

ABSTRACT

Title of thesis: **TOPIC MODELING FOR
WIKIPEDIA LINK DISAMBIGUATION**

Bradley Alan Skaggs, Master of Science, 2011

Thesis directed by: Professor Lise Getoor
Department of Computer Science

Many articles in the online encyclopedia Wikipedia have hyperlinks to ambiguous article titles. To improve the reader experience, any link to an ambiguous title should be replaced with a link to one of the unambiguous meanings. We propose a novel statistical topic model, which we refer to as the Link Text Topic Model (*LTTM*), that can suggest new link targets for existing ambiguous links in Wikipedia articles. For evaluation, we develop a method for extracting ground truth from snapshots of Wikipedia at different points in time. We evaluate *LTTM* on this ground truth, and demonstrate its superiority over existing link- and content-based approaches. Finally, we build a web service that uses *LTTM* to suggest unambiguous articles for human editors wanting to fix ambiguous links.

TOPIC MODELING FOR WIKIPEDIA LINK DISAMBIGUATION

by

Bradley Alan Skaggs

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2011

Advisory Committee:
Professor Lise Getoor, Chair/Advisor
Professor Jordan Boyd-Graber
Professor Hal Daumé III

© Copyright by
Bradley Alan Skaggs
2011

Acknowledgments

Wikipedia would not exist without its many millions of contributors. I would like to thank each of you for your part in writing humanity's greatest gift to the curious.

I would also like to thank Professor Lise Getoor and Professor Philip Resnik for their academic guidance in the long path of exploring this thesis. You both saw in this work something worth completing.

Finally, I would not have been able to write one word of this without the love, encouragement, and unflagging support of my wife Stephanie. Thank you for helping me finish!

Table of Contents

List of Figures	iv
1 Introduction	1
2 Background	3
2.1 Wikipedia	4
2.2 Word Sense Disambiguation	9
2.3 Topic Modeling	11
3 Link Text Topic Model	15
3.1 Generative Model	16
3.2 Posterior Inference	18
4 Alternate Disambiguation Techniques	20
4.1 Disambiguation Page Identification and Disambiguation Candidate Extraction	21
4.2 Disambiguation Algorithms	24
4.2.1 Baseline	24
4.2.2 Text-Based Approaches	26
4.2.2.1 Text Similarity	26
4.2.2.2 Latent Dirichlet Allocation	27
4.2.3 Link-Based Approaches	28
4.2.3.1 Random Walk with Restart	28
4.2.3.2 Link Relatedness	30
5 Evaluation and Results	31
5.1 Evaluation	32
5.2 Results	34
6 Disambiguation Web Service Implementation	35
7 Conclusions and Future Work	44
7.1 Conclusions	45
7.2 Future Work	45

List of Figures

2.1	The top of the “Apple” article at http://en.wikipedia.org/wiki/Apple links to the “Apple (disambiguation)” page.	6
2.2	The “Organ” disambiguation page, with links to disambiguation candidates, is available at http://en.wikipedia.org/wiki/Organ	6
2.3	Distribution of number of inlinks per disambiguation page in English Wikipedia	9
2.4	Plate diagram for Probabilistic Latent Semantic Analysis	13
2.5	Plate diagram for Latent Dirichlet Allocation	13
3.1	Plate diagram for the Link Text Topic Model. A prime next to a variable indicates it is used for ambiguous links.	18
4.1	The first page of the “Java (disambiguation)” article shows grouped disambiguation candidates. The whole list is viewable at http://en.wikipedia.org/wiki/Java_(disambiguation)	23
4.2	The distribution of inlinks with the text “organ” demonstrates the large class skew that can arise in disambiguation; 40 articles with a single inlink are not shown.	25
4.3	The distribution of inlinks with the text “organs” has the anatomical sense dominate rather than the musical sense; 10 articles with a single inlink are not shown.	25
5.1	An example disambiguation in English Wikipedia made between September 2010 and October 2010 shows that a link in the “Bare Wires” article has been disambiguated by an editor from “Organ” to “Organ (music)”.	34
5.2	Disambiguation-candidate-set size distribution among the 1,000 test ambiguous links in English Wikipedia	36
5.3	Accuracy of eight different disambiguation algorithms on English Wikipedia, at the top position and in the top three positions	36
5.4	Cumulative accuracy by rank of eight different disambiguation algorithms on English Wikipedia	37
5.5	Cumulative accuracy by rank for LTTM on all 36,009 disambiguated links	37
5.6	Counts of high-frequency links in four sample topics from a 1,000-topic LTTM model of Wikipedia representing Maryland, Southeast Asia, <i>The Simpsons</i> television show, and Greek mythology	38
6.1	When a user visits a Wikipedia page, the browser receives the text and extracts the links, sending them to the disambiguation server. The server returns the ambiguous links, which the browser then highlights.	40
6.2	Visiting the “Politeia (think tank)” page indicates that there is one ambiguous link on the page.	41

6.3	The changed color and border of this link indicates that it is ambiguous, and needs to be corrected.	41
6.4	When an ambiguous link is hovered over with the mouse, a set of disambiguation candidates appears, ranked by probability under the LTTM model.	41
6.5	Choosing one of the disambiguation candidates automatically creates the edit and edit summary for the change.	42
6.6	When the user's mouse hovers over an ambiguous link, the link is sent to Wikipedia to get the disambiguation candidates. They are displayed in the browser, and the link is sent to the disambiguation server, which scores suggestions and returns the scores to the browser, where they are displayed. The user clicks on a disambiguation candidate, and the changed text is sent to Wikipedia to be stored.	43

Chapter 1

Introduction

Wikipedia (wikipedia.org), the on-line, user-edited encyclopedia, is the sixth-most frequently visited website on the Internet, seen by 14% of global Internet users daily [3]. In total, Wikipedia has more than eight billion words in more than 19 million articles in more than 270 languages; the English language version of Wikipedia by itself has over two billion words in over 3.8 million distinct articles [38].

Wikipedia is the Internet's largest *wiki*, a website where almost any visitor may edit almost any article at almost any time. The MediaWiki software running Wikipedia makes every previous version of each article available at any time, while providing a standard view that defaults to the latest version of an article.

Any reader of Wikipedia can become an editor and make a change to an article. The extensive content of Wikipedia is the result of the collaboration of many millions of editors, some of whom contribute by writing complete articles, others by fixing typographical and grammatical errors, and still others by flagging non-neutral statements, identifying other stylistic issues, and correcting and refining content. There are 3.5 million edits per month made to English Wikipedia [35]. Although some editors have made more than ten thousand edits each, more than 95% of editors to English Wikipedia have made fewer than one hundred edits. As

of 2006, more than 30% of all edits were made by these least experienced editors [1, 20].

The breadth of coverage in Wikipedia, its diversity of contributors, and its almost complete record of changes, have made it more than just a simple reference encyclopedia; Medelyan et al. [22] give a thorough overview of the many efforts made to apply the data in Wikipedia towards applications in natural language processing, information retrieval, information extraction, and ontology building.

There is one perennial weakness in Wikipedia: the existence of hyperlinks to ambiguous terms. For example, the “Organ” article is not a regular article. Since both anatomical structures and musical instruments are commonly referred to by that term, there are separate articles for these two meanings (as well as several others). The “Organ” article itself is a *disambiguation page*, an article that contains a list of links to possible meanings of the title. Thus, any link to the “Organ” article should probably be corrected to link to one of these possible meanings.

If we could aid human editors in correcting these ambiguous links and pushing the results back into Wikipedia, we would expect to improve the performance of many of the ever-expanding collection of applications based on Wikipedia. The Freebase Project (freebase.org) and DBpedia (dbpedia.org) are two examples of semantic databases derived from data extracted from Wikipedia’s infoboxes, special information boxes that editors maintain in many Wikipedia articles. Improving link quality in Wikipedia should help these projects and many others.

In this thesis, we propose a novel statistical topic model, which we refer to as the *Link Text Topic Model* (LTTM), that can help aid human editors by suggest-

ing new target articles for existing ambiguous links in Wikipedia articles. Before describing this model, we first provide background information in three areas: the creation of ambiguous links in the Wikipedia editing process, relevant related work in word sense disambiguation, and a brief introduction to topic models and their applications. We then describe our new topic model and our proposed inference process. After that, we evaluate our technique alongside several other text- and link-based disambiguation techniques (including tf-idf text similarity, Random Walk with Restart, and Wikipedia Link Relatedness) on data derived from the history of edits to Wikipedia. Finally, we describe our web-based disambiguation service to aid Wikipedia editors, and propose future work.

Chapter 2

Background

2.1 Wikipedia

One main advantage an online encyclopedia like Wikipedia has over paper encyclopedias is the abundance of relevant in-text hyperlinks between articles; the Wikipedia Manual of Style suggests creating a link for the first instance of any word or phrase that a reader is likely to also want to read, since these links aid readers in the exploration of related topics [37]. Besides aiding readers, these links are also the fodder for semantic extraction tools. MediaWiki *wikitext*, the markup language in which Wikipedia articles are written, makes turning a word or phrase into a hyperlink a trivial action; an editor simply adds a matched pair of double square brackets around that word or phrase. For example, the following wikitext contains a link to the “Organ” and “Human body” articles: “The kidney is an [[organ]] in the [[human body]].”

When an editor adds a link in an article, the link should have a target article that is about the topic being referenced. However, this is complicated by polysemous words and phrases. Since an article title is how an article is referenced in a URL, the MediaWiki software that runs Wikipedia does not allow two or more articles to share an identical title. For example, the article about anatomical organs and the article about musical organs cannot both have the same title “Organ”. If multiple

articles could justifiably have the same title, there is a set of standard practices the Wikipedia community has to resolve title ambiguity. One option is for the article associated with the primary or earliest meaning of the title to be given that title, and any other articles be given different but still related titles. For example, the title “Apple” is assigned to the article about the fruit, and the article about the computer company has been given the title “Apple Inc.”. Another option is to add a word or phrase in parentheses expressing the specific sense of the title at the end of an ambiguous title. Following this model, the article about anatomical organs is called “Organ (anatomy)”, and the article about the family of musical instruments is called “Organ (music)”.

In cases where there is a clear dominant or root sense for the ambiguous title in question, a special italicized hyperlink is added at the top of the article to point either to other senses of the term (if there are only a few), or to a *disambiguation page*, a special article that lists the correct titles of articles that could be associated with the ambiguous title, known as *disambiguation candidates*. In English Wikipedia, a disambiguation page usually has “(disambiguation)” in its title. For example, in Figure 2.1, the primary “Apple” article about the fruit links to the “Apple (disambiguation)” page, which links in turn to “Apple”, “Apple Inc.”, and several other articles related the word “apple”. In the editions of Wikipedia in other languages, an equivalent term to “disambiguation” in the appropriate language is used.

In instances where there is no generally agreed upon dominant or root sense, the disambiguation page itself is usually given the ambiguous title. Figure 2.2


This article is about the fruit. For the technology company, see [Apple Inc.](#). For other uses, see [Apple \(disambiguation\)](#).

"Apple tree" redirects here. For other uses, see [Apple tree \(disambiguation\)](#).

The **apple** is the [pomaceous fruit](#) of the apple tree, species ***Malus domestica*** in the rose family ([Rosaceae](#)). It is one of the most widely cultivated tree fruits, and the most widely known of the many members of [genus Malus](#) that are used by humans.

The tree originated in [Western Asia](#), where its wild ancestor, the *Alma*, is still found today. There are more than 7,500 known [cultivars](#) of apples, resulting in a range of desired characteristics. Cultivars vary in their [yield](#) and the ultimate size of the tree, even when grown on the same [rootstock](#).^[2]

Apple



A typical apple

Figure 2.1: *The top of the “Apple” article at <http://en.wikipedia.org/wiki/Apple> links to the “Apple (disambiguation)” page.*

Organ may refer to the following:

- [Organ \(anatomy\)](#), a collection of tissues joined in structural unit to serve a common function
- [Organ \(music\)](#), a family of keyboard musical instruments characterized by sustained tone
 - [Pipe organ](#), a musical instrument that produces sound when pressurized air is driven through a series of pipes
 - [Theatre organ](#), a pipe organ originally designed specifically for imitation of an orchestra
 - [Electronic organ](#), an electronic keyboard instrument
- [Organs of state](#), branches of power within a government
- [division \(business\)](#) within an organization; i.e. Organs of United Nations
- [Organ pipe coral](#), a marine organism native to the Indian and Pacific Oceans
- *[Stenocereus thurberi](#)*, the organ pipe cactus plant
- [The Organ](#), an indie rock band
- [The Organ \(newspaper\)](#), an underground newspaper published in San Francisco
- [Organ, Hautes-Pyrénées](#), a commune in France
- *[Organ \(magazine\)](#)*, a UK music magazine run by Organart
- *[Organ \(film\)](#)*, a 1996 Japanese film

a free online dictionary

Wiktionary

Look up **organ** in Wiktionary, the free dictionary.


 *This [disambiguation page](#) lists articles associated with the same title. If an [internal link](#) led you here, you may wish to change the link to point directly to the intended article.*

Figure 2.2: *The “Organ” disambiguation page, with links to disambiguation candidates, is available at <http://en.wikipedia.org/wiki/Organ>.*

shows the “Organ” disambiguation page containing a short list of disambiguation candidates of the word Organ: “Organ (anatomy)”, “Organ (music)”, and several other senses.

These disambiguation pages are useful for users who reach pages directly by typing into Wikipedia’s search box a word or phrase that happens to be ambiguous, or by following a link from an external website. If the disambiguation page is assigned the ambiguous title, the reader can view the disambiguation candidates and pick the link to the article related to the meaning they intended. If, instead, one of the meanings is assigned the ambiguous title, the reader clicks the link at the top of the article to go to the disambiguation page and then clicks on the link to their intended meaning.

As useful as disambiguation pages are for aiding in searching, articles should not directly link to disambiguation pages in their text; it is almost always the case that an editor intended a link to a disambiguation page to be a link to one of the possible meanings of the phrase rather than the disambiguation page itself [36]. We assume that the primary way these undesired links to disambiguation pages are created is by an editor turning a word or phrase into a link, either after the original text was added to Wikipedia, or in the process of writing the text. For example, if an article about a musical band contained the word “organ”, an editor might turn that into “[[organ]]”, expecting the “Organ” article to be about the musical instrument. If the editor does not check to ensure that the linked article is not a disambiguation page and that the contents match their intended meaning, this ambiguous link will persist until corrected by another editor. To make a link with the same text as

the original link, but that instead points to one of the unambiguous meanings, the editor should have changed the link to be “`[[Organ (music)|organ]]`”; in wikitext, the pipe character separates the link destination from the text of the link visible to readers.

Due to the enormous breadth of coverage of Wikipedia, the English version had more than 196,000 disambiguation pages in April 2011. Each of these pages contains a notice that it is a disambiguation page and, if it follows the Wikipedia Manual of Style properly, a list of disambiguation candidates. It is easy for an editor to accidentally create a link to a disambiguation article rather than a more appropriate link to one of the article’s disambiguation candidates; in September 2010 there were 442 disambiguation pages in the English version of Wikipedia that each had more than 100 incoming links. Figure 2.3, shows the full distribution of the number of inlinks per disambiguation page. Since there are so many undesired links, an automated or semi-automated disambiguation system would be very helpful to editors who work on replacing these links to assist readers in moving quickly between relevant articles. Helping these editors fix ambiguous links aids other projects that extract information from Wikipedia. One example is Freebase, a web-based database derived from Wikipedia and other sources [7]. It permits complex queries of semi-structured information extracted from Wikipedia articles, corresponding to questions such as “What is the most populous city with a female mayor?” and “What British bands have an organ player?” In this last example, if the Wikipedia article for a British band with an organ player incorrectly linked to the “Organ” disambiguation page rather than to the “Organ (music)” page, that band would be incorrectly omitted

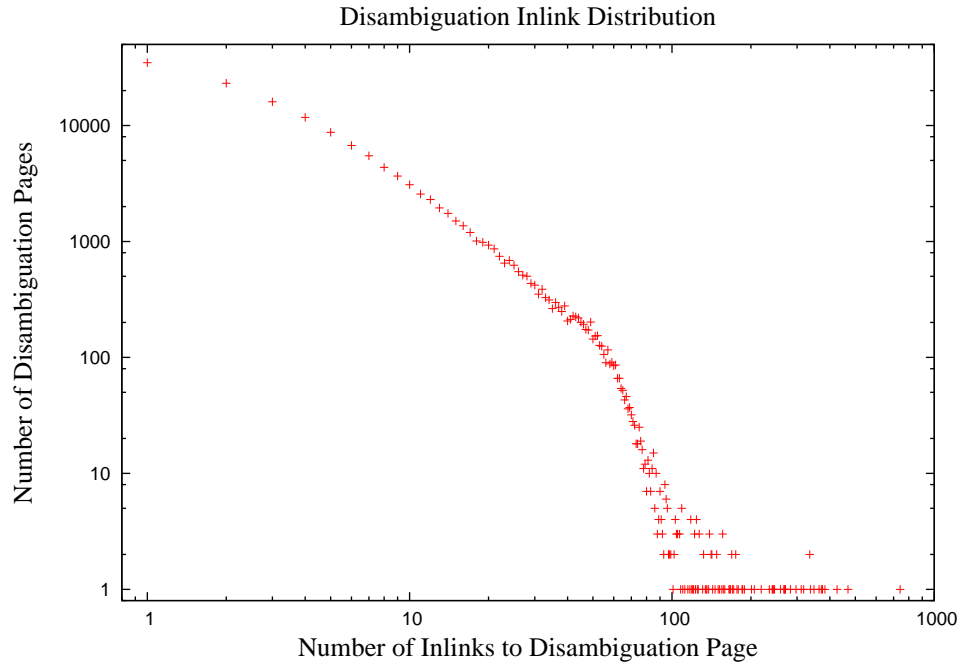


Figure 2.3: *Distribution of number of inlinks per disambiguation page in English Wikipedia*

from the results of this last query. Besides Freebase, Wikipedia is a growing data set for other natural language processing, artificial intelligence, and machine translation systems [9], and replacing ambiguous links should also improve the quality of these applications.

2.2 Word Sense Disambiguation

Wikipedia link disambiguation is related to the generic problem of word sense disambiguation, the process of determining which of several potential meanings a word has in a given context. These different meanings may be different parts of speech, so natural language processing applications involving sentence parsing or part-of-speech tagging need to address word sense disambiguation at some level.

In most applications involving word sense disambiguation, there is a strict

dichotomy between the topics uniquely identifying each document and the words being disambiguated. In link disambiguation, however, the set of topics and the set of link targets are the same.

The general word sense disambiguation problem has been studied for many years, and Agirre and Edmonds [2] provide recent in-depth coverage of many aspects of word sense disambiguation, from knowledge-based methods to unsupervised corpus-based methods, as well as the importance of word sense disambiguation to such natural language processing applications as machine translation. State-of-the-art word-sense-disambiguation techniques typically use the parts of speech and identity of the surrounding words to perform disambiguation.

Link disambiguation is similar to word sense disambiguation because picking the destination of an unambiguous link relates to picking the underlying meaning of the linked phrase. However, our techniques incorporate the richer structure of the link graph, rather than relying on just plain text. Most Wikipedia articles have many links; with 102 million links in total, there were more than 22 links per article on average in English Wikipedia in September 2010. There were 1.03 million ambiguous links in all.

Mihalcea [23] is the first to apply general word sense disambiguation to Wikipedia. Her system uses Wikipedia as a sense-tagged corpus and uses articles listed in disambiguation pages as classes for a naive Bayes classifier; the features used for each word are the part of speech and local context of links to each disambiguation candidate. A manual map was created between WordNet senses and Wikipedia articles for 51 words and used to evaluate the system against the SENSEVAL evaluations of

word sense disambiguation systems. The system showed a large improvement over the baseline system.

The “Wikify!” system of Mihalcea and Csomai [24] takes this word-sense-disambiguation system and applies it to the task of adding Wikipedia article hyperlinks to an existing text document, a process known as *wikification*. The system compares several methods of candidate extraction and candidate ranking, the most successful ranking algorithm being the ratio of the number of articles in which a specific candidate word or phrase from the document appears as a hyperlink compared to the number of article in which it appears regardless of status as a link or not. Once a phrase is identified as a link, it is put through Mihalcea’s previous disambiguation system.

Milne and Witten [26] take a different approach to disambiguation in their wikification system. They use a concept of Wikipedia Link Relatedness based on the amount of overlap of the sets of inlinking articles each disambiguation candidate has with inlinks to the other articles linked in the source text; we will describe and use this scoring system in Section 4.2.3.2.

2.3 Topic Modeling

Topic modeling is the process of describing documents in a text corpus in terms of a small number of *topics*, which are probability distributions over words. It is motivated by the problems associated with the extremely high dimensionality of the standard document-vector bag-of-words model. With hundreds of thousands to mil-

lions of words in a vocabulary, documents are treated as members of a huge vector space. Applying standard document similarity techniques such as cosine similarity to raw document-vectors can result in inadequate performance, since this approach suffers both from complete separation of related concepts as well as confusing polysemous words.

Before probabilistic topic models became popular, Latent Semantic Analysis (LSA) (also known as Latent Semantic Indexing (LSI)) was the dominant method of performing useful dimensionality reduction [11]. In LSA, singular vector decomposition (SVD) is performed on a term-document matrix X , yielding $X = U\Sigma V^T$, where U and V are orthogonal matrices and Σ is a diagonal matrix. The k largest entries in Σ correspond to the square roots of the non-zero eigenvalues of X^*X , and the corresponding row vectors in U and V are the best k -dimensional approximations of X under the Frobenius norm.

The probabilistic model behind LSA is not immediately obvious. Probabilistic Latent Semantic Analysis (PLSA) was a first attempt at putting LSA into a probabilistic framework [19]. In PLSA, a document is treated as a mixture of underlying topics. The topics are shared among all the documents, but in varying proportions. Each document has its own mixture of topics. Figure 2.4 shows PLSA using plate notation.

One downside to PLSA is that it is prone to overfitting, since the number of parameters grows linearly with the number of documents. In addition, it is difficult to evaluate the effectiveness of PLSA at the core task of document modeling, because it is impossible to assign a probability to a held-out document.

Latent Dirichlet Allocation (LDA) is a popular extension to PLSA that solves these shortcomings [6]. In LDA, the PLSA model is modified so that Dirichlet priors are placed on the topic distributions as well as the per-document topic mixtures. LDA is a true probabilistic generative model for describing how a corpus of documents is created, and its effectiveness for document modeling can be evaluated by measuring the perplexity of held out documents. Figure 2.5 shows LDA in plate notation.

There are several possible ways of inferring underlying topics in the LDA model. The original paper uses expectation maximization; other authors have used (collapsed) variational inference for performing inference in a batch on a single machine

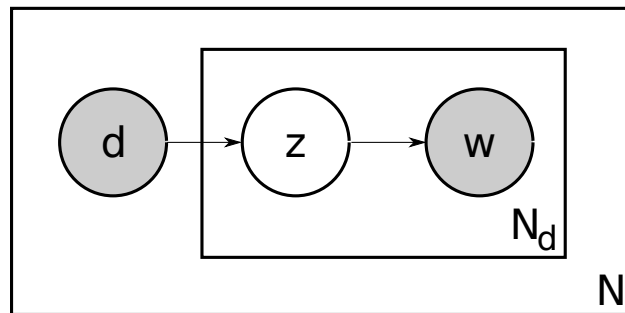


Figure 2.4: *Plate diagram for Probabilistic Latent Semantic Analysis*

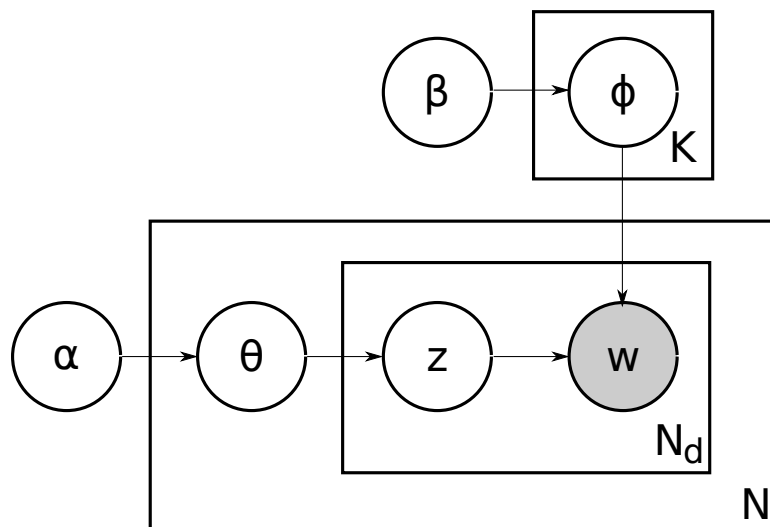


Figure 2.5: *Plate diagram for Latent Dirichlet Allocation*

[31], distributed among many machines, as well as in a streaming environment [30]. We will briefly describe the implementation of a Gibbs sampler for performing topic inference. See Heinrich [16] for a full exposition of the derivation.

In Gibbs sampling for LDA inference, the topic assignments for each term in the corpus are assigned at random to one of the K topics. Then, over several iterations, the topic assignment for each term is resampled based on the topic assignments in the current document and the count of topic assignments for that word over the entire corpus. This is a simple algorithm to implement, and recent computational improvements provide effective techniques to scale to thousands of topics or more with minimal performance penalty [39].

Regardless of the choice of inference algorithm, it is not instantly clear how to choose values for the hyperparameters α and β . Recent work has shown that in many text corpora, it is sufficient to pick a symmetric β , but that asymmetry in α does a reasonably good job of collecting stop words into a small number of topics, as well as resulting in better perplexity for held out documents [34]. Wallach [33] provides a straightforward algorithm for optimizing α and β by maximum likelihood estimation between rounds of Gibbs sampling.

Another frequent question brought up in performing inference with the LDA model is that of choosing the best way to pick the number of topics K ; practitioners frequently pick K via cross-validation with held-out data. The Hierarchical Dirichlet Process (HDP) is an alternative model that can be thought of as an extension to LDA with an infinite number of topics, a finitely many of which are actually used in the corpus [32]. In HDP, the number of topics used is not specified in advance; at each

stage of the inference process, the topic assignment for any word will likely reuse previously used topics, but there is a small probability that it will be assigned to an unused topic.

LDA has been applied to modeling graphical data; specifically, Latent Dirichlet Allocation for Graphs (LDA-G) uses the LDA generative model to describe how edges are created between nodes in a graph [17]. In LDA-G, a node is treated as a document, and the outlinks are treated as the words of the document. By performing inference on the model, latent groups in the graph can be discovered. This model has proved useful to applications such as identifying researchers in different topic areas based on a co-authorship graph [12].

Chapter 3

Link Text Topic Model

We shall describe a novel topic model based on LDA that provides a generative model for both the links in an article and the text of the links. We call this model the Link Text Topic Model (LTTM).

3.1 Generative Model

Instead of using LDA for modeling the words in an article, we will be modeling the creation of links between articles; this model will prove useful in our disambiguation task. As in LDA, we assume that each article has associated with it a mixture of shared topics drawn from a common Dirichlet distribution. However, instead of each topic being a distribution over words as in LDA, each topic in LTTM is now a distribution over articles. Furthermore, we stipulate that the text of each inlink to a specific article is drawn from a link-target-specific multinomial distribution over possible texts.

The generative story for LTTM is similar to LDA, and thus we will use similar notation. First, a global number of topics K is picked, as well as a total number of articles N and set of possible link texts of size V . Appropriate α , β , and γ vectors are chosen as parameters for Dirichlet distributions; α is K -dimensional, β is N -dimensional, and γ is V -dimensional. Following practical recommendations

related to performing inference on the LDA model, we will assume β and γ are symmetric, but we will let α be asymmetric. Furthermore, it is straightforward to place hyperpriors on these parameters, but in this exposition we will choose not to do so.

The article distribution for each of the K topics is chosen from a Dirichlet distribution parameterized by β . For each article r that is ever linked to, the distribution of possible link texts π_r is chosen from a Dirichlet distribution over link texts parameterized by γ .

To generate the links for an article d , we will pick an associated topic mixture θ ; as in LDA, the mixture is chosen from a Dirichlet distribution parameterized by α . Then, for each link, we pick the topic $z_i = k$ for the link from the topic mixture. We then choose the target $a_i = r$ for the link from the corresponding topic ϕ_k . Finally, we pick the text $t_i = l$ for the link from that target’s link-text distribution π_r . The identity of a_i and t_i is readily available for normal links, but we will assume that the link topic z_i , article specific topic mixture θ , and global topic distributions ϕ_k are latent and must be inferred. We will use z'_j , a'_j , and t'_j to refer to the j th ambiguous link in an article.

For all links, the text of the link is visible. For links to regular pages, we assume that the identity of the link is visible. However, we will assume that the true targets of links that are to disambiguation pages are latent. Figure 3.1 has the plate notation for this model, with the variables associated with the ambiguous links denoted with a prime.

This model is similar to the LDA-ER model used for entity resolution [5]; how-

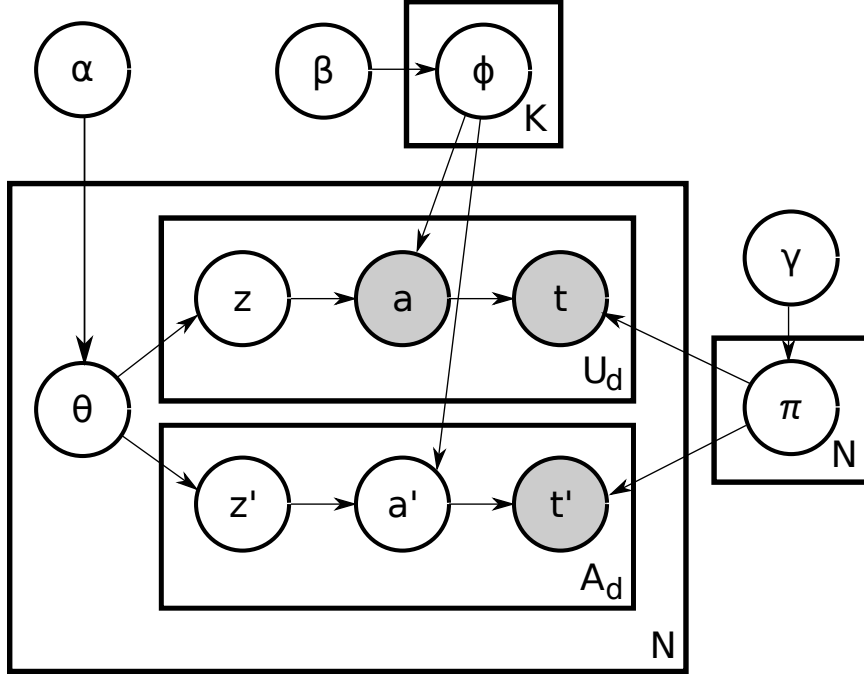


Figure 3.1: Plate diagram for the Link Text Topic Model. A prime next to a variable indicates it is used for ambiguous links.

ever, we assume that most link targets will be visible (rather than always latent as in LDA-ER). Because of this, we do not use a noise model for the link text, and instead use a multinomial with a Dirichlet prior. In addition, in our implementation of posterior inference, we do not require the entire link structure to fit in memory, α is no longer symmetric, we perform hyperparameter estimation for α , β , and γ , and we use a sampling improvement to scale easily to thousands of topics.

3.2 Posterior Inference

Given this model, it is possible to perform posterior inference to determine highly probable values of the latent variables. We use a collapsed Gibbs sampler to determine the values of z , z' , and a' . The values for θ , ϕ , and π are never explicitly

sampled, but instead are integrated out.

We will use the notation $n_{m,k,r,t}$ to represent the number of times in document m that topic k is used for a link to target article r with link text t , regardless as to whether or not the link is ambiguous. In addition, if any subscript is replaced with a “.”, then that subscript is being summed over. Finally, appending “; $\neg i$ ” means that the counts should not include the variables associated with link i in the corpus.

The value for topic z_i associated with the link a_i in position i in document m is resampled proportional to:

$$p(z_i = k | \vec{z}_{\neg i}, \vec{z}', \vec{a}, \vec{a}') = \frac{n_{(m,k,.,.;\neg i)} + \alpha_k}{\sum_{k'=1}^K n_{(m,k',.,.;\neg i)} + \alpha_{k'}} \cdot \frac{n_{(.,k,r,.;\neg i)} + \beta_r}{\sum_{r'=1}^N n_{(.,k,r',.;\neg i)} + \beta_{r'}} \quad (3.1)$$

$$\propto (n_{(m,k,.,.;\neg i)} + \alpha_k) \cdot \frac{n_{(.,k,r,.;\neg i)} + \beta_r}{\sum_{r'=1}^N n_{(.,k,r',.;\neg i)} + \beta_{r'}} \quad (3.2)$$

The value for topic z'_j associated with ambiguous link a'_j that currently has the value r in position j in document m is resampled proportional to an almost identical value:

$$p(z'_j = k | \vec{z}'_{\neg j}, \vec{z}, \vec{a}, \vec{a}') = \frac{n_{(m,k,.,.;\neg j)} + \alpha_k}{\sum_{k'=1}^K n_{(m,k',.,.;\neg j)} + \alpha_{k'}} \cdot \frac{n_{(.,k,r,.;\neg j)} + \beta_r}{\sum_{r'=1}^N n_{(.,k,r',.;\neg j)} + \beta_{r'}} \quad (3.3)$$

$$\propto (n_{(m,k,.,.;\neg j)} + \alpha_k) \cdot \frac{n_{(.,k,r,.;\neg j)} + \beta_r}{\sum_{r'=1}^N n_{(.,k,r',.;\neg j)} + \beta_{r'}} \quad (3.4)$$

Finally, the target a'_j of ambiguous link j is resampled proportional to:

$$p(a'_j = l | \vec{z}', \vec{z}, \vec{a}, \vec{a}'_{\neg j}, \vec{t}, \vec{t}') = \frac{n_{(.,k,r,.;\neg j)} + \beta_r}{\sum_{r'=1}^N n_{(.,k,r',.;\neg j)} + \beta_{r'}} \cdot \frac{n_{(.,r,t,;\neg j)} + \gamma_t}{\sum_{t'=1}^V n_{(.,r,t',;\neg j)} + \gamma_{t'}} \quad (3.5)$$

$$\propto (n_{(.,k,r,.;\neg j)} + \beta_r) \cdot \frac{n_{(.,r,t,;\neg j)} + \gamma_t}{\sum_{t'=1}^V n_{(.,r,t',;\neg j)} + \gamma_{t'}} \quad (3.6)$$

Under the model as described, it is possible that any article may be chosen for the link target a' of an ambiguous link; however, most will have a very small probability of being picked. Since it seems unreasonable for a disambiguation approach to suggest a target article that has never been associated with a given link text, we will restrict our sampler to only choose values for a'_j that have been used before in some a_i where $t'_j = t_i$; we will call this set of candidate values A'_j .

We have described our Gibbs sampler for sampling the topics and the links separately; instead we could use block Gibbs sampling for the (z'_j, a'_j) variable pair associated with each ambiguous link. However, by doing so, we would have to sample from $K \cdot |A'_j|$ values, where A'_j is the number of possible values for a'_j . By instead sampling each z'_j and a'_j separately, we only have to sample from $K + |A'_j|$ values per pair.

Finally, at the very end of this process, we do not actually care about the values for z and z' ; the only thing that matters for disambiguation is a' . Thus, we will not actually be making final predictions for the topic assignments; we will integrate them out and produce probability distributions over the possible article targets.

Disambiguation with this model is easy; the disambiguation candidate chosen for an ambiguous link is simply the most probable candidate. Furthermore, we can rank ambiguous links by our certainty in their disambiguated values by ranking them according to the probability of their most likely candidate.

Chapter 4

Alternate Disambiguation Techniques

We will compare LTTM to seven other algorithms for disambiguation: two simple baselines that predict popular link targets, three text-similarity approaches, a graph-based random-walk approach, and a link-based approach. Before describing the alternative approaches, we explain how to identify disambiguation pages and candidates.

4.1 Disambiguation Page Identification and Disambiguation Candidate Extraction

All the algorithms we consider require us to automatically identify ambiguous links and make a suggestion of a disambiguation candidate. Therefore, we need to find all the disambiguation pages and extract the disambiguation candidates from each page. Since the MediaWiki software has no special internal representation of disambiguation pages, we must be aware of the Wikipedia community standards to identify and extract the information we need.

In Wikipedia, the Manual of Style covers many aspects of article creation, ranging from the proper use of dashes to general layout guidelines for various kinds of articles. Specifically, there is a section dedicated to the layout of disambiguation pages [37]. Importantly, the guide indicates the various templates that can be placed

on a page to identify it as a disambiguation page. Thus, we can identify disambiguation pages by going through all articles and finding those pages that contain these templates.

The Wikipedia Manual of Style suggests that the various disambiguation candidates should be placed in a specially formatted list, ideally with only links to disambiguation candidates in the list; links to any other articles should be avoided, to make it easy for a reader to know what to click on in each line. For many pages, this is a simple one-level list, but for some topics, such as “Java (disambiguation)”, there is a complicated hierarchy (Figure 4.1). This hierarchy information potentially could be used in a hierarchical classification system, but we currently flatten such a list to treat all disambiguation candidates on equal footing.

The effectiveness of our algorithms rely on disambiguation pages following these guidelines. We will ignore any links that do not appear in list form; making suggestions to expand a disambiguation page would be an interesting problem in and of itself. A simple review of 100 randomly chosen English disambiguation pages shows that 93 had their disambiguation candidates appear all in list form; 7 had at least one disambiguation candidate appear only elsewhere in the page, but not in a list. However, 27 of the 100 had more than one link per line, with the extra links often being very general (for example, country names or years) that should not be treated as synonyms. Because of this, we consider the heuristic of treating all the links in a single line of a list as potential targets as too overly inclusive to be used effectively.

Instead of trying to derive possible candidates from disambiguation pages, we

Java is the most populous island in Indonesia.

Java may also refer to:

Look up *java* or *Java* in Wiktionary, the free dictionary.

Animals

- **Java Pipistrelle**, *Pipistrellus javanicus*, a species of pipistrelle bat
- **Java shark**, *Carcharhinus amboinensis*, also known as the **pigeye shark**
- **Java Sparrow**, *Padda oryzivora*, a popular cage-bird
- **Java (chicken)**, a rare breed that is one of the oldest American chickens

Literature

- *Java, seine Gestalt, Pflanzendecke, und sein innerer Bau* (Images of Light and Shadow from Java's interior) - a four volume treatise written by Dutch naturalist **Franz Wilhelm Junghuhn**, and considered the first formal articulation of **Pandeism**

Computer science

- **Java (programming language)**, an object-oriented high-level programming language
- **Java (software platform)**, a technology developed by Sun Microsystems for machine-independent software
 - **Java Platform, Standard Edition**, targets desktop environment
 - **Java Platform, Enterprise Edition**, targets server environment
 - **Java Platform, Micro Edition**, targets mobile devices and embedded systems
 - **Java Card**, targets smart cards and other small memory footprint devices
 - **Java Development Kit** (JDK), a software bundle from Sun Microsystems aimed at Java developers
 - **Java Virtual Machine** (JVM), part of the Java Platform that interprets (or possibly translates) Java bytecode
- **Java applet**, allows software to run in web browsers, and is accessible on most PCs
- **JavaScript**, a web scripting language with no direct relationship to the Java platform

Consumables

- **Java (cigarette)**, a brand of Russian cigarettes
- **Java coffee**, a variety of coffee grown on the island of Java, or American slang for coffee
- **Java**, a brand of **cachaça**, a type of alcohol

Entertainment

- **Java (band)**, a French band
- **Java (dance)**, a Parisian Bal-musette dance

Contents
1 Animals
2 Literature
3 Computer science
4 Consumables
5 Entertainment
6 Geography
7 Plants
8 Transportation

Figure 4.1: *The first page of the “Java (disambiguation)” article shows grouped disambiguation candidates. The whole list is viewable at [http://en.wikipedia.org/wiki/Java_\(disambiguation\)](http://en.wikipedia.org/wiki/Java_(disambiguation)).*

extract candidates from the text of links, which is both convenient and effective; if a specific link text has been used to link to a page before, it is reasonable to consider it as a potential link target in other contexts. For each link to an ambiguous page, we will construct a list of all possible articles that are linked to with the same link text; we will use this list as our disambiguation candidates.

4.2 Disambiguation Algorithms

4.2.1 Baseline

In a pattern classification problem with high class skew, a useful baseline is always picking the most frequent class, regardless of the feature values of a specific instance. In word sense disambiguation (see section 2.2), this is known as the *most-frequent sense baseline* and is common in evaluating word sense disambiguation techniques [14]. We will use two forms of this baseline; in the first we will predict the disambiguation candidate with the most inlinks of any kind, and in the second we will predict the disambiguation candidate with the most inlinks with the text of the ambiguous link in question. These approaches take no other features into consideration, so any useful disambiguation algorithm will hopefully perform better.

With link disambiguation, we see just such a class skew. Figure 4.2 demonstrates how one of the musical senses of the text “organ” accounts for most of the links. Thus, we use a most-common link baseline by comparing the number of links from other articles to the different disambiguation candidates for an ambiguous link. We simply pick the disambiguation candidate with the highest number of inlinks

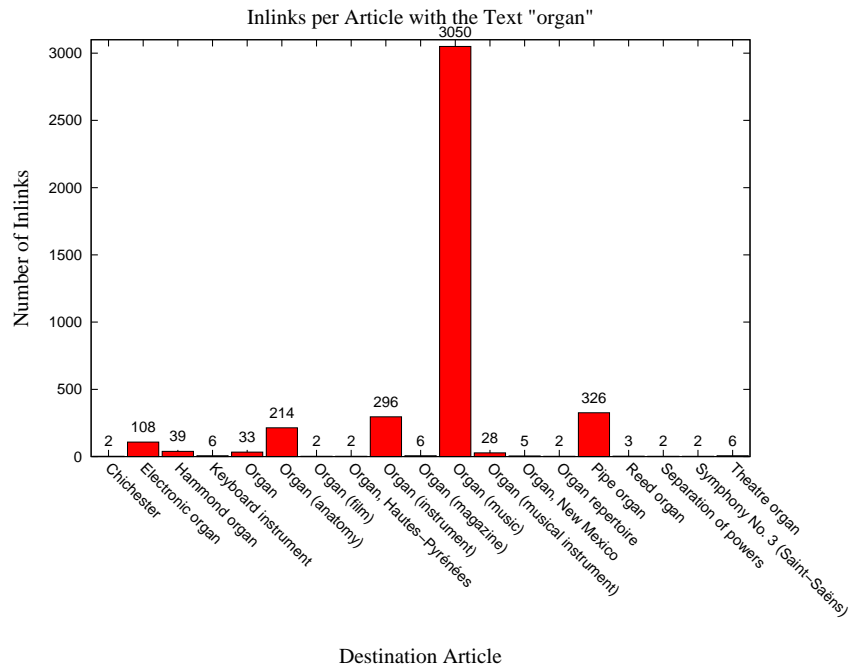


Figure 4.2: *The distribution of inlinks with the text “organ” demonstrates the large class skew that can arise in disambiguation; 40 articles with a single inlink are not shown.*

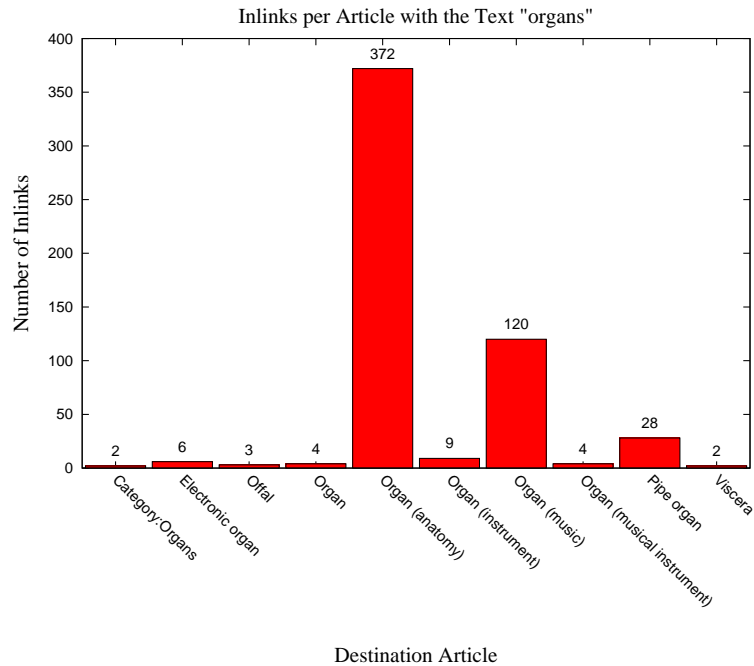


Figure 4.3: *The distribution of inlinks with the text “organs” has the anatomical sense dominate rather than the musical sense; 10 articles with a single inlink are not shown.*

with that text. For example, since “Organ (music)” had more inlinks than any page linked with “organ”, all links to “Organ” with the text “organ” would be replaced with links to “Organ (music)”.

However, it is important to note that the specific text used to link to a disambiguation page can alter the meaning. For example, Figure 4.3 shows that most links with the text “organs” link to the anatomical sense rather than the musical sense; thus, the text-specific most-frequent-class baseline would predict “Organ (anatomy)” for any link with “organs” as the link text.

4.2.2 Text-Based Approaches

The first non-trivial approaches we consider involve the text of the articles, but not explicitly the link structure.

4.2.2.1 Text Similarity

The first text-similarity technique we consider is **Jaccard similarity** on the sets of words found in articles; looking only at the text of the articles in question and not at the existence or frequency of any links, we pick the disambiguation candidate that has the highest Jaccard similarity between the set of words present in the candidate article page and the set of words in the source article. If $W(d)$ is the set of words in article d , and $W(d')$ is the set of words in article d' , then the Jaccard similarity is defined as the cardinality of the intersection of the two word sets divided by the cardinality of their union; $sim_{Jaccard}(d, d') = \frac{|W(d) \cap W(d')|}{|W(d) \cup W(d')|}$. The

Jaccard similarity ranges from zero, if the documents have no words in common, to one, if the documents each contain the exact same set of words.¹

A second text-similarity technique we consider is **tf-idf similarity**. In tf-idf similarity, we look at the cosine similarity of articles under a tf-idf weighting scheme. We let $tf_{t,d}$ be the term frequency of term t in article d , the number of times term t appears in article d . We let df_t be the document frequency of term t , the number of articles in which term t appears; we further let N be the total number of articles in Wikipedia. We then map each article d to a vector $\hat{w}_d = \langle w_{t,d} \rangle$ indexed by term t , where $w_{t,d} = (1 + \log tf_t) \cdot \log \frac{N}{df_t}$. To compare articles d and d' , we calculate the cosine similarity between the two vectors \hat{w}_d and $\hat{w}_{d'}$, $sim_{tf-idf}(\hat{w}_d, \hat{w}_{d'}) = \frac{\hat{w}_d \cdot \hat{w}_{d'}}{\|\hat{w}_d\| \|\hat{w}_{d'}\|}$. Using SMART notation, this is the *ltc.ltc* weighting scheme [21]. With this weighting, terms common to both articles that are also present in many articles contribute less to the similarity than terms common to both that are present in few other articles. There are other possible tf-idf weighting schemes, but this was the most effective of the several we considered.

4.2.2.2 Latent Dirichlet Allocation

A third text-similarity technique we use is one based on **LDA similarity**. We assume that the text of Wikipedia articles is generated by a 100-topic LDA model. We use the Gensim framework [29] for doing the model inference, since it can be done in a streaming fashion without fitting all the data in memory. Although other

¹For this and other text-based approaches, we use Lucene’s `StandardAnalyzer` class for tokenization.

packages can perform valuable hyperparameter estimation for α and β that has demonstrated improvements in other applications of LDA [34], we were unable to find an implementation that could do the estimation and still work in a streaming fashion, since we were limited to one standard desktop machine for our evaluation, which would reasonably be available to a typical Wikipedia editor.

With each Wikipedia article represented as a probability distribution over topics, we need some way to describe similarity between topic distributions associated with each article. We will use Jensen-Shannon divergence to compare these distributions; we will pick the disambiguation candidate for an ambiguous link that has the smallest topic divergence with the linking article.

4.2.3 Link-Based Approaches

Due to the rich link structure of Wikipedia, it is reasonable to consider disambiguation techniques based just on the links between articles.

4.2.3.1 Random Walk with Restart

The first link-based disambiguation technique we consider is **Random Walk with Restart (RWR)**, also known as Personalized PageRank [4]. In this approach, we rank disambiguation candidates by their probability of being visited in a modified random walk on the Wikipedia link graph originating with the linking article, after first removing the original link to the disambiguation page. Rather than always following random outlinks as in a normal random walk, a random outlink is followed

from a node with probability $1 - \alpha$, and with probability α , the random walk returns to the originating node and restarts a random walk. The effect is similar to that of PageRank [8], but the rankings are dependent on the originating node.

A standard argument applies for showing that there is a unique stationary distribution for this process. Let us assume that the originating article is identified by index 0, and that there are v articles in total. We let P be the transition matrix associated with a random walk without restart. This matrix is sparse; if there is a link from article i to article j , and n_i total distinct outgoing links from article i , then P_{ij} is equal to $\frac{1}{n_i}$. If there is no link from i to j , $P_{ij} = 0$. This is almost a stochastic matrix; to ensure that all rows sum to 1, let $P_{i0} = 1$ if there are no outgoing links from i (that is, if $n_i = 0$).

Matrix P is now stochastic, but it is not necessarily aperiodic. We follow the PageRank example of creating a new matrix $Q = (1 - \alpha)P + \alpha E$, where $E_{ij} = 1$ if $j = 0$, otherwise $E_{ij} = 0$. Matrix E is stochastic since there is exactly one unity entry in each row, the other entries being zero, so each row sums to one. Therefore, Q is stochastic since it is the weighted average of two stochastic matrices. Since a random walk can always jump back to the start node, any node that is reachable from the start is strongly connected to it. Since the start node can jump back to itself, any path starting at the start node can be lengthened by one, so the Q matrix is aperiodic. If Q is also irreducible, there exists a unique solution to the equation $\pi^T Q = \pi^T$, where π^T is a distribution over the article representing the steady state of the random walk, known as the stationary distribution. π^T is the eigenvector of Q associated with the eigenvalue of 1, and we calculate an approximation of this

value using the Bookmark-Coloring Algorithm described by Berkin [4].²

We are not the first to apply PageRank-related algorithms to natural-language-processing problems. TextRank is an algorithm for keyword extraction and text summarization based on building a graph from the text of a document with edges based on similarity metrics, and then ranking text nodes by their PageRank in this graph [25]. LexRank is a similar technique, where sentences as graph nodes are linked via undirected edges, weighted by cosine similarity of the text [13].

4.2.3.2 Link Relatedness

Another link-based disambiguation approach we consider is **Wikipedia Link Relatedness** [26], which is based on Normalized Google Distance [10]. If A is the set of links into article a , and B the set of links into b , and W the set of all Wikipedia articles, define:

$$relatedness(a, b) = \frac{\log(\max(|A|, |B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|, |B|))}$$

Relatedness would be zero if articles a and b have identical source articles linking in, and it would be infinite if there is no overlap between the two sets. One problem with the approach in [26] is that they use a weighted average of the relatedness score between all the links in the source document and each disambiguation candidate; if one of these scores is infinite, the average is thus infinite. As this

²Since $\delta^{(i)}$ for start node i is the initial vector in Berkin’s approach, it no longer matters if the matrix is actually irreducible; the value of any other connectivity classes besides the one containing the start node will always be zero.

average appears motivated by the uncertainty of the links being used (since the application was for wikification of completely unlinked text, rather than disambiguating existing links), we choose to simply take the smallest relatedness score rather than an average.

To incorporate this link relatedness into a disambiguation algorithm, we take the article and determine all outlinks, except to the ambiguous page in question. Then, we find the disambiguation candidate article with the minimum relatedness to any non-candidate article linked from the source article, plus the source article itself.

Chapter 5

Evaluation and Results

5.1 Evaluation

In order to use and evaluate the different disambiguation algorithms, we need the text content and outlinks for each article in Wikipedia. Periodically, the Wikimedia Foundation makes available for download XML snapshots of the contents of the different language editions of Wikipedia (<http://download.wikipedia.org>). These snapshots provide enough information to extract the data we need; they provide basic metadata about each article such as title and last modification date, in addition to the wikitext content of each article.¹

To determine ground truth, we find all links to disambiguation pages present at one point in time in English Wikipedia that were later removed by human editors and replaced with the same text but different targets. We identify disambiguation pages by finding those pages that included a disambiguation template, as discussed in Section 4.1.

To find our evaluation data, we identify all links to disambiguation pages in

¹It is possible to import these snapshots into a private installation of MediaWiki to create a mirror of Wikipedia. As part of this process, a list of links between pages is automatically generated. However, this list of links makes no distinction between links directly included in the wikitext of an article, and those links indirectly included via MediaWiki's template expansion mechanism. Since the links included via a template are duplicated for every page that includes that template, and since these links are not visible to a user when they are editing the wikitext for a page, we choose to ignore these links by extracting links directly from the source wikitext.

the September 2010 snapshot of English Wikipedia; we then find links with identical text but to a different target in the October 2010 snapshot. This results in 36,009 links, of which we pick 1,000 at random for our test set. We are thus only evaluating disambiguations that keep the same visible text; if the text is altered, we are not using it for evaluation. We do not take into account the location in the article or the surrounding text of a link, so we consider a link to be unchanged if it is deleted and a new one is added elsewhere in an article to the same target. To measure the accuracy of the various disambiguation techniques, we use them to make predictions for the new targets of the links based on data in the September snapshot; we consider a correct prediction to be one that matches the new target of the link in the October snapshot. Figure 5.1 is an example of an ambiguous link present in September that was fixed by October.

The motivating assumption for this evaluation technique is that blatant errors are not likely to persist in Wikipedia. The ease-of-editing at the heart of Wikipedia does allow for malicious users to corrupt the content of articles, as well as permit well-meaning users to mistakenly submit incorrect information. Friedhorsky et al. [28] have classified the kinds of damage that takes place in Wikipedia and assessed how long the damage persists. Using edit data and view logs, they estimate that 42% of all damaging edits to English Wikipedia are fixed on the next page view, and roughly 70% are fixed within ten page views. If an erroneous edit is made, it is likely to be corrected.

5.2 Results

On our 1,000-link test set, the most-frequent-candidate baseline achieved 30.1% accuracy, and the text-specific most-frequent-candidate baseline achieved 38.2% accuracy. For text similarity, the Jaccard-similarity approach was 33.5% accurate, the tf-idf approach was 38.5% accurate, and the LDA approach was only 28.7% accurate.

```
September 2010
== Personnel ==
;John Mayall's Bluesbreakers
* [[John Mayall]] - [[Singing|Lead vocals]], [[harmonica]],
[[piano]], [[harpsichord]], [[organ]], [[harmonium]],
[[guitar]]
* [[Mick Taylor]] - [[Lead guitar]], [[Hawaiian guitar]]
* Chris Mercer - [[Tenor saxophone|Tenor]], [[baritone
saxophone]]
* [[Dick Heckstall-Smith]] - Tenor, [[soprano saxophone]]
* [[Jon Hiseman]] - [[Drum kit|Drums]], [[percussion]]
* [[Henry Lowther]] - [[cornet]], [[violin]]
* [[Tony Reeves]] - [[string bass]], [[bass guitar]]

October 2010
== Personnel ==
;John Mayall's Bluesbreakers
* [[John Mayall]] - [[Singing|Lead vocals]], [[harmonica]],
[[piano]], [[harpsichord]], [[organ (music)|organ]],
[[harmonium]], [[guitar]]
* [[Mick Taylor]] - [[Lead guitar]], [[Hawaiian guitar]]
* Chris Mercer - [[Tenor saxophone|Tenor]], [[baritone
saxophone]]
* [[Dick Heckstall-Smith]] - Tenor, [[soprano saxophone]]
* [[Jon Hiseman]] - [[Drum kit|Drums]], [[percussion]]
* [[Henry Lowther]] - [[cornet]], [[violin]]
* [[Tony Reeves]] - [[string bass]], [[bass guitar]]
```

Figure 5.1: *An example disambiguation in English Wikipedia made between September 2010 and October 2010 shows that a link in the “Bare Wires” article has been disambiguated by an editor from “Organ” to “Organ (music)”.*

For link-based approaches, Random Walk with Restart was 53.2% accurate when there was a restart probability of 0.3, and Link Relatedness was 47.0% accurate. Finally, our novel LTTM approach with 1,000 topics was 61.9% accurate, the best of all approaches we considered. Figure 5.3 compares these results, which also demonstrate the improved accuracy of each technique when we consider a correct answer to be one in the top three suggestions.

We also analyzed the effect of altering the value for the α parameter in the random walk algorithm; adjusting this parameter had only a small change in accuracy over a wide range of values.

Since the scores produced by LTTM are probabilities, they are directly comparable across predictions; it is possible to say a specific link is more likely to have one target than another link is to have a different target based on their relative scores. We therefore considered ranking the chosen disambiguation candidates across all 1,000 links to see if any techniques were particularly good at the highest scores. As can be seen in Figure 5.4, LTTM was most effective at the highest scores. We see a similar pattern when we further applied the LTTM model to all possible disambiguated links, as shown in Figure 5.5.

In addition to the disambiguation predictions, performing inference on LTTM also produces each topic's distribution over links. For example, Figure 5.6 shows the highly probably links from four topics in the 1,000-topic LTTM model of Wikipedia.

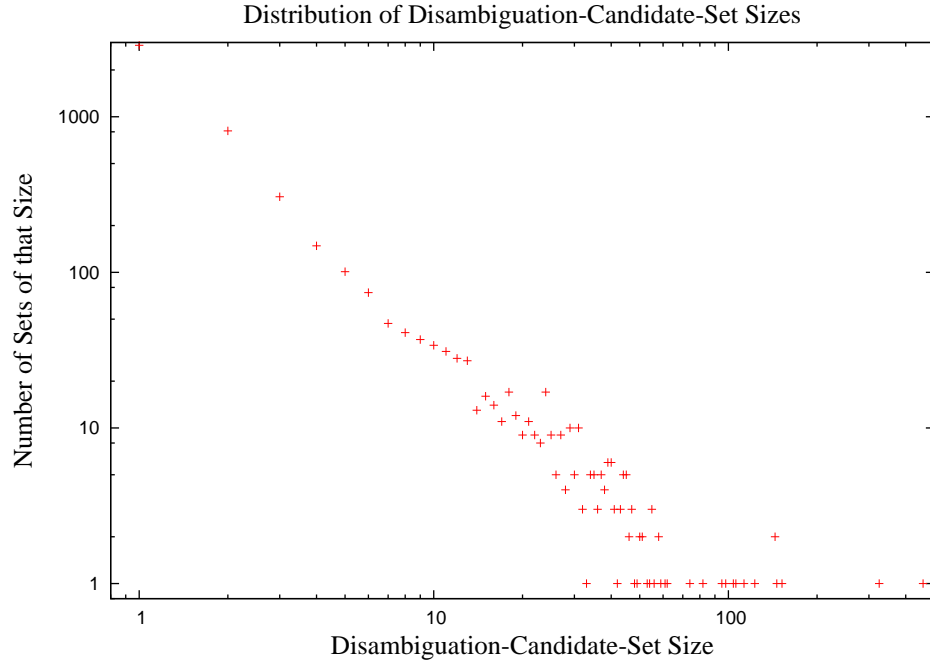


Figure 5.2: *Disambiguation-candidate-set size distribution among the 1,000 test ambiguous links in English Wikipedia*

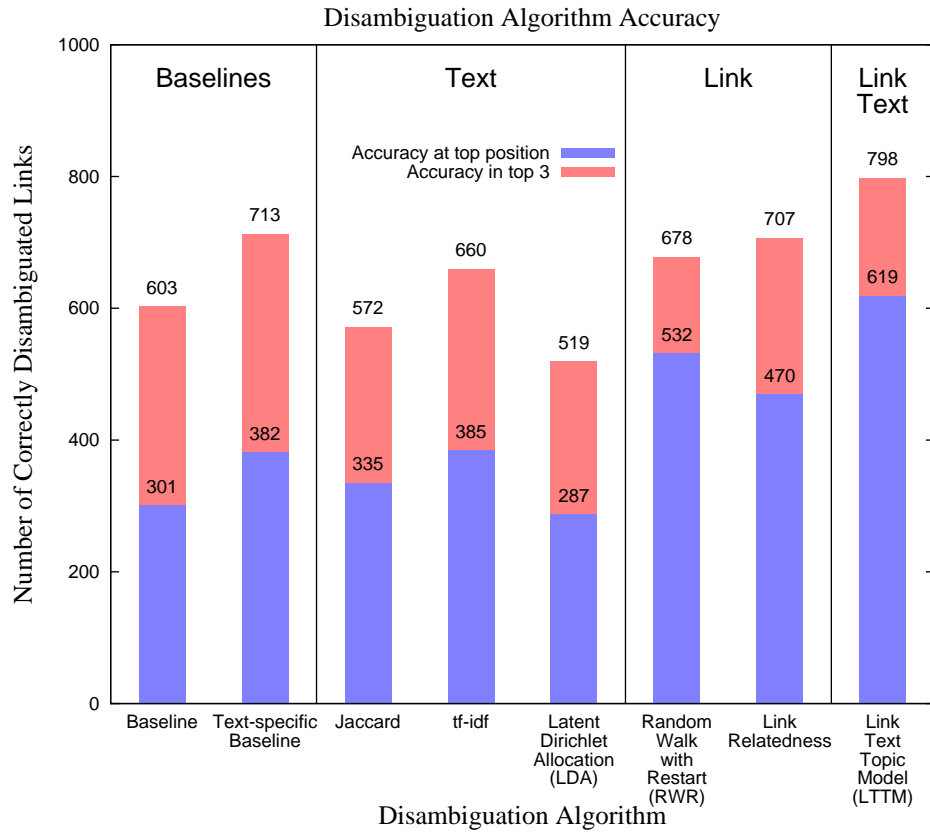


Figure 5.3: *Accuracy of eight different disambiguation algorithms on English Wikipedia, at the top position and in the top three positions*

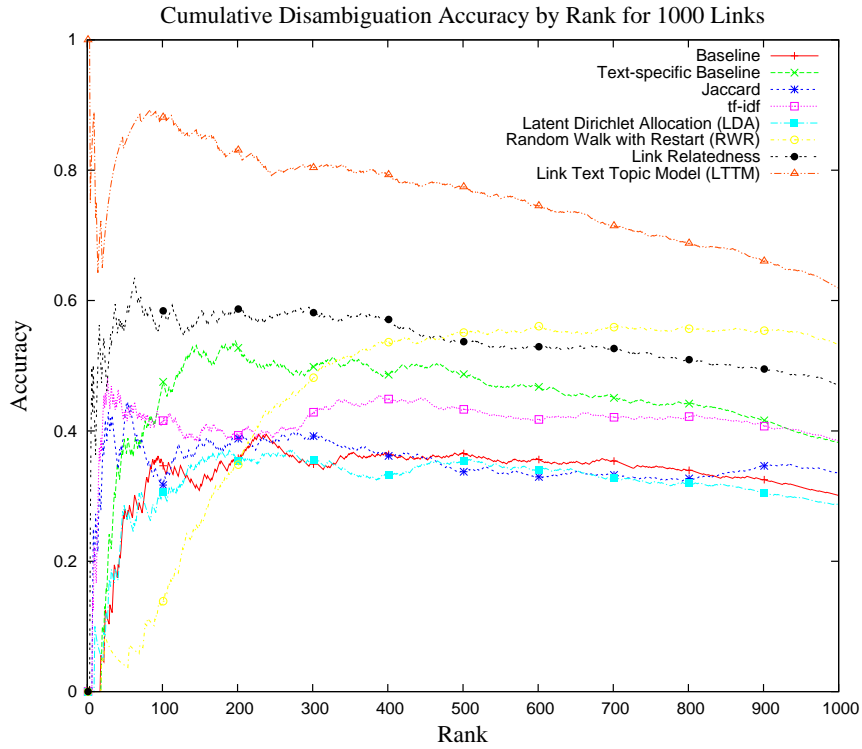


Figure 5.4: Cumulative accuracy by rank of eight different disambiguation algorithms on English Wikipedia

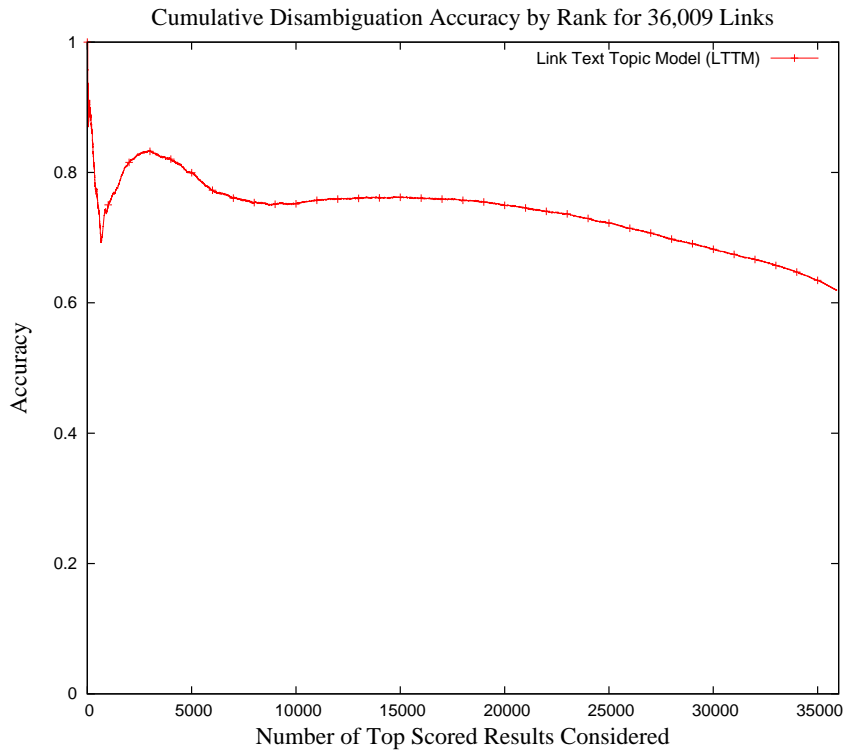


Figure 5.5: Cumulative accuracy by rank for LTTM on all 36,009 disambiguated links

Count	Title	Count	Title
5635	Maryland	6005	Thailand
3427	Baltimore	2127	Bangkok
2561	United States	1314	Tambon
1188	Washington, D.C.	1290	Laos
1163	National Register of Historic Places	946	Amphoe
697	Baltimore County, Maryland	945	Muban
690	Montgomery County, Maryland	780	Population
668	Freeway	693	Thesaban tambon
659	Prince George's County, Maryland	522	Burma
655	Anne Arundel County, Maryland	484	Bhumibol Adulyadej
655	Annapolis, Maryland	467	Thai language
645	Interstate Highway System	443	King Amphoe
612	U.S. state	442	Cambodia
598	Maryland House of Delegates	427	Chao Phraya River
541	Unincorporated area	411	Chulalongkorn
534	List of streets in Baltimore, Maryland	393	Chiang Mai Province
530	State highway	390	Thai people
490	Toll road	375	Thaksin Shinawatra
490	Interchange (road)	368	Chiang Mai
467	University of Maryland, College Park	355	Vientiane
3073	The Simpsons	3748	Greek mythology
947	List of recurring characters in The Simpsons	1660	Homer
907	Homer Simpson	1462	Zeus
720	Bart Simpson	1405	Ancient Greece
701	Futurama	1225	Apollo
658	Fox Broadcasting Company	1113	Greek language
552	Lisa Simpson	1080	Iliad
498	Marge Simpson	870	Odyssey
372	Matt Groening	864	Ancient Greek
352	Springfield (The Simpsons)	838	Athens
315	Mr. Burns	831	Dionysus
261	Nielsen ratings	826	Virgil
245	List of fictional locations in The Simpsons	823	Heracles
233	Ned Flanders	813	Troy
223	Simpson family	811	Trojan War
223	Krusty the Clown	802	Athena
217	List of recurring characters in Futurama	765	Ovid
216	List of media personalities in The Simpsons	734	Poseidon
207	The Simpsons Movie	710	Greece
206	IGN	697	Odysseus

Figure 5.6: *Counts of high-frequency links in four sample topics from a 1,000-topic LTMM model of Wikipedia representing Maryland, Southeast Asia, The Simpsons television show, and Greek mythology*

Chapter 6

Disambiguation Web Service Implementation

Building on the experimental effectiveness of the LTTM results, we constructed a web interface to aid Wikipedia editors in link disambiguation. We first created a web server to host the various tools we need. We used the Sinatra micro-web framework for the Ruby programming language to make it easy to both serve static JavaScript files, as well as respond to dynamic requests for disambiguation suggestions. We used the JRuby implementation of the Ruby programming language and the Hadoop Distributed File System for storage of our data. By building on Java-based technology, the system is able to run unchanged on a variety of operating systems.

We started with the XML database dumps available from the Wikimedia Foundation. We downloaded the latest XML file for English Wikipedia. We then processed the XML to add page metadata and article text to our database. We also processed every article to extract the links to other articles contained in the wikitext. We then performed inference on the links and their text using the LTTM model; we saved the topic distributions and the link text distributions for performing inference to disambiguate links on demand.

“Navigation Popups” is a pre-existing JavaScript addon to Wikipedia that provides several additions to the standard Wikipedia web interface. First, it provides

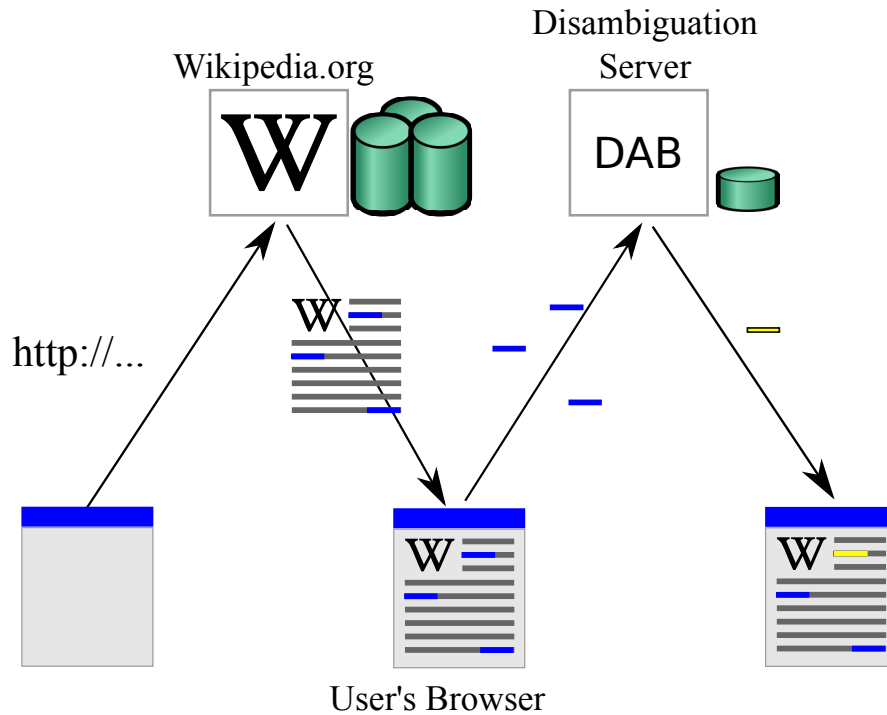


Figure 6.1: *When a user visits a Wikipedia page, the browser receives the text and extracts the links, sending them to the disambiguation server. The server returns the ambiguous links, which the browser then highlights.*

a short summary of the target page when a reader’s mouse pointer hovers over a link. Second, it provides rudimentary disambiguation capabilities. By hovering over an ambiguous link, the user may disambiguate it by clicking on one of the options in the displayed set of disambiguation candidates. This list is automatically extracted from the disambiguation page, and no recommendation is made. Also, there is no visual indication that a link is ambiguous until the reader’s mouse pointer is hovering over it.

We extended Navigation Popups in two ways: first, we provide visual highlights to indicate to the user the presence of ambiguous links; second, we calculate the LTM probabilities for the true destination of ambiguous links and return the highest-scoring disambiguation candidates by these probabilities.

Politeia (think tank)

From Wikipedia, the free encyclopedia

There was **one** ambiguous link found.

Politeia is a centre-right **British** political **think tank** that generally supports **free-market** based initiatives. Its Director, Sheila Lawlor, and Politeia 'have been and remain hugely influential in steering public policy debate gently in a right of centre direction'.^[1]

Its aim is to encourage reflection, discussion and debate about the place of the state in the daily lives of men and women across the range of issues which affect them, from **employment** and **tax** to **education**, **health** and **pensions**.

Figure 6.2: *Visiting the “Politeia (think tank)” page indicates that there is one ambiguous link on the page.*

Contributors and speakers for Politeia have included (among others):-

- [Boris Johnson](#), Mayor of London
- [William Hague](#), Foreign Secretary
- [Chris Daykin](#), HM Government Actuary (1989-2007)
- [Irwin Stelzer](#), Economics & Business Columnist, The Sunday Times
- [Michael Gove](#), Secretary of State for Education
- [Harold James](#), Professor of Economic History at Princeton University
- [Frank Field](#), Minister for Welfare Reform (1997-1998)

Figure 6.3: *The changed color and border of this link indicates that it is ambiguous, and needs to be corrected.*

- [Harold James](#), Professor of Economic History at Princeton University
 - [Frank Field](#), Minister for Welfare Reform (1997-1998)
 - [Theresa May](#), Prime Minister of the United Kingdom
 - [Dr Lucie Green](#), Director of the Centre for Health Economics and Policy
 - [Chris Vass](#), Director of the Centre for Health Economics and Policy
 - [Tim Harford](#), Director of the Centre for Health Economics and Policy
 - [George Mason](#), Director of the Centre for Health Economics and Policy
 - [Daniel Goffman](#), Director of the Centre for Health Economics and Policy
 - [John C. Hens](#), Director of the Centre for Health Economics and Policy
 - [Lord Goffman](#), Director of the Centre for Health Economics and Policy
 - [Lord Sainsbury](#), Director of the Centre for Health Economics and Policy
 - [Deepa Kundur](#), Director of the Centre for Health Economics and Policy
 - [John Emswiler](#), Director of the Centre for Health Economics and Policy
 - [Chris Emswiler](#), Director of the Centre for Health Economics and Policy
 - [Vito Tanzi](#), Director of the Centre for Health Economics and Policy
 - [Peter L. Jacobson](#), Director of the Centre for Health Economics and Policy
 - [Tony Hens](#), Director of the Centre for Health Economics and Policy
 - [Dr Liam Fox](#), Director of the Centre for Health Economics and Policy
 - [Chris Emswiler](#), Director of the Centre for Health Economics and Policy
- Frank Field** - actions - popups
Disambig, 414 bytes, 5 wikiLinks, 0 images, 0 categories, 84 weeks old
- Francis or Frank Field** may refer to:
- [Frank Field \(politician\)](#)
 - [Frank Field \(meteorologist\)](#)
 - [Frank Field \(cricketer\)](#) - English cricketer who took over 1,000 first-class wickets
 - [Frank Field \(Worcestershire cricketer\)](#) - another, less successful, English cricketer
 - [Francis Field \(St. Louis\)](#), stadium at Washington University
- [Frank Field \(politician\)](#) 0.999894321249034 [University of California](#)
- [Frank Field \(Australian politician\)](#) 0.0000341000731059285
- [Frank Field \(meteorologist\)](#) 0.0000247358296314857 [ation](#)
- [Frank Field \(cricketer\)](#) 0.0000206234084234256
- [Frank Field \(footballer\)](#) 0.0000140844593504505
- [Frank's Field](#) 0.000012134977491988
- [Frank Field](#) 2.96317185423016e-12
- [remove this link](#) [ons](#)

Figure 6.4: *When an ambiguous link is hovered over with the mouse, a set of disambiguation candidates appears, ranked by probability under the LTFM model.*

Editing Politeia (think tank)

Latest revision	Your text
Line 29:	Line 29:
* [[Michael Gove]], Secretary of State for Education	* [[Michael Gove]], Secretary of State for Education
* [[Harold James (historian) Harold James]], Professor of Economic History at Princeton University	* [[Harold James (historian) Harold James]], Professor of Economic History at Princeton University
- * [[Frank Field]], Minister for Welfare Reform (1997-1998)	+ * [[Frank Field (politician) Frank Field]], Minister for Welfare Reform (1997-1998)
* [[Theresa May]], Shadow Secretary of State for Work and Pensions	* [[Theresa May]], Shadow Secretary of State for Work and Pensions
* [[Dr Ludger Schuknecht]], European Central Bank	* [[Dr Ludger Schuknecht]], European Central Bank

B I [Advanced](#) [Special characters](#) [Help](#)

* [[Michael Gove]], Secretary of State for Education
 * [[Harold James (historian)|Harold James]], Professor of Economic History at Princeton University
 * [[Frank Field (politician)|Frank Field]], Minister for Welfare Reform (1997-1998)

Content that violates any copyrights will be deleted. Encyclopedic content must be [verifiable](#).

By clicking the "Save Page" button, you agree to the [Terms of Use](#), and you irrevocably agree to release your contribution under the [CC-BY-SA 3.0 License](#) and the [GFDL](#). You agree that a hyperlink or URL is sufficient attribution under the Creative Commons license.

[Edit summary](#) (Briefly describe the changes you have made)

Disambiguate [[Frank Field]] to [[Frank Field (politician)]] using [[en:Wikipedia:Tools/Navigation_popups|pop

Preview of edit summary: (Disambiguate Frank Field to Frank Field (politician) using popups)

This is a minor edit ([what's this?](#)) Watch this page

[Save page](#) [Show preview](#) [Show changes](#) [Cancel](#) | [Editing help](#) (opens in new window)

If you do not want your writing to be edited, used, and redistributed at will, then do not submit it here. All text that you did not write yourself, except brief excerpts, must be available under terms consistent with Wikipedia's [Terms of Use](#) before you submit it.

Figure 6.5: *Choosing one of the disambiguation candidates automatically creates the edit and edit summary for the change.*

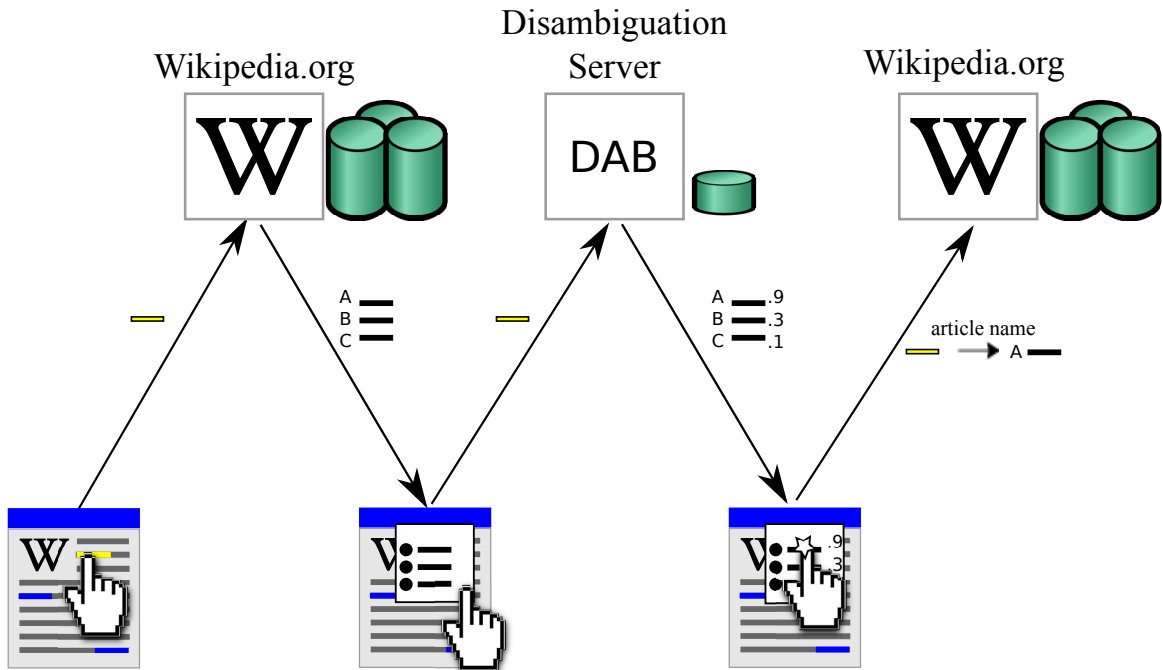


Figure 6.6: When the user's mouse hovers over an ambiguous link, the link is sent to Wikipedia to get the disambiguation candidates. They are displayed in the browser, and the link is sent to the disambiguation server, which scores suggestions and returns the scores to the browser, where they are displayed. The user clicks on a disambiguation candidate, and the changed text is sent to Wikipedia to be stored.

Figure 6.3 demonstrates the highlighting process. To use our extended popups, a user adds a few lines of JavaScript to their Wikipedia JavaScript user page. From then on, whenever the user visits a Wikipedia page, their browser loads and executes a script from our server, making a list of the links on the page, and sends it via an AJAX request to our server for analysis. The server queries the database to see if any are ambiguous, and the identity of any ambiguous links is returned to the user's browser. Then, the browser goes through the links and highlights the ambiguous ones in yellow, as well as providing the user with a count of ambiguous links at the top of the article. An example is illustrated in Figures 6.2 and 6.3.

Figures 6.4, 6.5, and 6.6 illustrate how a user changes an ambiguous link. The

highlights make it easy for a user to see ambiguous links on the page. The user then points their mouse at one such link, and another AJAX request is sent to our server with the list of other links on the page. Our server then performs inference on just the article in question using cached statistics from the other articles. This allows us to work with older copies of the pages for building the initial model, but then use the latest copy of the page being edited in case links have been changed. The distribution over disambiguation candidates is calculated and sent back to the user's browser, where the most likely candidates are displayed in the popup for the user to choose from; the links are listed with the highest-probability links first. When a choice is clicked, the edit is immediately processed and saved by Wikipedia.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Our evaluation technique of mining previous edits to Wikipedia avoids having humans manually assess several hundred disambiguation predictions, as has been done in previous work [26]. Instead, we leverage the work that has already been done by the many Wikipedia editors who have undertaken the manual disambiguation of links, and we can evaluate many thousands of disambiguation predictions without any further human interaction.

Semi-automated disambiguation is a promising approach to tackling the huge number of ambiguous links in Wikipedia. By building a system that incorporates the text and link structure of the rest of Wikipedia, we can be effective at improving this data source, to ultimately improve any application that uses it.

7.2 Future Work

The structure of the LTM model lends itself to modification. We see in the graphical model a structural component identical to LDA. Thus, it should be straightforward to replace that component with another topic model, such as the Hierarchical Dirichlet Process [32] which dynamically chooses an appropriate number

of topics. In addition, this non-parametric Bayesian model makes model selection easier. We hope to investigate such changes in the future.

There are several improvements to consider. One is to try new text similarity scores to see if they provide better results. Another approach would be to train a per-disambiguation-page classification algorithm such as a support vector machine, where the features are the words or links already existing on a page. For an individual disambiguation page with disambiguation candidates that have many inlinks, there may be enough data to train a support vector machine to predict the disambiguation candidate for a link from a given article. Furthermore, it would be possible to combine the scores from all the algorithms we considered into one score using a ranking support vector machine [18].

We would like to compare the effectiveness of these approaches on different language editions of Wikipedia. There may be different factors that affect how they perform: link density and article length are two examples. Also, we would like to incorporate the link structure of one language edition of Wikipedia when making disambiguation decisions for another language. At the word level, this kind of cross-language approach has been shown to be effective in word sense disambiguation [15]. Beyond additional link information, we could extend our LTTM approach by modeling the creation of the non-linked words at the same time as the links.

We do not take into account any features of the edits themselves in our algorithms; it may be that registered users do a better job of creating correct links, and perhaps experienced editors even more so. If true, we could take advantage of this link quality by modifying LTTM to incorporate the editor who added the link

as a visible variable that affects either the topic or a probability that a visible link is actually wrong. Also, the amount of time a link has lasted in an article is a good proxy for validity, as previous experiments have shown with vandalism [27].

Harnessing the edits of Wikipedia users in evaluating algorithms in natural language processing may prove fruitful in many areas. First, the effectiveness of the aggregate “wisdom of crowds” needs to be validated against standard metrics of inter-annotator agreement. Previous work on Wikipedia disambiguation techniques has used human judges to determine accuracy. One project used Amazon Mechanical Turk for the evaluation [26]; users from around the world were paid to assess 449 links in 50 documents. To apply that evaluation for the techniques we discuss here, a subset of the links disambiguated in our tests could be given to humans to manually disambiguate, and then compare the results. Or, we could boldly commit the suggestions to Wikipedia, and observe which of them are corrected.

Bibliography

- [1] B. Thomas Adler et al. *Measuring Author Contributions to the Wikipedia*. Tech. rep. School of Engineering, University of California, 2008.
- [2] Eneko Agirre and Philip Edmonds, eds. *Word Sense Disambiguation: Algorithms and Applications*. Vol. 33. Text, Speech and Language Technology. Springer, 2006. ISBN: 1402048084.
- [3] Alexa.com. *Wikipedia.org Site Info*. [Online; accessed 3-December-2011]. 2011. URL: <http://www.alexa.com/siteinfo/wikipedia.org>.
- [4] Pavel Berkhin. “Bookmark-coloring algorithm for personalized pagerank computing”. In: *Internet Math* 3.1 (2006), pp. 41–62.
- [5] Indrajit Bhattacharya and Lise Getoor. “A Latent Dirichlet Model for Unsupervised Entity Resolution”. In: *SIAM Conference on Data Mining (SDM)*. Winner of the Best Paper Award. 2006.
- [6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. “Latent dirichlet allocation”. In: *The Journal of Machine Learning Research* 3 (2003), pp. 993–1022.
- [7] Kurt Bollacker et al. “Freebase: a collaboratively created graph database for structuring human knowledge”. In: *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. Vancouver, Canada: ACM, 2008, pp. 1247–1250. ISBN: 978-1-60558-102-6. DOI: <http://doi.acm.org/10.1145/1376616.1376746>.
- [8] Sergey Brin and Lawrence Page. “The anatomy of a large-scale hypertextual Web search engine”. In: *Computer Networks and ISDN Systems* (1998). URL: <http://linkinghub.elsevier.com/retrieve/pii/S016975529800110X>.
- [9] Razvan Bunescu and Marius Pasca. “Using Encyclopedic Knowledge for Named Entity Disambiguation”. In: (2006), pp. 9–16. URL: <http://www.cs.utexas.edu/~ml/publication/paper.cgi?paper=encyc-eacl-06.ps.gz>.
- [10] Rudi L. Cilibrasi and Paul M. B. Vitanyi. “The Google Similarity Distance”. In: *IEEE Transactions on Knowledge and Data Engineering* 19.3 (Mar. 2007), pp. 370–383. ISSN: 1041-4347. URL: <http://dx.doi.org/10.1109/TKDE.2007.48>.
- [11] Scott C. Deerwester et al. “Indexing by Latent Semantic Analysis”. In: *Journal of the American Society of Information Science* 41.6 (1990), pp. 391–407.
- [12] Tina Eliassi-Rad and Keith Henderson. “Literature Search through Mixed-Membership Community Discovery”. In: *Advances in Social Computing* 6007.3 (2010), 7078. URL: <http://www.springerlink.com/index/H5L7W0780L56H077.pdf>.
- [13] Gunes Erkan and Dragomir R. Radev. “LexRank: Graph-based lexical centrality as salience in text summarization”. In: *Journal of Artificial Intelligence Research* 22 (2004), pp. 457–479.

- [14] William Gale, Kenneth W. Church, and David Yarowsky. “Estimating upper and lower bounds on the performance of word-sense disambiguation programs”. In: *Proceedings of the 30th annual meeting on Association for Computational Linguistics*. Newark, Delaware: Association for Computational Linguistics, 1992, pp. 249–256. DOI: <http://dx.doi.org/10.3115/981967.981999>.
- [15] William Gale, Kenneth W. Church, and David Yarowsky. “Using Bilingual Materials to Develop Word Sense Disambiguation Methods”. In: *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation: Empiricist vs. Rationalist Methods in MT*. Montreal, Canada, 1992, pp. 101–112.
- [16] Gregor Heinrich. *Parameter Estimation for Text Analysis*. Tech. rep. 2005. URL: <http://www.arbylon.net/publications/text-est2.pdf>.
- [17] Keith Henderson and Tina Eliassi-Rad. “Applying latent Dirichlet allocation to group discovery in large graphs”. In: *Proceedings of the 2009 ACM symposium on Applied Computing* (2009), pp. 1456–1461.
- [18] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. “Support Vector Learning for Ordinal Regression”. In: *International Conference on Artificial Neural Networks*. 1999, pp. 97–102.
- [19] Thomas Hofmann. “Probabilistic Latent Semantic Analysis”. In: *Proceedings of Uncertainty in Artificial Intelligence*. 1999, pp. 289–296.
- [20] Aniket Kittur et al. “Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie”. In: *25th Annual ACM Conference on Human Factors in Computing Systems (CHI 2007)* (2007). URL: <http://www.parc.com/research/publications/details.php?id=5904>.
- [21] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. ISBN: 0521865719.
- [22] Olena Medelyan et al. “Mining meaning from Wikipedia”. In: *International Journal of Human-Computer Studies* 67 (9 2009), pp. 716–754. ISSN: 1071-5819. URL: <http://dl.acm.org/citation.cfm?id=1618876.1619040>.
- [23] Rada Mihalcea. “Using Wikipedia for Automatic Word Sense Disambiguation”. In: *Proceedings of NAACL HLT* (2007). URL: <http://acl.ldc.upenn.edu/N/N07/N07-1025.pdf>.
- [24] Rada Mihalcea and Andras Csomai. “Wikify!: linking documents to encyclopedic knowledge”. In: *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. New York, NY, USA: ACM, 2007, pp. 233–242. ISBN: 9781595938039. URL: <http://dx.doi.org/10.1145/1321440.1321475>.

- [25] Rada Mihalcea and Paul Tarau. “TextRank: Bringing Order into Texts”. In: *Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain, 2004. URL: <http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Mihalcea.pdf>.
- [26] David Milne and Ian H. Witten. “Learning to link with wikipedia”. In: *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*. Napa Valley, California, USA: ACM, 2008, pp. 509–518. ISBN: 978-1-59593-991-3. URL: <http://dx.doi.org/10.1145/1458082.1458150>.
- [27] Martin Potthast, Benno Stein, and Robert Gerling. “Automatic Vandalism Detection in Wikipedia.” In: *European Conference on Information Retrieval*. Ed. by Craig Macdonald et al. Vol. 4956. Lecture Notes in Computer Science. Springer, Apr. 14, 2008, pp. 663–668. ISBN: 978-3-540-78645-0. URL: <http://dblp.uni-trier.de/db/conf/ecir/ecir2008.html#PotthastSG08>.
- [28] Reid Priedhorsky et al. “Creating, destroying, and restoring value in wikipedia”. In: *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work*. Sanibel Island, Florida, USA: ACM, 2007, pp. 259–268. ISBN: 978-1-59593-845-9. URL: <http://portal.acm.org/citation.cfm?id=1316624.1316663>.
- [29] Radim Řehůřek and Petr Sojka. “Software Framework for Topic Modelling with Large Corpora”. English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. <http://is.muni.cz/publication/884893/en>. Valletta, Malta: ELRA, May 2010, pp. 45–50.
- [30] Issei Sato, Kenichi Kurihara, and Hiroshi Nakagawa. “Deterministic Single-Pass Algorithm for LDA”. In: *Advances in Neural Information Processing Systems 23*. Ed. by J. Lafferty et al. 2010, pp. 2074–2082.
- [31] Yee Whye Teh, David Newman, and Max Welling. “A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation”. In: *Conference on Neural Information Processing Systems*. 2006. URL: http://books.nips.cc/papers/files/nips19/NIPS2006_0511.pdf.
- [32] Yee Whye Teh et al. “Hierarchical Dirichlet Processes”. In: *Journal of the American Statistical Association* 101.476 (2006), pp. 1566–1581. URL: <http://pubs.amstat.org/doi/abs/10.1198/016214506000000302>.
- [33] Hanna M. Wallach. “Structured Topic Models for Language”. PhD thesis. University of Cambridge, 2008.
- [34] Hanna M. Wallach, David Mimno, and Andrew McCallum. “Rethinking LDA: Why Priors Matter”. In: *Conference on Neural Information Processing Systems*. 2009. URL: http://books.nips.cc/papers/files/nips22/NIPS2009_0929.pdf.
- [35] Wikipedia. *Wikipedia Statistics English*. [Online; accessed 2-September-2011]. 2011. URL: <http://stats.wikimedia.org/EN/TablesWikipediaEN.htm>.

- [36] Wikipedia. *Wikipedia:Disambiguation* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 3-December-2011]. 2011. URL: <http://en.wikipedia.org/w/index.php?title=Wikipedia:Disambiguation&oldid=462966392>.
- [37] Wikipedia. *Wikipedia:Manual of Style (disambiguation pages)* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 3-December-2011]. 2011. URL: [http://en.wikipedia.org/w/index.php?title=Wikipedia:Manual_of_Style_\(disambiguation_pages\)&oldid=447636436](http://en.wikipedia.org/w/index.php?title=Wikipedia:Manual_of_Style_(disambiguation_pages)&oldid=447636436).
- [38] Wikipedia. *Wikipedia:Size comparisons* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 3-December-2011]. 2011. URL: http://en.wikipedia.org/w/index.php?title=Wikipedia:Size_comparisons&oldid=451578590.
- [39] Limin Yao, David Mimno, and Andrew McCallum. “Efficient methods for topic model inference on streaming document collections”. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '09. Paris, France: ACM, 2009, pp. 937–946. ISBN: 978-1-60558-495-9. URL: <http://doi.acm.org/10.1145/1557019.1557121>.