An Exploration of Entity Models, Collective Classification and Relation Description

Hema Raghavan, James Allan and Andrew McCallum
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{hema,mccallum,allan}@cs.umass.edu

ABSTRACT

Traditional information retrieval typically represents data using a bag of words; data mining typically uses a highly structured database representation. This paper explores the middle ground using a representation which we term *entity models*, in which questions about structured data may be posed and answered, but the complexities and task-specific restrictions of ontologies are avoided. An entity model is a language model or word distribution associated with an entity, such as a person, place or organization. Using these per-entity language models, entities may be clustered, links may be detected or described with a short summary, entities may be collectively classified, and question answering may be performed. On a corpus of entities extracted from newswire and the Web, we group entities by profession with 90% accuracy, improve accuracy further on the task of classifying politicians as liberal or conservative using collective classification and conditional random fields, and answer questions about "who a person is" with mean reciprocal rank (MRR) of 0.52.

1. INTRODUCTION

Information retrieval has traditionally been concerned with "document retrieval" from a document collection, each represented as a bag of words. A user has a specific information need, and the system provides a list of documents that satisfy all or parts of that information need. Typically the list is presented in an order of decreasing relevance, where relevance is determined by the system. Often it is the user's job to connect the pieces of information together in order to satisfy a precise information need. However, there is increasing interest in more structured and specific methods of satisfying information needs, such as question answering and data mining.

Data mining and link detection, on the other hand, have traditionally relied on structured data, organized into rich ontologies in relational databases. From this structured representation, data mining identifies patterns and trends. However, much data is provided in the form of free text, lacking this structure.

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

The standard approach to data mining from free text is two tiered: information extraction is applied to the corpus in order to obtain a structured database; this is followed by traditional data mining. This approach requires that an ontology be defined, and typically relies on (uncommon) high-accuracy extraction. It also removes the rich contextual language in which each entity mention occurs in the original text.

In this paper, we explore a middle ground between bag-of-words document retrieval and highly-structured datamining. We make minimal assumptions about ontological structure, and instead retain the contextual language around an entity to create a document-style representation of each entity. We call this representation an *entity language model*, or *entity model* for short. We apply several methods to these entity models in order to understand how they can be applied to mining information about these entities: grouping them, discovering links between them, classifying them and describing the semantics of the links.

Entity models are created by running an off-the-shelf named entity extractor over the corpus, and associating with each entity the words in a finite context around each of its mentions. Each entity is now described by a language model, (which in the case of this paper is a simple unigram word distribution). Although our information need may require distinguishing between politicians and atheletes, or require complex relations, we do not require that the extraction system have any knowledge of these. Basically we have a "document" for each entity, and can leverage the rich tools of traditional information retrieval in new ways.

Equipped with this view of the corpus, we set out to explore the capabilities of our approach in addressing the following types of information needs.

1. Questions regarding the description of the entity, for example, *Who is the managing director of Apricot Computers?* the answer to which is *Peter Horne*. Often the answer is explicitly present in the text, so that the task is one of finding the target phrase or sentence that contains the answer, as in traditional TREC-style question answering.

2. For some types of questions, the answer may not be explicitly present in the text, for example *What game does Martina Navratilova play?* the answer to which is *Tennis*. Often news articles, especially scorecards on the sports pages, mentioning Navratilova may mention the words Wimbledon or the US open, but not explicitly mention that the sport being referred to is Tennis. Correctly answering this question requires an extra level of indirection.

3. One may be interested in grouping entities into predefined

categories, or linking entities that are similar, or finding descriptions of why they are similar. Entities that are linked in this way may be linked because there exist actual social interactions between them. Alternatively, the links may not be explicit, but rather indicative of the fact that the linked entities are similar. Given similar entities, a user may also be interested in determining why they are similar.

4. The last type of information need is concerned with classifying an entity into various categories. Answers to questions such as *Is John Kerry liberal or conservative on gun control?* can be quite subtle and complex. Typically a document would discuss a senator's opinions on various issues. Using domain knowledge about correlations between issues, a human could easily arrive at the answer to the above question. Sentiment classification and opinion classification are related areas of research. Finding the answer to the above question can benefit from the use of relational "collective" classification schemes.

In this paper we aim to address all of these questions using entity models. In the following section we discuss past work which has ideas similar to our entity language models 2. We describe our model in detail in 3 and an evaluation of the same in section 5. In subsequent sections we address each of the above mentioned types of questions in turn. In section 7.1 we use entity language models to answer TREC style questions. We address the latter two types using document classification techniques . In section 7.3 we classify entities by their profession, whereas in section 8 we classify entities by their political bias. We give a description of our data in section 4 and explain similarity measures we experimented with in 6.

2. PAST WORK

Conrad and Utt [9] also considered breaking up the corpus into what they called pseudo documents. They ran an entity recognizer through a corpus. All paragraphs containing a mention of an entity were collapsed into a single document called a pseudo document. Their pseudo documents are very similar to our entity language models. They applied this to information visualization. We provide a more formal framework for the representation of these pseudo documents and extend the number of uses of this method of representation.

With the advent of the internet, the links between documents also play a significant role in determining the information contained in a document [5, 4]. Web retrieval also makes use of the links between documents to determine relevance, like in the HITS and PageRank algorithms. It has been found that tremendous gains can be obtained by exploiting these links. Similarly we expect that links between entities would also help. In this work we explore the possibility of generating indirect links between entities based on common words in their entity language models. This idea is similar to Conrad and Utt's indirect links, however the method of construction is different. They use these links for information visualization. We extend the link framework to relational classification

Although there has been a huge body of research on contextual information [25] we do not know of any work that studies the context of a named entity in particular. A lot of work in data mining tries to discover relationships and associations between entities [8, 11, 13], just as we do, but we know of no work that uses the probabilistic framework similar to the one that we have adopted.

3. ENTITY LANGUAGE MODELS

We define an entity language model (ELM) to be a probability distribution of words that are likely to be used to describe the named entity. For example, an entity model for *George W. Bush* would have *president*, *republican*, *conservative*, and other such words with high probability. It would also include names of strongly associated people (e.g., *Dick Cheney*), places (*Texas*), actions (*cut taxes*), and so on.

Given a large corpus of text, we construct a model for a named entity E as follows. First we find all occurrences of E in the corpus. We use a named entity extraction system to locate the entities and to provide an entity type (e.g., person, location, organization). If a name occurs as more than one type, we treat each type separately—e.g., *Ford* as an organization (company) and *Ford* as a person (Henry Ford).

The model is then computed from a maximum of m occurrences of E in the text and a window of $\pm n$ words surrounding each mention. We call these $(2n + 1)$ word windows *snippets*.

If we pool these m snippets into a combined "bag of words," we can calculate a maximum likelihood estimate for any word that appears around mentions of E. When snippets overlap we include the word only once. From this bag of words, a maximum likelihood language model for the entity E may be estimated as

$$P_{ml}(w|E) = \frac{cnt(w)}{N} \tag{1}$$

where $cnt(w)$ is the number of occurences of the word w in the "bag of words" created as above, and N is the total number of words in the entity language model. This model can be smoothed using the collection as the background as:

$$P(w|E) = \lambda P_{ml}(w|E) + (1 - \lambda)P(w|C) \tag{2}$$

A λ value of 0.6 was used throughout this paper.

4. DATA SETS AND TOOLS

For experiments in question answering, relationship modelling and traditional classification we used the TREC-8 corpus. We describe that data here. However for relational classification, we use a different data set which we describe later, in section 8.

To construct our entity language models we ran BBN's IdentiFinder[7] on the TREC-8 corpus to find names of people, places and organizations. The collection has 525,000 documents from the Foreign Broadcast Information Service (1996), Federal Register (1994), Financial Times (1992-1994) and Los-Angeles Times (1989-1990). IdentiFinder extracted 1,691,1654 entities, with a total of 14,688,360 occurrences. Since IdentiFinder outputs type information for each named entity that it extracts, we also have this information for each entity model.

For our classification experiments we needed to build categories for named entities. We chose the following categories by looking through a subset of the named entities in the TREC-8 corpus. Then a total of 162 entities were classified as

Politics	Political figures (48 entities)
Pop	Pop or rock music stars (12 entities)
Composers	Classical music composers (13 entities)
Actors	Movie actors (37 entities)
Sports	Tennis and basketball stars (52 entities)

The data was first split into two sets

- A training set (55 entities). The (arbitrarily chosen) training entities consisted of 15 entities each from the Sports, Politics, and Actors categories, 4 from the Pop, and 6 from the Composers category. Wherever initial labeled data for training was needed, it was obtained from this set.

	Values of n		
Entity	12	25	50
Marilyn Monroe	1.92	1.46	1.01
Martina Navratilova	2.25	1.68	1.50
Magic Johnson	1.68	1.39	1.11
Jimmy Carter	1.73	1.16	0.82
Dick Cheney	2.09	1.28	0.86

Figure 1: Clarity of five different models where snippets include n words to either size of the entity.

	Values of m				
Entity	50	100	300	500	1000
Marilyn Monroe	2.42	2.2	2.11	2.11	2.11
Martina Navratilova	2.50	2.40	2.25	2.25	2.25
Magic Johnson	2.21	2.00	1.75	1.68	1.68
Jimmy Carter	2.15	2.14	1.77	1.73	1.58
Dick Cheney	2.50	2.22	2.11	2.09	2.09

Figure 2: Clarity of 8 different models for different values of m

- A test set (107 entities). All testing was done using this group of entities.

The researchers are aware of the fact that this data set is small. Our intent though is to show that entity models have some value for classification and this data set is sufficient to support that belief.

5. INFORMATION CONTENT OF THE ENTITY LANGUAGE MODEL

In this section we perform an intrinsic evaluation of the entity language models. Entropy, Perplexity and Clarity are a few of the measures we can use for this purpose [20, 21] We chose clarity, which is defined as the KL-divergence between the model and the corpus.

$$Clarity = \sum_{w \in V} P(w|E) \log \frac{P(w|E)}{P(w|C)} \quad (3)$$

where V is the vocabulary of the corpus C. When the distribution of E is identical to C the $clarity$ score is zero. $P(w|C)$ is the probability of word w in the corpus C. It can be assumed to represent the distribution of words in general English.

Entity language models have two variable parameters- m and n. In the first set of experiments in this section we take $m = \infty$, which implies that all snippets were used. Figure 1 shows how the parameter values vary with n for $m = \infty$. If n is held constant at 12 and m is varied there is little decrease in clarity scores as seen in Figure 2.

Some names have less information than others. It is reasonable to assume that a name like *Janet* should have less information than say, *Janet Jackson*. A name like Janet is a common first name, whereas a mention of *Janet Jackson* most often represents one entity- the rock star. This idea is captured well by clarity as shown in figure 3. In the TREC-8 corpus there are 518 different names with Janet in them. Thus, clarity confirms with our expectation.

6. SIMILARITY MEASURES

Name	No. of variants	Clarity
Alice	3,844	0.85
Betty	506	0.79
Janet	518	0.44
Janet Jackson		2.32
Janet Weiss		2.69
Janet Reno		0.90

Figure 3: Clarity scores for some common first names and the much higher scores for *Janet* when combined with surnames.

In future sections we will often need to compute the similarity or distance (they are inversely related to each other) between two entity language models. Typically, the distance between documents in the vector-space model is computed using the cosine similarity measure or other vector-space measures. In the probabilistic setting, we would like to consider measures that compute the similarity(or distance) between probability distributions.

A number of similarity measures exist to compute the distance between probability distributions. One common measure is the Kullback-Liebler divergence and is defined as

$$D(p||q) = \sum_{w \in V} p(w) \log \frac{p(w)}{q(w)} \quad (4)$$

The KL-distance is not a distance measure as it is not symmetric and does not satisfy the triangle inequality. The Jensen-Shannon distance is a symmetric version of the KL distance given as

$$JS(p||q) = \frac{1}{2}(D(p||q) + D(q||p)) \quad (5)$$

The L_1 measure between two probability distributions is geometrically motivated and is given as

$$L_1(p, q) = \frac{1}{2} * \sum_{w \in V} |q(w) - p(w)| \quad (6)$$

We did some preliminary experiments on our training set to evaluate the L_1 measure, the KL-divergence and the Jensen Shannon measure. We found that the L_1 measure worked best for comparing two entity models. The L_1 measure is a distance measure, which we convert to a similarity metric with the following transformation

$$overlap(p, q) = 1 - L_1(p, q) \quad (7)$$

The overlap measure ranges from 0 for no overlap to 1 for perfect overlap.

We also did experiments where we asked human annotators to evaluate the L_1 measure. We compared 7 randomly chosen entities to 107 others and ranked the 469 pairs in order of increasing $overlap$ score. The list was given to each of two evaluators. They were asked to indicate the strength of the associations with a value from the set $(0, 0.25, 0.5, 1)$ with '0' indicating no similarity between the entities and a '1' indicating strong relationship. Whereas in the previous experiment associations were restricted to categories, here they were allowed to span categories. The scores of the two evaluators were averaged and binned. From Figure 4 it is apparent that for an overlap score of greater than 0.5 the annotator always found the entities to be related with a score of 1. Overlap appears to be slightly more conservative than a human judgement of the association between to entities. Nevertheless we can use overlap to determine the strength of the association between entities.

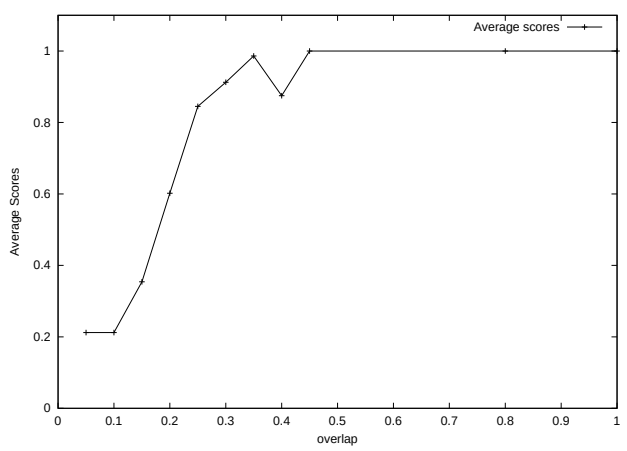

Figure 4: Relationship of evaluator assigned scores to a relationship with $overlap$.

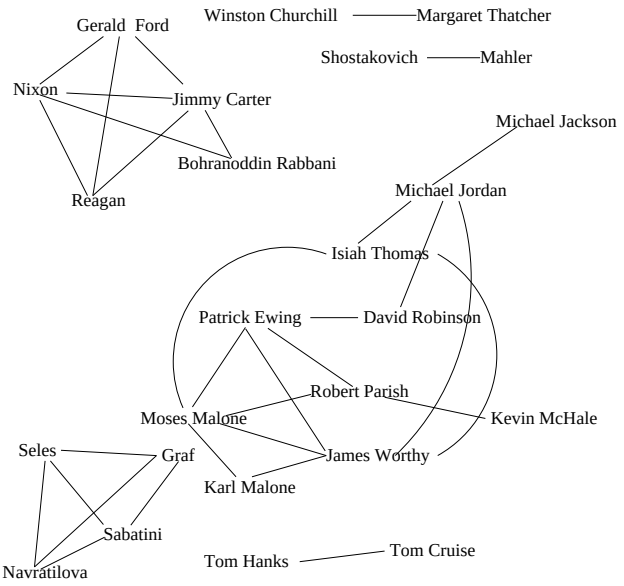

Figure 5: Network of entities, built with an overlap threshold of 0.20. The drawing was laid out by hand.

The overlap measure can be used to build a network of related entities as shown in Figure 5. The section on Relationship Description (section 7.2) attempts to extract the top terms that describe the similarity between two entities. The section on relational classification uses a network such as this one to classify named entities.

7. APPLICATIONS

7.1 Question Answering

In this section we explore answering questions of the type *Who wrote Margaret Thatcher's autobiography?*. These are traditional TREC [24] QA-like questions where the answer is present in a passage in the corpus.

Question Answering is the task of finding a specific answer to a question or query as opposed to retrieving an entire document for a query or question. Questions may be of several types depending on the type of the expected answer. In this case we consider only those questions which have entities as an answer. Such type of questions form a reasonable bulk of the questions in any TREC QA track. For example, in the TREC-8 QA [24] track 48.5% of the questions had named entities for answers-28% of the answers were names of people, 18.5% were locations, and 2% were organizations.

In the query likelihood method for traditional information retrieval the documents are ranked in the order of $P(Q|D)$. Typical QA systems [3] perform document retrieval or passage retrieval, and then look for the answer in the most likely passage. In question answering we wish to retrieve the answer E for a question Q. We propose to do this as

$$E = argmax_{E_i \in entities} P(Q|E_i) \quad (8)$$

That is, we wish to retrieve the entity E_i which is most likely to generate the answer. This would mean that $P(Q|E_i)$ be computed for all the entities in the corpus- a huge number(1.6 million in this case). Although, it should be feasible to index a collection of 1.6 million entities, it was beyond the scope of this paper. Because the number of entities is so large we did one round of document retrieval, and computed the above score for only those entities that occurred in the top N documents.

In order to evaluate the effectiveness of our idea, we used the protocol in the TREC-8 QA track. Each system could submit 5 answers to a question. For each question the system received points equal to the reciprocal of the rank at which the correct answer appeared. The Mean Reciprocal Rank (MRR) of a system is the average of the scores across all questions. For the TREC QA track, NIST provides the top 500 documents for each question, as returned by a document retrieval system. We used this list for the initial round where we build the list of entities to be scored as candidate answers. We also used the type information of the answer to a question in order to filter out entities that did not match the type information.

In the construction of the language models for this task, we chose to use $m = \infty$ because we do not have a principled way in which we choose which snippets to include in the model. In question answering this is crucial because the support of an answer may come from very few, sometimes even just one snippet. n was empirically chosen as 50.

Our system found the correct answer in the top 5 for 23 of 30 questions, and therefore obtained a MRR score of 0.52. We compared our performance to that of the top systems that year on the same set of questions. The median performance was 0.28, and the best was 0.78 (Cymphony). On our set of questions we would have been ranked fourth in that years evaluation.

Some of the errors were due to errors made by the tagger. For example all instances of *Agra* are tagged as organizations instead of locations. Some errors were due to the fact that the answer entity never appears in the top 50 documents returned by the document retrieval system. Both of these types of errors suggest that a potential place for improvement is to rank all answer types and all entities without using the two filters that we used, namely the document retrieval system and the question classification system.

7.2 Relationships between entities

Often we would like to ask of an entity *who or what is this entity like?*. While browsing through the news, if one came across a new name one might like to see who are the other names in the news that are similar to this one and why they are similar.

This is different from typical work on identifying relationships

Question	Answer
Who is the author of the book, The Iron Lady: A Biography of Margaret Thatcher?	Hugo Young
What is the name of the managing director of Apricot Computer?	Peter Horne
Where did Buzz Aldrin want to build a permanent manned space station?	moon
Which costume designer decided that Michael Jackson should only wear one glove?	Bill Whitten

Figure 6: Examples of questions where an entity model approach found the correct answer at rank 1

between entities, which looks at identifying the types of relationships between entities that co-occur in text. For example, some of the relationships to be identified in the ACE [22] task are EMPLOYEE-OF, WIFE-OF etc. Conrad et al's indirect links however do not rely on co-occurrence of entities, but rather on the similarity of what they call pseudo documents.

We saw in section 6 that the similarity between entities can be measured in many ways and that the $overlap$ measure was a good similarity metric for this purpose. We now attempt to model the similarity between two entities, such that the high probability terms are more descriptive of the relationship.

$$P^R(w|E_1, E_2) = \frac{min(P(w|E_1), P(w|E_2))}{\sum_{w \in V} min(P(w|E_1), P(w|E_2))} \quad (9)$$

The above equation computes a new distribution which captures the intersection between the distributions of E_1 and E_2 and then normalizes it to 1. If the maximum likelihood distributions are used for $P(w|E_1)$ and $P(w|E_2)$ we can smooth P^R as follows.

$$P_s^R(w|E_1, E_2) = \lambda P^R(w|E_1, E_2) + (1 - \lambda)P^R(w|C) \quad (10)$$

If we set $\lambda = overlap(P(w|E_1), P(w|E_2))$ then, when $E_1 = E_2$ we get $P_s^R(w|E_1, E_2) = P^R(w|E_1, E_2)$ and if $overlap(P(w|E_1), P(w|E_2)) = 0$ we get $P^R(w|E_1, E_2) = P^R(w|C)$. Therefore, if $overlap$ is zero, then the relationship is described by the model of general English.

We generated a set of entity pairs by randomly selecting 25 entities and then comparing each of those to 106 other entities. We discarded any of the 2,650 pairs that had an overlap score below 0.20, resulting in 69 pairs. For each pair E_1 and E_2 the top 10 terms with $P_s^R(w|E_1, E_2) > 0.01$ were selected. We call this a *relation description*. We asked two evaluators to mark the terms in the relation description as relevant or not depending on whether it described the relationship between the two entities. The evaluators were asked to make their judgements based on their a priori knowledge, and not to research the issue. One evaluator thought 63.8% of the terms extracted were relevant, and one thought 61.8 % were. When they worked together they found 61.8% terms relevant.

To score a relationship, we averaged the scores of all the terms in the relationship description. We have three scores for each pair: one from each judge and the result of adjudication. Figure 8 shows the distribution of the relationship scores. Most of the relationships have high scores (0.5-0.8), suggesting that a lot of the descriptive words are on target.

Figure 7 shows how the L_1 measure is used to measure similarity. For example, *Pete Sampras* is very similar to *Steffi Graf* (both are tennis players), he is less similar to *Michael Jordan* (a basketball player) and hardly similar to *Gerry Rawlings* (a politician).

E_1	E_2	overlap score	Actual relation	description of relation	
				Term	$P(w)$
Pete Sampras	Stefi Graf	0.39	Tennis players	champion	0.026
				wimbledon	0.023
				match	0.020
				tennis	0.019
				open	0.013
Pete Sampras	Michael Jordan	0.17	Sports players	player	0.015
Pete Sampras	Gerry Rawlings	0.01	None	*no terms extracted*	

Figure 7: Example of the top few terms in some sample relation descriptions.

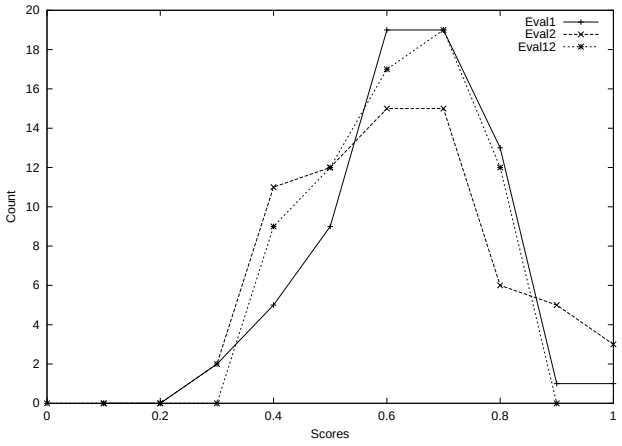

Figure 8: Score-wise distribution of *relation scores*. *Eval1* and *Eval2* denote the two evaluators; *Eval12* denotes the adjudicate result.

The table also shows the top terms in the relationship description. Note, that the evaluations for the relation descriptions are based on the a priori knowledge of our evaluators. For example, one of our evaluators marked the word *set* as non-relevant when it appeared in the description of the relation between two tennis players. However, the other evaluator, who was familiar with tennis, recognized this as a technical term in tennis parlance and therefore marked it as relevant. Sometimes there was a difference of opinion. For example, whether *money* is a descriptor of the similarity between two Hollywood celebrities is a matter of opinion.

7.3 Traditional Entity Classification

Given a name, it would be interesting to find out who this person is. For example, *Martina Navratilova* is a tennis player. We would like to decipher this, even if it is not explicitly mentioned in the text. If the word tennis is not mentioned next to any of the words in the neighborhood of *Martina Navratilova* but *Wimbledon* is, and say, Tennis and Wimbledon co-occur a lot in the vicinity of *Steffi Graf*, then we can say with a certain degree of certainty that *Tennis* has something to do with *Navratilova*. If we view this with a machine learning perspective, it is essentially a classification problem, where *Tennis* is a class label, and we have the knowledge (or training) that *Steffi Graf* is a tennis player, we can infer that *Navratilova*

is also one.

We tried four different classification algorithms- K-nearest neighbors, Centroid based clustering, Naive Bayes and Maximum Entropy [12, 16, 6].

7.4 Approach

For the training set we are given a set of entities E and their classes. A test entity E_t whose class needs to be determined is compared to each of the entities in the set E, and a list of K nearest neighbours is obtained. The entity E_t is put into the most commonly occurring class of these K nearest neighbors.

In the centroid based approach the entities in the training set E are split into groups by their class. If there are n classes represented from K_1 to K_n, we compute the centroid of each of these classes as shown below

$$P(w|K) = \frac{\sum_i^T P(w|E_i)}{\sum_w^V \sum_i^T P(w|E_i)} \tag{11}$$

Both approaches require that we compute the distance between two probability distributions. We stick to the L_1 metric, which we saw worked well for computing the distance between two entity language models.

The Naive Bayes classifier is described below:

Let $\mathcal{Y}$ denote a set of random variables and let $Y = (y_1, y_2 ... y_N)$, where $Y \in \mathcal{Y}$. Each y_i denotes the class label of entity i. Let $\mathcal{X}$ be the set of random variables on which we condition, such that $X = (x_1 ... x_N) \in \mathcal{X}$.

Let $x_i = \langle x_i^1, x_i^2, .. x_i^n \rangle$ be the counts of words in an entity language model with vocabulary size n. Classification then consists of selecting the label, y_i^*, with highest probability given the observed variables, $y_i^* = \arg\max_{y_i} P(y_i|x_i)$.

Naive Bayes makes an independence assumption among words given the label and defines

$$P(y_i|x_i) = \prod_j P_\theta(x_i^j|y_i)$$

for a given parameter setting, θ.

The above classifier uses Bayes rule. The priors are ignored. In this way each entity i is classified independent of the other.

The Maximum Entropy classifier is a discriminative approach and is given as,

$$P(y|x) = \frac{exp(\sum_i \lambda_i f_i(x, y))}{\sum_{x,y} exp(\sum_i \lambda_i f_i(x, y))} \tag{12}$$

$f_i(x, y)$ are feature functions. In this case the only features we used were the words in the text. We used the rainbow toolkit [17] to implement the Naive Bayes and Maximum Entropy classifiers.

7.5 Experiments

We ran 10 trials of the classification experiment as follows. In each run we randomly chose 40 out of 55 of our training instances, trained the classifier on these and tested on our test set of 107 instances. We then averaged the accuracies of these 10 runs. The table in Figure 9 shows the means and variances of each of the classifiers.

7.6 Discussion

The highest accuracy is obtained by the Naive Bayes classifier for $n = 12$ and $m = 500$. Also observe that the Politics and Sports categories exhibit low variance, whereas the Actors and Pop Stars categories, have higher variance. This is because Politicians and Sports person appear in the news for specific reasons. But, Actors and Pop stars tend to appear in the news for a wide variety of reasons, from films and music respectively to gossip. Therefore it seems reasonable that these would be hard to classify.

The natural question to ask then would be whether we can classify individual mentions of named entities. We tried classifying individual snippets using Naive Bayes and Maximum Entropy. We obtained accuracies of 86% and 69% respectively. The decrease in accuracy is easily explained as follows: each mention of a named entity may not contain sufficient information in a window of text around it to make a decision about its class accurately. However, by pooling in a bunch of mentions of named entities together with their surrounding text, we can make a more informed decision about its class.

8. RELATIONAL CLASSIFICATION

In the previous section the labels we assigned to an entity were independent of the label assigned to another entity. However, this is not necessarily true. Coming back to the *Navratilova* and *Graf* example, the decisions made for one entity influence those made for the other.

The network we built in section 6 had links between entities that were highly similar to each other. We would expect that labels of adjacent nodes in the graph to be strongly correlated to each other (that is captured by a K Nearest Neighbor approach also). Additionally we would also like to capture long range dependencies. Markov Random fields capture exactly these kind of dependencies. Any kind of graph lends itself to a Markov Random Field. We expect that Markov Random Fields would be useful in a setting where any of the classification schemes discussed in the previous section will not work well. We consider an example problem of classifying US government senators as liberal or conservative. If a senator A and a senator B are very similar on their opinion of say *abortion* one would expect that their views on another issue, say *gun control* would match.

Recent work on Relational Probability Trees and Relational Markov Nets [14, 5] have emphasized the importance of relational classification. However, in all those works the data is structured in the form of a database and some underlying link structure exists – either as hyperlinks or from the relational structure of the database. This basic structure is lacking when we are classifying a set of text documents and our only features are words. Our work also differs from similar work by Domingos [10] as we use Gibbs sampling for training and inference.

8.1 Data

We obtained the data for this problem as follows. We crawled *http://www.senate.gov* for the web-pages of all 105 senators. This was our test data. To obtain the true *conservative* and *liberal* scores of each of the senators we hired two undergraduates. They were asked to independently define what it meant to be conservative and liberal, and list a set of issues which would help determine the liberal or conservative bias of any person.

Ultimately the following set of 9 issues were chosen by both of them.

abortion	immigration	health
energy	education	economy
death penalty	civil rights	crime

The two evaluators were then asked to go through the list of senators and for each senator evaluate him or her on each of the 9 issues. They were asked to form their judgments on a five point scale from

Algorithm	m,n	Sports	Actors	Politics	Comp.	Pop	Average
5NN	∞,12	95 ± 3	81 ± 5	88 ± 1	**100 ± 0**	97 ± 4	90 ± 2
	300,12	$96 \pm$	95 ± 0	81 ± 2	78 ± 5	**100 ± 0**	90 + 1
Class Models	∞,12	91 ± 2	90 ± 4	89 ± 2	**100 ± 0**	**100 ± 0**	91 ± 1
	300,12	94 ± 0	**99 ± 2**	81 ± 1	86 ± 0	**100 ± 0**	91 ± 0
Naive Bayes	∞,12	94 ± 3	82 ± 5	**92 ± 1**	86 ± 0	79 ± 6	89 ± 1
	300,12	**97 ± 3**	87 ± 5	92 ± 3	86 ± 0	86 ± 7	**92 ± 2**
Max Ent	∞,12	84 ± 3	97 ± 2	88 ± 2	77 ± 7	56 ± 10	86 ± 2 *
	300,12	86 ± 3	96 ± 1	91 ± 1	81 ± 5	56 ± 8	87 ± 2

Figure 9: Percent accuracy for each of the classifiers. Boldface indicates the highest in the category. A two-tailed t-test was performed to compare the average accuracy of each classifier (last column) to Naive Bayes with $m = 300$,$n = 12$. Statistically significant differences (at $P < 0.05$) are indicated with an asterisk.

0-4, where 0 was most conservative and 4 the most liberal, for each of these issues, for each senator.

To do the evaluation the annotator first looked at the page dedicated to the given senator on issues2002.org. If the information on the page was sufficient to form an opinion, the senator was evaluated on that basis. If that information was insufficient, the annotator was allowed to perform a web search to arrive at the judgement. They were also asked to mark the overall leaning of the senator. The two annotators were asked to work independently and then where their scores differed by a value of 0.5 or more they were asked to resolve the conflicts, by working together. We also wanted to compare the entity model approach to what would be possible using purely structured data. The votes of each member of the senate on 365 bills of the 107th congress (2nd session) and 108th congress (1st session) were obtained from the above-mentioned site. For each entity we have the following attributes:
$E = (name, vote_1 ... vote_n,$ *Free-Text*$)$
The *Free-Text* attribute is obtained from the homepages of the senators as described earlier.

For the purpose of this paper, we considered the overall leanings of the senators only. Each of the values assigned by the annotators were rounded off to 0 or 1, depending on whether it was below or above 0.5 respectively.

8.2 Approach

When we did classification in section 7.3, we predicted the label of each entity independent of those of the remaining entities. Ignoring these dependencies is probably a reasonable assumption when trying to infer labels of entities where the class boundaries are more distinct. In a collective classification scheme, the labels of all entities would be decided simultaneously and not independently of each other. This scheme would take into consideration that there are complex relations between the entities and therefore their labels are decided in relation with each other. For example in a social network, the labels of one entity are highly correlated with those of its neighbours.

Let $G = (V, E)$ be an undirected graphical model. Let $W = (w_1 ... w_n)$, denote the random variable representing labels for each of the nodes. Let w denote an assignment of values to these random variables. The Graph G has a set of cliques C(G). W_c is the random variable representing a labeling of set of nodes in a clique $c \in C(G)$. The clique potential $\phi(w_c)$ determines how the members of a clique cohere to each other. A Markov Random Field is defined as

$$P(W) = \frac{\prod_{C \in C(G)} e^{\phi(w_c)}}{Z} \tag{13}$$

The probability of a configuration is obtained by a product over all cliques.

The generality of the model allows it to be applied to many different applications, outside of statistical physics. For example, in vision, it can capture that a given pixel is highly influenced by its neighbours. In social networks, it is valid to assume that the behaviour of a given individual is influenced by his or her neighbours. Neighbourhood, in a social network can be defined in many different ways - peers, family, people who think similarly etc. Behaviour can be anything: in our specific case we model biases of people, as either liberal or conservative. Applying a Markov Random Field to social networks has been suggested before [15, 10]. Domingos[10] successfully used a Markov Random Field for a collaborative filtering task.

We now extend the above model to a conditional Markov Random Field, and explain it in terms of the entity classification problem. Consider the space of entities to be classified. There are tied parameters associated with sets of entities. These sets are called clique templates. A clique template C defines a structure on the set of entities. Each template specifies a set of cliques of size 1 or more. A clique template defines the links between pairs of entities. For example, one clique template could define links between pairs of entities who use the word *pro-choice* often on their web-page. Many such clique templates may be defined. Consider that $\Im$ is a set of clique templates.

The conditional Markov Random Field is described below:

$$P(y|x) = \frac{1}{Z} \prod_{C \in \Im} \prod_{c \in C} \exp w_c.f_c(x_c, y_c) \tag{14}$$

where $Z = \sum_x \prod_{C \in \Im} \prod_{c \in C} \exp w_c.f_c$, $w_c = (w_c^1, w_c^2 .. w_c^n)$, $f_c = (f_c^1, f_c^2 .. f_c^n)$, $w_c.f_c = \sum_n w_c^i f_c^i$, f_c^i is a feature on a clique, and w_c^i is the weight of this feature. For example, f_c^i could take the value of 1 when all members of the clique satisfy a given property and be 0 otherwise. $w_c.f_c$ defines a clique potential, and determines how closely the members of the clique are correlated with each other. The weights for each of the features is learned during training. Note the similarity between equations 13 and 14.

In the model described by Equation 14, $P(y|x)$ assigns a label to all the entities simultaneously, conditioned on all the data of all of the entities. The model is a product over all cliques for each template, and the functions f are defined over all entities in the clique. In this way the model captures interdependecies between neighbours, where neighbourhood is defined by a clique template. For a more intricate explanation of the model refer Taskar et al's original work on Relational Markov Nets [5].

Clique templates are useful when the underlying relationships between the entities are explicit, like in a relational database, or a hyperlinked environment. But, when the relationships between

these entities is not clear, one solution is to consider that all entities are related to each other. Equation 14 reduces to

$$P(y|x) = \frac{1}{Z} \prod_{\forall (x_1,x_2) \in C} P_w(y_1, y_2|x_1, x_2)$$
$$P_w(y_1, y_2|x_1, x_2) = \exp w_c.f_c(x_1, x_2, y_1, y_2)$$

where $C = \{(i, j)|i \neq j; (i, j) \in C \Leftrightarrow (j, i) \notin C\}$
This template makes no assumptions about which entities are related to each other. It simply considers that each entity may be correlated to one or more of the others. The strength of the relationships between each pair of entities is determined by the clique potentials on the link between the pair.

The Log Likelihood function for this family of functions, for a given training set $X = (x_1...x_n)$, and a given parameter setting w is given by:

$$L(w, X) = \sum_{\forall x_1,x_2 \in X} \log P_w(y_1, y_2|x_1, x_2) \quad (15)$$

We use likelihood of the data as an objective function, and try to find the value w that maximizes this. Expanding the Log-Likelihood function in Equation 15 we get:

$$L(w, X) = \sum_{\forall x_1,x_2 \in X} (w.f(x_1, x_2, y_1, y_2) - \log(Z(x_1, x_2))) - \frac{w \cdot w}{2\sigma^2} + C \quad (16)$$

The function is concave, and is maximum when its gradient is zero. The gradient of this function is given by:

$$\bigtriangledown L(w, X) = \sum_{x_1,x_2 \in X} (f(x_1, x_2, y_1, y_2) - E[f(x_1, x_2, y_1, y_2)]) - \frac{w}{\sigma^2} \quad (17)$$

In Equation 16, σ is a gaussian smoothing parameter. From equation 17, it is clear that the function is maximized when the expected value of the feature counts, as assigned by the model, equals the feature counts observed in the training data.

The Likelihood function is maximized using Conjugate gradient. Although typical implementations of conjugate gradient code require that both the function and its gradient be calculated at each point, it is important to note that the function in equation 16 requires the calculation of Z, which is the sum of an exponential number of terms. But, a simple modification to conjugate gradient [1] where the line-minimization function is manipulated such that the step size α is computed only with the knowledge of the first derivative solves this problem. The trick is to do the line minimization to a point where the inner product of the direction of α and the value of the derivative at that point is zero. In Equation 16,

$$E[f(x_d, y_d)] = \sum_{y'} P_w(y'_d|x_d) f(x_d, y'_d)$$

The value of the expectation is the sum over all possible y', which for k possible labels for each y_i and n entities, is itself k^n, i.e it is exponential in n.

Using Gibbs sampling we obtain a set of samples. The expected feature count can be calculated as the average feature count of these samples. In this way, the function can be maximized without the calculation of Z which is a sum of an exponential number of terms.

Normally, the training data would be discarded after w is learnt. Instead we include the training data during inference, forcing their labels to be the known values.

That is, we want to estimate:
$argmax_y P(y|x)$
where
$y = (y_1...y_t, y'_1..y'_T)$
$x = (x_1...x_t, x'_1..x'_T)$ and
t is the number of training instances
T is the number of testing instances.

$y_1...y_t$ are set to be their known values. $P(y|x)$ is therefore evaluated over all possible values of $y'_1...y'_T$, an exponential number of possibilities. This problem is not new and there exist several approximate inference algorithms for such models. Taskar used loopy belief propogation [18]. However, he found that LBP did not do well with cliques [23] of size larger than 3. Hence we resort to sampling. MCMC methods [2] exist for calculating expectations in these models . We used Gibbs sampling [19], which is one kind of MCMC method.

We start with a random assignment of labels $y = (y_1..y_n)$. In each subsequent iteration we sample each y_i, keeping the labels of all other $y_j, i \neq j$ at the value sampled in the previous iteration as follows:

$$y_1^{t+1} \sim P(y_1|y_2^t...y_n^t, x_2^t...x_n^t)$$
$$y_2^{t+1} \sim P(y_1|y_1^t, y_3^t...y_n^t, x_1^t, x_3^t...x_n^t)$$
$$y_n^{t+1} \sim P(y_n^t|y_1^t...y_{n-1}^t, x_1^t...x_{n-1}^t)$$

By repeating the above for a sufficient number of iterations, the assumption is that we arrive at a final configuration y, which has high probability.

8.3 Experiments

Three types of features were used. The researchers expected that if two entities have the same opinion on a subject, they are similar. We chose the features based on that expectation.

For a given entity, we estimate the entity language model using all the pages on his or her home-page. Additionally we estimate a pseudo document of an *issue*, which is obtained by pooling together all fixed size windows of text around the mention of an issue word i. Examples of issues are Abortion, Gun Control, etc. The Issue pseudo-document is given as $P(w|, E, k)$ where P denotes the probability distribution of a model of senator E's opinions on issue k. Thus an entity can be considered as a mixture of issues.

1. The first feature is based only on the votes, and tries to encode our expectation that, if two people vote similarly, then their labels should be the same. It is given as:
   $f_k^1(x_i, x_j, y_i, y_j) = 1$ if E_i and E_j vote identically for vote k and $y_i = y_j$
   else $f_k = 0$

2. The second feature tries to encode the same in text: i.e, if two people's discussions of a given issue are very similar then their labels should be the same.
   $f_k^2(x_i, x_j, y_i, y_j) = 1$ if $overlap(P(w|k, i), P(w|k, j)) > 0.8$ and $y_i = y_j$ where $P(w|k, i)$ is the mixture component of topic k of the model of entity i.
   else $f_k = 0$
   We use $\approx$ to denote the similarity of the two distributions computed by the L_1 metric.

3. The third feature takes into account which words in the vocabulary contribute to a high similarity. Note that the second feature does not take this into account.
   $f_{k,m}^3(x_i, x_j, y_i, y_j) = 1$ if $P(m|k, i) > 0$ and

Data	30 senators		60 senators	
No votes	NB	MRFs	NB	MRF
5	51	90	57	84
10	78	87	70	87
15	73	90	71	85
20	82	90	75	88
40	87	93	84	94
80	58	81	51	81

Figure 10: Accuracy of classification for 2 different data set sizes. In each case 3 fold cross-validation was performed with a 50% split of training and testing data

$P(m|k,j) > 0$ and $y_i = y_j$
else $f_k = 0$
For example in feature f^3, if i =*abortion*, and k =*pro-life*, we expect that the learning algorithm will learn a higher weight, as opposed to the case when i =*abortion*, and k =*budget*:

The accuracies obtained using all three types of features are shown in figFor features f_2 and f_3 we used 3 issues only – abortion, environment and education. If a senators page did not contain a reasonable number of mentions of any of these issue words, then the senator was removed from the training or testing sets. We did all experiments on two sets of data, one having 30 senators and one with 60 senators to test how accuracy scales with increase in the number of entities. There does not seem to be a noticeable decrease in performance by increasing the number of entities.

When we used all three features we obtained a high accuraicy On closely examining the weights learned for the different features we observed that the weights for feature f_1 were the highest. It would be interesting to see what our performance would be if we used only the features obtained from pure text. When we use only features f_2 and f_3 we obtain an accuracy of 77%.

Our simple preliminary experiments using only the text on the home-page of a senator and a Naive Bayes classifier gave us an accuracy of 60% . Simple bag of words are useful when discriminating between distinct categories like Politics and Sports. But within such a specific domain, there is no clear distinction between the vocabulary of the two classes, viz. *conservatives* and *liberals* and therefore we need to go beyond bag of words.

9. DISCUSSION

We used a weakly structured representations entities, namely entity models for relational classification. We saw that performance was very high when we used the votes as features, a completely structured representation. However, by not restricting ourselves to structured features we have a mechanism for classification which is more flexible. By using newspaper snippets of entities we are not restricting ourselves to a particular domain, which in this case is the US Senate. Without confining to structure we can now classify other politicians, new senators with no voting record etc.

There are other computational aspects of the model that we have not discussed. Gibbs sampling, which we used for computing approximate expectations, has several convergence issues. For this work we ran Gibbs sampling for a certain fixed number of iterations. We leave the discussion of inference in these models for future work. We used a minimal set of features, in the future we can use some feature induction techniques in addition to more feature engineering.

10. CONCLUSIONS

Names of people, places and other entities appear in the news all the time. To deduce information about these named entities is an interesting problem and has many practical applications. For example, if you are browsing the news it would be interesting to be able to click on a name and get information about the entity associated with that name. A user might want to learn more about an entity by reading a summary of his or her identity, or might want specific questions about the entity answered. A user might also be interested in finding people related to the entity. It would also be interesting to find hidden attributes about an entity. In this paper we proposed a basic framework for representing an entity, and then went on to explore all of the above applications in turn, at times making modifications to the basic framework. The experiments, although small sample experiments in each case, showed promising results. Although this work is not complete in itself, we believe that it is a new direction for Information Retrieval and Data Mining research.

11. ACKNOWLEDGEMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

12. REFERENCES

[1] http://www.inference.phy.cam.ac.uk/mackay/macopt.html.

[2] *Learning in Graphical Models*, chapter Introduction to Monte Carlo Methods. MIT Press, 1999.

[3] S. Abney, M. Collins, and A. Singhal. Answer extraction. In *Proc. of the 6th ANLP Conference*, pages 296–301, 2000.

[4] E. Amitay. Using common hypertext links to identify the best phrasal description of target web documents. In *Proc. of the SIGIR'98 Post-Conference Workshop on Hypertext Information Retrieval for the Web*, 1998.

[5] P. A. B. Taskar and D. Koller. Discriminative probabilistic models for relational data. In *18th Conference on Uncertainty in Artificial Intelligence*, 2001.

[6] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[7] D. M. Bikel, R. L. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231, 1999.

[8] R. Byrd and Y. Ravin. Identifying and extracting relations from text. In *NLDB*, 1999.

[9] J. G. Conrad and M. H. Utt. A system for discovering relationships by feature extraction from text databases. In *Proc. of the 17th ACM-SIGIR Conference*, pages 260–270, 1994.

[10] P. Domingos and M. Richardson. Mining the network value of customers.

[11] J. Dorre, P. Gerstl, and R. Seiffert. Text mining: Finding nuggets in mountains of textual data. In *Proc. of the 5th ACM SIGKDD*, pages 398–401, 1999.

[12] E.-H. Han and G. Karypis. Centroid-based document classification: Analysis and experimental results. In *Principles of Data Mining and Knowledge Discovery*, pages 424–431, 2000.

[13] M. Hearst. Untangling text data mining. In *Proc. of ACL*, 1999.

[14] L. F. a. M. J.Neville, D.Jensen. Learning relational probability trees. Technical Report 02-55, University of Massachusetts, 2002.

[15] R. Kinderman and J. Snell. Markov random fields and their applications, 1980.

[16] D. D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In *Proceedings of ECML-98*, pages 4–15, 1998.

[17] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow, 1996.

[18] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. pages 467–475.

[19] D. G. S. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. pages 721–741, 1984.

[20] C. Stanley, F. Douglas, and B. Ronald. Evaluation metrics for language models. In *DARPA Broadcast News Transcription and Understanding Workshop.*, 1998.

[21] W. B. C. Steve Cronen-Townsend, Yun Zhou. Predicting query performance. 2002.

[22] B. M. Sundheim. Third message understanding evaluation and conference (muc-3): phase 1 status report. In *Proceedings of a workshop on Speech and natural language*, pages 301–305. Morgan Kaufmann Publishers Inc., 1991.

[23] B. Taskar. Personal communication.

[24] In *The Eighth Text REtrieval Conference (TREC 8)*. NIST, 1999. NIST Special Publication 500-246.

[25] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18(1):79–112, 2000.