Tagging of Name Records for Genealogical Data Browsing

Mike Perrow
IDIAP Research Institute
Rue du Simplon 4
Case Postale 592
CH-1920 Martigny
Switzerland
mike.perrow@idiap.ch

David Barber
IDIAP Research Institute
Rue du Simplon 4
Case Postale 592
CH-1920 Martigny
Switzerland
david.barber@idiap.ch

ABSTRACT

In this paper we present a method of parsing unstructured textual records briefly describing a person and their direct relatives, which we use in the construction of a browsing tool for genealogical data. The records have been created by researchers who are currently digitising a collection of historical archives stored at the Abbaye de Saint-Maurice, Switzerland. The string 'Beatrix, daughter of Johannes Trona, of Saillon' is a typical example of a record. We wish to annotate every term (word and symbol) in our records with a label which describes whether the term is a name (e.g. 'Beatrix'), a place (e.g. 'Saillon'), or a relationship (e.g. 'daughter'). Using this information, we are able to derive both a canonical form for each name (e.g. 'Beatrix Trona'), and the relationships between people. We build upon work developed for the cleaning and standardization of names for record linkage corpora, adding several enhancements to deal with our more difficult data, which contains common name structures of French, Italian and Latin, over hundreds of years. We present an approach to this problem that works interactively with a user to annotate the data set accurately, greatly reducing the human effort required. We do this by learning a Hidden Markov Model representing a record structure, and finding structural patterns in new records. Finally, we present a brief overview of a tool we are developing to help genealogical researchers browse and search the data.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*; J.4 [**Computer Applications**]: Social and Behavioural Sciences

General Terms

Algorithms, Management, Design, Experimentation, Standardization

JCDL'06, June 11–15, 2006, Chapel Hill, North Carolina, USA.

Keywords

genealogy, information retrieval, visualization, tagging

1. INTRODUCTION

Genealogy is the study of family history, finding and tracing a person's ancestors and descendants using data collected from historical documentation. A rich source of data is found in documents archived in monasteries, as monks have in the past acted as sources of public record, recording events such as births, marriages and transfers of property. Typically this data is not well-structured, so a genealogist sifting through these documents faces a massive challenge to find and verify information on a specific family. Now that equipment exists to safely scan these documents into a digital form, we have for the first time the opportunity to attempt to analyse the documents and create tools to help genealogical researchers.

We are working in partnership with the Fondation des Archives Historiques de l'Abbaye de Saint-Maurice [2], who are in the process of digitising historical documents dating back as far as the 10^{th} century, stored in the Abbaye de St. Maurice, Switzerland [1]. A page from one of these documents is shown in figure 1.

Researchers at the fondation are in the process of scanning each page of each document and manually recording summary information. This information includes a short translated (into modern French) summary of the document, a date for the document[1] and a list of names mentioned in the document, along with any given relationships (mother, sister, etc.) between the names. Table 1 shows a selection of these name records, along with their English translations. Since name conventions in this multi-lingual region have changed over the centuries, and also due to inconsistencies in annotations by archivists, the records are not in any standard format, so before we can use these records, they must be parsed. In this paper we focus on the task of extracting names and relationships from these records.

There are two main goals for this stage of our work. Firstly, we wish to be able to translate a name into a canonical form, which can be displayed in some visualisation of the data. The record 'Beatrix, fille de Johannes Trona' in a simple 'FirstName SurName' canonical form would be 'Beatrix Trona'. Secondly, we wish to identify family relationships mentioned in these records. In this example we would like

[1]The date of a document cannot always be determined precisely, so we often have a range of dates

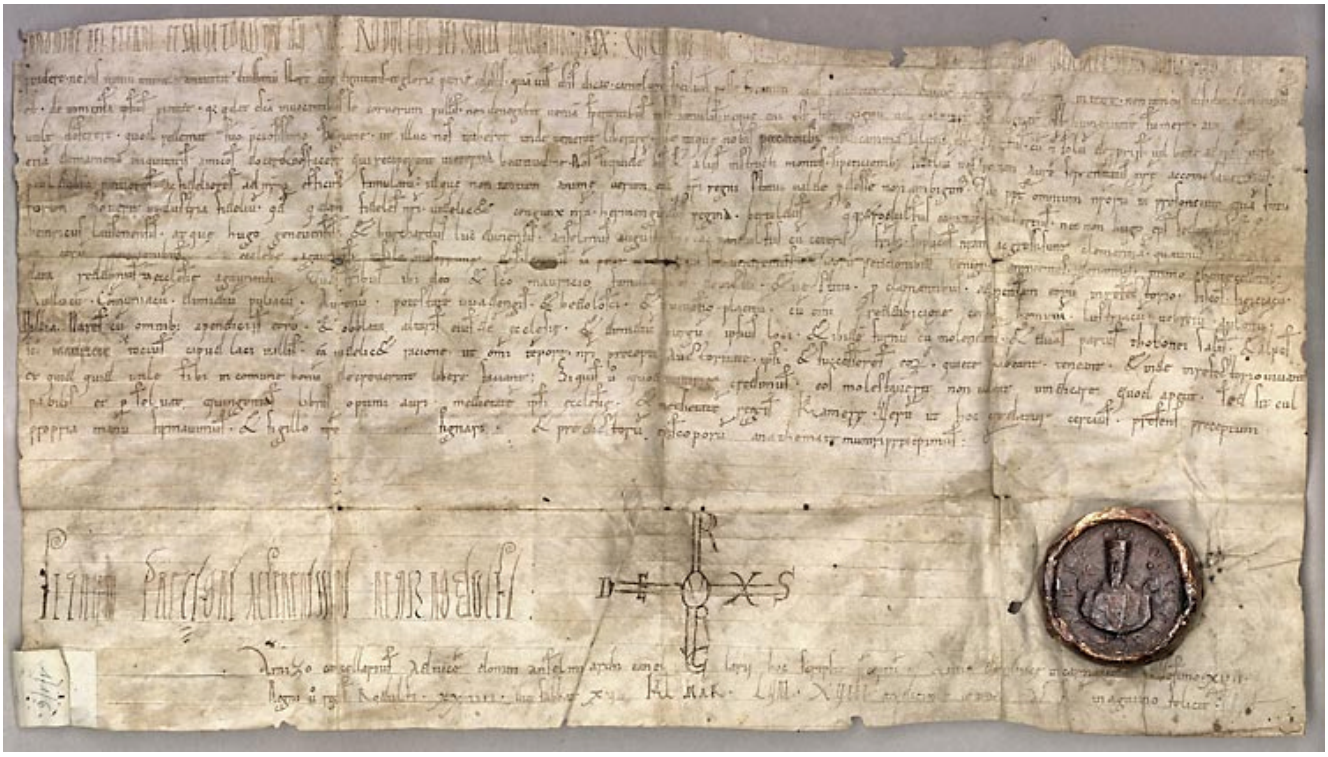

Figure 1: A page from an original document stored in the Abbey at Saint-Maurice .

to identify from the record two people, 'Beatrix Trona' and 'Johannes Trona' and the relationship between these two people, 'fille (daughter)'. With this information we can then present the data in a much simpler and easy to use form, as shown in figure 2. Our goal is thus to parse these records into a sequence of labels describing the various attributes of the person, such as their surname or profession. Figure 3 shows how a typical record should be parsed. In the future we plan to deal with the problem of *name disambiguation* or *person resolution*, the problem of identifying multiple people sharing the same name. This topic has been widely studied [13] [21] [12] [17]. In this work we do not attempt to solve this issue, and instead focus on the problem of matching names.

The goal of the work presented in this paper is to provide a tool for parsing these records, that can be used by a non-technical user, and can accurately predict the structure of labels for a previously unseen record. As new documents are summarised, our tool must be robust to changes in the terms used, and must adapt to new name structures quickly. Table 2 shows the selection of labels which we wish to assign to each term in the record string[2]. One of the compounding factors in this corpus is the fact that any single record may contain multiple names, places and or professions. In figure 3 we wish to extract the fact that there are two people, namely Beatrix Trona and Johannes Trona, the former being the daughter of the latter. Whilst the records are not in a standard format, it is clear that they follow some predictable patterns and that many terms (i.e. names and symbols) will be common across many records. Our approach is to try to predict the labelling of each new record by finding these patterns across the whole set of records.

[2]This selection may be readily changed to handle different corpora.

Table 2: A selection of labels used to model the records in our corpus.

Label	Description
FIRSTNAME	A First Name e.g. *Beatrix*
SURNAME	A Surname e.g. *Trona*
DE	*de, di, dou, du, etc.*
COMMA	The comma symbol ','
HYPHEN	The hyphen symbol '-'
PLACE	A place name e.g. *Salvan*
RELATION	A relationship e.g. *mother* or *daughter*
ROLE	A profession or role within the community e.g. *doctor* or *cantor*
CALLED	Used for aliases e.g. Anna *called* Ave
PERIOD	The period symbol '.'
(	Open brackets e.g. '(' or '{'
)	Close brackets e.g. ')' or '}'
AND	The word *'et'* and equivalent words

We build on an approach previously developed for cleaning name and address data [9], using Hidden Markov Models (HMMs) to learn the structures of the records. HMMs are probabilistic models that allow us to produce from a record a series of labels identifying the different parts of the record. We represent each possible label as a state, and the entire record can be thought of as a transition between states.

Table 1: A typical set of records for a document. The full corpus contains 80331 records, which we wish to parse into a standard format.

Original Records	English Translations
Perrodus de Salvan, clerc	Perrodus of Salvan, clerk
Petrus de Lydes, chantre de Saint-Maurice	Petrus of Lydes, cantor of Saint-Maurice
Trona, Johannes, de Salvan	Trona, Johannes, of Salvan
Martinus, fils de Johannes Trona, de Salvan	Martinus, son of Johannes Trona, of Salvan
Beatrix, fille de Johannes Trona, de Salvan	Beatrix, daughter of Johannes Trona, of Salvan
Lorio, Aymo	Lorio, Aymo
Jaqueta, femme d'Aymo Lorio	Jaqueta, wife of Aymo Lorio
Johanneta, fille d'Aymo Loryo	Johanneta, daughter of Aymo Loryo
Postelen, Petrus	Postelen, Petrus
Petrus, mari de Anna, veuve d'Aymo Loryo	Petrus, husband of Anna, widow of Aymo Loryo
Willermus Michie, de Combis Inferioribus	Willermus Michie of 'Combis Inferioribus'
Petrus dit de Castellione de Lugrino, donzel	Petrus named as Castellione of Lugrino, 'donzel'

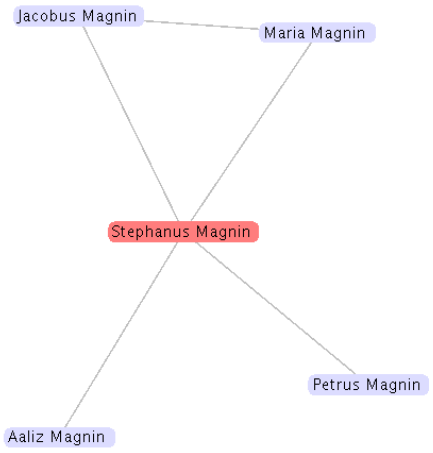

Figure 2: A view of the data from our prototype browser application, where a user can clearly see people related to each other, and select names to see information (documents summaries, sates, etc.) associated with that name.

In the example above, the record starts in state 'FIRSTNAME' and moves to state 'COMMA', and then to state 'RELATION' and so on. HMMs model the probabilities of transitions between states, allowing us to estimate the probability of being in a certain set of states given the observed sequence of words and symbols.

We expand on previous work by using an online learning algorithm to quickly begin learning these structures with very few hand annotations, eliminating the need for any previously hand annotated training data before the HMM is employed. In addition, we present a method for using uncertain annotations of parts of the record, and encoding rules into our annotations, using virtual evidence [8]. This has the advantage of combining rule-based and probabilistic approaches within the same model. Indeed, the professional

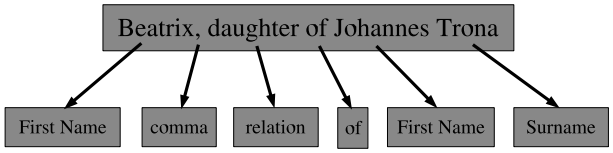

Figure 3: The corpus consists of 80331 records of the form shown in the upper line. We wish to parse each of the records (semi) automatically into a set of labelled terms such as shown in the bottom line.

archivists may also be uncertain as to the correct annotation of a record - for example 'Johannes de Saillon' might be interpreted as Johannes being the first name and 'de Saillon' being the surname. Alternatively, 'Saillon' may simply be the place where 'Johannes' was resident. Indeed, this ambiguity is not always resolvable since name conventions changed over time and often depended on the interpretation of the proffered name of the person by the creator of the document. Retaining the uncertainty of the annotation is useful since in future, the corpus may be searched for example for surname 'Saillon', and a hard assignment of 'Johannes de Saillon' to 'FIRSTNAME OF PLACE' may miss this record. A related issue is Record Linkage [11], whereby we wish to quickly and accurately identify records in our corpus which represent the same entity. In our case, we would like to identify which of our records represent the same person or set of people, when searching for family members. Previous work in record linkage has shown that identification of subcomponents of a record is vital to the accurate matching of records [23].

2. RELATED WORK

Our problem is one of *data cleaning*, which is the process of preparing unstructured information into a structured form for use in a data warehouse [20]. Our approach is to first tokenise the records into a sequence of terms. We can then find the relevant terms (names, relationships, professions, etc.) by estimating which terms are likely to correspond to the labels (surname, relationship, etc.) in which we are inter-

ested. Our problem is thus similar to that of Part-Of-Speech (POS) tagging [15] in natural language processing. The goal of part-of-speech tagging is to identify from a phrase (e.g. 'The cat sat on the mat') the set of tags describing each word grammatically (e.g. 'determiner noun verb preposition determiner noun'). In our case our tags are part-of-name-record tags, rather than part-of-speech tags. One approach to the problem of tagging parts of speech is to use a rule-based tagging system such as [7]. Rule based systems tag names by learning rules that describe the sequences of tags in a training set and matching them to new sequences. Unfortunately these are not generally stochastic systems, and thus are less useful for record linkage than a stochastic model would be.

A stochastic approach to POS tagging is the use of HMMs, as in [16]. HMMs are a probabilistic model which describe the sequence of tags and learn the probabilities of each tag following the previous tag. An HMM learns the probability of a noun following a verb, or of an adjective following a noun, etc. This is the approach taken in [9] to the problem of transforming names into a standard form for use in the **febrl** biomedical record linkage system. This is the problem of *name cleaning and standardization*, which is similar to our own, in that we wish to extract labels which represent name elements from an unstructured string. This work followed from [5] which used HMMs in a similar way to label postal addresses. It should be noted that our data contains much more 'noise' than typical name standardization data, in that the names in our records are collected over hundreds of years and follow forms of names found in multiple languages (Italian, Latin and medieval French, as well as modern French). In our task it is also important to extract relationship information from the records, for any subsequent use in genealogical analysis. The similar problem of identifying named entities (names, times, locations, organisations, etc.) occurring in documents was addressed using an HMM-based approach in [4]. The approach of [9] was to annotate by hand one hundred names used to initially train the HMM, then have the system guess at one thousand names, which were corrected by hand, and then have the system learn its parameters (probability distributions) from these names, and so on, until the parameters were considered good enough for a large corpus of names. We follow a similar approach, but train the HMM using a technique that allows us to use the structural information in all the corpus for training after only a few annotations have been made.

3. NAME PARSING USING HMMS

An HMM [19] is a probabilistic model which can be used to describe an ordered series of observations. In this case, our observations are the sequences of terms produced by a tokenisation of the record at word boundaries, transformed into lower case. The tokenisation of the name 'Perrodus de Salvan, clerc' produces the list of terms 'perrodus / de / salvan / , / clerc'. We assume that at each time-step t (corresponding to term t in the record) there is a hidden variable h_t which takes one of a small number of discrete values. In our case, h_t is the label (e.g. h_1 here is 'FIRSTNAME') associated with the observed term v_t (e.g. v_1 here is 'Perrodus').

Given the label i at time t, $h_t = i$, we define the probability of a transition to each possible label j at time $t+1$:

$$p(h_{t+1} = j | h_t = i) = A_{ji} \quad (1)$$

Similarly we define a distribution for the first label h_1:

$$p(h_1 = i) = \pi_i \quad (2)$$

Finally we define the probability of seeing a particular term given a particular label at time t:

$$p(v_t = i | h_t = j) = \beta_{ij} \quad (3)$$

In learning these parameters A, π and β, we are learning the structures of names. Given a good set of parameters, we can determine the most probable set of labels $h_{1:T}$ given a set of observations $v_{1:T}$, allowing our model to guess at the labels. We can use the Viterbi Algorithm [22] to efficiently calculate these guesses. Figure 4 shows a graphical representation of an HMM.

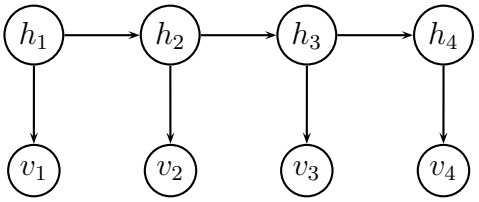

Figure 4: A Hidden Markov Model. Each variable v_t represents a term from the record (e.g. 'Johannes'), and h_t represents the label assigned to v_t (e.g. 'FIRSTNAME') where t indexes the term in a record. In training the model, we learn the transition probabilities $p(h_t|h_{t-1})$ and emission probabilities $p(v_t|h_t)$.

3.1 Training on all the data

Previous work on name standardisation using HMMs has involved training the model on a set of annotated names [9], and using a *bootstrapping* approach to annotating the data, wherin the user corrects the model's guesses on an increasingly large set of names and uses the new annotated data for further training. In this manner, the model is trained based always on a set of fully annotated data, and the parameters can therefore be learned by simply counting the frequencies of label-to-label transitions and label-to-term emissions. This is also true of previous work on record segmentation [5].

Our alternative approach takes advantage of the simple idea that if a term is annotated with a label in one name, it is highly likely that this term should be annotated with the same label everywhere in the data. We can annotate this label everywhere in the corpus, assuming that the annotations are the same unless explicitly told otherwise (by further annotations). In this way, we will use *all* the data in the corpus, containing largely only partially annotated records. To learn the parameters (the transition and emission probability tables) we use the Expectation Maximisation (EM) algorithm [10][3], which is an iterative algorithm that increases the likelihood of the corpus given the model parameters at each iteration. Using EM allows us to train our model on a much smaller amount of fully annotated data, but be able

[3]In the context of HMMs, this is equivalent to the Baum-Welch learning algorithm [3]

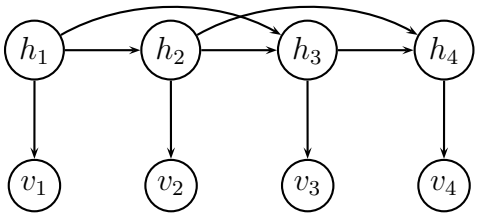

Figure 5: A Second-Order Hidden Markov Model. Each variable v_t represents a term from the record, and every variable h_t represents the label assigned to v_t. Now every label h_t is not only modelled to be dependant upon its neighbours h_{t-1} and h_{t+1}, but also on their neighbours too.

to capture structures found in the entire data set. Since we train on our entire data set, the difficulty of previously non-observed terms [9], [5] is much reduced, and we did not find it necessary to perform parameter smoothing.

3.2 Online Learning

Our approach iteratively learns the model parameters and then asks the user to correct a sample of guesses, repeating the following steps:

1. Train the model (using EM) on the current partially annotated corpus.
2. Present a number of records that have not yet been fully annotated by the user.
3. Use the Viterbi algorithm to produce the best-guess annotation for each of the presented records.
4. The user corrects any errors made by the Viterbi labelling of the presented records.

This is similar to the *bootstrapping* approach of [9], but differs in several ways. Firstly, we can start with a very small number of names (e.g. 5), rather than 100. The annotations for these names should be enough to improve the guesses for the next set of guesses. Secondly, we are always training our model on all of the data rather than just the fully annotated subset, and so we capture structural properties contained in the unannotated records.

3.3 Higher order HMMs

Consider the following two records:
'John, Robert' and 'John, brother of Robert'

In both cases, if the model has never before seen the term 'John', but it has seen the symbol ',', then the most likely state for the symbol 'John' will be the same, regardless of the rest of the name. This is due to the Markov property governing the hidden variables of the HMM. The variable h_1 is independent of the rest of the variables when v_1 and h_2 are known or observed. In general, the distribution $p(h_t = j|h_{t-1} = i)$ means that h_t is independent of h_{t-2} given h_{t-1}.

One straightforward enhancement to the model is to use a higher order HMM. In an order-k HMM, we encode a direct dependence between the hidden variables within k timesteps of each other, replacing the distribution $p(h_t|h_{t-1})$ with the distribution $p(h_t|h_{t-1}, h_{t-2}, ..., h_{t-k})$. A second order HMM is shown graphically in figure 5. A 2^{nd} order HMM was used for part-of-speech tagging in [6].

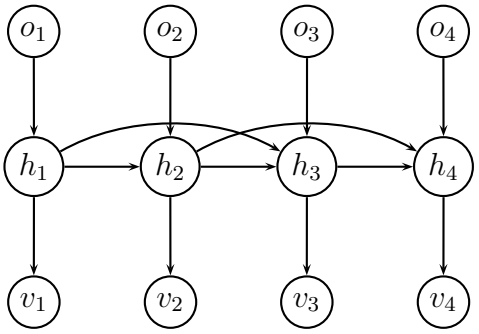

Figure 6: A Second-Order Hidden Markov Model with Virtual Evidence. Each variable v_t represents a term from the record, and every variable h_t represents the label assigned to v_t. In this model, every possible observation that could be made about h_t is modelled as a state of O_t. Each hidden variable h_t is now conditioned on the 'soft observation' O_t.

3.4 Virtual and Negative Evidence

So far we have discussed observations of a term where we know exactly what state the h_t variable is in given v_t. For example, if the term 'Johannes' is annotated by the user as a 'FIRSTNAME', then we set $p(h_t = `FIRSTNAME'|v_t = `Johannes') = 1$. We can call this type of observation *hard evidence* on the variable h_t. However, sometimes we cannot say for sure which label should be associated with a term, but we can say for certain that some labels definitely do not correspond to the term. Consider the record:
'Johannes de Saillon' ('Johannes of Saillon')

Here, without ever having seen the term 'Saillon' before, we cannot say for sure whether it is a surname or a place. We can however say with certainty that it is not a comma. We would like a general framework to encode this type of observation into our model, such that a user could specify rules e.g. *Unless otherwise specified, a term cannot be labelled as de/di unless it starts with a 'd'.* Similarly, if the same term is annotated by a user with two different labels in two different record, we would like to be able to apply this as an observation when we encounter the term in other records. We do this by representing our observation of a variable in terms of a distribution over hidden states. A normal observation is where we know exactly the state that a hidden variable takes. Here we can observe that variable h_t is equally likely to be in any state *except* a certain state, or that it is equally likely to be in one of two states, but is definitely not in any other state.

There are several frameworks for modelling this kind of uncertain observation [8]. Here we use the framework of Virtual Evidence [18], where an uncertain observation is recast as a certain observation on another auxiliary variable. This results in a minor modification to the standard HMM framework, which we present briefly here for a 1^{st} order HMM. For a more detailed explanation of this Virtual Evidence approach, see [8].

Consider that we have at each timestep an observation variable O_t, which can take one of K values, each value representing a possible 'soft observation' of the variable h_t. A soft observation of the variable h_t may be that the variable is in a specific state, that it is in one of a collection of states,

Table 3: Evaluation Records

Number of records	500
Number of terms	3483
Number of unique terms	716
Mean record length	6.966 terms

or that it is definately *not* in a specified state. We can now redefine the distributions of each h_t variable as follows:

$$p(h_1 = i) = p(h_1 = i|O_t = k)p(O_t = k) \quad (4)$$

$$p(h_t = j|h_{t-1} = i) = p(h_t = j|h_{t-1} = i, O_t = k)p(O_t = k) \quad (5)$$

We wish that each observation O_t influences the state of h_t based on the observation, so we define a distribution ev_k that represents this evidence.

$$p(h_t = j|h_{t-1} = i, O_t = k) \propto A_{ji}ev_k(j) \quad (6)$$

When we observe a term that we know cannot be labelled with a certain label l, we associate a state k' of O_t with an evidence distribution that assigns equal prior probability $ev_{k'}(j)$ to every state j of h_t except the state corresponding to label l. Thus we have $ev_{k'}(l) = 0$ and $p(h_t = l|O_t = k') = 0$. O is always observed, and thus $p(O_t)$ can be counted directly from the data. Thus

$$p(h_t = j|h_{t-1} = i)^{new} \propto p(h_t = j|h_{t-1} = i)^{old}ev_t(j) \quad (7)$$

The resulting model is shown graphically in figure 6.

4. EVALUATION

To evaluate our work we have fully hand-annotated a sample of 500 records selected uniformly at random from the full 80331 record corpus. We shall refer to this set of annotations as our ground-truth annotations. Table 3 describes the characteristics of the records we annotated.

Using our ground-truth annotations, we wish to simulate the real-world use of this system to evaluate the different features of our system. We present a framework which simulates the task of annotating a dataset of records, 5 records at a time where, at each iteration, the user corrects guesses made by the current model. For the purposes of this current evaluation, we do not consider user labelling errors. To evaluate our system we repeatedly iterate through the following steps:

1. Train the model on the current partially annotated corpus.
2. Present 5 records that have not yet been fully annotated by the user, randomly from the dataset.
3. Use the Viterbi algorithm [19] to produce the best-guess annotation for each of the 5 records.
4. Compare the Viterbi labelling to the ground-truth annotations.
5. Record any errors.
6. Annotate the 5 records using the ground-truth data.

We iterate through these steps until every record in our dataset is fully annotated. This corresponds to a user being repeatedly presented with 5 records and their labels as guessed by the model, and being asked to correct the guesses. Using this approach we can directly measure how many terms a user has to annotate before the model can guess the annotations of every record with a very high accuracy.

4.1 Measurements

As each batch of 5 records is annotated we measure:

1. The percentage of labels guessed correctly from the presented 5 records.
2. How many errors the current model makes on annotating the whole dataset, compared to the ground truth.

After the entire set of records has been correctly annotated by the procedure of the previous section, we measure also the total number of corrections a user would have had to have made to obtain a perfect annotation of the records.

It is useful to know how much work the user would have to do if all our system did was record terms currently annotated by the user – for example 'Johannes' is a 'FIRSTNAME', etc. and we use these in all subsequent records so that where 'Johannes' appears, we assume that this is in fact a 'FIRSTNAME'. This will serve as our baseline result. In our baseline we assume that all unseen terms are incorrectly guessed. In the future we would like to compare our labelling method to other non-HMM methods such as a rule-based part-of-speech tagger.

4.2 Training on all the data

We evaluated our approach of training the HMM using EM against training based on frequency counting of the annotated data (where only fully annotated records are used to train the model, as used in [9] and [5]), on datasets of varying sizes (between 100 and 500 records). For each dataset size, we randomly sampled 50 subsets of the given size from our 500 annotated records[4], and ran our evaluation task on a second-order model being trained by EM, and also a second-order model trained using the simple frequency counting method. We calculated the cumulative amount of corrections made by the user to fully annotate every record correctly, using each model. In table 4 we present our results.

From the results in table 4 we can see that as the sample size increases, both models begin to dramatically outperform the baseline performance. It can also be seen that as the sample size increases, the performance of the EM-trained model increases relative to the performance of the model trained with frequency counting, ranging from 13% better with 100 samples, up to 18% better with 500 samples.

4.3 1^{st} vs 2^{nd} Order Models

We compared the performance of the first and second order models over 50 randomly selected subsets of 100 records. The percentage of incorrectly labelled terms from the 5 records presented to the user at each iteration of the learning procedure is presented in Figure 7, along with the number of

[4]For a sample size of 500 records, each 'subset' consists of the same 500 records in a different order. The variance in the results is caused by the random order in which the records are presented to the user.

Table 4: EM vs Frequency Counting. We use 50 random subsamples of varying sizes, from our 500 annotated records. For each subsample, we run our evaluation task using a second-order model trained with EM and another second-order model trained with frequency counting, and record the total number of corrections made by the user. Here we present the mean number of corrections and standard deviation for each model and sample size. Below, we also show the ratio of the number of corrections to the sample size, to show the scaling properties of each model.

Sample Size	100		200		300		400		500	
Baseline	205.4	± 9.2	332.6	± 9.4	439.0	± 9.0	529.2	± 8.4	614.6	± 4.0
Counting	97.5	± 9.7	141.7	± 11.3	175.9	± 10.5	206.8	± 10.5	230.2	± 10.6
EM	85.3	± 7.7	119.6	± 8.5	146.1	± 10.0	168.8	± 8.6	189.5	± 10.4
Baseline / sample size	2.05	± 0.09	1.66	± 0.05	1.46	± 0.03	1.32	± 0.02	1.23	± 0.01
Counting / sample size	0.98	± 0.08	0.71	± 0.06	0.59	± 0.04	0.52	± 0.03	0.46	± 0.02
EM / sample size	0.85	± 0.08	0.60	± 0.04	0.49	± 0.03	0.42	± 0.02	0.38	± 0.02

incorrectly predicted terms from all 100 records after each iteration. The left hand side of figure 7 clearly shows that the second order model is making better guesses at the labelling of names presented to the user, and the right hand side shows that the second order model is modelling the whole dataset better. The curvature of the baseline results can be explained by the fact that as more annotations are added, the frequency of previously annotated terms (which the baseline is assumed to remember) in each new record increases. Moreover, the most frequent terms will tend to be annotated early in the process, so the first few iterations using even the baseline method show a steep improvement.

For each subset of 100 records, after all 100 records had been correctly annotated by the evaluation procedure, we calculated the cumulative amount of corrections made by the user. These are: Baseline (mean 204, standard deviation 9), Order 1 (mean 131.7, standard deviation 9.5), Order 2 (mean 86.5, standard deviation 8.2) corrections, showing that the 2nd order HMM clearly outperforms the baseline and first order HMM.

4.4 Virtual Evidence

In our evaluation of virtual evidence, we used seven observation rules. Six of these rules were of the form: 'Unless otherwise stated, this label is not COMMA unless the term contains a comma symbol'. A rule like this was created for the states corresponding to the terms ',' , 'et' , '(' , ')' , '-' and '.'. The last rule was: 'Unless otherwise stated, this label is not DE/DI unless the term starts with the letter d'. To evaluate virtual evidence, we used a second-order HMM, both with and without the use of virtual evidence. We compared the performance over 50 randomly selected subsets of 100 records. Figure 8 shows the relative performance at each iteration of the test.

For each model and sample, after all 100 records had been correctly annotated by the evaluation procedure, we calculated the amount of corrections made by the user. With Virtual Evidence, the mean number of corrections is 72 (Standard deviation 8.1). Without Virtual Evidence, the mean number of corrections is 87 (Standard deviation 7.8). For each sample, we plot the results of both models against each other in figure 8, where it can be seen that the model using Virtual Evidence outperforms the model which does not use Virtual Evidence.

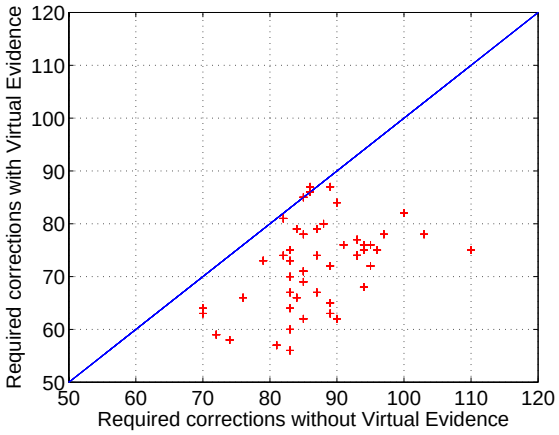

Figure 8: A second order HMM trained with Virtual Evidence is compared to one trained without Virtual Evidence. Each point represents one uniformly sampled subset of 100 records taken from the corpus. For each subset, both models are evaluated, and the number of label corrections needed to fully annotate the set using each model is recorded.

5. THE BROWSING TOOL

With labelled records, we can start to explore our data further. We have built a prototype browser, shown in figure 10. This browser allows a user to instantly see and navigate between people and their relations, and displays information associated with a selected name at a glance. When a user selects a name in the browser, a summary of information regarding the name (known immediate relatives) is displayed, along with the summaries for any documents containing the name. The browser visualisation has been built using the prefuse [14] toolkit.

We use our name record model in this browser for two purposes:

1. to display a name in canonical form ('FIRSTNAME SURNAME', e.g. 'Johannes Trona')

2. to identify possible relationships between people.

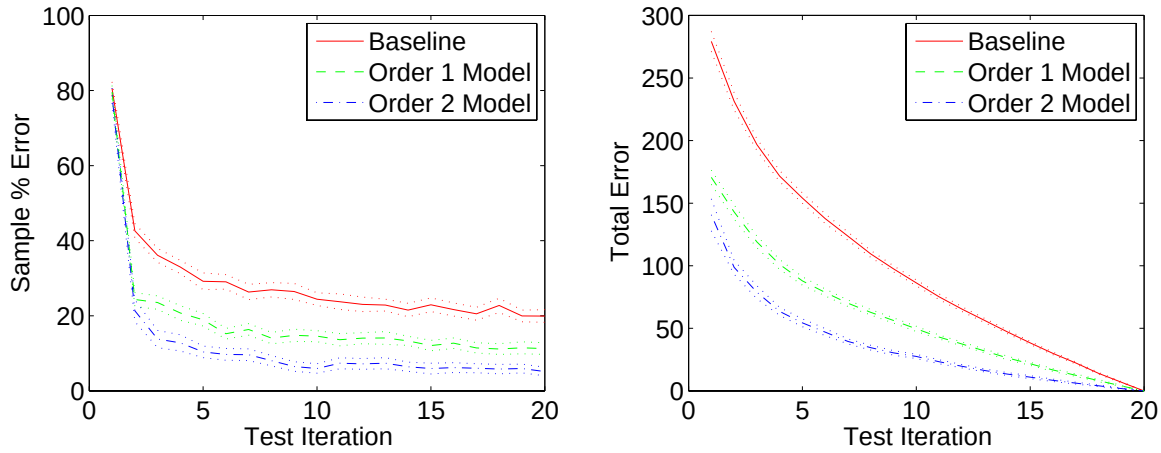

Figure 7: First and second order model results. Both models tested on names selected in random order from a uniformly sampled subset of 100 records. Mean results are shown, along with the standard error of the mean shown as a dotted boundary. Left: shows the error on the 5 records presented at each iteration, measured as a percentage of the number of terms to be labelled across the 5 records. Right: shows how well the model can guess the labelling for the whole corpus after each iteration. The error is a simple sum of incorrectly guessed labels.

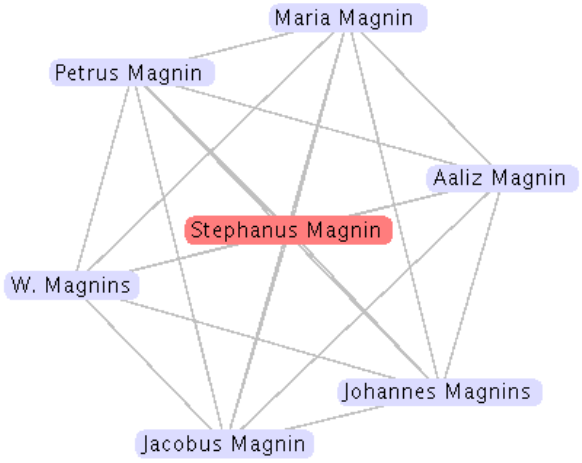

Figure 9: A screenshot from the prototype browsing tool. With Surnames in records identified, we can find people from the same time period and/or the same place with similar Surnames. Here we have found people who might be related to 'Stephanus Magnin', so the browser displays people with the Surnames 'Magnin' or 'Magnins'.

Most of the name records only contain a single first name and surname, so converting these records into a canonical form is a straightforward process. For records with two first names and a single surname (e.g. 'Adeleta, wife of Stephanus Magnin'), the surname is usually common to both people, but this is not certain. By taking into account the relationship in the record we can be more certain of whether the two people should share the surname (e.g. in 'Adeleta, wife of Stephanus Magnin', the relationship 'wife' should imply a common surname, but in 'Adeleta, niece of Johannes Trona', the relationship 'niece' makes it less likely that the surnames should agree.

With the records parsed, we can start to use the names within the records to identify possible family relations. We have explicitly labelled relationships mentioned within the records themselves(e.g. 'Adeleta, niece of Johannes Trona' specifies a relationship between Adeleta and Johannes Trona), but we also wish to identify possible relations between multiple documents (e.g. 'Adeleta Trona' and 'Trona, Johannes' may be members of the same family if they are mentioned in the same time period and in the same region). We also wish to identify where two names may refer to the same person (e.g. 'Trona, J.' and 'Johannes Trona'). Our model tells us which terms in each record refer to surnames, so we can find people with identical or similar[5] surnames, as shown in figure 9. We have currently implemented a very simple surname matcher which simply looks for lexically similar surnames. We intend to study more sophisticated matching techniques. We intend to evaluate phonetic methods, which we will attempt to adapt to handle the cross-lingual nature of this data (containing names from Italy, France and Switzerland). This work will be challenging as we do not know which language should be used to pronounce each name.

One particular shortcoming of our approach was found in the inability of the model to distinguish between male and female names. Consider the record 'Petrus, wife of Maria, daughter of Stephanus Magnin'. In this case, the relationship between Petrus and Stephanus is aambiguousto our model, as Petrus could equally be the 'wife of the daughter of', or simply the 'daughter' of Stephanus. Without knowledge of name genders one cannot make this distinc-

[5]Family names commonly morph over long periods of time, especially in days when literacy was not common and names were only known phonetically.

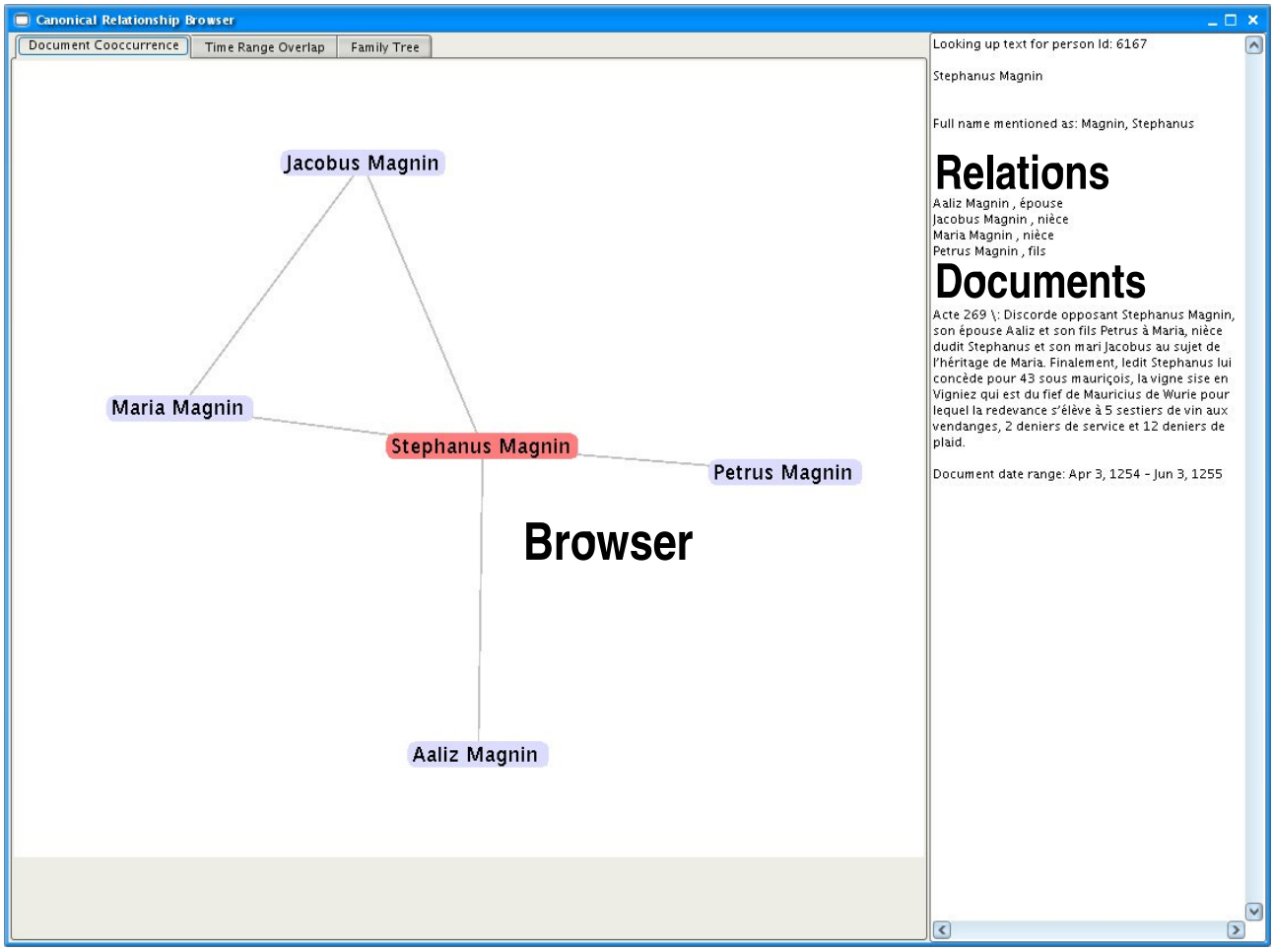

Figure 10: A screenshot of the prototype browsing tool. Names and relationships can be viewed and selected graphically. When a user selects a name, relevant information is loaded into the text pane on the right side of the tool, including labelled relationships (e.g. mother, sister, etc.) and a text summary of every document containing the name.

tion. Fortunately, this issue can be addressed by using a more sophisticated set of labels (e.g. having 'MALEFIRSTNAME' and 'FEMALEFIRSTNAME' instead of simply 'FIRSTNAME').

6. CONCLUSIONS

In this paper we presented initial work in a project which aims to make genealogical information taken from historical archives as easy to search as websites on the internet. Our prototype system already makes finding information on certain people mentioned in the documents extremely easy, and we foresee this tool being very useful for many genealogical researchers in the near future. We presented a method for annotating the short name records taken from summaries of our documents. We followed a method developed for name cleaning and standardization, using Hidden Markov Models. In contrast to previous studies, we used a method of training which could use all of the records, annotated and unannotated, to learn record structures. Finally, we designed a method for incorporating rules governing the labelling of terms which allow a user to be more expressive in annotation.

We showed that using a higher-order model dramatically reduced the required user effort. Training with EM was seen to scale well with the number of records to be annotated, and was shown to outperform the simple frequency-counting method, especially as the number of records to be annotated was increased (and thus the relative size of the unannotated data to the annotated data increased). In larger corpora, the larger relative size of unannotated to annotated data, along with the smaller number of terms relative to the number of records, gives us hope that this scaling property will continue to hold as the corpus size is increased into the tens of thousands. In future work we hope to evaluate our methods over a much larger set of records.

We showed that by using rules encoded with Virtual Evidence, the performance of our model was increased. This framework for incorporating known syntactic rules into the HMM model is promising, as it could easily be extended to incorporate more sophisticated rules. In future work we intend to evaluate the performance of this method of using virtual evidence on a more general task such as part-of-speech tagging, rather than in our current task-specific evaluation framework presented in this work.

We intend to continue developing our browsing tool to allow genealogists to efficiently extract useful information from this data. Future work on the tool will involve the visualisation of probabilistic linkage information (e.g. closer or thicker links between names to indicate more probable relations), giving users the ability to enter information back into the tool easily as relationships are found and verified, and more sophisticated identification of likely surname morphing. We also intend to develop methods which are robust to errors in user labelling.

7. ACKNOWLEDGEMENTS

The authors acknowledge financial support provided by the Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM)2. The NCCR is managed by the Swiss National Science Foundation on behalf of the Federal Authorities.

We would also like to acknowledge the hard work and dedication of the Fondation des Archives Historiques de l'Abbaye de Saint-Maurice [2], who have been digitising and summarising the data collected at the abbaye, and other nearby abbayes, for over two years.

8. REFERENCES

[1] Abbeye de saint-maurice. `http://www.abbaye-stmaurice.ch/`.

[2] Fondation des archives historiques de l'abbaye de saint-maurice. `http://www.aasm.ch/`.

[3] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

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

[5] Vinayak Borkar, Kaustubh Deshmukh, and Sunita Sarawagi. Automatic segmentation of text into structured records. In *SIGMOD '01: Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 175–186, New York, NY, USA, 2001. ACM Press.

[6] Thorsten Brants. Tnt – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference*, Seattle, WA., 2000.

[7] Eric Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento, IT, 1992.

[8] Hei Chan and Adnan Darwiche. On the revision of probabilistic beliefs using uncertain evidence. *Artif. Intell.*, 163(1):67–90, 2005.

[9] T. Churches, P. Christen, K. Lim, and J. Zhu. Preparation of name and address data for record linkage using Hidden Markov Models. *BMC Medical Informatics and Decision Making*, 2, 2002.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[11] Lifang Gu, Rohan Baxter, Deanne Vickers, and Chris Rainsford. Record linkage: Current practice and future directions. Technical Report 03/83, CSIRO Mathematical and Information Sciences, April 2003.

[12] Hui Han, Lee Giles, Hongyuan Zha, Cheng Li, and Kostas Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 296–305, New York, NY, USA, 2004. ACM Press.

[13] Hui Han, Hongyuan Zha, and C. Lee Giles. Name disambiguation in author citations using a k-way spectral clustering method. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 334–343, New York, NY, USA, 2005. ACM Press.

[14] Jeffrey Heer, Stuart K. Card, and James A. Landay. prefuse: a toolkit for interactive information visualization. In *CHI '05: Proceeding of the SIGCHI conference on Human factors in computing systems*, pages 421–430, New York, NY, USA, 2005. ACM Press.

[15] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, chapter 8. Prentice-Hall, 2000.

[16] Julian Kupiec. Robust part-of-speech tagging using a Hidden Markov Model. *Computer Speech and Language*, 6:225–242, 1992.

[17] Bradley Malin, Edoardo Airoldi, and Kathleen M. Carley. A network analysis model for disambiguation of names in lists. *Comput. Math. Organ. Theory*, 11(2):119–139, 2005.

[18] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[19] Lawrence R. Rabiner. *A tutorial on Hidden Markov Models and selected applications in speech recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[20] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.

[21] M.D. Taffet. Looking ahead to person resolution. In *Proceedings of 4th Annual Workshop on Technology for Family History and Genealogical Research*, pages 11–15, Provo, Utah, 2004.

[22] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. on Info. Theory*, IT-13:260–269, 1967.

[23] W. Winkler. Matching and record linkage. *Business Survey Methods*, pages 355–384, 1995.