First-Order Probabilistic Models for Information Extraction

Bhaskara Marthi
Computer Science Div.
University of California
Berkeley, CA 94720-1776
bhaskara@cs.berkeley.edu

Brian Milch
Computer Science Div.
University of California
Berkeley, CA 94720-1776
milch@cs.berkeley.edu

Stuart Russell
Computer Science Div.
University of California
Berkeley, CA 94720-1776
russell@cs.berkeley.edu

Abstract

Information extraction (IE) is the problem of constructing a knowledge base from a corpus of text documents. In this paper, we argue that first-order probabilistic models (FOPMs) are a promising framework for IE, for two main reasons. First, FOPMs allow us to reason explicitly about entites that are mentioned in multiple documents, and compute the probability that two strings refer to the same entity — thus addressing the problem of *coreference* or *record linkage* in a principled way. Second, FOPMs allow us to resolve ambiguities in a text passage using information from the whole corpus, rather than disambiguating based on local cues alone and then trying to merge the results into a coherent knowledge base. This paper presents a comprehensive FOPM for a bibliographic database, and explains how the desired inference patterns emerge from the model.

1 Introduction

1.1 Information extraction

Information extraction (IE) is the problem of constructing a knowledge base from a corpus of text documents. Some IE systems extract information from ordinary English prose: for instance, the Message Understanding Conferences [DARPA, 1998] have evaluated systems that extract information about changes of corporate management, airplane crashes, and rocket launches from *Wall Street Journal* articles. Other systems extract information that is presented in highly formatted headers, lists, and tables rather than in complete sentences. For instance, Citeseer [Lawrence *et al.*, 1999a] and Cora [McCallum *et al.*, 2000b] build databases of academic publications; FlipDog [Cohen *et al.*, 2000a] builds a database of job openings from companies' employment web pages; and Froogle [Google Inc., 2003] builds a database of product offers from online stores.

Natural language prose is notoriously ambiguous, and even highly formatted documents (such as web pages listing job openings) can be hard to interpret automatically. An even harder task is combining information from multiple documents into a single coherent knowledge base. In this paper, we argue that first-order probabilistic models (FOPMs) are a promising framework for IE. Because FOPMs allow us to explicitly represent uncertainty about how many objects are in the world and what relations hold between them, we can use a single probabilistic model for everything from parsing or segmenting the text, to inferring object attributes, to inferring relations between objects.

1.2 Advantages of a comprehensive model

One advantage of using such a comprehensive probabilistic model is that we can reason explicitly about *identity uncertainty* — for instance, whether two citations refer to the same publication. This problem has been treated extensively in natural language processing under the name *coreference resolution*, but methods for resolving coreference across documents remain mostly heuristic. In the bibliography domain, resolving identity uncertainty is important both to avoid having duplicate entries for publications and authors in our final database, and so we can assemble more complete descriptions of publications and authors from multiple citations.

A further advantage of having a comprehensive probabilistic model is that we can use cross-document information to disambiguate text. For example, suppose we see a citation that begins, "Wauchope, K. Eucalyptus: Integrating Natural Language Input with a Graphical User Interface". Is "Eucalyptus" part of the title, or is it the author's middle name? If we see other similar citations where the formatting clearly indicates that "Eucalyptus" is part of the title, then the most likely explanation is that all these citations refer to a single publication with "Eucalyptus" in the title, rather than there being two publications, one with "Eucalyptus" in the title and one without. Conversely, if we see another paper by "K. E. Wauchope", it is more likely that "Eucalyptus" is a middle name. As discussed in Section 3.2, a FOPM for the bibliography domain allows this kind of cross-citation disambiguation. Such disambiguation would not be possible if we just chose the most likely segmentation for each citation based on local cues, and passed these results to another layer of the system for merging into a coherent database. That is, processes that are normally bottom-up and opaque to the higher levels of the systems should instead be *cognitively penetrable*, to borrow a phrase from [Pylyshyn, 1984].

1.3 Knowledge base functionality

Once we have created a knowledge base, what would we like to do with it? One application is allowing a user to browse the data and follow hyperlinks between entities: for instance, from a paper, to one of its authors, to other papers by that author. We would also like to support queries about an entity's attributes, such as an author's full name or the page numbers of a journal paper. Finally, we would like to support structured search queries, like "Find all papers by Mike Jordan in UAI '97". One possible answer to such a query is "the system has not seen any citations to such a paper". However, we would like our system to distinguish between the case where it has simply not seen any evidence for the existence of such a paper, and the case where it is very sure no such paper exists—perhaps because it has parsed Mike Jordan's publications page (or the UAI '97 conference program) and seen no such paper. Thus, our knowledge base will need to do more than just store lists of known entities and their attributes.

1.4 Paper overview

Pasula et al. [Pasula *et al.*, 2003] have already applied a FOPM to the bibliography domain. However, that paper discusses a simple model where the only entities are publications and authors, and results are reported only for resolving coreference among citations. The purpose of this paper is to bring the general IE problem to the attention of the FOPM community, and to show how a FOPM can serve as a comprehensive model for an IE task. We use the bibliography domain as our example, but we believe the advantages of a FOPM for coreference resolution and joint disambiguation will be even more important in more complex domains.

We do not assume any particular representation language for the FOPM in this paper. Instead, we focus on the properties of the model itself, particulary how it supports the kinds of reasoning discussed above. Our notation is based on that used in relational probability models (RPMs) [Pfeffer, 2000], but we are not concerned about whether all the complexities of the model can be expressed by an RPM. Later in the paper, we briefly discuss features that would be desirable in a first-order probabilistic language for specifying IE models.

2 Model for the Bibliography Domain

In this section, we describe our probabilistic model of the citation domain. The model, which is an expanded version of the one presented in [Pasula *et al.*, 2003], includes several classes of objects – authors, publications, collections, citation groups, and citations – and its possible worlds consist of the objects and their attributes and relations.

We do not discuss inference or learning in this section, and indeed, exact inference in the model is probably intractable. However, rather than building many approximating assumptions into the model itself, we choose to make the model as rich as possible and perform any approximations during inference. The parameters will be learnt either using Monte-Carlo EM [Tanner and Wei, 1990] or using supervised methods.

2.1 Classes and attributes

Our model has the following generative structure. First, the set of Author objects, and the set of Collection objects are generated independently. Next, the set of Publication objects is generated conditional on the Authors and Collections. After this, CitationGroup objects are generated conditional on the Authors and Collections, and finally, Citation objects are generated from the CitationGroups. We now describe each of these parts in more detail.

Authors

The number of authors who write papers in this field is chosen from a slowly decreasing log-normal prior. Each Author object has an attribute name, which is chosen from a mixture of a letter bigram distribution with a distribution that chooses from a set of commonly occurring names. There is also a multinomial attribute area, which specifies the field this author usually writes papers in (to be more realistic, we could also have multiple such attributes).

Publications

Each publication has attributes area and type which are chosen according to multinomial distributions. Example types include books, conference papers, and journal papers (alternatively, we could have subclasses of publication corresponding to each type, in which case there would be 'class uncertainty'). Publications also have a compound attribute authorList, generated as follows: first, the length of the list is chosen. Next, for each position i in the list, a reference attribute authorList[i] is chosen (by *reference attribute*, we mean an attribute whose value is another object). Most of the time, this attribute is chosen uniformly from the set of authors whose area attribute equals this publication's area, but there is also some probability of choosing uniformly from all the authors. The attribute title is generated from an n-gram model, conditioned on area (this captures the fact that each area has its own commonly used technical terms).

If the publication is of a type that is usually part of a larger collection, such as a conference paper, the collection reference attribute is set, again depending on area, and date and publisher are set to equal collection.date and collection.publisher, respectively. If not, date is generated from a prior distribution, and publisher is chosen uniformly from the set of publishers. A publication may also have other attributes, such as a number for a technical report, which are chosen using appropriate prior distributions.

Publishers

This class has name and city attributes. Instances for the commonly used publishers are included as evidence, and there is a prior that allows for previously unseen publishers.

Collections

A Collection is a journal issue, a book of conference proceedings, or a book that is a collection of articles. It has string attributes name and date, a multinomial attribute type, and a reference attribute publisher.

Citation Groups

Citations often occur in groups. Examples include a reference list at the end of a paper, a bibliography on a particular topic, the publications section of a researcher's homepage, or the table of contents of conference proceedings. The CitationGroup class captures some of the structure present in these groups. To begin with, there is is an attribute type, which takes values in $\{\text{refList}, \text{bibliography}, \text{tableOfContents}, \text{homePage}, \text{other}\}$. Next, there is a multinomial attribute style, depending on type, that selects from a dictionary of common bibliography styles (there will also be an 'other' style, to model styles that are not in the dictionary).

The CitationGroup class also contains a compound variable publicationList, which is a list of Publication objects. If type $\in \{\text{refList}, \text{other}\}$, this is generated by picking the list length and then sampling independently from a uniform distribution over the publications. If type = bibliography, then the CitationGroup has an area attribute and we sample only from publications with the same area value.

If type = homePage (the case of tableOfContents is analogous), then there is a reference attribute author and a Boolean attribute exhaustive. If exhaustive, then publicationList is the set of Publication objects p such that p.author = author. If not, we need a model for selecting a subset of this set (we assume that there is no repetition within such lists). A simple way to do this is to independently include each member with some probability θ, but more complicated distributions are possible, for example to list only publications before a certain date.

Finally, this class contains a compound variable citationList, of the same length as publicationList. The elements of this list are Citation objects, and each element depends on the corresponding element in publicationList, in a manner specified in the next section.

Citations

A citation is generated conditional on the cited publication, which is the value of the citation's pub attribute. In any CitationList object ℓ, we require that $\ell.\text{citationList}[i].\text{pub} = \ell.\text{publicationList}[i]$. A Citation object also has several 'as cited' attributes that correspond to how the true attributes of the publication are 'corrupted' while creating this citation. As an example, the conditional distribution of titleAsCited given pub.title includes probabilities of misspelling based on edit distance, of abbreviating common technical terms (e.g. "HMM"), and of dropping words like "the". Once again, we have an elementwise dependency between two lists, this time between authorsAsCited and pub.authorList.

There is also an attribute parse that specifies how the various parts are ordered to produce the citation text. It depends on the style attribute of the containing citation list, as well as on pub.type and, if necessary, pub.collection.type (since, for example, journal articles are usually cited differently from conference papers). We use a PCFG for this, but other models such as HMMs are possible.

Finally, there is an attribute text, which will usually be observed. This attribute has a deterministic distribution, which involves filling in the structure found in parse with the text of the asCited attributes.

2.2 Examples

We have specified a rich probabilistic model of the citation domain, but this richness comes at a computational cost. We now argue that this cost is justified, by giving some examples where the model leads to plausible conclusions that would be difficult to reach using simpler methods. Of course, empirical tests would be needed to make the argument conclusive.

In Figure 1, the journal name could potentially refer to either *Journal of Artificial Intelligence Research*, or *Artificial Intelligence Journal*. Suppose the model has previously come across the table of contents for *AIJ* 1996, which is known to be an exhaustive list. None of the citations in that list resembles this one, and so the model would yield a low probability for the hypothesis that one of those papers produced this citation. If the model has not seen an exhaustive list for *JAIR*, it is free to hypothesize the existence of a paper from *JAIR* 1996 whose title is very similar to this one, and would conclude that the paper was published in *JAIR*.[1]

In Figure 2, the model would assign high probability to the event of the citations referring to the same publication, as they have the same title and year of publication. As a result, information from both citations will be combined when inferring the attributes of the underlying publication — the first citation contains the correct conference name, while the second one contains the author's full name, which could be useful if there are other Hegers in the knowledge base.

3 Properties of the Model

3.1 Handling identity uncertainty

One desirable property of our model is that it allows us to reason explicitly about whether two citations refer to the same publication, or whether two papers are written by the same author. For example, although the two citations in Figure 2 look different, we are quite sure they refer to the same publication. In this section, we explain how our model can yield the same conclusion.

A simple scenario

To build intuition, we begin with a very simple scenario, isomorphic to the "balls in an urn" example in [Russell, 2001]. Suppose a library contains n books $b_1, \ldots, b_n$. For now, the only attribute of a book that we will consider is its title: for any b_i, let $P(b_i.\text{title} = x) = P_X(x)$. We create a citation list by repeatedly selecting a book uniformly at random from the library, writing down its title (with some probability of making an error), and returning the book to the shelf. For any citation c, let $P(c.\text{text} = y \mid c.\text{pub.title} = x) = P_Y(y|x)$. Thus, P_Y models the process by which titles are corrupted as we write them down.

Now suppose we are looking at a citation list with two citations c_1 and c_2, whose text strings are y_1 and y_2. We have two hypotheses about whether the citations refer to the same book:

$$H_1: \quad c_1.\text{pub} = c_2.\text{pub}$$
$$H_2: \quad c_1.\text{pub} \neq c_2.\text{pub}$$

We can evaluate the posterior probability that the citations co-refer by comparing the joint probabilities of the two hy-

[1] A third possibility, that this is a previously unseen journal, would be deemed unlikely thanks to the Occam's razor effect discussed in the next section.

```
Helzerman, R. A., and Harper, M. P. 1996.  MUSE CSP: An extension to the constraint
satisfaction problem.  Journal of Artificial Intelligence
```

Figure 1: Disambiguating a journal name

```
Heger, M. (1994).  Consideration of risk in reinforcement learning.  In Proceedings
of the Eleventh International Conference on Machine Learning, pages 105-111, San
Francisco, CA. Morgan Kaufmann.

[Heger, 1994] Heger, Matthias 1994.  Consideration of risk in reinforcement learn-
ing.  In Proceedings of the Machine Learning Conference.  To appear.
```

Figure 2: Combining information from multiple citations

potheses with the evidence:

$$p_1 = P(H_1, c_1.\mathsf{text} = y_1, c_2.\mathsf{text} = y_2)$$
$$p_2 = P(H_2, c_1.\mathsf{text} = y_1, c_2.\mathsf{text} = y_2)$$

Since we choose books uniformly from the n books in the library, the prior probability of H_1 is $1/n$.

$$p_1 = \frac{1}{n} P(c_1.\mathsf{text} = y_1, c_2.\mathsf{text} = y_2 \mid H_1)$$
$$p_2 = \frac{n-1}{n} P(c_1.\mathsf{text} = y_1, c_2.\mathsf{text} = y_2 \mid H_2)$$

To compute $P(c_1.\mathsf{text} = y_1, c_2.\mathsf{text} = y_2 \mid H_1)$, we must sum over all possible values x for $c_1.\mathsf{pub.title}$. To compute $P(c_1.\mathsf{text} = y_1, c_2.\mathsf{text} = y_2 \mid H_2)$, we must sum over both $c_1.\mathsf{pub.title}$ and $c_2.\mathsf{pub.title}$. The results are as follows:

$$p_1 = \frac{1}{n} \sum_x P_X(x) P_Y(y_1|x) P_Y(y_2|x) \quad (1)$$

$$p_2 = \frac{n-1}{n} \left(\sum_{x_1} P_X(x_1) P_Y(y_1|x_1) \right) \left(\sum_{x_2} P_X(x_2) P_Y(y_2|x_2) \right) \quad (2)$$

Occam's razor

So which is greater, p_1 or p_2? Of course, the answer depends on our probability models for book titles and string corruptions, as well as on n. We can gain some insight by considering the case where no string corruption occurs: $P_Y(y_1|x_1) = 1$ if $y_1 = x_1$ and 0 otherwise. Obviously, under this model, H_1 has probability zero when $y_1 \neq y_2$. So suppose $y_1 = y_2 = y$. Then all the terms in the summations where $x \neq y$ are zero, and we have:

$$p_1 = \frac{1}{n} P_X(y)$$
$$p_2 = \frac{n-1}{n} P_X(y)^2$$

These equations make sense: if H_1 is true, then there is at least one book with title y, but if H_2 is true, there are at least *two* books with title y, so the title probability is squared.

The fact that the title probability is squared in p_2 penalizes H_2 for constraining the values of more hidden variables than H_1 does. The penalty is especially strong because a reasonable prior over publication titles has high entropy: the probability of a typical title might be 10^{-7}. Then if we are selecting from a library of 100,000 books, the posterior probability of H_1 is about 100 times that of H_2. The posterior probabilities only become equal when the library size is about 10^7. Thus, Occam's razor — a preference for hypotheses that explain the observed data using few hidden objects — arises naturally from our model. This effect has been analyzed in the literature on Bayesian model selection since the work of Jeffreys [Jeffreys, 1939]; see [MacKay, 1992] for a more recent overview of the topic.

On the other hand, Occam's razor does not always dominate the computation. Suppose that instead of choosing books from a library and writing down their titles, we are choosing people from a phone book and writing down their first names. The distribution over first names has much lower entropy than the distribution over book titles: for instance, the 1990 census indicated that between 1% and 2% of people in the U.S. were named Mary. So if we select from a phone book with 100,000 entries and get two people named Mary, then p_1 is about 10^{-7} and p_2 is about 10^{-4}: the probability that the two occurrences of Mary are two different people is about 0.999.

String corruptions

Now let us return to the case where the citation text may be an imperfect copy of the book's title. For instance, suppose y_1 = "Doctor Zhivago" and y_2 = "Doctor Zivago". For concreteness, assume $P_X(y_1) = P_X(y_2) = 10^{-7}$; writing "Zhivago" as "Zivago" or vice versa has probability 10^{-3}; and writing the titles correctly has probability close to 1. Also, to make the computations simple, assume all other strings are either extremely unlikely titles, or extremely unlikely to be transcribed as "Doctor Zhivago" or "Doctor Zivago". Then when we substitute into Equations (1) and (2), most of the terms in the summations are near zero, and we can approximate the probabilities as follows:

$$p_1 \approx \frac{1}{n} \left((P_X(y_1) \cdot 1 \cdot 10^{-3}) + (P_X(y_2) \cdot 10^{-3} \cdot 1) \right)$$
$$\approx \frac{1}{n} (2 \cdot 10^{-10})$$
$$p_2 \approx \frac{n-1}{n} (P_X(y_1) \cdot 1)(P_X(y_2) \cdot 1)$$
$$\approx \frac{n-1}{n} (10^{-14})$$

Thus, H_1 has greater posterior probability than H_2 if there are fewer than about 20,000 books in the library. The Occam's razor effect appears here too: H_2 must "pay the cost" of generating each observed title independently, whereas H_1 only "pays" for one title generation and one copying error.

Of course, if y_1 and y_2 are quite different strings, such as "Doctor Zhivago" and "Doctor Dolittle", then the specific set of copying errors necessary to transform one to the other will be less likely than the generation of the title itself, and H_2 will have greater posterior probability.

Unknown numbers of publications

So far, we have assumed the number of books in the library is a known value n. It does not complicate things much to make the number of books a random variable N, with a prior distribution $P_N(n)$. Then, to evaluate hypotheses about coreference, we must sum over the possible values of N. Equations (1) and (2) become:

$$\begin{aligned} p_1 &= \sum_n P_N(n)\left(\frac{1}{n}\right)\sum_x P_X(x)P_Y(y_1|x)P_Y(y_2|x) \\ p_2 &= \sum_n P_N(n)\left(\frac{n-1}{n}\right)\left(\sum_{x_1} P_X(x_1)P_Y(y_1|x_1)\right) \\ &\quad \left(\sum_{x_2} P_X(x_2)P_Y(y_2|x_2)\right) \end{aligned}$$

We can also obtain a posterior distribution over N given the observed citations. This involves summing over all possible mappings from citations to publications, as well as summing over publication titles. Formally, let $\mathbf{x} = x_1, \ldots, x_N$ range over assignments of titles to all the publications. Suppose we have seen K citations. Let $\mathbf{y} = y_1, \ldots, y_K$ be the observed titles of the citations and let $\omega = \omega_1, \ldots, \omega_K$ range over mappings from citations to publications. Then $P(N = n|\mathbf{y})$ is proportional to:

$$P_N(n)\sum_{\mathbf{x}}\left(\prod_{i=1}^{n} P_X(x_i)\right)\sum_{\omega}\left(\frac{1}{n}\right)^K\left(\prod_{i=1}^{K} P_Y(y_i|x_{\omega_i})\right)$$

This is analogous to the equation given for balls in an urn in [Russell, 2001]. Intuitively, if we observe the same titles over and over, we will believe there are few books in the library; if we very seldom see the same title twice, we will believe the library is large.

Identity uncertainty in complex models

This section has discussed identity uncertainty in a simplified scenario: writing down the titles of books from a library. Working with the complete bibliography model described in Section 2 introduces two complications. First, the probability models for publication attributes and citation strings are more complex. If c is a citation, then c.text depends not only on c.pub.title, but also on c.pub.author[1].name, c.pub.date, c.pub.collection.name, and so on. So to compute the probability that two particular citations co-refer, we need to sum over the possible values of many complex and simple attributes (in practice, we must approximate these sums). Furthermore, two citations of the same publication may differ from each other not because of errors, but simply because they use different formatting and abbreviations.

The second complication is that we are dealing with identity uncertainty for all classes simultaneously: publications, authors, publishers, etc. We may be uncertain not just about whether c_1.pub.author[1] $= c_2$.pub.author[3], but also about whether c_2.pub even has a third author, and whether c_2.pub $= c_1$.pub. We can make sense of all this uncertainty if we think in terms of distributions over logical interpretations (possible worlds). However, these multiple layers of identity uncertainty pose challenges for both representation languages and inference algorithms.

3.2 Cross-citation disambiguation

Another useful property of our model is that it can resolve ambiguities in a citation by using information from other citations. For example, consider the citations in Figure 3. The first citation is ambiguous: it could be that the author's name is K. Eucalyptus Wauchope, or "Eucalyptus" could be part of the paper's title. Of course, a human reader who knew of Kenneth Wauchope and his Eucalyptus system — perhaps from seeing other citations of this paper — would have no trouble seeing that "Eucalyptus" is part of the title. In this section, we show how our model can also disambiguate the first citation using other citations, such as the second one in Figure 3.

Ambiguity given a single citation

To begin with, suppose we observe only the first citation c_1, whose text is y_1. There are two likely hypotheses:

$$A_1 = \begin{cases} c_1.\mathsf{authorsAsCited}[1] = \text{“Wauchope, K.”} \\ c_1.\mathsf{titleAsCited} = \text{“Eucalyptus: Integrating...”} \end{cases}$$
$$A_2 = \begin{cases} c_1.\mathsf{authorsAsCited}[1] = \text{“Wauchope, K. Eucalyptus”} \\ c_1.\mathsf{titleAsCited} = \text{“Integrating...”} \end{cases}$$

We can compare the joint probabilities:

$$\begin{aligned} q_1 = P(A_1, c_1.\mathsf{text} = y_1) = P(A_1)P(c_1.\mathsf{text} = y_1|A_1) \\ q_2 = P(A_2, c_1.\mathsf{text} = y_1) = P(A_2)P(c_1.\mathsf{text} = y_1|A_2) \end{aligned}$$

Suppose our our title model and our author name model assign about the same probability to an unusual word like "Eucalyptus". Then $P(A_1) \approx P(A_2)$. And if the author-title separator is about equally likely to be a period or a colon, then $P(c_1.\mathsf{text} = y_1|A_1) \approx P(c_1.\mathsf{text} = y_1|A_2)$. So $q_1 \approx q_2$.

Using a second citation

Thus, looking at c_1 alone, a reasonable model assigns equal posterior probabilities to the two hypotheses. But suppose we also observe c_2 (the second citation in Figure 3), whose text is y_2. An ideal model would specify that an institution is unlikely to issue multiple tech reports with the same number: so unless the first publication was issued by some other "NRL" rather than the Naval Research Laboratory, the two citations must co-refer. However, in the model described in Section 2, tech report numbers are chosen independently for each publication. So we must rely on Occam's razor to give high probability to the hypothesis that c_1.pub $= c_2$.pub. As shown in Section 3.1, our model prefers this hypothesis because it requires the tech report number (and most of the title) to be generated only once rather than twice.

```
Wauchope, K. Eucalyptus: Integrating Natural Language Input with a Graphical User
Interface. NRL Report NRL/FR/5510-94-9711 (1994).

Kenneth Wauchope (1994). Eucalyptus: Integrating natural language input with a
graphical user interface. NRL Report NRL/FR/5510-94-9711, Naval Research Laboratory,
Washington, DC, 39pp.
```

Figure 3: A pair of citations where the second helps to disambiguate the first.

So most of the posterior probability mass is on worlds where c_1 and c_2 corefer. In y_2, the date is a clear delimiter between the author list and the title, so with probability close to one:

$$\begin{aligned} &c_2.\mathsf{authorsAsCited}[1] = \text{“Kenneth Wauchope”} \\ &c_2.\mathsf{titleAsCited} = \text{“Eucalyptus: Integrating...”} \end{aligned} \quad (3)$$

This is consistent with A_1: if the publication attributes are $c_1.\mathsf{pub.authorList}[1].\mathsf{name}$ = “Kenneth Wauchope” and $c_1.\mathsf{pub.title}$ = “Eucalyptus: Integrating...”, then the c_1 attributes in A_1 and the c_2 attributes in (3) have high probability. Note that this explanation only requires the word “Eucalyptus” to be generated once, as part of the title. On the other hand, if A_2 is true, then “Eucalyptus” occurs in the author name in c_1 and the title in c_2. This is not impossible: it could be that “Eucalyptus” was inserted accidentally in one of the citations; or perhaps both the true title and the true author name include the word “Eucalyptus”, but it was accidentally deleted from the title in c_1. But these explanations are orders of magnitude less likely than the explanation consistent with A_1, so A_1 has greater posterior probability.

Thus, when local cues are insufficient for parsing a citation, our model gives a probability “bonus” to parses that are consistent with the parses of other co-referring citations. Parsing is done as part of the overall inference process, incorporating such top-down information. Note that this approach does not require lists of known author names, paper titles, or journal titles: we are just taking a potentially large set of unlabeled citations and using them to disambiguate each other.

A more difficult example

We must admit that it took some effort to find a citation where the the distinction between authors and title was truly ambiguous. However, there are other domains where fewer formatting cues are available, and word or character n-gram models are less helpful for distinguishing the values of different attributes. As an extreme example, the radio station WPTC displays the artists and titles of songs on its playlist in two unlabeled columns: [2]

```
The Used           Maybe Memories
From Zero          Smack
V Ice              Nothing is Real
Burnt by the Sun   Soundtrack to the
                     Worst Movie Ever
Tsunami Bomb       Take the Reigns
Squirt             Mr. Normal
```

The reader is challenged to tell which column is which. Clearly, it would help to find other mentions of these artists and titles where their roles are less ambiguous.

[2] `http://www.pct.edu/wptc/playlist2.html`

4 Desiderata for a FOPL

In section 2, we gave an informal description of our model. Our current implementation essentially requires the details of the model to be hardcoded in. Such an approach will not scale as we build models for many different IE tasks: it would be desirable to have a declarative language for specifying such models. Based on our experience in modeling this domain, here are some of the features we think such a first-order probabilistic language (FOPL) should have:

- A probability distribution over possible worlds which contain objects, functions, and relations.
- Uncertainty about the number of objects in the world, and the ability to make inferences about the existence or nonexistence of objects having particular properties.
- Uncertainty about the relational structure of the world. It is often, as in the citation domain, not possible to specify this structure beforehand.
- The ability to answer queries about all aspects of the world, including the relational and object structure.
- The ability to represent common types of compound objects such as lists and finite sets, and common probability distributions for dependencies between them, such as models for selecting a subset of a set, and models for elementwise dependencies between lists
- The ability to represent probabilistic dependencies that don’t have a natural generative structure, such as the dependence between authors, topics, and papers.
- An efficient inference algorithm with provable guarantees on accuracy and computational complexity, and ways to adjust the tradeoff between these two.
- The ability to incorporate domain knowledge into the inference algorithm. For example, in MCMC this knowledge can be used to design a proposal distribution.
- A learning procedure which allows priors over the parameters.

5 Inference

Because exact inference in our model is intractable, we use MCMC [Gilks *et al.*, 1996; Andrieu *et al.*, 2003] as our inference procedure. Specifically, we use a Metropolis-Hastings proposal distribution, the details of which are described in [Pasula *et al.*, 2003]. This proposal includes moves that create and destroy objects, as well as moves that change the attributes of existing objects. This last type of move includes changes to the parse tree of a citation, thus allowing top-down information to be used to resolve uncertainty about the parse.

An important point is that, for most queries, if an object is not referred to by any other objects in the current state, then we don't need to waste time resampling its attributes. This allows us to reason efficiently about worlds with a large number of unseen papers. However, if we are answering queries like "How many papers has Mike Jordan published at UAI?", we are forced to sample attributes of all papers, and so these queries are more difficult.

Designing efficient general-purpose MCMC algorithms for first-order models remains a challenging open problem. We are investigating several possibilities for speeding convergence. *Query-dependent sampling* is based on the idea that when answering a query that only depends on the marginal distribution of a small subset of the variables, we should focus our sampling near those variables. [Marthi *et al.*, 2002] described how to do this for a specific graph structure, but the idea is more broadly applicable. *Rao-Blackwellization* is a technique that can be used when some of the variables are amenable to exact inference conditional on their Markov blanket. These variables then don't need to be sampled, as we can marginalize them out. Finally, a common approximation technique is to replace a distribution by a reweighted distribution over its k most likely values. This is useful for sampling variables with large domains, such as parse trees.

Besides sampling, the other major family of approximate inference algorithms is that of variational approximations. In the future, we hope to apply generalized variational inference [Xing and Russell, 2003] and generalized belief propagation [Yedidia *et al.*, 2001] in this domain, and compare their performance to MCMC.

6 Related Work

6.1 Existing work in IE

A great deal of work on extracting information from news articles is described in the MUC proceedings (most recently [DARPA, 1998]); examples of work on highly formatted text include [McCallum *et al.*, 2000b; Lafferty *et al.*, 2001; Cohen *et al.*, 2002]. However, most IE work has not focused on combining information from multiple documents. IE researchers have made considerable progress on resolving coreference *within* documents, e.g., between nouns and pronouns; see [Harabagiu *et al.*, 2001] and references therein. There has been less work on cross-document coreference resolution, but [Bagga and Baldwin, 1999] describes a method for detecting mentions of the same event in different news stories, and [Lawrence *et al.*, 1999b; McCallum *et al.*, 2000a] discuss coreference among citations.

There has been considerable work on *record linkage*, the task of finding and merging duplicate entries in databases [Fellegi and Sunter, 1969; Cohen *et al.*, 2000b; Bilenko and Mooney, 2002]. However, record linkage algorithms typically take database tuples as input, while we are starting with unsegmented text. Of course, one could do IE to obtain database tuples and then find duplicates with a record linkage algorithm. But then one would not be able to disambiguate text by finding other mentions of the same entities, as our proposed system does.

Our work can be seen as a fusion of information extraction, which deals with the relationship between facts and text, and data mining, which deals with statistical regularities in the facts themselves. Nahm and Mooney [Nahm and Mooney, 2000] have implemented such a combined system, called DISCOTEX, for extracting information about job openings from newsgroup postings. Their system learns association rules between fields (analogous to our prior model over object attributes) and uses these rules to improve the recall of an IE system. Another example of using domain knowledge to improve IE is the DATAMOLD system [Borkar *et al.*, 2001], which was applied to parsing postal addresses. DATAMOLD has a database of containment relationships between cities, provinces, and countries, and prefers parses that include city-country pairs where the city is known to be in that country. If we used a FOPM for this task, we would hope to infer the geographic relationships while parsing the addresses.

6.2 Bayesian modeling

Another way to think about our probabilistic model would be to say that all the unobserved attributes are parameters of the model: then the prior distributions over these parameters become parameter priors, and the problem of choosing how many hidden objects there are (or computing a posterior distribution over the number of hidden objects) is one of model selection (or model averaging). This Bayesian model selection problem has been tackled, for example, by [Green, 1995] using an MCMC inference method.

Researchers in other branches of AI have used similar models where the observed data is generated by first generating some hidden objects, then generating a correspondence between observations and hidden objects, and finally generating the values of the observations conditioned on their corresponding hidden objects. Applications of such models include robot localization [Anguelov *et al.*, 2002], recovering the 3D structure of an object from multiple images [Dellaert *et al.*, 2003], and finding stochastically repeated patterns (motifs) in DNA sequences [Xing *et al.*, 2003]. However, not all these models are fully Bayesian: [Dellaert *et al.*, 2003] estimate the positions of visual features (corner points, etc.) on objects using maximum likelihood. They note that this strategy is feasible only because they assume that in each image, the mapping from observed features to actual features is one-to-one. Thus, there is no question about the number of hidden objects (features), and no need for the Occam's razor effect provided by a fully Bayesian approach.

7 Conclusions

We have argued that first-order probabilistic models are a useful, probably necessary, component of any system that extracts complex relational information from unstructured text data. We presented an example of such a model for one particular information extraction task. Many desirable features of plausible reasoning, such as a preference for simple explanations and the combination of top-down and bottom-up information, which are lacking in most nonrelational or nonprobabilistic IE systems, occur naturally in our model.

Some of the directions we plan to pursue in the future include defining a representation language that allows such models to be specified declaratively, scaling up the inference procedure to handle large knowledge bases, and tackling domains where the observed text is even less structured.

References

[Andrieu *et al.*, 2003] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

[Anguelov *et al.*, 2002] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, and S. Thrun. Learning hierarchical object maps of non-stationary environments with mobile robots. In *Proc. 18th UAI*, 2002.

[Bagga and Baldwin, 1999] A. Bagga and B. Baldwin. Cross-document event coreference: Annotations, experiments, and observations. In *Proc. ACL-99 Workshop on Coreference and Its Applications*, pages 1–8, 1999.

[Bilenko and Mooney, 2002] M. Bilenko and R. J. Mooney. Learning to combine trained distance metrics for duplicate detection in databases. Technical Report AI 02-296, AI Lab, Univ. of Texas at Austin, 2002.

[Borkar *et al.*, 2001] V. Borkar, K. Deshmukh, and S. Sarawagi. Automatic segmentation of text into structured records. In *Proc. ACM SIGMOD Conf.*, 2001.

[Cohen *et al.*, 2000a] W. Cohen, A. McCallum, and D. Quass. Learning to understand the Web. *IEEE Data Engineering Bulletin*, 23(3):17–24, 2000.

[Cohen *et al.*, 2000b] W. W. Cohen, H. Kautz, and D. McAllester. Hardening soft information sources. In *Proc. 6th KDD*, pages 255–259, 2000.

[Cohen *et al.*, 2002] W. W. Cohen, M. Hurst, and L. S. Jensen. A flexible learning system for wrapping tables and lists in HTML documents. In *Proc. 11th WWW*, 2002.

[DARPA, 1998] DARPA, editor. *Proc. 7th Message Understanding Conference (MUC-7)*, Fairfax, VA, 1998. Morgan Kaufman.

[Dellaert *et al.*, 2003] F. Dellaert, S. M. Seitz, C. E. Thorpe, and S. Thrun. EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Machine Learning*, 50:45–71, 2003.

[Fellegi and Sunter, 1969] I. Fellegi and A. Sunter. A theory for record linkage. *JASA*, 64:1183–1210, 1969.

[Gilks *et al.*, 1996] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, 1996.

[Google Inc., 2003] Google Inc. Froogle. http://froogle.google.com, 2003.

[Green, 1995] P. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

[Harabagiu *et al.*, 2001] S. Harabagiu, R. Bunescu, and S. Maiorano. Text and knowledge mining for coreference resolution. In *Proc. 2nd NAACL*, pages 55–62, 2001.

[Jeffreys, 1939] H. Jeffreys. *Theory of Probability*. Clarendon Press, Oxford, 1939.

[Lafferty *et al.*, 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th ICML*, pages 282–289, 2001.

[Lawrence *et al.*, 1999a] S. Lawrence, C. L. Giles, and K. Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67–71, 1999.

[Lawrence *et al.*, 1999b] S. Lawrence, C. L. Giles, and K. D. Bollacker. Autonomous citation matching. In *Proc. 3rd Int'l Conf. on Autonomous Agents*, pages 392–393, 1999.

[MacKay, 1992] D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[Marthi *et al.*, 2002] B. Marthi, H. Pasula, S. Russell, and Y. Peres. Decayed MCMC filtering. In *Proc. 18th UAI*, pages 319–326, 2002.

[McCallum *et al.*, 2000a] A. McCallum, K. Nigam, and L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proc. 6th KDD*, pages 169–178, 2000.

[McCallum *et al.*, 2000b] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of Internet portals with machine learning. *Information Retrieval*, 3:127–163, 2000.

[Nahm and Mooney, 2000] U. Y. Nahm and R. J. Mooney. A mutually beneficial integration of data mining and information extraction. In *Proc. 17th AAAI*, pages 627–632, 2000.

[Pasula *et al.*, 2003] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *NIPS 15*. MIT Press, Cambridge, MA, 2003.

[Pfeffer, 2000] A. Pfeffer. *Probabilistic Reasoning for Complex Systems*. PhD thesis, Stanford, 2000.

[Pylyshyn, 1984] Z. W. Pylyshyn. *Computation and Cognition: Toward a Foundation for Cognitive Science*. MIT Press, Cambridge, MA, 1984.

[Russell, 2001] S. Russell. Identity uncertainty. In *Proc. 9th Int'l Fuzzy Systems Assoc. World Congress*, 2001.

[Tanner and Wei, 1990] M. A. Tanner and G. C. G. Wei. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *JASA*, 85:699–704, 1990.

[Xing and Russell, 2003] E. P. Xing and S. Russell. On generalized variational inference, with application to relational probability models. Submitted, 2003.

[Xing *et al.*, 2003] E. P. Xing, M. I. Jordan, R. M. Karp, and S. Russell. A hierarchical Bayesian Markovian model for motifs in biopolymer sequences. In *NIPS 15*. MIT Press, Cambridge, MA, 2003.

[Yedidia *et al.*, 2001] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS 13*. MIT Press, Cambridge, MA, 2001.