Robust Reading: Identification and Tracing of Ambiguous Names

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 the name identification 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 its most general form, our model assumes: (1) a joint distribution over entities, (2) an “author” model, that assumes that at least one mention of an entity 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. 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 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 processing. 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, such 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 the document discusses Kennedy’s batting statistics. A similar problem exists for other entity types, such as locations, organizations etc. Ad hoc solutions to this problem, as we show, fail to provide a reliable and accurate solution.

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 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 “rep-
resentative” 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. Our experimental results suggest that the entity identity problem can be solved accurately, giving accuracies (F1) 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 this problem. Several works in NLP and Databases, though, have addressed some aspects of it. 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 document based on their appearance and local context. (Charniak, 2001) presents a solution to the problem of name structure recognition by incorporating coreference information. 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. Some of very few works we are aware of that works directly with text data and across documents, are (Bagga and Baldwin, 1998; Mann and Varowsky, 2003), which consider one aspect of the problem – that of distinguishing occurrences of identical names in different documents, and only of people.

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 collection of documents \( D = \{d_1, d_2, \ldots, d_m\} \), each of which may contain mentions (i.e. real occurrences) of \( |T| \) types of entities. In the current evaluation we consider \( T = \{\text{Person, Location, Organization}\} \).

An entity refers to the “real” concept behind a mention and can be viewed as a unique identifier to a real-world object. 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 \( E^d = \{e_i^d\}_{i=1}^{|e^d|} \) is the set of entities mentioned in document \( d \). \( M \) denotes the collection of all possible mentions and \( M^d_i = \{m_i^d\}_{i=1}^{|m_i^d|} \) is the set of mentions in document \( d \). \( M^d_i(1 \leq i \leq |d|) \) is the set of mentions that refer to entity \( e_i^d \in E^d \). 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 \( e_i^d \) in document \( d \) we distinguish the one occurring first, \( r_1^d \in M^d, \) as the representative of \( e_i^d \). In practice, \( r_1^d \) is usually the longest mention of \( e_i^d \) in the document as well, and other mentions are variations of it. Representatives are 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 \( R^d = \{r_i^d\}_{i=1}^{|R|} \subseteq M^d \) 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 \( n \in W \)) and an ordered list of attributes, \( A = \{a_1, \ldots, a_p\} \), which depends on the entity type. Attributes used in the current evaluation include both internal attributes, such as, for People, \( \{\text{title, firstname, middlename, lastname, gender}\} \) as well as contextual attributes such as \( \{\text{time, location, proper-names}\} \). Proper-names refer to a list of proper names that occur around the mention in the document. All attributes are of string value and the values could be missing or unknown\(^2\).

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

\(^3\)The observed writing of a mention is its identifying writing. For entities, it is a standard representation of them, i.e. the full name of a person.

\(^2\)Contextual attributes are not part of the current evaluation, and will be evaluated in the next step of this work.
Figure 1: Generating a document

how entities (of different types) are distributed into a document and reflects their co-occurrence dependencies.

(2) The number of entities in a document, \( \text{size}(E^d) \), and the number of mentions of each entity in \( E^d, \) \( \text{size}(M^d) \), need to be decided. The current evaluation 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 \( \text{size}(E^d) \) entities \( E^d \subseteq E \) is selected to appear in a document \( d \), according to \( P(E^d) \). For each entity \( e^d \in E^d \), a representative \( r^d \in R \) is chosen according to \( P(r^d|e^d) \), generating \( R^d \). Then mentions \( M^d \) of an entity are generated from each representative \( r^d \in R^d \) — each mention \( m^d \in M^d \) is independently transformed from \( r^d \) according to the appearance probability \( P(m^d|r^d) \). 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:

\[
E^d = \arg\max_{E \subseteq E} P(E^d, R^d|M^d, \theta) = \arg\max_{E \subseteq E} P(E^d, R^d|E^d, M^d, \theta),
\]

where \( \theta \) 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 computed simply as follows:

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

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

\[
e^*_m = \arg\max_{e \in E} P(e|m, \theta) = \arg\max_{e \in E} P(e)P(m|e).
\]
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 \(e^d_1\) with a prior probability \(P(e^d_1)\). Representatives and mentions are generated in the same way as in Model II. Therefore, a more general model for the distribution over documents is:

\[
P(d) = P(e^d_1)P(e^d_1|e^d_2) \prod_{i=2}^{D} P(e^d_i|e^d_{i-1})P(e^d_i|e^d_1) \times \prod_{(i,j) \in E^d} P(m^d_j|m^d_i)
\]

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

### 3.2 Inference Algorithms

The fundamental problem in robust reading can be solved as inference with the models: given a mention \(m\), seek the most likely entity \(e \in E\) for \(m\) according to Equ. (3) for Model I or Equ. (2) for Model II and III. Instead of all entities in the real world, \(E\) can be viewed without loss as the set of entities in a closed document collection that we use to train the model parameters and it is known after training. 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 and 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 chosen 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 groups with the largest \(P(m|r)\). In the second step, each representative \(r^d_1 \in R^d\) is assigned to its most likely entity according to \(e = \arg \max_{e \in E} P(e) \times P(r|e)\). This algorithm has a time complexity of \(O(|M^d| + |E| \times |M^d|)\).

Model III has a similar 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 are given by \(P(e), P(e|e), P(e|e)\) respectively. Here we seek the most likely sequence of entities given those representatives in their appearing order using the Viterbi algorithm. The total time complexity is \(O(|M^d|^2 + |E|^2 \times |M^d|)\). The \(|E|^2\) component can be simplified by filtering out unlikely entities for a representative according to their appearance similarity.

### 3.3 Discussion

Besides different assumptions, some fundamental differences exist 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, it relies on other mentions in the same document for clustering. 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 entity assignments to the five mentions in the models could be as follows:

- **For Model I**, \(\text{mentions}(e_1) = \emptyset\), \(\text{mentions}(e_2) = \{m_1, m_2, m_3\}\) 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**, \(\text{mentions}(e_1) = \emptyset\), \(\text{mentions}(e_2) = \{m_1, m_2\}\) and \(\text{mentions}(e_3) = \{m_4, m_5\}\). Local dependency (appearance similarity) between mentions inside each document enforces the constraint that they should refer to the same entity, like “Steve Bush” and “Bush” in \(d_2\).

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

### 3.4 Other Tasks

Other aspects of “Robust Reading” can be solved based on the above inference problem.

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

\[m_1 \sim m_2 \iff \arg \max_{e \in E} P(e, m_1) = \arg \max_{e \in E} P(e, m_2)\]
for Model I and
\[ m_1 \sim m_2 \iff \text{argmax}_{e \in E} P(E^{d_1}, R^{d_1}, M^{d_1}) = \text{argmax}_{e \in E} P(E^{d_2}, R^{d_2}, M^{d_2}). \]

for Model II and III.

Name Expansion: Given a mention \( m^g \) in a query \( q \), decide whether mention \( m \) in the document collection \( D \) is a ‘legal’ expansion of \( m^g \):
\[ m^g \rightarrow m \iff c_{m^g} = \text{argmax}_{e \in E} P(E^g, R^g). \]

Here it’s assumed that we already know the possible mentions of \( e^* \) after training the models with \( D \).

Prominence: Given a name \( n \in W \), the most prominent entity for \( n \) is given by \( (P(e) \text{ given by the prior distribution } P_E \text{ and } P(n|e) \text{ given by the appearance model}) \):
\[ e^* = \text{argmax}_{e \in E} P(e) P(n|e). \]

4 Learning the Models

Confining 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 \( \theta \) (several underlying probability 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 \( (E^d_0, R^d_0) \) is assigned to each document \( d \) by an initialization algorithm. After this step, we can assume that the documents are annotated with \( D_0 = \{(E^d_0, R^d_0, M^d)\} \).
2. In the M-step, we seek the model parameter \( \theta_{i+1} \) that maximizes \( P(D_i|\theta) \). Given the “labels” supplied in the previous I- or E-step, this amounts to the maximum likelihood estimation. (to be described in Sec. 4.3).
3. In the E-step, we seek \( (E^d_{i+1}, R^d_{i+1}) \) for each document \( d \) that maximizes \( P(D_{i+1}|\theta_{i+1}) \) where \( D_{i+1} = \{(E^d_{i+1}, R^d_{i+1}, M^d)\} \). It’s the same inference problem as in Sec. 3.2.
4. Stopping Criterion: If no increase is achieved over \( P(D_i|\theta_i) \), 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 one, but much simpler in the sense that it does not have the notions of documents and representatives. So in the E-step we only seek the most likely 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 algorithms stop in our experiments.

4.2 Initialization

The purpose of the initial step is to acquire an initial guess of document structures and the set of entities \( E \) in a closed collection of documents \( D \). The hope is to find all entities without loss so duplicate entities are allowed. For all the models, we use the same algorithm:

A local clustering is performed to group mentions inside each document: simple heuristics are applied to calculating the similarity between mentions; and pairs of mentions with similarity above a threshold are then 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 cluster.

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 documents have already been annotated \( D = \{(e, r, m)\}^n \) from previous I- or E-step, several underlying probability distributions of the relaxed models are estimated by maximum likelihood estimation in each M-step. The model parameters include a set of prior probabilities for entities \( P_E \), a set of transition probabilities for entity pairs \( P_{E|E} \) (only in Model III) and the appearance probabilities \( P_{E|W} \) of each name in the name space \( W \) being transformed from another.

- The prior distribution \( P_E \) is modelled as a multinomial distribution. Given a set of labelled entity-mention pairs \( \{(e_i, m_i)\}^n \),
\[ P(e) = \frac{\text{freq}(e)}{n} \]
where \( \text{freq}(e) \) denotes the number of pairs containing entity \( e \).

- Given all the entities appearing in \( D \), the transition probability \( P(e|e) \) is estimated by
\[ P(e_2|e_1) \sim P(\text{wrt}(e_2)|\text{wrt}(e_1)) = \frac{\text{doc}(\text{wrt}(e_2), \text{wrt}(e_1))}{\text{doc}(\text{wrt}(e_1))}. \]

Here, the conditional probability between two real-world entities \( P(e_2|e_1) \) is backed off to the one between the identifying writings of the two entities \( P(\text{wrt}(e_2)|\text{wrt}(e_1)) \) in the document set \( D \) to avoid

Note that the performance of the initialization algorithm is 97.3% precision and 10.1% recall (measures are defined later.)
The appearance probability can be modelled differently by using other string similarity measures. We will compare the model described here with some other non-learning similarity metrics later.

4 The appearance probability can be modelled differently by using other string similarity metrics between names. We will compare the model described here with some other non-learning similarity metrics later.

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.

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. The maximum likelihood estimation of the transformation probability \( P(t, k) (t \in TT, a_k \in A) \) from annotated representative-mention pairs \{\((r, m)\}\} is:

\[
P(t, k) = \frac{\text{freq}(r, m) : v'_k \rightarrow v_k^n}{n}
\]

\( v'_k \rightarrow v_k^n \) 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 the three models on identifying three entity types (People, Locations, Organization); (2) comparing our induced similarity measure between names (the appearance model) with other similarity measures; (3) evaluating the contribution of the global nature of our model, and finally, (4) evaluating our models on name expansion and prominence ranking.

5.1 Methodology

We randomly selected 300 documents from 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 the 64 million possible mention pairs, most are trivial non-matching one — the appearances of the two mentions are very different. Therefore, direct evaluation over all those pairs always get almost 100% accuracy in our experiments. To avoid this, only the 130,000 pairs of matching mentions that correspond to the same entity are used to evaluate the performance of the models. 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 predicted pairs is then compared with the annotated 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: The appearance model induces a "similarity" measure between names, which is estimated during the training process. In order to understand whether the behavior of the generative model is dominated by the quality of the induced pairwise similarity or by the global aspects (for example, inference with the aid of the document structure), we (1) replace this measure by two other "local" similarity measures, and (2) compare three possible decision mechanisms – pairwise classification, straightforward clustering over local similarity, and our global model. To obtain the similarity required by pairwise classification and clustering, we use this formula \( sim_a(n_1, n_2) = P(n_1|n_2) \) to convert the appearance probability described in Sec. 4.3 to it.

The first similarity measure we use is a simple baseline approach: two names are similar iff they have identical writings (that is, \( sim_b(n_1, n_2) = 1 \) if \( n_1, n_2 \) are identical or 0 otherwise). The second one is a state-of-art similarity measure \( sim_a(n_1, n_2) \in [0, 1] \) 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).

Pairwise classification is done by pairing two mentions iff the similarity between them is above a fixed threshold. For Clustering, a graph-based clustering al-
Algorithm is used. Two nodes in the graph are connected if the similarity between the corresponding mentions is above a threshold. In evaluation, any two mentions belonging to the same connected component are paired the same way as we did in Sec. 5.1 and all those pairs are then longing to the same connected component are paired the number averages all entity types; numbers in parentheses represent People, Location and Organization respectively.

Finally, we evaluate the baseline and the SoftTFIDF measure in the context of Model II, where the appearance model is replaced. We found that the probabilities directly converted from the SoftTFIDF similarity behave badly so we adopt this formula \( P(n_1|n_2) = \frac{1}{1 + \exp(-(n_1-n_2))} \) instead of acquire \( P(n_1|n_2) \) needed by Model II. Those probabilities are fixed as we estimate other model parameters in training.

5.2 Results

The bottom line result is given in Tab. 1. All the similarity measures are compared in the context of the three levels of decisions – local decision (pairwise), clustering and our probabilistic model II. Only the best results in the experiments, achieved by trying different thresholds in pairwise classification and clustering, are shown.

The behavior across rows indicates that, locally, 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 the global model, and that the appearance model behaves better with it than a fixed similarity measure does. A second observation is that the Location appearance model 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 discriminative 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 example, taken from the corpus, illustrates the advantage of this model.
The average accuracy of name expansion of Model II is shown in Tab. 5. Here is an example:

**Query**: Who is Gore?

**Expansions**: Vice President Al Gore, Al Gore, Gore.

**Prominence Ranking**: We refer to Example 3.1 and use the ranking of the entities with regard to the value of \( P(e) = P(n|e) \) (shown in brackets) by Model II as follows.

**Input: George Bush**
1. George Bush (0.0448) 2. George W. Bush (0.0058)

**Input: Bush**
1. George W. Bush (0.0047) 2. George Bush (0.0015)
3. Steve Bush (0.0002)

### 6 Conclusion and Future Work

This paper presents an unsupervised learning approach to several aspects of the “robust reading” problem – cross-document identification and tracing of ambiguous 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 with a state-of-the-art discriminative model that does not take a global view. The experiments exhibit encouraging results and the advantages 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 of training the model; that is, when a new document is observed, coming, how to update existing model parameters; (3) integration of this work with other aspects of general coreference resolution (e.g., other terms like pronouns that refer to an entity) and named entity recognition (which we now take as given); and (4) scalability issues in applying the system to large corpora.

### Acknowledgments

This research is supported by NSF grants ITR-IIS-0085836, ITR-IIS-0085980 and IIS-9984168 and an ONR MURI Award.

### References

A. Bagga and B. Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In ACL.

M. Bilenko and R. Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. In KDD.

E. Charniak. 2001. Unsupervised learning of name structure from coreference data. In NAACL.

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 IWeb 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.