# Collective Context-Aware Topic Models for Entity Disambiguation

Prithviraj Sen*
IBM Research - Almaden
San Jose, CA, USA
senp@us.ibm.com

## ABSTRACT

A crucial step in adding structure to unstructured data is to identify references to entities and disambiguate them. Such disambiguated references can help enhance readability and draw similarities across different pieces of running text in an automated fashion. Previous research has tackled this problem by first forming a catalog of entities from a knowledge base, such as Wikipedia, and then using this catalog to disambiguate references in unseen text. However, most of the previously proposed models either do not use all text in the knowledge base, potentially missing out on discriminative features, or do not exploit word-entity proximity to learn high-quality catalogs. In this work, we propose topic models that keep track of the context of every word in the knowledge base; so that words appearing within the same context as an entity are more likely to be associated with that entity. Thus, our topic models utilize all text present in the knowledge base and help learn high-quality catalogs. Our models also learn groups of co-occurring entities thus enabling collective disambiguation. Unlike most previous topic models, our models are non-parametric and do not require the user to specify the exact number of groups present in the knowledge base. In experiments performed on an extract of Wikipedia containing almost 60,000 references, our models outperform SVM-based baselines by as much as 18% in terms of disambiguation accuracy translating to an increment of almost 11,000 correctly disambiguated references.

## Categories and Subject Descriptors

H.4.m [**Information Systems**]: Miscellaneous; G.3 [**Mathematics of Computing**]: Probability and Statistics; I.7 [**Computing Methodologies**]: Document and Text Processing

## Keywords

Entity Disambiguation, Topic Models

## 1. INTRODUCTION

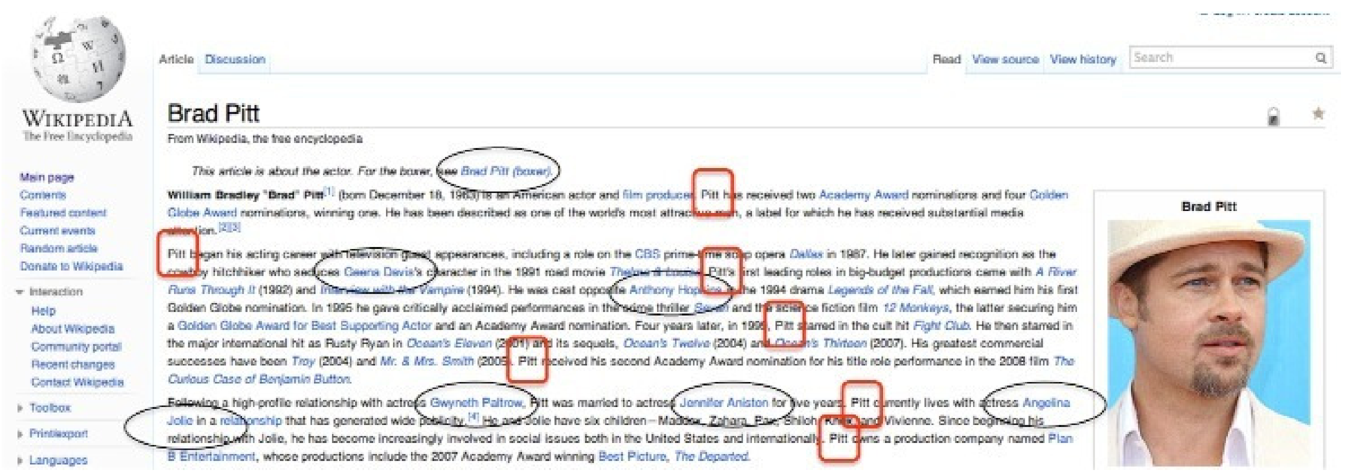
An important step in data cleaning is performing *entity resolution* [11] wherein, we take mentions or *references* of entities strewn in different data sources and match them against each other. Since entity resolution allows one to draw connections between different datasets, essentially allowing us to join them, and thus create larger, more dependable datasets in an automatic fashion, it has been the topic of fervent research for almost the past half a century.

*Work done while author was at Yahoo! Labs, Bangalore.

We are interested in a particular variant of entity resolution known as *entity disambiguation* [9, 16, 17, 18, 6, 10, 26]. In entity disambiguation, we are given a collection of documents containing words and references. The challenge lies in identifying which reference belongs to which entity. Prior research has tackled this problem by dividing it into two steps: 1) learn a *catalog* of entities containing quantities that can help in the disambiguation process and, 2) use the learnt catalog to disambiguate references in unseen documents. Usually, quantities that help disambiguate references most commonly include *word-entity associations*, i.e., storing with each entity the most common words that appear alongside it, and *groups of entities* that appear frequently within the same document. The most common approach to learning a catalog is from a pre-annotated knowledge base, usually Wikipedia [9, 16, 17, 18, 6, 26]. To date, prior work has used the text in Wikipedia pages, links and anchor text within Wikipedia (*intra-wiki* links), Wikipedia's disambiguation and redirect pages, and the Wikipedia category hierarchy for entity disambiguation.

The words appearing in a given piece of running text form one of the main sources of evidence that we can utilize to disambiguate references appearing in it. One approach to leveraging this evidence, is to use the learnt word-entity associations and the words appearing in text to identify which entities the references are referring to. One of the challenges in learning word-entity associations for entity disambiguation lies in the fact that knowledge bases such as Wikipedia are organized into separate documents and most documents contain references to more than one entities. For instance, Figure 1 shows Brad Pitt's Wikipedia page and here we have circled all the blue text indicating intra-wiki links to other peoples' Wikipedia pages. In this case, the anchor text represents references and the pages to which the links point to denote the entity being referred to. For example, the first reference is to a different Brad Pitt, the second to Geena Davis and so on. In such a setting, if one were interested in associating each word on the page with exactly one entity amongst all the entities being referred to by the page, then one needs to go through a non-trivial inference process.

Previous work has circumvented this problem of associating words to entities by utilizing a *window*-ing strategy [10, 6, 17, 16]. Given a user-defined window length, for each reference in a Wikipedia page, take only the words around the reference within the window and associate these with the entity being referred to. This strategy has two very significant drawbacks. First, as the reader has probably also guessed, windowing leads to a significant loss of discriminative features since a unit of discourse can easily exceed the window length. More importantly, determining the correct window length seems crucial but not an easy task. If we set the window to be too large then we risk including noise and a small window will lead to loss of features. Indeed, in the past, various works have used



Brad Pitt

From Wikipedia, the free encyclopedia

*This article is about the actor. For the boxer, see Brad Pitt (boxer).*

**William Bradley "Brad" Pitt**[1] (born December 18, 1963) is an American actor and film producer. Pitt has received two Academy Award nominations and four Golden Globe Award nominations, winning one. He has been described as one of the world's most attractive men, a label for which he has received substantial media attention.[2][3]

Pitt began his acting career with television guest appearances, including a role on the CBS prime-time soap opera *Dallas* in 1987. He later gained recognition as the cowboy hitchhiker who seduces Geena Davis's character in the 1991 road movie *Thelma & Louise*. Pitt's first leading roles in big-budget productions came with *A River Runs Through It* (1992) and *Interview with the Vampire* (1994). He was cast opposite Anthony Hopkins in the 1994 drama *Legends of the Fall*, which earned him his first Golden Globe nomination. In 1995 he gave critically acclaimed performances in the crime thriller *Seven* and the science fiction film *12 Monkeys*, the latter securing him a Golden Globe Award for Best Supporting Actor and an Academy Award nomination. Four years later, in 1999, Pitt starred in the cult hit *Fight Club*. He then starred in the major international hit as Rusty Ryan in *Ocean's Eleven* (2001) and its sequels, *Ocean's Twelve* (2004) and *Ocean's Thirteen* (2007). His greatest commercial successes have been *Troy* (2004) and *Mr. & Mrs. Smith* (2005). Pitt received his second Academy Award nomination for his title role performance in the 2008 film *The Curious Case of Benjamin Button*.

Following a high-profile relationship with actress Gwyneth Paltrow, Pitt was married to actress Jennifer Aniston for five years. Pitt currently lives with actress Angelina Jolie in a relationship that has generated wide publicity.[4] He and Jolie have six children – Maddox, Zahara, Pax, Shiloh, Knox and Vivienne. Since beginning his relationship with Jolie, he has become increasingly involved in social issues both in the United States and internationally. Pitt owns a production company named Plan B Entertainment, whose productions include the 2007 Academy Award winning Best Picture, *The Departed*.

**Figure 1: A Wikipedia article with annotated (black ellipses) and un-annotated references (red squares).**

widely divergent window length settings, e.g., 6 [17], 10 [10] and 55 [6], besides [16] which does not specify its window length. Note that, in knowledge bases such as Wikipedia, not all references are intra-wiki links with their destinations clearly indicating the entity being referred to. In most Wikipedia pages, there also exist many references that are not part of anchor text and to make the distinction clear, we sometimes refer to these as *un-annotated references*. Going back to Figure 1, besides the black ellipses highlighting references in anchor text, there also exist many other references that refer to "Brad Pitt" only by his last name "Pitt" and these appear in plain text which we have highlighted using red rectangles. A windowing scheme, which only concentrates on annotated references, will ignore such un-annotated references since these usually do not fall within any windows as opposed to an approach that considers all text appearing in the knowledge base.

Recently, [26] proposed topic models that attempt to use all text in the knowledge base instead of just using windows. However, these models do not exploit *proximity* to learn *word-entity* associations. If we see words $w$ and $w'$ occurring in a document containing a reference $r$ to entity $e$ then intuitively speaking, we expect the word that appears closer to $r$ to be a stronger indicator of $e$. Note that, the use of windows automatically exploits proximity. Further, [26] also performs *collective disambiguation*. Collective disambiguation can be helpful when entities with similar-sounding names appear with different sets of entities. For instance, on its own a "Chris Columbus" reference may be ambiguous but if we know that "Daniel Radcliffe" also appears alongside it then we can be more certain that its the film-maker[1] who is being referred to and not the explorer. Unfortunately, for this [26] requires a category hierarchy, viz. the Wikipedia category hierarchy which being crowd-sourced is noisy and entails expensive hierarchy pruning procedures. Note that, collective disambiguation has also been utilized previously in [16], and also in [18, 9] to a certain extent.

In this paper, we achieve the best of both worlds by proposing topic models that utilize all text in the knowledge base, thus going beyond window-based approaches, and exploit proximity to learn word-entity associations, thus going beyond previously proposed topic models for entity disambiguation. Our main contribution, the *context-aware* topic model takes proximity into account by associating a distinct, non-uniform probability distribution with every word in the knowledge base. This distribution defines how likely it is to associate the word with an entity being referred to by the page to which the word belongs to. The distribution is computed by taking into account which entities appear within the same *context*, e.g., paragraph, as the word and which do not, thus taking into account word-entity proximity. Our context-aware topic model is capable of reasoning with multiple levels of contexts simultaneously. Other examples of contexts include documents and sentences. Besides utilizing words, our context-aware topic model also performs collective disambiguation [16, 9, 18] by learning *groups* of co-occurring entities from the knowledge base. Note that, we learn groups without the aid of a category hierarchy thus making the proposed models capable of learning from more general knowledge bases which may not come with a category hierarchy. However, one complication here is to estimate the number of groups and to address this we borrow from the literature on non-parametric topic models [28]. The result is an almost parameter-free model that does not need to be fed the exact number of groups as an extra parameter.

Note that originally, topic models were proposed for unsupervised learning [4]. The entity disambiguation problem however, is better posed as a supervised learning problem, i.e., learn from a knowledge base and test on unseen text. One of the defining aspects of this work is the use of different topic models for learning and testing. To be more precise, we use a *collection* of topic models to learn various useful parameters from the knowledge base and it is this collection that we refer to as the context-aware topic model. In the end, we show that our collection of topic models outperforms baselines by almost 18% with respect to disambiguation accuracy on a dataset containing almost 60,000 references spread over roughly 16,000 entities, which translates to an increment of almost 11,000 correctly disambiguated references.

The rest of the paper is organized as follows: In the next section, we begin with some notation and discuss pros and cons of adapting existing topic models for entity disambiguation. This section also serves as a primer for topic models. In Section 3 we present our main contribution, the context-aware topic model, including the model that learns groups, its non-parametric variant and the model we use to test with. In Section 4, we describe our dataset on which we perform experiments and report comparisons with previously proposed models. We conclude the paper with a description of related work in Section 5 and pointers to future work in Section 6.

[1]Chris Columbus directed the first two Harry Potter movies and Daniel Radcliffe is the name of the child actor who played Harry.

## 2. ADAPTING TOPIC MODELS FOR ENTITY DISAMBIGUATION

Let $\mathcal{D}$ denote a knowledge base and $e$ denote an entity. $\mathcal{D}$ consists of a set of documents such that each $d \in \mathcal{D}$ describes a distinct entity denoted by $e_d$. Moreover, let $E = \{e_d | d \in \mathcal{D}\}$ denote the set of entities described in $\mathcal{D}$. We now introduce the concept of an *annotated* word. One of our main goals in this paper is to learn word-entity associations. Each document $d \in \mathcal{D}$ consists of two components, the words in $d$ and out-going links connecting $d$ to other documents in $\mathcal{D}$. The anchor text associated with links emanating from $d$ is very useful since these words provide examples of word-entity associations. Let $d_1 \xrightarrow{l} d_2$ denote a link $l$ from $d_1$ to $d_2$ and let $w$ represent a word in $l$'s anchor text in $d_1$. We denote the association between $w$ and $e_{d_2}$ by saying $w$ is *annotated* with $e_{d_2}$. Also, $w_e = e_{d_2}$ denotes the entity $w$ is annotated with. We use $A_d$ to denote the set of all anchor text words present in $d$.

Consider for instance, Brad Pitt's Wikipedia page in Figure 1. This is one of many pages in the Wikipedia extract which we use to experiment with later and contains pages describing various people including sports-persons, actors, musicians and scholars. In Figure 1, we have highlighted most of the intra-wiki links that point from this page to other pages in Wikipedia in black ellipses. Thus, it is easy to associate the words "Brad", "Pitt" and "boxer" with a different Brad Pitt (who is a boxer) since the first intra-wiki link points to his page. Besides the anchor text in this page, there is a lot of other text that is not part of any intra-wiki links. Included among these, are many references to the primary entity described in this page (these are highlighted in red rectangles). If one were to capture these un-annotated references to Brad Pitt, then one would first need to *learn* word-entity associations between anchor text and the entities they point to, and use these associations to annotate the un-annotated reference words with entities. This is where topic models and inference come in.

**Topic models** were originally proposed to capture latent *topics* in text corpora [4]. Intuitively, a topic is a collection of words that co-occur in the same documents. A topic can be represented using a multinomial distribution over vocabulary $V$. More precisely, topic $\phi$ is such that $\phi : V \rightarrow [0,1]$ so that it maps each word in the vocabulary to a probability and $\sum_{w \in V} \phi(w) = 1$ so that $\phi$ forms a legal distribution. To learn a catalog for entity disambiguation, we are more interested in capturing word-entity associations and we can do this using *entity-specific* topics. In most of the topic models introduced in this paper, we will use $\phi_e$ to denote the entity-specific topic for entity $e$ and the words which $\phi_e$ maps to high probabilities are precisely the words $e$ is associated with.

**Gibbs sampling** (GS) [12] belongs to an accurate class of approximate inference algorithms known as Markov chain Monte Carlo (MCMC). Moreover, GS is also the fastest MCMC technique known and very easy to set up for topic models. The crucial step in GS is the *sampling step*, which defines how to sample for a word's annotation given the annotations for the rest of the corpus. The structure of GS consists of outer loops wherein we go over the whole corpus sampling for every word's annotation. Having performed an adequate number of these (burn-in), GS converges. A joint sample is a snapshot where every word has been annotated using which we can compute entity-specific topics $\phi_e$, $\forall e \in E$. We refer to this as the *$\phi$ computation step*. After burn-in, we collect a sufficient number of joint samples and average the value of $\phi_e$ across these.

### 2.1 LDA and its Shortcomings

Numerous topic models have been proposed to date and some of these can be adapted for entity disambiguation. In what follows, we adapt one of the earliest topic models, viz. latent Dirichlet allocation (LDA) [4], to learn entity-specific topics. Subsequently, to illustrate the shortcomings of LDA we present a better performing topic model and improve on this model in the next section.

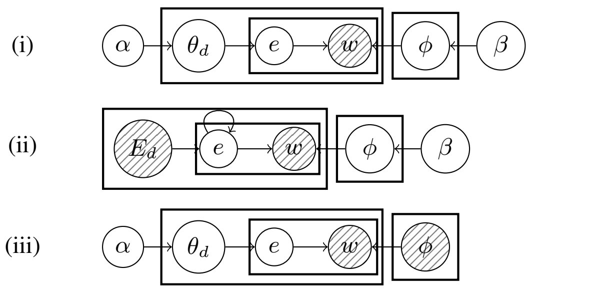


**Figure 2: Plate models for (i) LDA (ii) Contiguous-Entity model and (iii) the testing model.**

Perhaps the easiest way to understand LDA is via its generative semantics. Figure 2 (i) shows LDA's plate model representation [7] from which it is easy to read off the generative semantics. In a plate model, unshaded and shaded nodes represent unobserved and observed random variables, respectively, while boxes represent repetitions. Let $E_d = \{e_d\} \cup \{w_e | w \in A_d\}$, represent the set of relevant entities either described in or linked to from document $d$. In Figure 2 (i), $\theta_d$ represents a multinomial distribution over $E_d$ sampled from a Dirichlet distribution ($\mathtt{Dir}$) whose parameters are given by $\alpha$. More specifically, $\theta_d : E_d \rightarrow [0,1]$ and $\forall e \in E_d$, $\theta_d(e)$ is the probability of a random word in $d$ to be associated with $e \in E_d$[2]. The box enclosing $\theta_d$ represents a loop over all $d \in \mathcal{D}$. The next box represents a loop over words $w \in d$. Essentially, to generate a word $w$ we need to first choose an annotation $e$ with probability $\theta_d(e)$ multiplied by the probability for generating $w$ from $e$'s entity-specific topic $\phi_e(w)$. Note that, since the words in the knowledge base's documents comprise our observations, $w$ is a shaded node in Figure 2 (i). The last box indicates that for each $e \in E$ there is a distinct $\phi_e$ and $\mathtt{Dir}(\beta)$ is their common prior.

Having discussed LDA's generative semantics, we now need to learn $\phi_e$ and for this we need an inference procedure. To derive Gibbs sampling (GS) for any topic model, one usually needs to go through a tedious (but straightforward) derivation which has been well illustrated in prior works [15, 13]. For the sake of brevity, we refrain from delving into such details in this paper. Algorithm 1 describes GS for LDA by specifying its sampling and $\phi$ computation steps, respectively. In Algorithm 1, $\sim$ denotes "sample". Thus, the sampling step defines how to sample an annotating entity $e$ for $w$ and the probability of choosing $e$ is defined in terms of counts. Whenever a count variable has a superscript "$-$", that means we discount the current word from the count. Thus, when sampling for $w$'s annotation, $n_e^- = \sum_{d \in \mathcal{D}} \sum_{w' \neq w, w' \in d} \delta(w'_e = e)$ whereas $n_e = \sum_{d \in \mathcal{D}} \sum_{w' \in d} \delta(w'_e = e)$ where $\delta(\mathtt{pred})$ is 1 iff $\mathtt{pred}$ holds else is 0. Similarly, $n_{d,e}^- = \sum_{w' \in d, w' \neq w} \delta(w'_e = e)$ and $n_{w,e} = \sum_{d \in \mathcal{D}} \sum_{w' \in d} \delta(w'_e = e) \delta(\mathtt{term}(w') = \mathtt{term}(w))$ where $\mathtt{term}(w)$ returns the term from the vocabulary that $w$ is an instance of. For example, if $w$ is an instance of the word "graph" then $n_{w,e}$ would count the number of instances of "graph" in the whole knowledge base that were annotated with entity $e$. Note that, we never sample for anchor words, their annotations are "clamped" to the link's destination entity. Also, GS is bootstrapped by assign-

[2]Note that, $\forall e \notin E_d : \theta_d(e) = 0$ and this follows from Wikipedia's manual of style [1] that mandates any entity ever referred to in $d$ should either be $e_d$ or be the destination of some intra-wiki link emanating from $d$.

```
Algorithm 1: Gibbs Sampling for LDA
  Sampling step:
      $w_e \sim (\alpha + n^{-}_{d,e}) \frac{\beta + n^{-}_{w,e}}{|V|\beta + n^{-}_{e}}, \ \forall d \in \mathcal{D}, \forall w \in d, w \notin A_d$
  $\phi$ computation:
      $\phi_e(w) \propto \beta + n_{w,e}, \ \sum_{w \in V} \phi_e(w) = 1, \ \forall e \in E$
```

```
Algorithm 2: Gibbs sampling for CE
  Sampling step:
      $w_e \sim (\delta(e = w^{\leftarrow}_{e}) + \frac{1}{|E_d|}) \frac{\beta + n^{-}_{w,e}}{|V|\beta + n^{-}_{e}}, \ \forall d \in \mathcal{D}, \forall w \in d, w \notin A_d$
  $\phi$ computation:
      $\phi_e(w) \propto \beta + n_{w,e}, \ \sum_{w \in V} \phi_e(w) = 1, \ \forall e \in E$
```

```
Algorithm 3: Gibbs Sampling for testing
  Sampling step:
      $w_e \sim (n^{-}_{d,e} + \alpha)\phi_e(w), \ \forall d \in \mathcal{D}_{\text{test}}, \forall w \in d$
```

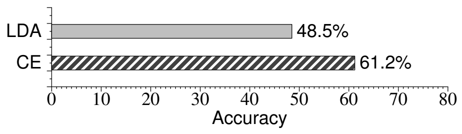


**Figure 3: Gibbs sampling algorithms and results of the preliminary experiment.**

ing all non-anchor $w \in d$ to entities in $E_d$ uniformly at random. A number of techniques are available to set hyperparameters $\alpha$ and $\beta$, and for all the models in this paper we use Minka's updates [19].

One of the shortcomings of LDA is that it does not take into account proximity of links and words. For instance in Figure 1 intuitively speaking, one would expect it to be much more likely that the word "lives" is associated with the entity "Angelina Jolie" as opposed to the entity "Anthony Hopkins", since he appears in a different paragraph. LDA considers the document to be a bag of words and uses the same probabilities for annotating words no matter where the word lies with respect to links. To rectify this, we introduce the contiguous entity model (CE) in Figure 2 (ii) that takes proximity into account. CE assumes a document is a *sequence* and flips an unbiased coin to annotated a word. If the coin falls heads then we sample an annotating entity uniformly at random from $E_d$, but if the coin falls tails then we choose the annotation of the previous word that appears in $d$. In terms of probabilities, $\Pr(w_e) = \frac{1}{2}\frac{1}{|E_d|}\delta(w_e = e) + \frac{1}{2}\delta(w_e = w^{\leftarrow}_{e}), \ \forall e \in E_d$, where $w^{\leftarrow}_{e}$ is the annotation of the previous word. $\because \ w^{\leftarrow}_{e} \in E_d, \ \Pr(w_e) = \frac{|E_d|+1}{2|E_d|}\delta(w_e = w^{\leftarrow}_{e}) + \frac{1}{2|E_d|}\delta(w_e = e), \ \forall e \in E_d \setminus \{w^{\leftarrow}_{e}\}$. The conditioning on the previous word's annotation is depicted by a self-loop on $e$ in Figure 2 (ii). GS for CE is shown in Algorithm 2, where the sampling step is $|E_d| + 1$ times more likely to choose the previous word's annotation a priori. Note that, the notion of a previous word does not hold for the first word in document $d$, and we annotate this with $e_d$ since the first word in a Wikipedia page is usually the first name of the entity being described.

We now proceed towards comparing the two models introduced so far and for this we will require a topic model (shown in Figure 2 (iii)) that can use the learnt entity-specific topics to disambiguate references in unseen data. As the reader may have already noticed, it resembles LDA in every way except that $\phi$ is shaded since in this case these have already been learnt. Further, Algorithm 3 depicts GS for testing and here we only describe the sampling step. Note that, $\alpha$ in Algorithm 3 is a smoothing term, one can replace it with any small number while testing or with the final learnt value of hyperparameter $\alpha$ learnt by running LDA (Figure 2 (i)).

## 2.2 Comparison

We assume that the held-out test corpus $\mathcal{D}_{\text{test}}$ consists of documents containing running text (and no links). We also assume that, the references to entities in the test documents have been located by using a named-entity recognition tool. For each reference, our task is to *identify* the correct entity from the knowledge base and for this purpose, we utilize the learnt entity-specific topics in conjunction with the testing topic model. The metric we report is the percentage of correctly disambiguated references. For references consisting of multiple words, we require that a majority of the annotations belong to the ground truth entity for it to be adjudged a correctly disambiguated reference.

The dataset we use for our experiment is an extract of Wikipedia describing people and is described in full detail in Section 4. Essentially, we downloaded the html source for this extract, stripped out all the html code keeping only the words and the intra-wiki links, and partitioned each page into two parts. We construct a training corpus comprising all the first parts of the pages and present this to either LDA or CE to learn entity-specific topics from. In this training phase, the intra-wiki links provide entity-word asscociations to learn from. While testing, we present the second parts of the pages to the testing model along with a learnt set of $\phi_e$s. In this phase, we do not provide the intra-wiki links to the testing model but use these as ground truth annotations instead.

The bar graph in Figure 3 shows the results. CE outperforms LDA by $\approx 13\%$ and this is especially surprising given that LDA has more parameters, $\alpha, \beta$ as opposed to CE's $\beta$. We also compared against the author-topic model (AT) [24], which was originally proposed to model collaboratively written documents such as scientific publications. To adapt AT for entity disambiguation, we simply replace authors with entities. AT, like LDA, does not handle proximity. Moreover (adapted) AT posits that, an entity is a distribution over latent topics and topics are distributions over words. Nevertheless, it is possible to recover $\phi_e$s given the two learnt (sets of) distributions. Unfortunately, this results in a severely loss of sparsity which implies that almost every entity can now produce every word in the vocabulary. This drops disambiguation accuracy on held-out documents to almost $0\%$.

From these experiments, it seems clear that the best topic models for entity disambiguation should 1) account for proximity while learning and 2) should directly model $\phi_e$ to maintain sparsity. Neither LDA nor AT do both of these. In the next section, we develop on CE's rudimentary handling of proximity to produce improved models besides adding a group learning component to it to enable collective disambiguation.

## 3. COLLECTIVE CONTEXT-AWARE TOPIC MODELS

We begin by revisiting the Contiguous-Entity model from the last section. Currently, this model only takes into account one level of context, i.e. the previous word's entity. In reality however, units of discourse can easily span larger syntactic units such as sentences and paragraphs. We next present a general model, wherein *multiple levels of context* can be included and it is left upto the topic model's inference mechanism to optimally choose among them, so that coherent entity-specific topics can be learnt from the knowledge base. Following that, we present group-learning topic models to exploit collective disambiguation.

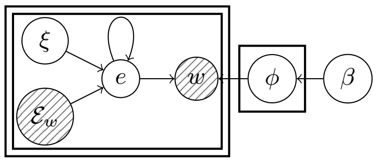


**Figure 4: Context-aware topic model**

```
Algorithm 4: Gibbs sampling for CA
  Sampling step:
    $w_e \sim \frac{\beta+n^-_{w,e}}{|V|\beta+n^-_e}\sum_\xi \frac{\delta(e\in\mathcal{E}_{w\xi})}{|\mathcal{E}_{w\xi}|},\ \forall d\in\mathcal{D},\forall w\in d, w\notin A_d$
  $\phi$ computation:
    $\phi_e(w)\propto\beta+n_{w,e},\ \sum_{w\in V}\phi_e(w)=1,\ \forall e\in E$
```

## 3.1 Context-Aware Topic Models

Recall that, in the CE model we use a coin flip to choose between the entity of the previous word and any entity from the set $E_d$. To extend this process, we will need the following: the idea of a multi-headed coin and the concept of a *context*. Intuitively speaking, a *context* corresponds to a syntactic unit of discourse such as a paragraph or a sentence. For the purposes of learning word-entity associations however, we are interested in connecting text to entities. Thus it is useful to note that, given a document in a knowledge base and a word in it, it is straightforward to collect all the entities corresponding to a particular context for that word. Consider for instance, the paragraph context. We first locate the paragraph the word is present in, and then collect all anchor text from the paragraph and the destination entities they point to. This set of entities pointed to from the paragraph forms the paragraph context for the word. Sentence contexts can be defined similarly. Thus, for the purposes of this paper, a context for a word is denoted by a set of entities. Note that, for $w\in d$, we always include $e_d$ in the context no matter which context (sentence or paragraph) it is. It should be clear now that $E_d$, as was defined in the last section, represents the document-level context.

Let $c$ denote a context, examples are the whole document, paragraphs, sentences and previous word. Let $\mathcal{C}$ denote a set of contexts we are interested in modeling. For instance, $\mathcal{C}=\{\text{doc},\text{prev}\}$ denotes the document context and the previous word which is exactly what the CE model takes into account. For our experiments that we report in Section 4, we set $\mathcal{C}=\{\text{doc},\text{para},\text{prev}\}$. Given such a set of contexts, one can define the *Context-Aware* topic model (CA) whose plate model is shown in Figure 4. The generative semantics of CA are as follows. In Figure 4, $\xi$ represents an unbiased multi-headed coin with $|\mathcal{C}|$ heads. Everytime we want to annotate a word in the knowledge base $\mathcal{D}$, we flip $\xi$ to choose a context. In Figure 4, $\mathcal{E}_w$ denotes a hyperset of entities such that $\forall c\in\mathcal{C},\ \mathcal{E}_{wc}\in\mathcal{E}_w$ contains the entities pertaining to context $c$ for word $w$. The remainder of the generative process is thus, to choose an entity $e$ uniformly at random from $\mathcal{E}_{w\xi}$ and generate $w$ from $e$ using $\phi_e$. Note that, the self-loop on $e$ in Figure 4 is included only if prev is included in $\mathcal{C}$.

It would be cumbersome to keep track of various counts associated with each context. Thus, to design a simple and efficient Gibbs sampling procedure, we prefer to marginalize over the $\xi$ variable and eliminate it from our analysis. First note that, all the contexts we have mentioned until now share a convenient containment relationship in that each context is always contained in another context (e.g., the sentence the word is in is contained in the paragraph and so on). GS is possible even without this relationship, but is more complicated and not assuming this relationship may have implications on how fast GS mixes and converges. Our goal is now to generate a distribution that lets us choose entities from, to annotate word $w\in d$:

$$\Pr(e)=\sum_\xi \Pr(\xi)\Pr(e|\mathcal{E}_{w\xi})=\sum_\xi\frac{1}{|\mathcal{C}|}\frac{\delta(e\in\mathcal{E}_{w\xi})}{|\mathcal{E}_{w\xi}|}\propto\sum_\xi\frac{\delta(e\in\mathcal{E}_{w\xi})}{|\mathcal{E}_{w\xi}|}$$

which means that, to compute the likelihood of choosing $e$ we simply need to know which contexts of $w$ $e$ is present in and what the sizes of those contexts are. For hierarchical contexts, we simply need to know which is the smallest context $e$ is present in and we can then compute its probability. Of course, this probability needs to be combined with $e$'s likelihood of generating $w$ and Algorithm 4 describes GS for CA.

## 3.2 Learning Groups of Entities

Until now, we have concentrated on learning entity-specific topics from the knowledge base. We would now like to shift focus towards performing collective disambiguation and for this we will also need to learn groups of co-occurring entities from the knowledge base. The knowledge base $\mathcal{D}$ provides ample opportunities to learn groups. For instance, just the fact that the same Wikipedia page refers to multiple entities through intra-wiki links can be used to learn groups, or we could first run any of the entity-specific topic learning models we have introduced so far to annotate all words in the knowledge base and subsequently, run the group-learning topic model we will introduce shortly to learn groups. However, the second approach risks transferring errors in annotations to errors in the learnt groups and can also be inefficient simply because we are utilizing a much larger set of annotations. Thus, we take the former approach.

We represent groups the same way we represent entity-specific topics, i.e., using multinomial distributions. Let $\lambda$ represent a group such that $\lambda: E\rightarrow[0,1]$ and $\sum_{e\in E}\lambda(e)=1$ where once again, $E$ is the set of entities described in $\mathcal{D}$. Thus, the entities mapped to (relatively) larger values by $\lambda$ are precisely the ones which are part of this group. As often happens, since we will be describing topic models to learn groups, $\lambda$s we learn will usually end up being sparse assuming we model these directly through our topic model. One issue we will need to address later is, how many groups we wish to learn from $\mathcal{D}$? This question will be more clearly answered in the next subsection; for now we assume the user has a good idea of what this number is and feeds it as a parameter $G$.

Figure 5 describes the plate model of a group-learning topic model. In this plate model, the notion of document $d$ is slightly different. As we discussed earlier, we will be using the natural grouping of intra-wiki links in Wikipedia to learn groups. More pre-

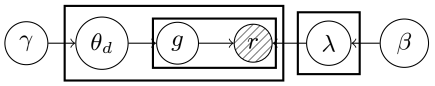


**Figure 5: Group-learning topic model**

```
Algorithm 5: Gibbs Sampling for learning groups
  Sampling step:
    $r_g \sim (\gamma_g+n^-_{d,g})\frac{\beta+n^-_{\text{dest}(r),g}}{|E|\beta+n^-_g},\ \forall d\in\mathcal{D},\forall r\in d$
  $\phi$ computation:
    $\lambda_g(e)\propto\beta+n_{e,g},\ \sum_{e\in E}\lambda_g(e)=1,\ \forall g=1\dots G$
```

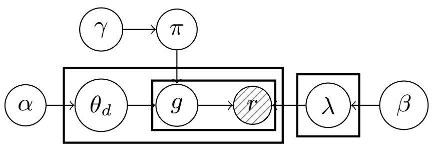


**Figure 6: Non-parametric group-learning topic model**

```
Algorithm 6: Gibbs sampling for learning groups (non-param.)
  Sampling step: $\forall d \in \mathcal{D}, \forall r \in d$ do
      $r_g \sim \begin{cases} n^{-}_{d,g} \frac{\beta + n^{-}_{\text{dest}(r),g}}{|E|\beta + n^{-}_g} & \text{if } g \in G_d \\ \frac{m^{-}_g + \gamma_g}{\sum_g m^{-}_g + \sum_g \gamma_g} \alpha \frac{\beta + n^{-}_{\text{dest}(r),g}}{|E|\beta + n^{-}_g} & \text{if } g \notin G_d \end{cases}$
  $\phi$ computation:
      $\lambda_g(e) \propto \beta + n_{e,g},\ \sum_{e \in E} \lambda_g(e) = 1,\ \text{if } \exists r \text{ s.t. } r_e = g$
```

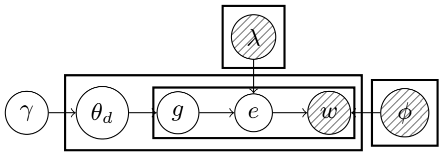


**Figure 7: Collective entity disambiguation model**

```
Algorithm 7: Gibbs Sampling with $\lambda_g$s and $\phi_e$s
  Sampling step:
      $w_e \leftarrow \langle g, e\rangle \sim \lambda_g(e)\phi_e(w)(n^{-}_{d,g} + \gamma_g),\ \forall d \in \mathcal{D}_{\text{test}}, \forall w \in d$
```

cisely, given a document from $\mathcal{D}$, we remove all words and simply collect all the destination entities of the links appearing in this document and form a set out of these entities. This set of entities and the entity described in this document (essentially recreating $E_d$) is what forms a document $d$ for group learning purposes and each entity mentioned in it is essentially a reference $r$. The plate model described in Figure 5 is essentially LDA run in such a setup. The generative model is thus as follows. To generate $d$, we first generate a distribution over groups $1 \ldots G$ referring to it by $\theta_d$. To generate each reference $r \in d$, we simply choose a group $g$ using $\texttt{mult}(\theta_d)$ and then generate the entity $e$ represented by $r$ from $\lambda_g(e)$. $\texttt{Dir}(\beta)$ represents the common prior for all $\lambda_g,\ g = 1 \ldots G$. GS for this model is shown in Algorithm 5 which uses counts analogous to Algorithm 1 where we described GS for LDA. Here, dest($r$) denotes the entity $r$ represents and $r_g$ denotes the group $r$ has been annotated with. Thus, $n_{e,g}$ is the number of times a reference pointing to $e$ has been assigned group $g$ in the corpus, $n_g$ is the number of times $g$ has been used to annotate any reference in the corpus and $n^{-}_{d,g}$ is the number of times references in $d$ have been annotated with $g$ not counting the current reference for which this sampling step is being performed. $\gamma$ and $\beta$ are familiar hyperparameters. In this case, $\gamma$ is to be treated as a vector with $\gamma_g$ denoting the relative preference for using group $g$ to annotate with. Both $\gamma$ and $\beta$ can be learnt using Minka's updates [19] just as in the previous models.

## 3.3 Learning the Number of Groups

The main issue with our group-learning topic model is how to set $G$. In the recent past, there have been proposals for non-parametric variants of LDA wherein the model automatically determines the number of topics required [28]. Fortunately, since the group-learning model we introduced earlier resembles LDA these techniques are directly applicable. Unfortunately, they are difficult to control. When we tried them on our extract of Wikipedia, these either grossly under-estimated or over-estimated the number of groups. Motivated by this and by practical considerations, we now present a version of a non-parametric group-learning topic model that is easy to apply and does not depend overly on the user to specify the correct number of groups.

Notice that, even though the user may not know the exact, correct number of groups of entities in the knowledge base, s/he may still be able to specify a loose upper-bound on $G$. The topic model itself can then choose how many groups it needs to fit the data well. For instance, for our experiments in Section 4, we used an upper-bound of roughly $|E|/4$ groups to produce good results. In a knowledge base with about $16,500$ entities, one can hardly expect more than 5000 groups (which is possible but one hopes unlikely).

In the model we present next, the assumption is that the user has specified the upper-bound over the number of groups using a parameter $\hat{G}$ and this model has close connections to the work presented in [28]. Figure 6 shows the plate model and the generative assumptions are as follows. Besides the normal LDA generative process, we will also need a global prior distribution over groups. Let $\pi$ denote a multinomial distribution such that $\pi : 1 \ldots \hat{G} \rightarrow [0,1]$ and $\sum_{g=1}^{\hat{G}} \pi(g) = 1$. We need $\pi$ because we would like to choose groups on demand as and when required by a document. Essentially, the generative process now changes to as follows. We first generate $\lambda_g \sim \texttt{Dir}(\beta),\ \forall g = 1 \ldots \hat{G}$ and $\pi \sim \texttt{Dir}(\gamma)$. Just as in LDA, we also generate for $\theta_d \sim \texttt{Dir}(\alpha),\ \forall d \in \mathcal{D}$. For each reference $r \in d$, we now choose $g$ using the following strategy:

$$g \sim \begin{cases} \texttt{mult}(\theta_d) & \text{if } g \in G_d \\ \texttt{mult}(\pi) & \text{o.w.} \end{cases}$$

where $G_d$ denotes the set of groups currently used to annotate references in $d$. Finally, we generate dest($r$) from $\lambda_g$.

In our earlier group learning topic model (Algorithm 5), one factor that goes into deciding which group $g$ is used to annotate reference $r$ is how many times $g$ has been used to annotate other references pointing to dest($r$) in the corpus. This is in fact, the numerator in the second term in the sampling step in Algorithm 5. This leads to a clustering effect where the same group gets chosen to annotate references pointing to $e$. In the modified model we just presented, such a clustering happens at a document level too where, when we include $g$ to annotate a reference in document $d$ for the first time one is likely to choose a group that is used relatively more often in the other documents in the corpus. This clustering effect keeps some degree of control over how many groups we end up using from $\hat{G}$ and it is usually the case, that the total number of groups used to annotate references with in the corpus is much smaller than $\hat{G}$. In the end, we only compute $\lambda$s for groups that have been used to annotate at least one reference in the corpus. GS for the above model is described in Algorithm 6 where the only new count variable is $m^{-}_g$ which is the number of documents (discounting the one whose reference we are sampling for) where group $g$ has been used to annotated references. In the sampling step, the first clause gets invoked if $g$ is already in use in $d$, otherwise we invoke the second clause involving hyperparameter $\alpha$ and the new count $m^{-}_g$. The derivation for this sampling step is described in Appendix A, which also discusses how to set the hyperparameters $\alpha, \beta, \gamma$.

## 3.4 Collective Entity Disambiguation

Having defined the topic models we need to learn entity-specific topics and groups from the knowledge base $\mathcal{D}$, we now need a topic model that can disambiguate references from running text using these learnt quantities. Figure 7 describes the plate model for such a topic model and Algorithm 7 describes its GS inference. The generative assumptions are as follows. We assume each unseen test document $d$ is a distribution over groups and generate a distribution $\theta_d \sim \mathtt{Dir}(\gamma)$. Subsequently, to annotate each word in $d$ we first choose a group $g \sim \mathtt{mult}(\theta_d)$, an entity $e$ from $g$ $e \sim \mathtt{mult}(\lambda_g)$ using the learnt groups and finally, we generate the word using the entity-specific topics $w \sim \mathtt{mult}(\phi_e)$. The GS (Algorithm 7) for this model utilizes a set of learnt $\lambda_g$,$\phi_e$ and assigns to each word $w$ a $\langle g, e\rangle$ pair consisting of a group and an entity. Even though, we only check whether we assigned the correct entity to the words in the reference, we need the group assignments to the words to estimate $n^{-}_{d,g}$ which is present in the last term of the sampling step. Hence, the assignment of a group-entity pair instead of annotating words with just entities. Note that, both group-learning models described in this section also learn $\gamma$ hyperparameters, which is a vector with $\gamma_g$ denoting the relative preference for group $g$, and the same $\gamma$ is required to run GS in Algorithm 7.

## 3.5 Further Details

The Gibbs sampling inference algorithms we have described so far are fairly simple and can be implemented in a straightforward manner using a few nested loops. However, to achieve an efficient implementation one needs to keep a few things in mind such as caching counts, i.e., $n_{d,e}, n_{w,e}, n_e, n_{d,g}, n_{e,g}, n_g$ and $m_g$, in appropriately chosen data structures (hashtables or integer arrays) for quick lookup. Moreover, note that it may not be possible to store all entity-specific topics, i.e, $\phi_e$, $\forall e \in E$, in a double matrix. $|E|\times|V|$ is usually too large. This is where we need to exploit sparsity in $\phi_e$, which usually holds in the case of $\phi_e$s learnt with topic models. Further optimizations for speeding up topic models such as clever re-ordering of entities while GS is being performed have also been explored in prior works [23].

## 4. EXPERIMENTS

In this section, we empirically compare the proposed approaches. We begin by describing the dataset and the approaches we compare against, followed by reporting on their performance.

## 4.1 The Dataset

The dataset we use for our experiments is an extract of Wikipedia describing people and includes sportspeople (basketball, football, baseball and tennis players), actors, musicians and scholars. We downloaded the source of these webpages and then stripped them of their html code leaving only the words and intra-wiki links. The final corpus contained 16,548 pages with each page describing a different entity interconnected by 59,161 intra-wiki links with a vocabulary size of 221,782 and a total of 9,958,728 words in the corpus. For our experiments, we required disjoint training and test sets so that we could learn entity-specific topics and groups from the former and test on the latter. To ensure that we faced no unknown entities while testing, we split each Wikipedia page horizontally and performed four-fold cross validation. More specifically, from each page we separate out the first paragraph and divide the remaining paragraphs into four roughly equal (disjoint) parts. The $i^{th}$ part from each page then forms part of the test set while the remaining three parts along with the first paragraph forms the training set. This is repeated for $i = 1 \ldots 4$ and the numbers we report are averages over all four folds of cross validation. The metric we report is disambiguation accuracy, defined as the fraction of references correctly disambiguated expressed as a percentage.

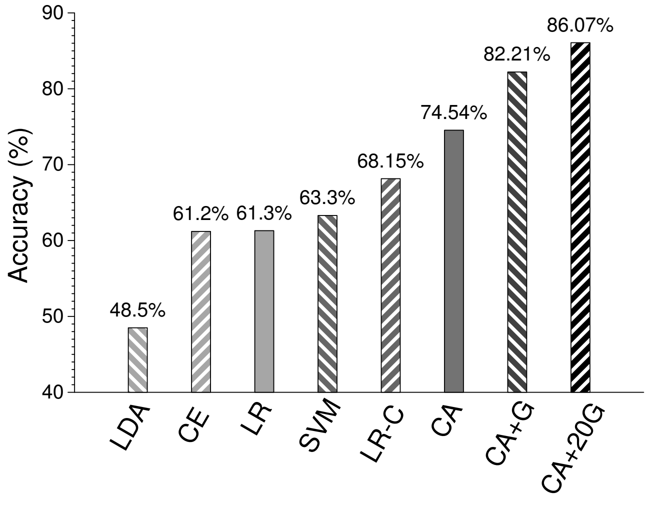


**Figure 8: Performance against discriminative baselines.**

## 4.2 Algorithms Compared

Of the baseline approaches we consider, two are based on logistic regression and a third one is the SVM-based approach proposed in [16]. To adapt discriminative approaches, [16] uses a windowing scheme where references are first extracted from the text that needs to be disambiguated. A window of pre-specified number of words is extracted from around each reference. This window (set to 20 for our experiments) of words is then compared against a candidate entity's descriptive page in the knowledge base and converted into a high-dimensional feature vector. A suitably trained SVM is then used to assign this reference-candidate entity pair a score. Finally, the highest scoring candidate entity is deemed the disambiguation of the reference. [16] also proposes performing collective disambiguation by looking at all the references in the test document and assigning them entities so that their *relatedness* [18] improves. However, when we tried this approach the SVM (libsvm [8]) became too slow to be viable[3]. Instead, we used a scalable implementation of logistic regression [22] to implement this technique. In our experiments, we denote the content-only SVM-based approach of [16] by "SVM", its content-only logistic regression couterpart by "LR" and collective logistic regression-based approach by "LR-C".

Of the topic models proposed in this paper, we (re-)report results of LDA and CE tested with the testing topic model from Section 2, CA and combinations of CA and the group-learning models tested with the testing topic model proposed in Section 3. We denote the combination of CA and the non-parametric group-learning model by "CA+G", and CA with the group-learning model that requires specifying the number of groups by "CA+$k$G", where $k$ is the specified group count.

For all the Gibbs sampling procedures we run, we use 1000 outer iterations for burn-in and subsequently collect 10 samples, each separated by 100 outer iterations so that consecutive samples are not dependent. All our experiments were run on a dual core 2.4GHz Xeon processor machine with 16GB RAM. Our code is implemented in JAVA.

## 4.3 Performance

Figure 8 charts the performanace of the various approaches. Even though LDA and CE perform the poorest, CE still manages to perform as well as LR perhaps indicating that the windowing-based

[3]Took over two days to train.

approach of adapting discriminative models may be losing out on useful features that fall outside the window. Next comes SVM and it is interesting to see that SVM performs better than LR perhaps indicating that SVMs are better classifiers than logistic regression, although this may very well be a statement about the particular logistic regression implementation we were using [22]. After that is the collective approach of using logistic regression combined with the relatedness measure which improves upon SVM's performance by about 5%. After that, comes the topic models we proposed in the previous section. CA and CA+G (the non-parametric version) stand at approximately 75% and 82%, showing that incorporating collective disambiguation provides a boost of almost 8%. Finally, we also report the best result of learning groups and combining it with CA by specifying the number of groups. As it turns out, the best result was obtained at the setting $G = 20$ and CA+20G obtains a disambiguation accuracy of more than 86% which is 23% higher than the best content-only windowing approach (SVM) and 18% higher than the discriminative collective technique (LR-C).

We also report disambiguation accuracies obtained by CA+$k$G for varying $k$ in Figure 9. In this plot, the x-axis shows $k$ (in logscale), y-axis depicts disambiguation accuracy obtained and the horizontal blue line shows the accuracy of CA+G. Clearly, there is an optimal number of groups but setting it higher than this number reduces accuracy fairly gradually. From Figure 8, we know that CA+G achieves 3.86% less than the best fixed count group-learning topic model, that and the fact that CA+G beats all the other approaches implies CA+G's performance is quite satisfactory. Although, this also shows that if we had more knowledge about how many groups there were in the knowledge base then we could gain an additional boost.

We looked further into the performance of CA+G to see how well it was identifying groups and across the four-fold cross validation from each of the different training sets it identified 777, 773, 756 and 780 groups, respectively, which is an average of 771.5 groups with a standard deviation of 1.2%, a fairly stable estimate. This seems very different from the optimal number of groups determined by CA+$k$G found on the basis of optimal disambiguation accuracy. However on further investigation, we found out that most of 700+ groups determined by CA+G comprised groups formed out of entities for whom we did not have too many intra-wiki links pointing to them in our extract of Wikipedia, thus indicating that these might be fleeting or unreliable groups anyway.

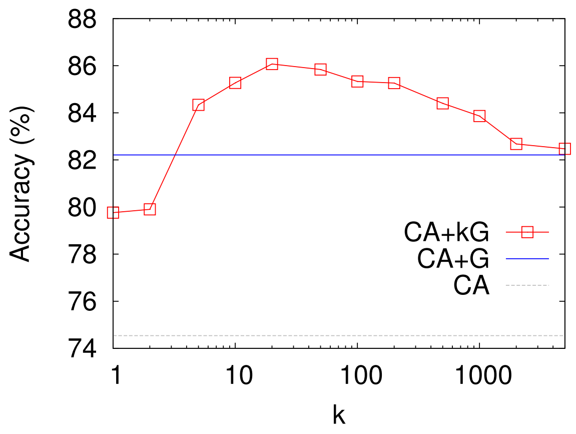


**Figure 9: Results at varying number of groups.**

## 4.4 Topics and Groups

Finally, we also report some of the entity-specific topics and groups learnt by the models proposed in this paper. Figure 10 shows a few of the groups discovered by listing the top-5 entities in order of decreasing membership probabilities. Most of these intuitively make sense. Note that, Albert Einstein is a member of two separate groups of physicists one of which was interested in measuring the speed of light and the other in the nature of light. The actors group contains both actors who played the character of Albus Dumbledore in the Harry Potter movies (Richard Harris followed by Michael Gambon). Similarly, Figure 11 shows the top-11 most probable words in a few of the topics discovered. Notice how Albert Einstein's topic includes "relativity", Alan Turing's topic includes "Bletchley" (of Bletchley Park), Paul Erdos's includes "graph" and the previously mentioned actor Richard Harris's topic includes "Dumbledore".

## 4.5 Discussion

In this section, we reported on the disambiguation accuracies obtained by the topic models proposed in this paper and how their performance compared with state-of-the-art baselines from the literature. We showed that even without any special knowledge and armed with just a knowledge base it is possible to obtain an improvement of about 14% in disambiguation accuracy over the best performing baseline (CA+G vs. LR-C). We also showed that if we had a good idea of certain parameters (e.g., the number of groups present in the knowledge base) we could approach improvements of close to 18% compared to the best performing baselines (CA+20G vs. LR-C). In absolute terms also, the disambiguation accuracies obtained by CA+G (82.21%) and CA+20G (86.07%) are quite respectable especially considering the simplicity of all the Gibbs sampling procedures reported in the paper and how easy they are to implement.

Performance in terms of accuracies however, form a story only half told. What also matters is the time taken to run these procedures. All our topic models (CE, CA, CA+G, CA+$k$G) run in the 8-10 hour time range on the dataset used for our experiments, making all of these models viable offline techniques. The question obviously is, how much time would these models take to learn from a larger knowledge base that describes more than the 16,548 entities that we were dealing with. Our topic models scale between linearly and quadratically with the size of the dataset (in terms of the number of entities), and this is where one would want to utilize faster inference techniques such as the distributed ones proposed for topic models in [21, 27] or the variational inference techniques evaluated in [2]. Of these two options, distributed algorithms for topic models is still a topic of active research and currently these algorithms only work for LDA, to the best of our knowledge. Different topic models (such as the ones described in this paper), will likely require new strategies to run inference in a distributed setting.

## 5. RELATED WORK

Most of the prior work from entity disambiguation has already been reviewed in Section 1. As mentioned in the introduction, [26] is the only prior work that proposes using topic models. However, to implement collective disambiguation they attempt to learn from Wikipedia's category hierarchy which is crowd-sourced thus necessitating pruning. In the end, [26] states that pruning results in a three-level hierarchy whose bottom-most level contains entities and topmost level contains one root. The middle level contains

(Physicists: Light)

| | |
|---|---|
| Richard Feynman | 8.5% |
| **Albert Einstein** | 8.3% |
| Stephen Hawking | 8.2% |
| Robert Millikan | 5.4% |
| Albert Michelson | 5.3% |

(Physicists: Matter)

| | |
|---|---|
| **Albert Einstein** | 12% |
| Niels Bohr | 12% |
| Arnold Sommerfeld | 4.9% |
| Werner Heisenberg | 4.7% |
| Enrico Fermi | 4.0% |

(Mathematicians)

| | |
|---|---|
| Paul Erdos | 24% |
| Bela Bollobas | 6.5% |
| Timothy Gowers | 4.3% |
| Laszlo Lovasz | 2.8% |
| George Szekeres | 2.6% |

(Computer Scientists)

| | |
|---|---|
| Norbert Wiener | 15% |
| Heinz von Foerster | 7.1% |
| William Ross Ashby | 6.6% |
| Alan Turing | 5.7% |
| John von Neumann | 4.2% |

(Actors)

| | |
|---|---|
| Peter O'Toole | 16% |
| Richard Burton | 11% |
| Richard Harris | 9.3% |
| Emlyn Williams | 3.0% |
| Michael Gambon | 2.2% |

(Basketball Players)

| | |
|---|---|
| Larry Bird | 7.2% |
| Michael Jordan | 6.8% |
| Clyde Drexler | 5.0% |
| Wilt Chamberlain | 5.0% |
| Charles Barkley | 4.0% |

**Figure 10: A few of the more popular groups.**

eight non-leaf nodes which form groups. This is a very coarse hierarchy compared to what CA+G learns with groups numbering in excess of 700. It is not clear how [26]'s techniques can provide fine-grained hierarchies. Our approach of learning groups is closer to computing relatedness scores [16, 18] since relatedness is computed from Wikipedia's hyperlinks which is what we use too.

Among other works from closely related areas, word sense disambiguation [5] and entity resolution [3, 25] have also made use of topic models. The main differentiating factor between these areas and entity disambiguation is the assumption of the presence of a learnable catalog. Topic models for entity resolution and word sense disambiguation usually assume an unsupervised setting. Thus, none of [3, 5, 25] attempt to learn entity-specific topics nor groups of entities prior to disambiguating references in unseen text.

Our non-parametric group learning topic model is a simplification of hierarchical Dirichlet processes (HDP) [28] keeping in mind practical considerations. Whereas our model assumes that an upper bound on the number of groups has been specified and that these groups are available for use when a document needs them, an HDP is in theory, able to *generate* new groups on demand. We argue however that, given the number of entities in the knowledge base it is possible to specify a loose upper bound on the number of groups. Our experiments show that, even when we ran CA+G with $\hat{G} = 5000$ the model used only about 770 groups and stabilized. In our experience, HDPs are extremely difficult to control. While some parameter settings send it on a group generation spree, others do not let it generate nearly enough groups.

Our CE model is similar to topic segmentation [29, 30, 14]. The initial LDA model treated a document as a bag of words and did not model how topics transitioned in text. Later works have improved upon this by disallowing topic transitions mid-sentence [14], or by conditioning the topic of a word on the previous word [29], or by modeling n-grams [30]. It is however, difficult to see how to extend these models so that they can handle multiple levels of context simultaneously like the proposed CA model does.

## 6. CONCLUSION

In this paper, we described topic models for entity disambiguation. Whereas previous work has illustrated the effective use of topic models, those models did not incorporate aspects such as word-entity proximity which helps learn high-quality, discriminative catalogs. Our proposed topic models improve on previous work by not only incorporating proximity but also learning groups of co-occurring entities that help perform collective disambiguation

Albert Einstein

| | |
|---|---|
| einstein | 2.0% |
| theory | 0.9% |
| physics | 0.8% |
| albert | 0.7% |
| relativity | 0.7% |
| work | 0.5% |
| light | 0.4% |
| einsteins | 0.4% |
| general | 0.4% |
| time | 0.3% |
| quantum | 0.3% |

Paul Erdos

| | |
|---|---|
| erds | 3.2% |
| theory | 1.9% |
| number | 1.5% |
| paul | 1.2% |
| graph | 1.0% |
| papers | 1.0% |
| mathematician | 0.9% |
| combinatorics | 0.6% |
| mathematics | 0.6% |
| university | 0.6% |
| theorem | 0.6% |

Stephen Hawking

| | |
|---|---|
| hawking | 1.7% |
| universe | 0.8% |
| stephen | 0.7% |
| science | 0.5% |
| cosmology | 0.4% |
| cambridge | 0.4% |
| hawkings | 0.4% |
| theoretical | 0.4% |
| time | 0.3% |
| life | 0.3% |
| university | 0.3% |

Alan Turing

| | |
|---|---|
| turing | 2.7% |
| computer | 1.0% |
| alan | 0.8% |
| machine | 0.7% |
| park | 0.7% |
| work | 0.6% |
| turings | 0.6% |
| bletchley | 0.5% |
| computing | 0.5% |
| worked | 0.5% |
| war | 0.4% |

Kurt Godel

| | |
|---|---|
| gdel | 2.6% |
| logic | 1.6% |
| kurt | 1.1% |
| theory | 1.0% |
| mathematical | 1.0% |
| arithmetic | 0.9% |
| set | 0.8% |
| gdels | 0.6% |
| axioms | 0.5% |
| system | 0.5% |
| theorem | 0.4% |

Richard Harris

| | |
|---|---|
| harris | 2.2% |
| role | 0.9% |
| film | 0.8% |
| actor | 0.7% |
| play | 0.5% |
| richard | 0.5% |
| potter | 0.5% |
| adaptation | 0.4% |
| films | 0.4% |
| children | 0.3% |
| dumbledore | 0.3% |

**Figure 11: Topics of six popular people.**

without requiring the user to specify the exact number of groups. In this respect, our proposed models are non-parametric extensions of topic models where determining the correct number of groups is part of the learning process and all the remaining parameters are learnt by optimizing joint likelihood of the data. We also showed how to use two different topic models to learn interesting and useful parameters from the knowledge base. These parameters are subsequently combined, in a principled manner, using a third topic model to obtain excellent disambiguation performance on unseen text. In the end, disambiguation performance is only limited by the quality of the input knowledge base. One metric to measure the quality of a knowledge base is coverage. For a rough count, Wikipedia contains descriptions of a few million entities whereas the planet contains a few billion people. Even though we do not expect all entities to be mentioned in unstructured text, we do hope to reduce such an acute dependency on available knowledge bases in future work.

## 7. REFERENCES

[1] Wikipedia manual of style. `http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style`.

[2] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In *UAI*, 2009.

[3] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. In *SDM*, 2006.

[4] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *JMLR*, 2003.

[5] J. Boyd-Graber, D. Blei, and X. Zhu. A topic model for word sense disambiguation. In *EMNLP*, 2007.

[6] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *European Association for Computational Linguistics*, 2006.

[7] W. L. Buntine. Operations for learning with graphical models. *JAIR*, 1994.

[8] C.-C. Chang and C.-J. Lin. LIBSVM – A Library for Support Vector Machines. `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[9] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *EMNLP*, 2007.

[10] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, J. Y. Zien, J. Y. Zien. Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. *WWW*'03.

[11] I. Fellegi and A. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 1969.

[12] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *PAMI*, 1984.

[13] T. Griffiths and M. Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences*, 2004.

[14] A. Gruber, M. Rosen-Zvi, and Y. Weiss. Hidden topic markov models. In *AISTATS*, 2007.

[15] G. Heinrich. Parameter estimation for text analysis. Technical report, 2005.

[16] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *KDD*, 2009.

[17] R. Mihalcea and A. Csomai. Wikify! linking documents to encyclopedic knowledge. In *CIKM*, 2007.

[18] D. Milne and I. Witten. Learning to link with Wikipedia. In *CIKM*, 2008.

[19] T. P. Minka. Estimating a dirichlet distribution. Technical report, Microsoft Research, 2003.

[20] R. M. Neal. Bayesian mixture modeling. In *Workshop on Max. Entropy and Bayesian Methods of Stat. Analysis*, 1992.

[21] D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed algorithms for topic model. *JMLR*, 2009.

[22] A. M. Paul Komarek. Making logistic regression a core data mining tool with tr-irls. In *ICDM*, 2005.

[23] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *KDD*, 2008.

[24] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*, 2004.

[25] L. Shu, B. Long, and W. Meng. A latent topic model for complete entity resolution. In *ICDE*, 2009.

[26] S. Singh, K. Kumar, R. Rastogi, P. Sen, S. Sengamedu. Entity disambiguation with hierarchical topic models. *KDD*'11.

[27] A. Smola and S. Narayanamurthy. An architecture for parallel topic models. In *VLDB*, 2010.

[28] Y. Teh, M. Jordan, M. Beal, D. Blei. Hierarchical dirichlet processes. *J. of the American Statistical Association*, 2005.

[29] H. Wallach. Topic modeling: Beyond bag of words. In *ICML*, 2006.

[30] X. Wang, A. Mccallum, and X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *ICDM*, 2007.

## APPENDIX

## A. INFERENCE FOR GROUP DISCOVERY

To set up Gibbs sampling for the model shown in Figure 6, we need show how to sample a group for a reference given annotations to all other references in the corpus. More precisely, we need to compute for a given reference $r$, the conditional distribution $\Pr(r_e = g|\mathbf{r}^-, \mathbf{g}^-; \alpha, \beta, \gamma)$ $, \forall g \in G$, where $\mathbf{r}^-$ and $\mathbf{g}^-$ represent annotations to the rest of the references in the corpus. For this, we will need the joint probability distribution which can be simply read off the plate model (Figure 6):

$$\Pr(\mathbf{r}, \mathbf{g}, \Theta, \pi, \Lambda, ; \alpha, \beta, \gamma) \tag{1}$$

$$= \Pr(\pi|\gamma) \left[\prod_{g=1}^{G} \Pr(\lambda_g|\beta)\right] \left[\prod_{d=1}^{D} \Pr(\theta_d|\alpha) \prod_{r\in d} \Pr(g|\theta_d, \pi) \Pr(r|\lambda_g)\right]$$

$$\propto \left[\prod_{g=1}^{G} \pi_g^{m_g+\gamma_g-1}\right] \left[\prod_{\substack{g=1\\e=1}}^{G,E} \lambda_{ge}^{n_{ge}+\beta_e-1}\right] \left[\prod_{\substack{d=1\\g\in G_d}}^{D} \theta_{dg}^{n_{dg}+\alpha/|G_d|-1}\right]$$

where $G_d$ is the set of groups used by document $d$ to assign references within it to, $m_g$ is the number of documents using group $g$ across the corpus, $n_{ge}$ is the number of references which refer to entity $e$ assigned to $g$ across the corpus and $n_{dg}$ is the number of references in $d$ assigned to $g$. Note that, the last term has a $\frac{\alpha}{|G_d|}$ in the exponent because we assume the prior on $\theta_d$ is a symmetric Dirichlet prior whose parameters add upto $\alpha$.

We will now get rid of the useless parameters by integrating out $\pi$, $\Lambda$ and $\Theta$. To do this, we will need the following identities:

$$\int \prod_{g=1}^{G} \pi_g^{m_g+\gamma_g-1} d\pi = \frac{\prod_{g=1}^{G} \Gamma(m_g+\gamma_g)}{\Gamma(\sum_g m_g + \sum_g \gamma_g)} \tag{2}$$

$$\int \prod_{e=1}^{E} \lambda_{ge}^{n_{ge}+\beta_e-1} d\lambda_g = \frac{\prod_{e=1}^{E} \Gamma(n_{ge}+\beta_e)}{\Gamma(n_g + \sum_e \beta_e)} \tag{3}$$

$$\begin{aligned}\lim_{G_d\to\infty} \int \prod_{g\in G_d} \theta_{dg}^{n_{dg}+\alpha/|G_d|-1} d\theta_d &= \frac{\prod_{g\in G_d} \Gamma(n_{dg}+\frac{\alpha}{|G_d|})}{\Gamma(\sum_g n_{dg}+\alpha)} \\ &= \frac{\prod_{g\in G_d} \Gamma(n_{dg}+\frac{\alpha}{|G_d|})}{\Gamma(\sum_g n_{dg}+\alpha)} \\ \text{and finally, setting } \lim_{G_d\to\infty} &= \frac{\prod_{g\in G_d} \Gamma(n_{dg})}{\Gamma(n_d+\alpha)} \end{aligned} \tag{4}$$

where, to set the limit, we have simply set $\alpha/|G_d|$ to 0. Note that, in Equation 4, if we add up the terms $n_{dg}$ in the numerator then this differs from the denominator by exactly $\alpha$. We will later pretend that $\alpha$ is the probability mass of picking a new group from $\theta_d$ [20].

We now substitute Equations 2, 3 and 4 into Equation 1:

$$\begin{aligned}\Pr(\mathbf{r}, \mathbf{g}; \alpha, \beta, \gamma) \propto\ & \frac{\prod_{g=1}^{G} \Gamma(m_g+\gamma_g)}{\Gamma(\sum_g m_g + \sum_g \gamma_g)} \\ & \prod_{g=1}^{G} \frac{\prod_{e=1}^{E} \Gamma(n_{ge}+\beta_e)}{\Gamma(n_g+\sum_e \beta_e)} \prod_{d=1}^{D} \frac{\prod_{g\in G_d} \Gamma(n_{dg})}{\Gamma(n_d+\alpha)}\end{aligned} \tag{5}$$

Having described the joint distribution in terms of counts, to compute $\Pr(r_e = g|\mathbf{r}_-, \mathbf{g}_-; \alpha, \beta, \gamma)$ we now need to imagine a corpus without $r$. This corpus' joint distribution will be of the same form as Equation 5 with a few terms missing. If $r$ were annotated with a group $g$ already in use in the document $r$ belongs to then there are differences only in the second and third terms of Equation 5. On the other hand, if $g$ is not in use in the document $r$ belongs to, then there are differences in the first and second terms only. Dividing the joint of the full corpus by the joint of the corpus with $r$ missing gives us the conditional probability distribution we need [13]:

$$\Pr(d \ni r_e \leftarrow g|\mathbf{r}^-, \mathbf{g}^-; \alpha, \beta, \gamma)$$

$$\propto \begin{cases} \frac{n^-_{g,\text{dest}(r)}+\beta_{\text{dest(r)}}}{n^-_g+\sum_e \beta_e} n^-_{d,g} & \text{if } g \in G_d \\ \frac{n^-_{g,\text{dest}(r)}+\beta_{\text{dest}(r)}}{n^-_g+\sum_e \beta_e} \alpha \frac{m^-_g+\gamma_g}{\sum_g m^-_g+\sum_g \gamma_g} & \text{if } g \notin G_d \end{cases}$$

To recover $\Lambda$, which will be used while testing with the topic model described in Figure 7, we simply collect all $g$ which have been assigned at least one reference in the corpus and compute the corresponding multinomials describing the groups:

$$\forall g \in G \text{ s.t. } n_g > 0 : \lambda_g(e) \propto n_{g,e} + \beta_e, \ \sum_e \lambda_g(e) = 1$$

To learn the hyperparameters $\alpha, \beta$ and $\gamma$, we simply pretend that at the time of updating these the number of groups in use were the number of groups we had originally begun sampling with [20]. Thus, the group discovery process does not interfere with learning the hyperparameters and we can simply use Minka's updates [19].