IDIAP RESEARCH REPORT

Probabilistic Tagging of Unstructured Genealogical Records

Mike Perrow [a]   David Barber [a]

IDIAP–RR 05-86

December 2005

submitted for publication

[a] IDIAP Research Institute, 1920 Martigny,Switzerland

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 05-86

Probabilistic Tagging of Unstructured Genealogical Records

Mike Perrow David Barber

December 2005

submitted for publication

Abstract. In this paper we present a method of parsing unstructured textual records briefly describing a person and their direct relatives. The string 'Stephanus, brother of Johannes Magnin, from 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. 'Stephanus'), a place (e.g. 'Saillon'), or a relationship (e.g. 'brother'). 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.

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'

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

Label	Description
First Name	First Name
Surname	A Surname
de / di	de, di, dou, du, etc.
comma	a comma
hyphen	The hyphen symbol '-'
Place	A place name
relation	mother, daughter, son, etc.
role	A profession or role within the community e.g. doctor, cantor
called	Used for aliases e.g. Anna *called Ave*
period	The period symbol '.'
(	Open brackets e.g. ({
)	Close brackets
and	the word 'et' and equivalent words
other	Any miscellaneous term

1 Introduction

Our corpus consists of a list of strings, each containing a brief record of a person and their relationship to other people. These records have been extracted from historical documents dating back as far as the 13^{th} century from the Abbeye de St. Maurice in Switzerland [1] and are currently being digitised by the Fondation des Archives Historiques de l'Abbaye de Saint-Maurice[2]. Each record is a summary of a person mentioned in a particular historical text and, amongst other things, their relationship with other people mentioned in the document. Typically, the list of people mentioned in a document will include several members from the same family. We wish to parse these records to allow easy browsing and comparison of the various names contained in the records, allowing this data to be used for genealogical analysis. It is our goal 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. 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. Table 1 shows a selection of the 80331 records along with their English translations. We wish to parse these records into a sequence of labels describing the various attributes of the person, such as their surname or profession. Figure 1 shows a typical record. Our goal is to produce from this record the sequence of labeled terms contained in the record. Once parsed, records may then be used for searching, matching, and can be presented in a standard form (e.g. the name in standard form is 'Beatrix Trona'). Table 2 shows the selection of labels which we wish to assign to each term in the record string[1]. 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 1 we

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

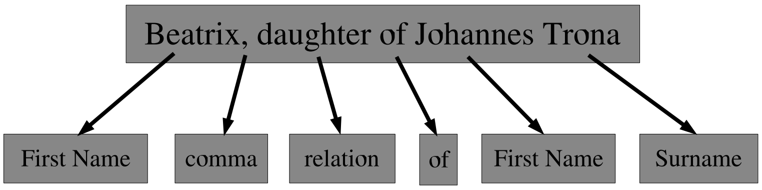

Figure 1: The corpus consists of 80331 records of the form shown in the upper line. We wish to parse each of the records (semi) automatically into a set of labeled terms such as shown in the bottom line.

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.

We build on an approach previously developed for cleaning name and address data[7], 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. 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 [6]. This has the advantage of combining rule-based and probabilistic approaches within the same model. Indeed, the professional 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 [9], 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 [15].

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 [14]. 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 interested. Our problem is thus similar to that of Part-Of-Speech (POS) tagging [10] 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 speech is to use a rule-based system such as [5]. 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 [11]. 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 [7] 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 [4] 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 approach of [7] 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 [13] 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. Tokenising the name 'Perrodus de Salvan, clerc' produces the list '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. 'SURNAME') associated with observation v_t (e.g. 'Magnin').

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

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

Similarly we define a distribution for the first label h_1:

$$p(h_1 = i) = \pi_i \tag{2}$$

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

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

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 to efficiently calculate these guesses. Figure 2 shows a graphical representation of an HMM.

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 [7], 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

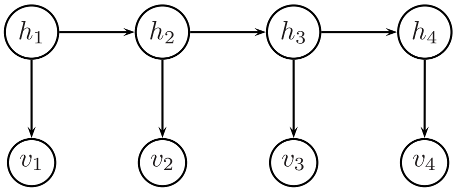

Figure 2: A Hidden Markov Model. Each variable v_t represents a term from the record (e.g. 'Johannes'), and h_t represents the label assigned to v_t (e.g. 'FIRSTNAME') where t indexes the term in a record. In training the model, we learn the transition probabilities $p(h_t|h_{t-1})$ and emission probabilities $p(v_t|h_t)$.

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 [4].

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 [8][2], 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 to capture structures found in the entire data set. Since we train on our entire data set, the difficulty of previously non-observed terms [7], [4] 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 [7], 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.

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

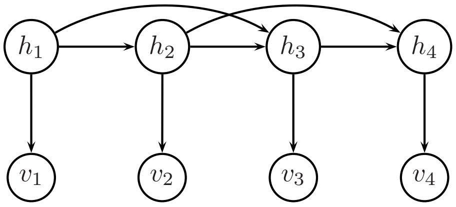

Figure 3: A Second-Order Hidden Markov Model. Each variable v_t represents a term from the record, and every variable h_t represents the label assigned to v_t. Now every label h_t is not only modelled to be dependant upon its neighbours h_{t-1} and h_{t+1}, but also on their neighbours too.

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, the distribution $p(h_t|h_{t-1})$ is replaced with the distribution $p(h_t|h_{t-1}, h_{t-2}, ..., h_{t-k})$, which encodes a direct dependence between the hidden variables within k timesteps of each other. A second order HMM is shown graphically in figure 3.

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 $p(h_t = \text{`}FIRSTNAME' | v_t = \text{`}Johannes') = 1$. 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 following 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 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 [6]. Here we use the framework of Virtual Evidence [12], 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.

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. To evaluate our system we repeatedly iterate through the following steps:

Table 3: Evaluation Records

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

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[13] to produce the best-guess annotation for each of the 5 records.
4. Compare the Viterbi labellings 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 new batch of 5 records is annotated in our test 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.

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 [7] and [4]), on datasets of varying sizes. For each dataset size, we randomly sampled 50 subsets of the given size from our 500 annotated records, 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 the results of this evaluation. 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.

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 and standard deviation for each model and sample size.

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

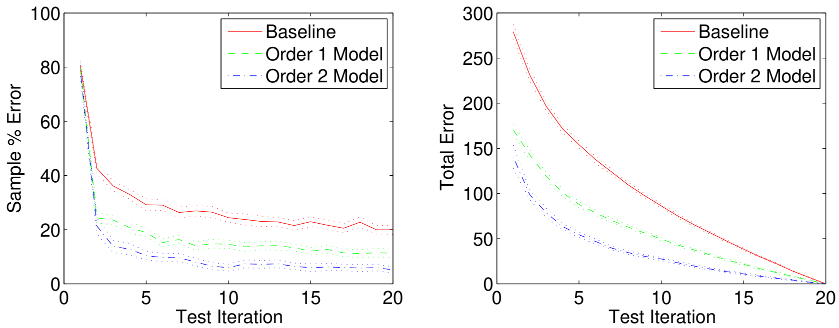

Figure 4: First and second order model results. Both models tested on names selected in random order from a uniformly sampled subset of 100 records. Mean results are shown, along with the standard error of the mean shown as a dotted boundary. Left: shows the error on the 5 records presented at each iteration, 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.

4.3 1st vs 2nd 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 4, along with the number of incorrectly predicted terms from all 100 records after each iteration. The left hand side of figure 4 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.

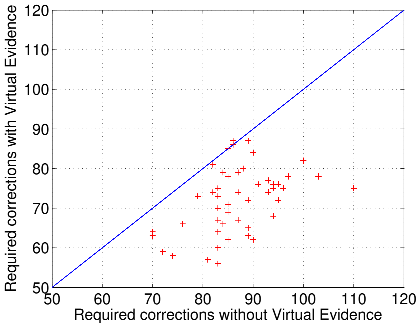

Figure 5: A second order HMM trained with Virtual Evidence is compared to one trained without Virtual Evidence. Each point represents one uniformly sampled subset of 100 records taken from the corpus. For each subset, both models are evaluated, and the number of label corrections needed to fully annotate the set using each model is recorded.

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 5 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 5, where it can be seen that the model using Virtual Evidence outperforms the model which does not use Virtual Evidence.

5 Conclusions

In this paper we presented a method for annotating a set of short genealogical records created by summarising historical documents. We followed a method developed for name cleaning and standardization, using Hidden Markov Models. In contrast to previous studies, we created a more expressive second-order model, and 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 labelling which allow a user to be more expressive in annotating a data set. 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). 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.

6 Acknowledgements

We would 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 abbey for over two years.

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] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

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

[13] 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.

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

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