

Robust Reading of Ambiguous Writing

Xin Li Paul Morie Dan Roth

Department of Computer Science
University of Illinois, Urbana, IL 61801
{xli1, morie, danr}@uiuc.edu

Abstract

A given entity, representing a person, a location or an organization, may be mentioned in text in multiple, ambiguous ways. Understanding natural language requires identifying whether different mentions of a name, within and across documents, represent the same entity. We develop an unsupervised learning approach that is shown to resolve accurately several aspects of the name identity and tracing problem. At the heart of our approach is a generative model of how documents are generated and how names are “sprinkled” into them; in particular, we model appearance similarity between names representing the same entity, contextual correlation among entities, and co-occurrence probabilities of entities within a document. We show how to estimate the model and do inference with it and how this resolves several aspects of the problem from the perspective of applications such as questions answering.

1 Introduction

Reading and understanding text is a task that requires the ability to disambiguate at several levels, abstracting away details and using background knowledge in a variety of ways. One of the difficulties that humans resolve instantaneously and unconsciously is that of reading names. Most names of people, locations, organizations and others, have multiple writings that are being used freely within and across documents.

The variability in writing a given concept, along with the fact that different concepts may have very similar writings, poses a significant challenge to progress in natural language related tasks. Consider, for example, an open domain question answering system (Voorhees, 2002) that attempts, given a question like *When was President Kennedy born?* to search a large collection of articles in order to pinpoint the concise answer: *on May 29, 1917*. The sentence, and even the document that contains the answer, may not contain the name “President Kennedy”; it may refer to this entity as “Kennedy”, “JFK” or “John Fitzgerald Kennedy”. Other documents may state that “John F. Kennedy, Jr. was born on November 25, 1960”, but this fact refers

to our target entity’s son. Other mentions as “Senator Kennedy” or “Mrs. Kennedy” are even closer to the writing of the target entity, but clearly refer to different entities. Even the statement “John Kennedy, born 5-29-1941” turns out to refer to a different entity, as one can tell observing that this document discusses Kennedy’s batting statistics. A similar problem exists for other entity types, such as locations, organization etc. Ad hoc solutions to this problem, as we show, fail to provide a reliable and accurate solution to this problem.

This paper presents the first attempt to apply a unified approach to all major aspects of this problem, presented here from the perspective of the question answering task:

(1) *Entity Identity* - do mentions *A* and *B* (typically, occurring in different documents, or in a question and a document, etc.) refer to the same entity? This problem requires both identifying when different writings refer to the same entity, and when very similar or identical writings refer to different entities. (2) *Name Expansion* - given a writing of a name (say, in a question), find other likely writings of the same name. (3) *Prominence* - given question *What is Bush’s foreign policy?*, and given that any large collection of documents may contain several Bush’s, there is a need to identify the most prominent, or relevant “Bush”, perhaps taking into account also some contextual information.

At the heart of our approach is a global probabilistic view on how documents are generated and how names (of different entity types) are “sprinkled” into them. In its most general form, our model assumes: (1) a joint distribution over entities, so that a document that mentions “President Kennedy” is more likely to mention “Oswald” or “White House” than “Roger Clemens”; (2) an “author” model, that makes sure that at least one mention of a name in a document is easily identifiable, and then generates other mentions via (3) an appearance model, governing how mentions are transformed from the “representative” mention.

Our goal is to learn the model from a large corpus and use it to support **robust reading** - enabling “on the fly” identification and tracing of entities.

This work presents the first study of our proposed model and several relaxations of it. Given a collection of documents we learn the models in an unsupervised way; that is, the system is not told during training whether two mentions represent the same entity. We only assume the

ability to recognize names, using a named entity recognizer run as a preprocessor. We define several inferences that correspond to the solutions we seek, and evaluate the models by performing these inferences against a large corpus we annotated (the corpus will be available on the web). Our experimental results suggest that the problem can be solved accurately, giving accuracies (F_1) close to 90%, depending on the specific task, as opposed to 80% given by state of the art ad-hoc approaches.

Previous work in the context of question answering has not addressed the problem studied here. Several works in NLP and Databases, though, have addressed some aspects of the problem. From the natural language perspective, there has been a lot of work on the related problem of coreference resolution (Soon et al., 2001; Ng and Cardie, 2003; Kehler, 2002) - which aims at linking occurrences of noun phrases and pronouns within a given document based on their appearance and local context. In the context of databases, several works have looked at the problem of record linkage - recognizing duplicate records in a database (Cohen and Richman, 2002; Hernandez and Stolfo, 1995; Bilenko and Mooney, 2003). Specifically, (Pasula et al., 2002) considers the problem of identity uncertainty in the context of citation matching and suggests a probabilistic model for that. Perhaps the work that is most related to ours in terms of the problem definition and in the sense that it works with text data and across documents, is (Mann and Yarowsky, 2003), which considers the problem of distinguishing occurrences of identical names in different documents. Like ours, this problem is global, but they consider only one aspect of the identity problem, and only for identical names of *people* (e.g., do occurrences of “Jim Clark” in different documents refer to the same person or not).

The rest of this paper is organized as follows: We formalize the “robust reading” problem in Sec. 2. Sec. 3 describes a generative view of documents’ creation and three practical probabilistic models designed based on it, and discusses inference in these models. Sec. 4 illustrates how to learn these models in an unsupervised setting, and Sec. 5 describes the experimental study. Sec. 6 concludes.

2 Robust Reading

We consider reading a large number of documents $D = \{d_1, d_2, \dots, d_m\}$, each of which may contain *mentions* (i.e. real occurrences) of $|T|$ types of *entities*. In the current evaluation we consider $T = \{Person, Location, Organization\}$.

An *entity* refers to the “real” concept behind the mention and can be viewed as a unique identifier to an object in the real world. Examples might be the person “John F. Kennedy” who became a president, “White House” – the residence of the US presidents, etc. E denotes

the collection of all possible entities in the world and $E_d = \{e_{di}\}_1^{l_d}$ is the set of entities mentioned in document d . M denotes the collection of all possible mentions and $M_d = \{m_{di}\}_1^{n_d}$ is the set of mentions in document d . $M_{di}(1 \leq i \leq l_d)$ is the set of mentions that refer to entity $e_{di} \in E_d$. For example, for entity “John F. Kennedy”, the corresponding set of mentions in a document may contain “Kennedy”, “J. F. Kennedy” and “President Kennedy”. Among all mentions of an entity e_{di} in document d we distinguish the one occurring first, $r_{di} \in M_{di}$, as the *representative* of e_{di} . In practice, the representative is usually the longest mention of an entity in the document as well, and other mentions are variations of it. Representatives can be viewed as a typical representation of an entity mentioned in a specific time and place. For example, “President J.F.Kennedy” and “Congressman John Kennedy” may be representatives of “John F. Kennedy” in different documents. R denotes the collection of all possible representatives and $R_d = \{r_{di}\}_1^{l_d} \subseteq M_d$ is the set of representatives in document d . This way, each document is represented as the collection of its entities, representatives and mentions $d = \{E_d, R_d, M_d\}$.

Elements in the name space $W = E \cup R \cup M$ each have an identifying writing (denoted as $wrt(n)$ for $n \in W$)¹ and an ordered list of attributes, $A = \{a_1, \dots, a_p\}$, which depends on the entity type. Attributes used in the current evaluation include both *internal* attributes, such as, for *People*, $\{title, firstname, middlename, lastname, gender\}$ as well as *contextual* attributes such as $\{time, location, proper-names\}$. *Proper-names* refer to a list of proper names that occur around the mention in the document. All attributes are of string value and can be empty when the values are missing or unknown².

The fundamental problem we address in robust reading is to decide what entities are mentioned in a given document (given the observed set M_d) and what the most likely assignment of entity to each mention is.

3 A Model of Document Generation

We define a probability distribution over documents $d = \{E_d, R_d, M_d\}$, by describing how documents are being generated. In its most general form the model has the following three components:

(1) A joint probability distribution $P(E_d)$ that governs how entities (of different types) are distributed into a document and reflects their co-occurrence dependencies.

(2) The number of entities in a document, $size(E_d)$, and the number of mentions of each entity in E_d , $size(M_{di})$, need to be decided. The current evaluation

¹The observed writing of a mention is its identifying writing, i.e., “President Kennedy”. For entities, it is a standard representation of them, i.e. the full name of a person.

²Contextual attributes are not part of the current evaluation, and will be evaluated in the next step of this work.

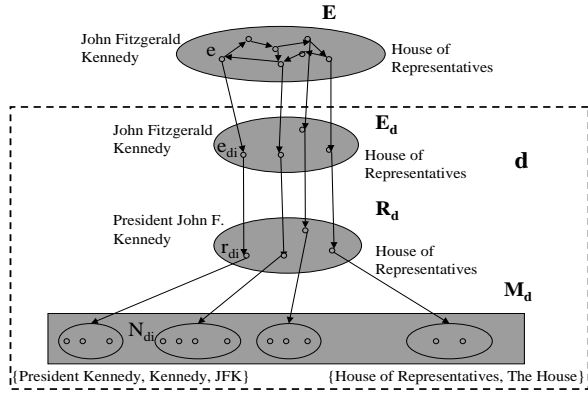


Figure 1: Generating a document

makes the simplifying assumption that these numbers are determined uniformly over a small plausible range.

(3) The *appearance probability* of a name generated (transformed) from its representative is modelled as a product distribution over relational transformations of attribute values. This model captures the similarity between appearances of two names. In the current evaluation the same appearance model is used to calculate both the probability $P(r|e)$ that generates a representative r given an entity e and the probability $P(m|r)$ that generates a mention m given a representative r . Attribute transformations are relational, in the sense that the distribution is over transformation types and independent of the specific names.

Given these, a document d is assumed to be generated as follows (see Fig. 1): A set of $size(E_d)$ entities $E_d \subseteq E$ is selected to appear in a document d , according to $P(E_d)$. For each entity $e_{di} \in E_d$, a representative $r_{di} \in R$ is chosen according to $P(r_{di}|e_{di})$, generating R_d . Then mentions M_{di} of an entity are generated from each representative $r_{di} \in R_d$ – each mention $m_{dj} \in M_{di}$ is independently transformed from r_{di} according to the appearance probability $P(m_{dj}|r_{di})$, after $size(M_{di})$ is determined. Assuming conditional independency between M_d and E_d given R_d , the probability distribution over documents is therefore

$$P(d) = P(E_d, R_d, M_d) = P(E_d)P(R_d|E_d)P(M_d|R_d),$$

and the probability of the document collection D is:

$$P(D) = \prod_{d \in D} P(d).$$

Given a mention m in a document d (M_d is the set of observed mentions in d), the key inference problem is to determine the most likely entity e_m^* that corresponds to it. This is done by computing:

$$E_d = \operatorname{argmax}_{E' \subseteq E} P(E_d, R_d | M_d, \theta) \quad (1)$$

$$= \operatorname{argmax}_{E' \subseteq E} P(E_d, R_d, M_d | \theta), \quad (2)$$

where θ is the learned model’s parameters. This gives the assignment of the most likely entity e_m^* for m .

3.1 Relaxations of the Model

In order to simplify model estimation and to evaluate some assumptions, several relaxations are made to form three simpler probabilistic models.

Model I: (the simplest model) The key relaxation here is in losing the notion of an “author” – rather than first choosing a representative for each document, mentions are generated independently and directly given an entity.

That is, an entity e_i is selected from E according to the prior probability $P(e_i)$; then its actual mention m_i is selected according to $P(m_i|e_i)$. Also, an entity is selected into a document independently of other entities. In this way, the probability of the whole document set can be written in a simpler way:

$$P(D) = P(\{(e_i, m_i)\}_{i=1}^n) = \prod_{i=1}^n P(e_i)P(m_i|e_i),$$

and the inference problem for the most likely entity given m is:

$$e^* = \operatorname{argmax}_{e \in E} P(e|m, \theta) \quad (3)$$

$$= \operatorname{argmax}_{e \in E} P(e)P(m|e) \quad (4)$$

Model II: (more expressive) The major relaxation made here is in assuming a simple model of choosing entities to appear in documents. Thus, in order to generate a document d , after we decide $size(E_d)$ and $\{size(M_{d1}, size(M_{d2}), \dots)\}$ according to uniform distributions, each entity e_{di} is selected into d independently of others according to $P(e_{di})$. Next, the representative r_{di} for each entity e_{di} is selected according to $P(r_{di}|e_{di})$ and for each representative the actual mentions are selected independently according to $P(m_{dj}|r_{dj})$. Here, we have individual documents along with representatives, and the distribution over documents is:

$$\begin{aligned} P(d) &= P(E_d, R_d, M_d) = P(E_d)P(R_d|E_d)P(M_d|R_d) \\ &= [P(size(E_d))] \prod_{i=1}^{|E_d|=l_d} P(e_{di}) \\ &\times [P(size(M_{d1}), size(M_{d2}), \dots)] \prod_{i=1}^{|E_d|=l_d} P(r_{di}|e_{di}) \\ &\times \prod_{(r_{dj}, m_{dj})} P(m_{dj}|r_{dj}) \\ &\approx \prod_{i=1}^{|E_d|=l_d} [P(e_{di})P(r_{di}|e_{di})] \prod_{(r_{dj}, m_{dj})} P(m_{dj}|r_{dj}). \end{aligned}$$

after we ignore the size components. The inference problem here is the same as in Equ. (2).

Model III: This model performs the least relaxation. After deciding $size(E_d)$ according to a uniform distribution, instead of assuming independency among entities which does not hold in reality (For example, “Gore” and “George. W. Bush” occur together frequently, but “Gore” and “Steve. Bush” do not), we select entities using a graph based algorithm: entities in E are viewed

as nodes in a weighted directed graph with edges (i, j) labelled $P(e_j|e_i)$ representing the probability that entity e_j is chosen into a document that contains entity e_i . We distribute entities to E_d via a random walk on this graph starting from e_{d1} with a prior probability $P(e_{d1})$. Representatives and mentions are generated in the same way as in Model II. Therefore, a more general model for the distribution over documents is:

$$P(d) \approx P(e_{d1})P(r_{d1}|e_{d1}) \prod_{i=2}^{|E_d|=l_d} [P(e_{di}|e_{di-1})P(r_{di}|e_{di})] \\ \times \prod_{(r_{dj}, m_{dj})} P(m_{dj}|r_{dj}).$$

The inference problem is the same as in Equ. (2).

3.2 Inference

The fundamental problem in robust reading can be solved as inference with the models: given a mention m , seek the most probable entity $e \in E$ for m according to Equ. (4) for Model I or Equ. (2) for Model II and III. The inference algorithm for Model I (with time complexity $O(|E|)$) is simple and direct: just compute $P(e, m)$ for each candidate entity $e \in E$ and then choose the one with the highest value. Due to exponential number of possible assignments of E_d, R_d to M_d in Model II and III, precise inference is infeasible. Approximate algorithms are therefore designed:

In Model II, we adopt a two-step algorithm: First, we seek the representatives R_d for the mentions M_d in document d by sequentially clustering the mentions according to the appearance model. The first mention in each group is treated as the representative. Specifically, when considering a mention $m \in M_d$, $P(m|r)$ is computed for each representative r that have already been created and a fixed threshold is then used to decide whether to create a new group for m or to add it to one of the existing group with the highest $P(m|r)$ value. In the second step, each representative $r_{di} \in R_d$ is assigned to its most likely entity according to $e^* = \operatorname{argmax}_{e \in E} P(e) * P(r|e)$ ³. This algorithm has a total time complexity of $O(|M_d|^2 + |E| * |R_d|)$.

Model III has a similar two-step algorithm as Model II. The only difference is that we need to consider the global dependency between entities. Thus in the second step, instead of seeking an entity e for each representative r separately, we determine a set of entities E_d for R_d in a Hidden Markov Model with entities in E as hidden states and R_d as observations. The prior probabilities, the transitive probabilities and the observation probabilities for this HMM are given by $P(e)$, $P(e_j|e_i)$ and $P(r|e)$ respectively. In this step we seek the most likely sequence

³ E is known after learning the model in a closed document collection that d belongs to.

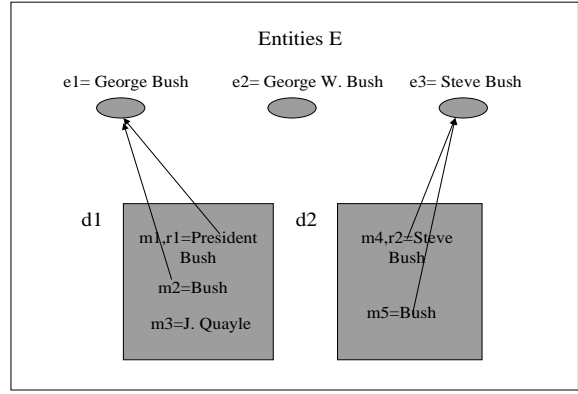


Figure 2: An conceptual example. The arrows represent correct assignment of entities to mentions. r_1, r_2 are representatives.

of entities given those representatives in their appearing order using the Viterbi algorithm. The total time complexity is $O(|M_d|^2 + |E|^2 * |R_d|)$.

3.3 Discussion

Besides different assumptions of the models, there are some fundamental differences in inference with the models as well. In Model I, the entity of a mention is determined completely independently of other mentions, while in Model II the way of figuring out the entity relies on local similarity among mentions in the same document. In Model III, it is not only related to other mentions but to a global dependency over entities. The following conceptual example illustrates those differences as in Fig. 2.

Example 3.1 Given $E = \{\text{George Bush, George W. Bush, Steve Bush}\}$, documents d_1, d_2 and 5 mentions in them, and suppose the prior probability of entity “George W. Bush” is higher than those of the other two entities, the probable assignment of entities to mentions in the three models could be as follows:

For Model I, $\text{mentions}(e_1) = \phi$, $\text{mentions}(e_2) = \{m_1, m_2, m_3\}$ and $\text{mentions}(e_3) = \{m_4\}$. The result is caused by the fact that a mention tends to be assigned to the entity with higher prior probability when the appearance similarity is not distinctive.

For Model II, $\text{mentions}(e_1) = \phi$, $\text{mentions}(e_2) = \{m_1, m_2\}$ and $\text{mentions}(e_3) = \{m_4, m_5\}$. Local dependency (appearance similarity) among mentions inside each document enforces constraints that they should refer to the same entity, like “Steve Bush” and “Bush” in d_2 .

For Model III, $\text{mentions}(e_1) = \{m_1, m_2\}$, $\text{mentions}(e_2) = \phi$, $\text{mentions}(e_3) = \{m_4, m_5\}$. With the help of global dependency among entities, for example, “George Bush” and “J. Quayle”, an entity can be distinguished from another entity with a similar writing.

3.4 Other Tasks

The three basic problems related to “Robust Reading” can be solved based on the solutions to the key inference problem above.

Entity Identity: Given two mentions $m_1 \in d_1, m_2 \in d_2$, determine whether they correspond to the same entity ($m_1 \sim m_2$) by:

$$m_1 \sim m_2 \text{ iff } \operatorname{argmax}_{e \in E} P(e, m_1) = \operatorname{argmax}_{e \in E} P(e, m_2)$$

for Model I and

$$m_1 \sim m_2 \text{ iff } \operatorname{argmax}_{e \in E} P(E_{d_1}, R_{d_1}, M_{d_1}) = \operatorname{argmax}_{e \in E} P(E_{d_2}, R_{d_2}, M_{d_2}).$$

for Model II and III.

Name Expansion: Given a mention m_q in a query q , decide whether mention m in the document collection D is a ‘legal’ expansion of m_q :

$$m_q \rightarrow m \quad \text{iff } e_{m_q}^* = \operatorname{argmax}_{e \in E} P(E_q, R_q, M_q) \\ \& m \in \text{mentions}(e^*).$$

We assume here that we already know the possible mentions of e^* after learning the models in D .

Prominence: Given a name $n \in W$, the most prominent entity for n is given by:

$$e^* = \operatorname{argmax}_{e \in E} P(e)P(n|e).$$

$P(e)$ is given by the prior distribution P_E and $P(n|e)$ is given by the appearance model.

4 Learning the Models

Confined by the labor of annotating data, we learn the probabilistic models in an unsupervised way given a collection of documents; that is, the system is not told during training whether two mentions represent the same entity. A greedy search algorithm modified after the standard EM algorithm (We call it Truncated EM algorithm) is adopted here to avoid complex computation.

Given a set of documents D to be studied and the observed mentions M_d in each document, this algorithm iteratively updates the model parameter θ (several underlying probabilistic distributions described before) and the structure (that is, E_d and R_d) of each document d . Different from the standard EM algorithm, in the E-step, it seeks the most likely E_d and R_d for each document rather than the expected assignment.

4.1 Truncated EM Algorithm

The basic framework of the Truncated EM algorithm to learn Model II and III is as follows:

1. In the initial (I-) step, an initial E_d^0 and R_d^0 is assigned to each document d using an initialization algorithm. After this step, we can assume that we have labelled documents $D^0 = \{(E_d^0, R_d^0, M_d)\}$.
2. In the M-step, we seek the model parameter θ^{t+1} that maximizes $P(D^t|\theta)$. Given the ‘labels’ supplied by the model in the previous I- or E-step, this amounts to the maximum likelihood estimation as described in Sec. 4.3.
3. In the E-step, we seek (E_d^{t+1}, R_d^{t+1}) for each document d that maximizes $P(D^{t+1}|\theta^{t+1})$ where $D^{t+1} = \{(E_d^{t+1}, R_d^{t+1}, M_d)\}$. It’s the same inference problem in Sec. 3.2.
4. Stopping Criterion: If no increase is achieved over $P(D^t|\theta^t)$, the algorithm exits. Otherwise the algorithm will iterate over the M-step and E-step.

The algorithm for Model I is similar to the above algorithm but much simpler in the sense that it does not have the notions of documents and representatives. So in the E-step we only need to seek the most possible entity e for each mention $m \in D$ and this simplifies the parameter estimation in the M-step accordingly. It usually takes 3 – 10 iterations before the algorithm stops for all the models in our experiments.

4.2 Initialization

The purpose of the initial step is to acquire an initial guess of document structures and to seek the set of entities E in a closed collection of documents D . The hope is to find all entities without loss even if repeated entities might be created. For all the models, we use the same algorithm:

First, a local clustering is performed to group all mentions inside each document. A set of simple heuristics of matching attributes is applied to calculating the similarity between mentions and pairs of mentions with similarity above a threshold are clustered together. The first mention in each group is chosen as the representative (only in Model II and III) and an entity having the same writing with the representative is created for each cluster⁴.

For all the models, the set of entities created in different documents become the global entity set E in the following M- and E-steps.

4.3 Estimating the Model Parameters

In the learning process, assuming we have obtained labelled documents $D = \{(e, r, m)\}_1^n$ from previous I- or E-step, several probability distributions underlying the relaxed models are estimated according to maximum likelihood estimation in each M-step. The model parameters include a prior distribution over entities P_E , a tran-

⁴Note that the performance of the initialization algorithm is 97.3% precision and 10.1% recall, measures defined in our later experimental study in Sec. 5.

sitive probability distribution over pairs of entities $P_{E|E}$ (only in Model III) and the appearance probability $P_{W|W}$ of a name in the name space W being transformed from another name.

- The prior distribution P_E is modelled as a multinomial distribution. Given a set of labelled entity-mention pairs $\{(e_i, m_i)\}_1^n$,

$$P(e) = \frac{freq(e)}{n}$$

where $freq(e)$ denotes the number of pairs containing entity e .

- Given all the entities appearing in D , The transitive probability between entities $P(e|e)$ is estimated by

$$\begin{aligned} P(e_2|e_1) &\sim P(wrt(e_2)|wrt(e_1)) \\ &= \frac{doc^\#(wrt(e_2),wrt(e_1))}{doc^\#(wrt(e_1))}. \end{aligned}$$

Here, the conditional probability between two real entities $P(e_2|e_1)$ is backed off to the conditional probability between the identifying writings of the two entities $P(wrt(e_2)|wrt(e_1))$ in the document set D to avoid sparsity problem. Given $D = \{d_1, d_2, \dots, d_m\}$. And $doc^\#(w_1, w_2, \dots)$ denotes the number of documents having the co-occurrence of writings w_1, w_2, \dots .

- Appearance Probability, the probability of one name being transformed from another, denoted as $P(n_2|n_1)$ ($n_1, n_2 \in W$), is modelled as a product of the transformation probabilities over attribute values. The transformation probability for each attribute in A is further modelled as a multi-nomial distribution over a set of predetermined “typical” transformation types that depend on the entity types: $TT = \{copy, missing, typical, non - typical\}$ ⁵.

Suppose $n_1 = (a_1 = v_1, a_2 = v_2, \dots, a_p = v_p)$ and $n_2 = (a_1 = v'_1, a_2 = v'_2, \dots, a_p = v'_p)$ are two names belonging to the same entity type, the transformation probabilities $P_{M|R}$, $P_{R|E}$ and $P_{M|E}$, are all modelled as a product distribution (naive Bayes) over attributes:

$$P(n_2|n_1) = \prod_{k=1}^p P(v'_k|v_k).$$

We manually collected typical and non-typical transformations for attributes such as *titles*, *first names*, *last names*, *organizations* and *locations* from multiple sources such as U.S. government census and online dictionaries. For other attributes like *gender*, only **copy** transformation is allowed. Assuming multi-nomial distribution for each attribute, the maximum likelihood estimation of the transformation probability $P(t, k)$ ($t \in TT, a_k \in A$) from labelled representative-mention pairs $\{(r, m)\}_1^n$ is:

⁵**copy** denotes v'_k is exactly the same as v_k ; **missing** denotes “missing value” for v'_k ; **typical** denotes v'_k is a typical variation of v_k , for example, “Prof.” for “Professor”, “Andy” for “Andrew”; **non-typical** denotes a non-typical transformation.

$$P(t, k) = \frac{freq(r, m) : v_k^r \rightarrow_t v_k^m}{n} \quad (5)$$

$v_k^r \rightarrow_t v_k^m$ denotes the transformation from attribute a_k of r to that of m is of type t . Simple smoothing is performed here for unseen transformations.

5 Experimental Study

Our experimental study focuses on (1) evaluating our three models on the name identity task using three entity types (People, Locations, Organization); (2) comparing our induced similarity measure between names with other similarity measures; (3) evaluating the contribution of the global nature of our model, and (4) evaluating our models on name expansion and prominence ranking.

5.1 Methodology

We collected 300 documents from randomly sampled 1998-2000 New York Times articles in the TREC corpus (Voorhees, 2002). The documents were annotated by a named entity tagger for People, Locations and Organizations. The annotation was then corrected and each name mention was labelled with its corresponding entity by two annotators. In total, about 8,000 mentions of named entities which correspond to about 2,000 entities were labelled. The training process gets to see only the 300 documents and extracts attribute values for each mention. No supervision is supplied. These records are used to learn the probabilistic models. In testing, 130,000 pairs of mentions that correspond to the same entity are generated, and are used to evaluate the models’ performance. Since the probabilistic models are learned in an unsupervised setting, testing can be viewed simply as the evaluation of the learned model, and is thus done on the same data. The same setting was used for all models and all comparison performed (see below).

To evaluate the performance, we pair two mentions iff the learned model determined that they correspond to the same entity. The list of pairs is then compared with the annotated list of pairs. We measure Precision (P) – Percentage of correctly predicted pairs, Recall (R) – Percentage of correct pairs that were predicted, and $F_1 = \frac{2PR}{P+R}$.

Comparisons: Our global model induces a “similarity” measure between names – the appearance model. In order to understand whether the behavior of our model is dominated by the quality of the induced pairwise similarity or by the global aspects of the model we (1) replace this measure by two other “local” similarity measures and (2) study the performance on entity identity at three levels – local decision, straightforward clustering over local similarity, and our global model.

The first similarity measure we use is a simple baseline algorithm according to which two names are similar iff

All(P/L/O)	Identity	SoftTFIDF	Appearance
Pairwise	70.7 (64.7/64.1/83.7)	82.1 (79.9/77.3/89.5)	81.5 (83.6/70.9/90.7)
Clustering	70.7 (64.7/64.1/83.7)	79.8 (70.6/76.7/91.0)	79.6 (70.9/76.1/91.0)
Model II	70.7 (64.7/64.1/83.7)	82.5 (79.8/77.4/90.2)	89.0 (92.7/81.9/92.9)

Table 1: **Comparison of different decision levels and similarity measures.** Three similarity measures are evaluated (rows) across three decision levels (columns). Performance is evaluated by the F_1 values over the whole test set. The first number averages all entity types; numbers in parentheses represent People, Location and Organization respectively.

they have identical writings. The second is a state-of-art similarity measure for entity names (SoftTFIDF with Jaro-Winkler distance and $\theta = 0.9$); it was ranked the best measure in a recent study (Cohen et al., 2003).

Local decision (**Pairwise**) is done by pairing two mentions iff the similarity between them is above a fixed threshold. For **Clustering**, a graph-based clustering algorithm is used, where two mentions are paired iff they belong to the same connected component. Finally, we use the baseline and the SoftTFIDF in the context of Model II, where the appearance model is replaced by the similarity measure⁶.

5.2 Results

The bottom line result is given in Tab. 1. All local similarity measures are compared in the context of the three levels of processing – local decision, clustering and our probabilistic model II.

The behavior across rows indicates that our unsupervised learning based appearance model is about the same as the state-of-the-art SoftTFIDF similarity. The behavior across columns, though, shows the contribution of our global model, and that the local appearance model behaves better with it than a fixed similarity measure does.

A secondary observation is that our appearance model for Location is not as good as the one for People and Organization, probably due to the attribute transformation types chosen.

Tab. 2 presents a more detailed evaluation of the different approaches on the entity identity task. All the three probabilistic models outperform the discriminatory approaches in this experiment, an indication of the effectiveness of the generative model.

We note that although Model III is more expressive and reasonable than model II, it does not always perform better. Indeed, the global dependency among entities in Model III achieves two-folded outcomes: it achieves better precision but, may degrade the recall. The following

⁶Note that both the appearance model $s(n_1, n_2) = P(n_1|n_2)$ and the SoftTFIDF similarity measure are not symmetric. Also, we found that the SoftTFIDF similarity measure behaves very badly in the context of the probabilistic model, and improved it by converting it to $P(n_1|n_2) = \frac{e^{c \times s(n_1, n_2)} - 1}{e^c - 1}$. c was set to 10 in the experiments.

Entity Type	Mod	InDoc F_1 (%)	InterDoc F_1 (%)	All		
				R(%)	P(%)	F_1 (%)
All	B	86.0	68.8	58.5	85.5	70.7
	D	86.5	78.9	66.4	95.8	79.8
	I	96.3	85.0	79.0	94.1	86.2
	II	96.5	88.1	85.9	92.2	89.0
	III	96.5	87.9	84.4	93.6	88.9
P	B	82.4	59.0	48.5	86.3	64.7
	D	82.4	67.1	54.5	91.5	70.6
	I	96.2	84.8	80.6	94.8	87.4
	II	96.4	91.7	94.0	91.5	92.7
	III	96.4	88.9	89.8	91.3	90.5
L	B	88.8	63.0	54.8	75.0	64.1
	D	91.4	76.0	61.3	95.9	76.7
	I	92.9	78.9	70.9	89.1	79.5
	II	93.8	81.4	76.2	88.1	81.9
	III	93.8	82.8	76.0	91.2	83.3
O	B	95.3	82.8	72.6	96.4	83.7
	D	95.8	90.7	83.9	98.9	91.1
	I	98.8	91.8	86.5	98.5	92.3
	II	98.5	92.5	88.6	97.5	92.9
	III	98.8	93.0	88.5	98.6	93.4

Table 2: **Performance of different approaches over all test examples.** B, D, I, II and III denote the baseline model, the SoftTFIDF similarity model with clustering, and the three probabilistic models. *All, P, L, O* denote all entities, People, Locations and Organizations respectively. We distinguish between pairs of mentions that are inside the same document (*InDoc*, 15% of the pairs) or not (*InterDoc*).

example, taken from the corpus, illustrates the advantage of this model.

Example 5.1 “*Sherman Williams*” is mentioned along with the baseball team “*Dallas Cowboys*” in eight out of 300 documents, while “*Jeff Williams*” is mentioned along with “*LA Dodgers*” in two documents.

In all the models except Model III, “*Jeff Williams*” is judged to correspond to the same entity as “*Sherman Williams*” since they are quite similar and the prior probability of the latter is higher than the former. Only in Model III, due to the dependency between “*Jeff Williams*” and “*Dodgers*”, the system identifies it as corresponding to a different entity than “*Sherman Williams*”.

While this exhibits the better precision achieved by Model III, the recall may go down. The reason is that the global dependency among entities in Model III enforces restrictions over possible grouping of similar mentions; in addition, with a limited document set, estimating this global dependency cannot be done accurately, especially in the setting that entities themselves need to be found when learning the model. We expect that Model III will dominate Model II when we have enough data to estimate a more accurate global dependencies.

Hard Cases: To analyze the experimental results further, we evaluated separately two types of harder cases of the entity identity task: (1) mentions with *different* writings that refer to the same entity; and (2) mentions with *similar* writings that refer to different entities. Model II and III outperform other models in those two cases as well.

Tab. 3 presents F_1 performance of different approaches in the first case. The best F_1 value is only 73.1%, indicating that appearance similarity and global dependency are not sufficient to solve this problem when the writings are very different. Tab. 4 shows the performance of different approaches for disambiguating *similar* writings that

Model	B	D	I	II	III
Peop	0	77.9	79.2	86.0	82.6
Loc	0	30.4	55.1	58.5	61.5
Org	0	77.7	69.5	71.7	71.2
All	0	63.3	68.4	73.1	72.5

Table 3: **Identifying different writings of the same entity** (F_1). We filter out identical writings and report only on cases of *different* writings of the same entity. The test set contains 46,376 matching pairs (but in different writings) in the whole data set.

Model	B	D	I	II	III
Peop	75.2	83.0	60.8	89.7	88.0
Loc	86.5	80.7	80.0	90.3	90.3
Org	80.0	89.4	71.0	93.1	92.6
All	78.7	78.9	68.1	90.7	89.7

Table 4: **Identifying similar writings of different entities.** (F_1) The test set contains 39,837 pairs of mentions that associated with different entities in the 300 documents and have at least one token in common.

correspond to different entities.

Both these cases exhibit the difficulty of the problem, and that our approach provides a significant improvement over the state of the art similarity measure — column *D* vs. column *II* in Tab. 4. It also shows that it is necessary to use contextual attributes of the names, which are not yet included in this evaluation.

5.3 Other Tasks

In the following experiments, we evaluate our generative model on other tasks related to robust reading. We present results only for Model II.

Name Expansion: Given a mention m (for example, in a IR query q), we find the most likely entity $e \in E$ for m using our inference algorithm. All unique mentions of the entity in the documents are output as the expansions of m . The accuracy of Name Expansion for a given mention is defined as the number of correct expansions over the total number of names output. The average accuracy of Name Expansion of Model II is shown in Tab. 5. Here is an example of a query: **Query:** Who is *Gore* ? **Expansions:** Vice President Al Gore, Al Gore, Gore.

Prominence Information: We refer to Example 3.1 and use it to exemplify quantitatively how our system supports prominence ranking. The following examples show the ranking of entities with regard to the value of $P(e) * P(m|e)$ (shown in the brackets) using Model II, given a query name m .

Input: George Bush

1. George Bush(2.49E-4)
2. George W. Bush(6.64E-7)

Input: Bush

1. George W. Bush(5.13E-7)
2. George Bush(1.42E-7)
3. Steve Bush(5.69E-10)

Entity Type	People	Location	Organization
Accuracy(%)	90.6	100	100

Table 5: **Accuracy of Name Expansion.** Accuracy is averaged over 30 randomly chosen queries for each entity type.

6 Conclusion and Future Work

This paper presents an unsupervised learning approach to several aspects of the “robust reading” problem – cross-document resolution of ambiguous writings of names. We developed a model that describes the natural generation process of a document and the process of how names are “sprinkled” into them, taking into account dependencies between entities across types and an “author” model. Several relaxations of this model were developed and studied experimentally, and compared to a state-of-the-art model that does not take a global view. The experiments exhibit good results and show the advantage of several aspects of our model.

This work is a preliminary exploration of the robust reading problem. There are several critical issues that our model can support, but were not included in this preliminary evaluation. Some of the issues that will be included in future steps are: (1) integration with more contextual information (like time and place) related to the target entities, both to support a better model and to allow temporal tracing of entities; (2) studying an incremental approach learning the model; that is, when a new document is observed, coming, how can we update our model parameters and the corresponding knowledge base? (3) integration of this work with other aspect of coreference resolution (e.g., other terms like pronouns that refer to an entity) and named entity recognition (which we now take as a given); and (4) scalability issues in applying the system to very large corpora.

References

- M. Bilenko and R. Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. In *KDD*.
- W. Cohen and J. Richman. 2002. Learning to match and cluster large high-dimensional data sets for data integration. In *KDD*.
- W. Cohen, P. Ravikumar, and S. Fienberg. 2003. A comparison of string metrics for name-matching tasks. In *IIWeb Workshop 2003*.
- M. Hernandez and S. Stolfo. 1995. The merge/purge problem for large databases. In *SIGMOD*.
- A. Kehler. 2002. *Coherence, Reference, and the Theory of Grammar*. CSLI Publications.
- G. Mann and D. Yarowsky. 2003. Unsupervised personal name disambiguation. In *CoNLL*.
- V. Ng and C. Cardie. 2003. Improving machine learning approaches to coreference resolution. In *ACL*.
- H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. 2002. Identity uncertainty and citation matching. In *NIPS*.
- W. Soon, H. Ng, and D. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics (Special Issue on Computational Anaphora Resolution)*, 27:521–544.

E. Voorhees. 2002. Overview of the TREC-2002 question answering track. In *Proceedings of TREC*, pages 115–123.