\lambda, \delta_b)$ is $\delta_a = \delta_\lambda = \delta_b = 1000000$.

4.3. Evaluation Results

Table 5 gives the evaluation results on the two data sets for test scenario 1 where both questions and their metadata are available, and Table 6 reports the evaluation

ACCEPTED MANUSCRIPT

Table 5: Evaluation results on scenario 1 (full metadata)

Method	**Yahoo! Answers**		**Baidu Knows**	
	A_{micro}	A_{macro}	A_{micro}	A_{macro}
LDA	53.62	64.55	60.83	70.48
Cosine	61.81	69.99	67.10	75.56
AIDA	63.68	70.21	-	-
Wikifier2.0	68.29	73.35	-	-
ET Model	74.11	79.25	62.17	72.31
CA+20G	74.30	79.29	70.73	77.63
Our Model	**78.10**	**82.37**	**73.40**	**79.27**

results for test scenario 2 where only questions are available. The results of AIDA and Wikifier2.0 are only available on Yahoo data, because their open-source tools only support English.

Results in Table 5 and results in Table 6 are consistent. From the two tables, we can see that in both of the two scenarios, our model outperforms all the baseline methods on both the English data and the Chinese data, and the improvements are statistically significant (sign-test, p value < 0.05) on both metrics.

Without any priori knowledge, LDA performed the poorest. The result demonstrates the importance of priori knowledge for the NED task. Cosine comes to the next, due to the few common words between the mention context in CQA data and entity context in Wikipedia article. AIDA is a graph based method. Its performance is not satisfied in this task, and dropped a lot in senario 2. This may because the built graph tends to be sparse in CQA data, due to the shortness of a question document. Wikifier2.0 performs much worse than ET Model, CA+20G and our model, which is consistent with the conclusion that collective methods are better than individual methods drawn by the existing work. ET Model worked almost equally well as CA+20G on English data but performed poorly on Chinese data. This is because there is a lot of noise in the hyperlinks of Baidu Baike articles. Our model significantly outperforms CA+20G and ET Model, indicating that we cannot simply apply the existing models

ACCEPTED MANUSCRIPT

Table 6: Evaluation results on scenario 2 (only question)

Method	**Yahoo! Answers**		**Baidu Knows**	
	A_{micro}	A_{macro}	A_{micro}	A_{macro}
LDA	53.42	64.45	60.49	70.46
Cosine	55.78	67.26	65.05	73.88
AIDA	59.01	67.17	-	-
Wikifier2.0	67.25	72.66	-	-
ET Model	72.58	78.99	60.27	71.06
CA+20G	72.78	79.34	69.85	76.63
Our Model	**77.49**	**81.26**	**73.35**	**79.40**

learned on knowledge bases to questions in CQA due to the language gap. But if we take them as priori knowledge in a generative model trained on CQA, the problem can be well solved.

Results in Table 6 are consistently worse than results in Table 5. Particularly, the performances of Cosine and AIDA drop significantly when only questions are available. The results tell us that NED becomes more challenging in tasks like question retrieval, question recommendation, and question routing. On the other hand, without metadata, the performance of our method only slightly drops, which demonstrates the power of the priori knowledge and the robustness of our method.

4.4. Discussion

We first use examples to study why our model outperforms the best performing baseline method CA+20G, and investigate how the word-entity associations Φ learned by our model differs from that learned by CA+20G from Wikipedia. Then, we conduct an experiment to investigate how the estimation of the different parameters affect the results. Finally, we discuss some possible applications of the proposed method.

4.4.1. Qualitative comparison between Model 2 and CA+20G

The success of our model comes from its capability on leveraging different types of priori knowledge estimated from the metadata of CQA and Wikipedia. Table 7 gives two examples to further justify this claim.

ACCEPTED MANUSCRIPT

Table 7: Examples for comparison between Model 2 and CA+20G

(a) Disambiguation results of Model 2 and CA+20G for "CK"

Result	**Label**	**Model 2**	**CA+20G**
	Calvin Klein	Calvin Klein	Citizen Kane
Context	Question: *Do the "CK in 2 u" for him **smell** the same as the women's **perfume***		
	Answer: *Nope. It's actually **smell** different. It's kinda **soft**, not sport though. ...*		
	Category: *Men's Health*		

(b) Disambiguation results of Model 2 and CA+20G for "NWS"

Result	**Label**	**Model 2**	**CA+20G**
	National Weather Service	National Weather Service	News Corporation
Context	Question: *NWS says that a **storm** is moving east and northeast but from what I see it is moving east*		

Table 7(a) shows an example from scenario 1. Mention "CK" in the question refers to entity "Calvin Klein" in Wikipedia. CA+20G failed on this case, because regarding to "Calvin Klein", the language of CQA and the language of Wikipdia are quite different. For example, "CK" in this case can be well disambiguated from words like "smell" and "soft". These words, however, are seldom used by people in Wikipedia when they talk about "Calvin Klein". Therefore, the association of these words with "Calvin Klein" cannot be learned by CA+20G. On the other hand, "Calvin Klein" is more likely to appear in category *Men's Health* than "Citizen Kane" which is a film. Our model leveraged the category information, and therefore can successfully recognize the correct referent entity for this mention.

Table 7(b) shows an example from scenario 2. "NWS" refers to "National Weather Service" in the question. CA+20G failed on this case due to the language gap between CQA and Wikipedia, and the little information provided by the short question. On the other hand, our method learned that 1) the most possible category for the question is *Weather*; 2) "National Weather Service" is more likely to appear in category *Weather* than "News Corporation". It leveraged these extra information in inference and successfully disambiguated mention "NWS" for this case.

ACCEPTED MANUSCRIPT

Table 8: Comparison between the learnt Φ with the prior B^0

Entity	**Top 10 words from Φ**	**Top 10 words from B^0**
Apple Inc.	apple; iphone; **use**; **application**; **criticism**; **music**; **store**; product; **user**; itunes	apple; apple inc.; company; jobs; product; iphone; year; introduce; announce; mac
Java (programming language)	java; use; class; **need**; method; **try**; **download**; **run**; void; static	java; class; method; use; code; object; program; sun; call; applet
Eclipse	eclipse; **see**; pilate; **take**; **jesus**; **when**; **where**; time; duration; darken	eclipse; darkness; crucifixion; solar; occur; hour; year; account; minute; moon

4.4.2. Comparison between Φ and B^0

An interesting observation is that by taking prior word-entity associations estimated from Wikipedia as priori knowledge, our model can select relevant words for an entity from both CQA and Wikipedia, and finally learn a comprehensive representation of the relationship of words and entities. Table 8 shows some examples. It compares the word-entity associations learned by our model and CA+20G (i.e., B^0) by listing top 10 words for some entities. From the table, we can see that the results are really interesting. For example, for entity "Eclipse", from the top 10 words learned from Wikipedia, we know that people in Wikipedia talk more about the phenomenon and the principle of eclipse, while in CQA, besides these, people also care about "when" and "where" they can go to "see" eclipse. Through learning, our model realizes the connection between CQA and Wikipedia, and renders a more complete description about an entity.

4.4.3. Effect of the estimated parameters

Our model leveraged three prior parameters (A^0, Λ^0, and B^0) estimated from the metadata of CQA and Wikipedia. To see how the estimation of different parameters affect the results, it is instructive to perform parameter elimination. Based on our model, we implemented 3 alternative models by replacing each prior parameter one by

ACCEPTED MANUSCRIPT

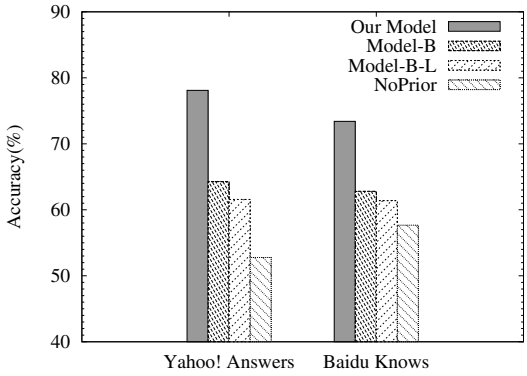


Figure 3: Effect of different parameters

one with a fixed hyperparameter 0.01. We denoted the model without B^0 as **Model-B**, the model without B^0 and Λ^0 as **Model-B-L**, and the model without any priori knowledge as **NoPrior**. We used the same data sets to train and test these models. Figure 3 shows their performances on scenario 1 in terms of micro-averaged accuracy.

Results on the two data sets are consistent. The accuracy decreased with the prior parameter elimination. Without any priori knowledge, **NoPrior** performed the worst on both English (52.78%) and Chinese data (57.68%), even though it has the same three-level generative process with our model. This demonstrates the importance of priori knowledge for the NED task, and the effectiveness of the way we leverage the priori knowledge. The accuracy dropped dramatically when we eliminated B^0. The decline was 13.84% and 10.59% respectively. It suggests that B^0 has more effects on the task, compared with A^0 and Λ^0. The contextual information reflected by the word-entity distributions B^0 is essential for the NED task. On the other hand, the quality of A^0 and Λ^0 needs to be improved. We estimated them by leveraging a naive Bayes classifier trained on the noise CQA data. Tested on 100 manually labeled questions, the accuracy of the classifier was only 71%. We argue that further refine the estimation method may further improve their effects on the task. We leave it for future work.

4.4.4. Possible applications

Experiments on two scenarios (metadata is available or not) have verified the advantages of our method on efficacy and robustness. Our model was designed to the task

ACCEPTED MANUSCRIPT

of NED for questions in CQA, but can be also applied to other text in CQA such as question description and answers. The proposed method can benefit many applications such as information extraction in CQA, question retrieval, and related question recommendation. When applying our method to these applications, one needs to prepare his/her own training data for covering the entities in the used knowledge base, so as to learn a comprehensive entity-word distributions Φ.

Performance in terms of accuracies however, form a story only half told. What also matters in practice is the time taken to run these procedures. The training time of our model is about 3 days on a workstation with one Intel Core i7 4785T processor. Since the training can be done offline, we believe that the training time is not critical to the real world usage as the online inference on a new question is quick. Our topic model scales quadratically with the size of the dataset (in terms of the number of entities). One can utilize faster inference techniques [32, 33, 34] for help. Distributed algorithms for topic models is still a topic of active research, and different topic models (such as our model) will likely require new strategies to run inference in a distributed setting.

5. Conclusion

We propose a new topic model for disambiguating named entity mentions appearing in questions in CQA. The model simulates users' behavior in CQA and performs learning in a weakly supervised manner. It incorporates different types of priori knowledge estimated from metadata of CQA and metadata of Wikipedia into learning. Experimental results show that the model can significantly outperform state-of-the-art named entity disambiguation methods when we apply them to questions in CQA.

This work suggests some interesting directions for future work. For examples, we can further explore different machine learning techniques to improve the priori knowledge estimation. It would be interesting to investigate the potential impact of additional information in CQA such as answer quality, and modeling the interaction between questions and answers may further improve the NED performance.

ACCEPTED MANUSCRIPT

Acknowledgments.

This work was supported by the National Natural Science Foundation of China (Grand Nos.U1636211, 61672081, 61370126), Beijing Advanced Innovation Center for Imaging Technology (No.BAICIT-2016001), National High Technology Research and Development Program of China (No.2015AA016004),the Fund of the State Key Laboratory of Software Development Environment (No.SKLSDE-2015ZX-16).

References

References

[1] K. Zhang, W. Wu, H. Wu, Z. Li, M. Zhou, Question retrieval with high quality answers in community question answering, in: Proceedings of the 23rd ACM Conference on Information and Knowledge Management, 2014, pp. 371–380.

[2] K. Sun, Y. Cao, X. Song, Y.-I. Song, X. Wang, C.-Y. Lin, Learning to recommend questions based on user ratings, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management, 2009, pp. 751–758.

[3] B. Li, I. King, Routing questions to appropriate answerers in community question answering services, in: Proceedings of the 19th ACM Conference on Information and Knowledge Management, 2010, pp. 1585–1588.

[4] M. A. Khalid, V. Jijkoun, M. De Rijke, The impact of named entity normalization on information retrieval for question answering, in: Advances in Information Retrieval, Springer, 2008, pp. 705–710.

[5] R. C. Bunescu, M. Pasca, Using encyclopedic knowledge for named entity disambiguation, in: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, Vol. 6, 2006, pp. 9–16.

[6] S. Cucerzan, Large-scale named entity disambiguation based on wikipedia data, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Vol. 7, 2007, pp. 708–716.

ACCEPTED MANUSCRIPT

[7] R. Mihalcea, A. Csomai, Wikify!: linking documents to encyclopedic knowledge, in: Proceedings of the 16th ACM International Conference on Information and Knowledge Management, 2007, pp. 233–242.

[8] L. Ratinov, D. Roth, D. Downey, M. Anderson, Local and global algorithms for disambiguation to wikipedia, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 1375–1384.

[9] P. Sen, Collective context-aware topic models for entity disambiguation, in: Proceedings of the 21st international conference on World Wide Web, 2012, pp. 729–738.

[10] A. Alhelbawy, R. Gaizauskas, Graph ranking for collective named entity disambiguation, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 75–80.

[11] D. Milne, I. H. Witten, Learning to link with wikipedia, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, 2008, pp. 509–518.

[12] P. Ferragina, U. Scaiella, Tagme: on-the-fly annotation of short text fragments (by wikipedia entities), in: Proceedings of the 19th ACM International Conference on Information and knowledge management, 2010, pp. 1625–1628.

[13] 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 Conference on Empirical Methods in Natural Language Processing, 2011, pp. 782–792.

[14] X. Cheng, D. Roth, Relational inference for wikification, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, p. 1787C1796.

ACCEPTED MANUSCRIPT

[15] F. Wang, Z. Wang, S. Wang, Z. Li, Exploiting description knowledge for keyphrase extraction, in: Proceedings of the 13th Pacific Rim International Conference on Artificial Intelligence, 2014, pp. 130–142.

[16] M. Pershina, Y. He, R. Grishman, Personalized page rank for named entity disambiguation, in: Proceedings of the 2015 Annual Conference of the North American Chapter of the ACL, NAACL HLT, Vol. 14, 2015, pp. 238–243.

[17] Z. He, S. Liu, M. Li, M. Zhou, L. Zhang, H. Wang, Learning entity representation for entity disambiguation., in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 2013, pp. 30–34.

[18] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, X. Wang, Modeling mention, context and entity with neural networks for entity disambiguation, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2015, pp. 1333–1339.

[19] H. Huang, L. Heck, H. Ji, Leveraging deep neural networks and knowledge graphs for entity disambiguation, in: arXiv preprint arXiv:1504.07678, 2015.

[20] I. Yamada, H. Shindo, H. Takeda, Y. Takefuji, Joint learning of the embedding of words and entities for named entity disambiguation, in: arXiv preprint arXiv:1601.01343, 2016.

[21] E. Meij, W. Weerkamp, M. de Rijke, Adding semantics to microblog posts, in: Proceedings of the 5th ACM international conference on Web Search and Data Mmining, 2012, pp. 563–572.

[22] X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, Y. Lu, Entity linking for tweets., in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 2013, pp. 1304–1311.

[23] R. Blanco, G. Ottaviano, E. Meij, Fast and space-efficient entity linking for queries, in: Proceedings of the 8th ACM International Conference on Web Search and Data Mining, 2015, pp. 179–188.

ACCEPTED MANUSCRIPT

[24] B. Gu, V. S. Sheng, A robust regularization path algorithm for -support vector classification, IEEE Transactions on Neural Networks & Learning Systems 1 (2016) 1–8.

[25] B. Gu, V. S. Sheng, S. Li, Bi-parameter space partition for cost-sensitive svm, in: International Conference on Artificial Intelligence, 2015, pp. 3532–3539.

[26] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, in: Journal of machine Learning research, Vol. 3, 2003, pp. 993–1022.

[27] D. Ramage, D. Hall, R. Nallapati, C. D. Manning, Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora, in: Proceedings of the 2009 Conference on Empirical Methods on Natural Language Processing, 2009, pp. 248–256.

[28] X. Han, L. Sun, A generative entity-mention model for linking entities with knowledge base, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 945–954.

[29] Y. Li, C. Wang, F. Han, J. Han, D. Roth, X. Yan, Mining evidences for named entity disambiguation, in: Proceedings of the 19th ACM SIGKDD international conference on Knowledge Discovery and Data Mining, 2013, pp. 1070–1078.

[30] Y. Li, S. Tan, H. Sun, J. Han, D. Roth, X. Yan, Entity disambiguation with linkless knowledge bases, in: Proceedings of the 25th International Conference on World Wide Web, 2016, pp. 1261–1270.

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

[32] A. Smola, S. Narayanamurthy, An architecture for parallel topic models, in: Proceedings of the VLDB Endowment, Vol. 3, 2010, pp. 703–710.

ACCEPTED MANUSCRIPT

[33] J. Yuan, F. Gao, Q. Ho, W. Dai, J. Wei, X. Zheng, E. P. Xing, T.-Y. Liu, W.-Y. Ma, Lightlda: Big topic models on modest computer clusters, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 1351–1361.

[34] D. Newman, A. Asuncion, P. Smyth, M. Welling, Distributed algorithms for topic models, in: Journal of Machine Learning Research, Vol. 10, 2009, pp. 1801–1828.

ACCEPTED MANUSCRIPT