Searching for the Blackbox: Unsupervised Recovery of Relational Schema from Unstructured Airplane Crash Reports

Evan Donahue
Dept. of Computer Science
Brown University
emdonahu@cs.brown.edu

Abstract

We present a two-stage, unsupervised statistical method for first extracting a relational database schema from a collection of unstructured documents that express common characteristics, and then populating a database described by the extracted schema. In particular, we present a corpus of 1759 news articles about airplane crashes, learn a schema containing flight number, plane model number, and number of passengers killed, and subsequently populate a database defined by that schema. The first stage of our approach operates in two phases, first using PCFG phrasal nodes to cull some of the variance in the local linguistic contexts surrounding the data we intend to extract, and then applying a Gibbs sampling clustering process to discover common patterns that form the relational schema. This paper will focus primarily on the first stage of our method, as it is the best developed at the time of writing.

1 Introduction

Robust methods of imposing structured relations on unlabeled data will likely prove crucial in developing strong text generation methods as well as have potential applications in a wide range of research areas. Current largely supervised methods often rely on fine-grained, domain-specific named entity taggers and hand-crafted schema that, while often highly effective in their domains, must be effectively redesigned for each new domain, if such redesign is even practical given the semantic limitations of NER tagging methods. In this paper, we present an unsupervised statistical approach for extracting a nuanced and semantically relevant relational schema from a collection of documents with similar characteristics, and for laying the groundwork for populating a database with information described by that schema. By examining at the linguistic context surrounding a single piece of information[1] as it is expressed throughout a corpus of related documents, we can begin to build a statistical profile of how that type of information is expressed, and by relating pieces of information that are frequently expressed the same way, the relational structure of our corpus starts to become clear. In particular for the purposes of this paper, our corpus will consist of 1759 news reports of plane crashes, and by examining numbers and named entities that show up frequently in the context of a given crash, we can begin to guess which information is important and common to a plane crash report, and collect categories of lexical patterns that will characterize the different types of information that form the columns in a relational schema.

2 A Formal Model

2.1 The Data

Our corpus consists of 1759 mainstream news articles reporting on airplane crashes. In order to illustrate the workings of our algorithm, we will use the following sentence concerning Continental Flight 3407 from our corpus as a consistent example

[1] A piece of information for our purposes is a specific number or a basic named entity (person, location, organization)

throughout the paper:

> *Co-pilot Rebecca Shaw pulled an all-nighter before she got on the commuter plane that nosedived into a house near Buffalo, New York, killing all 49 passengers on an icy February night.*

While the question of what information is "important" will vary depending on the context and is a question perhaps best left to the philosophers, for our purposes we propose the working definition that numeric information and information representing "named entities" (people, locations, and organizations) are all candidates for inclusion in a relational schema for a given set of documents. Taken together, numbers and named entities that share common linguistic contexts will be referred to as belonging to the same semantic "attribute" and specific numbers and named entities will be referred to as "values" for the remainder of this paper in reference to the terminology for columns and values repsectively in a relational database schema. In the case of our example sentence, we observe three possible candidates for values and by extension attributes, denoted by the underlined portions:

> *Co-pilot Rebecca Shaw pulled an all-nighter before she got on the commuter plane that nosedived into a house near Buffalo, New York, killing all 49 passengers on an icy February night.*

The basic insight of our approach is that given a large number of articles about plane crashes, we might expect to see the phrases "Co-pilot PERSON" "near LOCATION" and "killing all N passengers" many times, making them promising candidates for attributes, and we will be able to tell that "killing all N passengers" represents the same information as, for instance, "N people died" because both will take the value 49 when speaking about Continental flight 3407. Our basic high-level approach is as follows:

- Partition the documents into clusters representing distinct events (in our case, different crashes)[2]
- Within each cluster, identify common values such as "Rebecca Shaw" and "49" and collect into groups the immediate linguistic context[3] of each expressed value[4].
- Take each linguistic context type[5] a document in a topic modelling sense and apply a Gibbs sampling topic-modelling-like operation to these "documents" across flight clusters to collect global categories of similar linguistic contexts that tend to express the same attributes.
- Use the clusters of linguistic contexts as a statistical basis for identifying latent attribute values expressed in new documents, thereby filling the "database."

2.2 Local Lingustic Contexts

Our process operates on an intuitively defined notion of a "linguistic context" that characterizes the way that a certain attribute is lexicalized. Here we attempt to more formally define the notion of a linguistic context as it applies to our approach. Fundamentally, we want to capture a context that is limited enough that it will appear with reasonable probability across a number of documents, yet expansive enough to usefully identify only the relevant attribute being expressed. A reasonable place to start looking for such a context, we contend, is in a set of basic PCFG parse trees for our corpus generated using the Charniak parser [1999]. For our example datum of the 49 people killed on Continental flight

[2] We did this by searching for the flight number using regular expressions, which was fairly successful. While ours was a simple domain-specific approach, it is easy to imagine a more principled approach involving a clustering step based on the vector of value candidates in each article and indeed our later steps even indirectly depend on individual events expressing identifyable signatures of the type that would be useful in principled clustering.

[3] What constitutes the "immediate linguistic context" is one of our central objects of study. At present we are using elementary trees created in the context of parsing sentences with a Tree Substitution Grammar with some success.

[4] By value we mean to refer not to the number 49 in general, but to the fact that 49 people died on flight 3407. For this reason every "value" will be considered as a pair $\langle entity, flightnumber \rangle$ so we can be reasonably sure that two contexts expressing the same value are aligned to precisely the same entity

[5] As opposed to token. (killing all 49 passengers) and (killing all 78 passengers) are different tokens, but we count them as the same random variable.

3407, the immediate context of the parse is as follows:

> (VP (VBG killing) (NP (NP (DT all) (CD 49) (NNS passengers)) (PP (IN on) (NP (DT an) (JJ icy) (NNP February) (NN night)))))

This passage fairly unambiguously represents the number-of-deaths attribute in question, but walking far enough up the tree to encapsulate the main verb, as was done here, would likely yield too sparse a dataset. Unless many planes crash on "an ice February night," this representation needs to be simplified. Intuitively, we might want something like just

> (VP (VBG killing) (NP (NP (DT all) (CD 49) (NNS passengers))

but notice that the VP tag is opened without being closed. It is difficult to specify in a principled fashion exactly what portions of the tree we are interested in that balance the need to capture enough context with the need to saturate the context space. Nevertheless, experiments show encouraging results, and as we discuss in our Future Work section, our recent work with Tree Substitution Grammars (TSGs) shows promise for mitigating the weaknesses of PCFGs [2011].

3 The Model

3.1 Formal Problem Statement

For reference in the following equations, we provide this table of variables[6]:

Notation	Meaning
$\mathbb{A}$	Set of attributes[7]
A	Distribution over attributes in $\mathbb{A}$.
γ_A	Hyperparameter specifying how to smooth, weight, and shape the prior distribution over attributes A
N	Total number of observed linguistic context types. Equivalent to $\|\mathbf{L}\,\|$.
M	Total number of observed values.
$\mathbf{L}_j$	Random variable distributed over possible attribute labels for linguistic context j.
$\mathbf{L}^{(-j)}$	Random vector of contexts excluding context j.
L_j	Predicted attribute label for context j.
$\mathbf{V}_j$	Vector of counts of specific numeric and named entity values associated with context type j.
$\hat{\mathbf{V}}_j$	Frequency distribution over specific numeric and named entity values associated with context type j.
γ_V	Hyperparameter specifying smoothing over value probabilities.
$\mathbb{V}_a$	Vector of counts over all values associated with contexts labeled as $a \in \mathbb{A}$ given the vector of draws L
$\hat{\mathbb{V}}_a$	Empirical distribution over all values associated with contexts labeled as $a \in \mathbb{A}$ given the vector of draws L

Table 1: Notation Reference Guide

Our problem formally breaks down into two subproblems: the task of establishing meaningful attributes and the task of populating a database defined by those attributes with accurate values. For the first problem, we can formally define the relevant variables as follows:

- Let $\mathbb{A}$ be the finite set of possible attributes with length[8] $|\mathbb{A}|$. For example, the category of the number of people that die in any given plane crash may form an attribute (column) in our database.

- Let
$$\mathbf{L} = \langle \mathbf{L}_1, ..., \mathbf{L}_N \rangle \quad (1)$$
be a random vector distributed over possible labelings of linguistic context types where each dimension is a random variable ranging over the values in $\mathbb{A}$,
$$\mathbf{L}_j \in \mathbb{A} \quad (2)$$
In our case, if context j is the expression "all [N] people were killed" $\mathbf{L}_j$ would be the distribution of its proper attribute over the categories of the number of people that died, who the co-pilot was, what the flight number was, etc.

[6] The proceedure we use for the first stage task closely follows the Gibbs sampling framework for Naive Bayes that Resnik and Hardisty [2009] describe and so the notation has been extended where applicable for consistency. We will use bold for vectors, block lettering for sets and hat notation for empirical distributions and maximum likelihood estimators

[8] Currently $|\mathbb{A}|$ is a parameter of the model, but it could be a potentially useful extension to learn $|\mathbb{A}|$ from the data

- Let $\mathbf{V}$ be a vector such that each $\mathbf{V}_j$ is a vector of unnormalized counts[9] of each observed value appearing in context j, and let $\hat{\mathbf{V}}$ be the vector of normalized empirical frequency distributions over the values such that

  $$\hat{\mathbf{V}} = \left\langle \frac{\mathbf{V}_1}{\sum_{i=1}^{M} \mathbf{V}_{1_i}}, ..., \frac{\mathbf{V}_N}{\sum_{i=1}^{M} \mathbf{V}_{N_i}} \right\rangle \quad (3)$$

  If context j is again "all [N] people were killed," $\hat{\mathbf{V}}_j$ will be the vector of counts of how often any given value, such as 49, 109, 78, etc. showed up in an expression in place of [N]

- Let $\mathbb{V}$ be the set of count vectors and $\hat{\mathbb{V}}$ the corresponding set of empirical distributions over values associated with contexts belonging to the same attribute, so

  $$\hat{\mathbb{V}}_a = \left\{ \sum_{j=1}^{N} \frac{\mathbf{V}_j}{\sum_{i=1}^{M} \mathbf{V}_{j_i}} \mathbb{I}(\mathbf{L}_j = a) : a \in \mathbb{A} \right\} \quad (4)$$

  So if we take attribute a to be the category representing the number of people killed, we might expect "all [N] people were killed" and "[N] people died" both to be part of that attribute. $\mathbb{V}_a$ will then consist of the collected counts of each value that appeared in any of the associated contexts (e.g. 49, 109, 78, 34, 65 etc.). $\hat{\mathbb{V}}_a$ is the corresponding normalized distribution over values obtained by dividing each count in $\mathbb{V}_a$ by the total of all counts.

With this notation, we can describe the generative story that we will use to sample and adjust our model at every iteration of the Gibbs process.

3.2 Generative Story

Our goal at each iteration is to select an attribute $a \in \mathbb{A}$ for each linguistic context j characterized by its value distribution $\hat{\mathbf{V}}_j$ based on the conditional distribution

$$\mathbf{L}_j = \arg\max_{a \in \mathbb{A}} P(a|\hat{\mathbf{V}}_j) = \arg\max_{a \in \mathbb{A}} \frac{P(\hat{\mathbf{V}}_j|a)P(a)}{P(\hat{\mathbf{V}}_j)} \quad (5)$$

$$\arg\max_{a \in \mathbb{A}} P(\hat{\mathbf{V}}_j|a)P(a) \quad (6)$$

To make this problem tractable, we propose a Naïve Bayes assumption that each value associated with a given context is generated independently of all other values. The Naïve Bayes independence assumption allows us to compute

$$P(\hat{\mathbf{V}}_j|\mathbf{L}_j = a) \quad (7)$$

as a multinomial likelihood due to the independence of values being generated, and gives us an easy way to compose a generative model for our sampling proceedure. Given our notation, we propose the following genarative story for our first stage model:

- To generate a linguistic context, we first need to select an attribute $a \in \mathbb{A}$. We do this by first drawing a distribution A over attributes from

  $$A \sim Dirichlet(\gamma_A) \quad (8)$$

  We draw from the Dirichlet to impose an increasing cost on a clustering result that spreads contexts too widely over attributes. Our aim is to guarantee that the full range of contexts interconnected by common semantic categories will be compelled to cluster densely in the first few attributes, since we do not expect a huge number of useful semantic categories to define a given set of related articles. We next sample our distribution A to determine the attribute label of a given document with

  $$\mathbf{L}_j \sim Multinomial(N, A) \quad (9)$$

- Now given our attribute we need to sample the specific values associated with contexts under this attribute. We do this by drawing a distribution $\hat{\mathbb{V}}_a$ over values associated with attribute a, so

  $$\hat{\mathbb{V}}_a \sim Dirichlet(\gamma_V) \quad (10)$$

[9] To formally understand value counts as a vector we must imagine a global ordering such that each value has a unique sequential index i associated with it such that $\mathbf{V}_{j_i}$ represents the count of the ith value appearing in the jth context

The Dirichlet distribution here provides a way to give greater weight to words that show up frequently, allowing us to gradually collect the contexts that express the same values in the same attributes. From here we are almost ready to sample the values for a given context, save for one important assumption: we assume that each value is chosen independently, as per the Naïve Bayes assumption, so that we can model the sampling distribution with a multinomial like so:

$$\hat{\mathbf{V}}_j \sim Multinomial(N, \hat{\mathbb{V}}_a) \tag{11}$$

This sampling proceedure leaves us with a complete generative story for sampling a set of linguistic contexts grouped by attribute, and serves as the necessary precondition for our sampling proceedure.

3.3 Gibbs Sampling

The Gibbs sampling process requires us to formalize our problem as a graphical model that we can walk through probabilistically such that we asymptotically approach the expected state. In this case, we can define our state space as a k-dimensional random vector

$$\langle Z_1, ..., Z_k \rangle \tag{12}$$

where each dimension is a latent random variable from our generative model. Each state, consequently, is a sample

$$\langle Z_1 = z_1, ..., Z_k = z_k \rangle \tag{13}$$

drawn from the joint distribution of the random vector. In particular, we want to sample from the joint distribution of $P(\mathbf{L}, \hat{\mathbb{V}})$. Our generative model tells us how to sample from the distribution

$$P(L_i | L_1, L_2, ..., L_{i-1}, L_{i+1}, ..., L_N, \hat{\mathbb{V}}_{a_1}, ..., \hat{\mathbb{V}}_{a_{|\mathbb{A}|}}) \tag{14}$$

so to walk through the state space we can iteratively sample each L_i and $\hat{\mathbb{V}}_i$ one at a time and conditioned on all the other variables, in the sense that to sample at iteration t+1 we have

$$P(L_i^{(t+1)} | L_1^{(t+1)}, L_2^{(t+1)}, ..., L_{i-1}^{(t+1)}, L_{i+1}^{(t)}, ..., L_N^{(t)}, \hat{\mathbb{V}}_1^{(t)}, ..., \hat{\mathbb{V}}_{|\mathbb{A}|}^{(t)})$$

Resnik and Hardisty [2009] present a more detailed derivation of the sampling proceedure, but due to concerns for length here we present only the final proceedure for sampling a new state and some intuition regarding why this proceedure yields the results we expect.

3.4 Sampling From L

For each $\mathbf{L}_j$, we write

$$P(\mathbf{L}_j = a | \mathbf{L}^{(-j)}, \hat{\mathbb{V}}) = \frac{(\sum_{i=1}^{M} \mathbb{V}_{a_i}) + \gamma_A - 1}{N + (\sum_{x \in \mathbb{A}} \gamma_{A_x}) - 1} \prod_{i=1}^{M} \theta_{a,i}^{\mathbf{V}_{j,i}}$$

where the term

$$\frac{(\sum_{i=1}^{M} \mathbb{V}_{a_i}) + \gamma_A - 1}{N + (\sum_{x \in \mathbb{A}} \gamma_{A_x}) - 1} \tag{15}$$

is proportional to

$$\frac{(\sum_{i=1}^{M} \mathbb{V}_{a_i})}{N} \tag{16}$$

which is the multinomial maximum likelihood estimate for the number of contexts claimed by a given attribute given the current state. This term ensures that our contexts tend to collect in a small number of attributes, rather than loosing relevant contexts to random variability. The hyperparameter expression

$$\frac{\gamma_A - 1}{(\sum_{x \in \mathbb{A}} \gamma_{A_x}) - 1} \tag{17}$$

acts as a smoothing constant to determine how much weight to give to the likelihood term as well as to define how sharply the Dirichlet penalizes assignment to a wider range of attributes. The term

$$\prod_{i=1}^{M} \theta_{a,i}^{\mathbf{V}_{j,i}} \tag{18}$$

represents the multinomial likelihood of the distribution of values associated with a context given the distribution over values belonging to the attribute a. This expression ensures that the documents cluster meaningfully into attributes that express strong preferences for the values they represent. To actually sample from this distribution we simply select $\hat{\mathbf{L}}_j = a$ with probability

$$\frac{P(\mathbf{L}_j = a | \mathbf{L}^{(-j)}, \theta)}{\sum_{x \in \mathbb{A}} P(\mathbf{L}_j = x | \mathbf{L}^{(-j)}, \theta)} \tag{19}$$

. In order for attributes to represent meaningful categories however, we have to estimate their distributions over values as well.

3.5 Sampling from $\hat{\mathbb{V}}$

For each $\hat{\mathbb{V}}_a$, we write

$$\hat{\mathbb{V}}_a|\mathbf{L}, \hat{\mathbb{V}}^{(-a)} \sim Dirichlet(\mathbf{t}) \tag{20}$$

where

$$\mathbf{t} = \langle \hat{\mathbb{V}}_{a,1} + \gamma_{V1}, ..., \hat{\mathbb{V}}_{a,M} + \gamma_{VM} \rangle \tag{21}$$

This distribution guarantees that our newly sampled $\hat{\mathbb{V}}_a$ focuses its probability mass on the most common values expressed within attribute a. As we go back and forth between sampling $\mathbf{L}$ and $\hat{\mathbb{V}}$, the attributes begin to collect a consistent vocabulary of values shared among a related set of contexts, and the contexts begin to gravitate towards specific attributes that represent their values. As the sampling process proceeds in iterations, the state walk grows increasingly close to the expected value of the random vector of our parameters

$$\langle \mathbf{L}, \hat{\mathbb{V}} \rangle \tag{22}$$

and so we can learn from this the expected values of the individual labels, which are our main point of interest in this process.

3.6 Maximum Likelihood Estimate of L

As the Gibbs sampler walks it gets closer and closer to the expected value of the state vector, and so generates observations more in line with what we expect the true distribution to produce. We are interested in the maximum likelihood estimator $\hat{\mathbf{L}}_j$ of each label, as this will be our best guess at the correct label given the data. If we let the process run for a *burn-in* period before beginning our sampling process we can eliminate the first few highly random samples, at which point each sample will be more or less a sample from the true joint distribution to within a reasonable degree of approximation. Given a number of samples from the approximated "true" distribution, and given the assumption that each attribute is sampled independently of the last one[10], we can express the maximum likelihood estimator as the mode of the observed sampled attribute labels. Since we can see that since in the case of the multinomial

$$\mathbb{E}[x_i] = \mathbb{E}\left[n\frac{x_i}{n}\right] = np_i \tag{23}$$

the mode corresponds to the highest probability, which is what we want.

4 Results

Using the linguistic context underneath the first phrasal node above each value, we see reasonable results for numeric data:

Number of People Killed
[N] people
the [N] [N] mph
the [N] victims
[N] victims
the [N] victims [N] passengers
[N] lives
Flight Number
Flight [N]
flight [N]
Plane Model No.
A[N]
Airbus A[N]
The Airbus A[N]

Table 2: Example attribute grammars recovered using PCFG phrasal nodes

Named entities proved a harder problem due to higher variability in the structure of their phrasal nodes. In order to compensate, we have been working recently to deploy TSGs as detailed in our Future Work section.

5 Relationship to Previous Work

Information extraction from natural language to structured records is a fairly wide category, and as such there are many examples of previous work in the area. An exhaustive list would clearly be impractical, so we selectively draw comparisons that highlight the distinguishing features of this project and provide a sense of how this work might fit in to

[10] Since Gibbs sampling relies on an iterative process of improvement, it is certainly not true that one state arises independently of the last one. Some implementations attempt to remedy this *auto-correlation* by sampling only every n steps under the assumption that the additional mixing time will make draws less dependent on previous draws.

the general landscape that has already been developed for this type of information extraction.

Snyder and Barzilay [2007] present a multilabel classification representation of the task of aligning database records with information expressed in accompanying natural language for the stated purposes of potentially aiding language generation or training information extraction systems. The paper articulates a supervised approach based on already existing databases and matching natural language. Our task can be viewed as complimentary in that, with sufficiently structured data, we present an unsupervised approach to creating the database schema and populating the database, implicitly aligning records and linguistic contexts in the process. Our model does have one major domain restriction, however, namely that we require our training data to be completely separable into distinct records; because flight number makes up part of the data vector for every observed value, we need to be able to decide reliably what flight a given value represents. In our experiment, because the structure of the news article dictates that it generally refers to only one plane crash-the basic unit of the record on which we base all attributes-we were able to easily separate articles by flight number and so guarantee that each cluster belonged to a single flight number for the purposes of defining our feature space. Snyder and Barzilay perform database matching over a much more diverse and integrated domain that our model does not account for.

The particular nature of our dataset, while it does limit the scope of possible domains to those in which such a corpus is available, also enables us to ask a much different set of questions than researchers in this area have typically asked. Named entity based extraction is not new. Wick Cullota and McCallum [2006] demonstrate a method of extracting comprehensive records from web pages based on the cooccurrence of numeric and fine-grained NER-tagged data. The real point of departure in our work has been the use of correlated linguistic contexts from natural language made possible by the simplified nature of the elementary TSG trees. We accumulate correlated numeric and NER-tagged records, but we also accrue statistical information about a much wider feature space over expressive contexts that will hopefully allow this work to serve as a starting point for further investigation into dealing with noisy natural language data when trying to extract clear relational schema.

6 Future Work

In order to cull some of the variability in local contexts that persist in the PCFG rules, we have been working with promising results with Tree Substitution Grammars. Cohn Blunsom and Goldwater [2011] present a method for inducing a TSG from a treebank using a Gibbs sampling process to remove sections of a parse tree specific to a given sentence in order to find common parse structures. By eliminating areas of the parse tree that have high variability in favor of substitution points, TSG methods tend to produce common linguistic contexts that show great promise as substitutes for our PCFG phrasal nodes. Tree Substitution Grammars seem ideal for achieving a balance between capturing the complexity of structure in natural language and creating simple and meaningful atomic elements-elementary trees-that can serve as useful units for statistical tasks involving common linguistic structures, and hopefully we will shortly have results competitive with our numeric results.

Acknowledgements

We (I) thank Professor Eugene Charniak for advising this thesis, Professor Sudderth for agreeing to read it, and the BLLIP research group for providing a productive environment for this work. Special thanks to Micha Elsner for offering much advise early in the project, Ben Swanson for supporting and offering advice about the Tree Separated Grammars portion of the project, and especially to Ben Leib who has had significant involvement with most areas of the project and without whom the project certainly would not be as far along as it is at present. Additionally, we thank Vazheh Moussavi for endlessly helping to refine the notation and language for a rather notationally verbose explanation of Gibbs sampling.

References

Eugene Charniak. 1999. A maximum-entropy-inspired parser. pages 132–139.

Trevor Cohn, Phil Blunsom, and Sharon Goldwater. 2011. Inducing tree-substitution grammars. *Journal of Machine Learning Research*.

Philip Resnik and Eric Hardisty. 2009. Gibbs sampling for the uninitiated.

Benjamin Snyder and Regina Barzilay. 2007. Database-text alignment via structured multilabel classification. In *In Proc. of the International Joint Conference on Artificial Intelligence*, pages 1713–1718.

Michael Wick, Aron Culotta, and Andrew McCallum. 2006. Learning field compatibilities to extract database records from unstructured text. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 603–611, Sydney, Australia, July. Association for Computational Linguistics.