

Reviewer's report

Title: Preparation of name and address data for record linkage using hidden Markov models

Authors:

Dr Tim Churches (tchur@doh.health.nsw.gov.au)
Dr Peter Christen (peter.christen@anu.edu.au)
Kim Lim (klim@doh.health.nsw.gov.au)
Justin X Zhu (u3167614@student.anu.edu.au)

Version: 1 **Date:** 3 Dec 2002

Reviewer: Dr William E. Yancey

Level of interest: A paper whose findings are important to those with closely related research interests

Advice on publication: Accept after discretionary revisions

In the paper "Preparation of Name and Address Data for Record Linkage Using Hidden Markov Models," the authors Churches et al. describe the standardization procedure based on hidden Markov models that is performed by their software Febrl. The hidden Markov approach is distinguished from the rule-based approach to standardization. They also wish to distinguish their approach from the hidden Markov method described in "Automatic Segmentation of Text into Structured Records" by Borkar et al. The advantage of their hidden Markov approach using their software package is that it is purported to be more easily implemented than rule-based methods that require special programming by skilled staff. They apply their procedure to standardize addresses and names from some data files, and they report the surprising result that they were more successful standardizing addresses than names. Since on the surface it would appear that names are easier to parse than addresses, it is interesting to try to follow their procedure to speculate on the reasons for these findings.

Tokenization

A distinguishing feature of the authors' method is a preprocessing of the component strings into tokens or observation symbols. This amounts to a kind of preliminary parsing without using the Markov structure by identifying the part of the address or name of each string without considering preceding string classifications (although it can group strings using a forward-looking greedy algorithm). These tokens will form the set of possible symbol emissions for each state of the Markov model. This tokenization is said to distinguish this method from that of Borkar et al. which is said to use each unique word as a symbol. While this would soon make the calculation of an observation probability array infeasible, in fact Borkar et al. (pp.180--181) show that they usually replace classes of strings with a few symbols as well. By contrast, the symbols in Churches et al. have a more semantic character rather than a rational expression string grouping. The Febrl software evidently saves the user from having to determine his/her own set of symbols. On the other hand, it is not clear if the user can modify the set in Table 3.

In order to assign these observation symbols to strings, the Febrl algorithm uses a few rules and look-up tables. For the wayfare (street) names and locality names, it indicates that one can use pretty comprehensive postal or government lists covering the geographic area of the data set under

examination. On the other hand, it is not indicated what sort of lists are used to assign tokens for female given names, male given names, or surnames, or how to select among these if the same name appears in more than one list. This could be one source of the different performance for address parsing and name parsing.

Moreover, this may not be the most useful choice for name observation symbols. Perhaps the authors are being too hard on themselves when computing name standardization accuracy. They indicate that about 9% of the names were of the form either "givenname givenname surname" or "givenname surname surname." Evidently a standardization is deemed incorrect if a name in one of these forms is parsed to a name in the other form. But it is not clear to me how even the humans parsing the list of names for the training data values can make this distinction. If a woman has a middle name such as "Leslie" or "Ashley", is that a given name, or is it her maiden surname, or is it her mother's maiden surname that has been given to her? Is there a rationale for trying to make the distinction? With some names, like "Spangler Arlington Brugh," presumably none of the names will be found in a list of given names. The only information available to parse the name is the word order, in which case one would be more likely parse it into the pattern "firstname middlename lastname".

Modeling

It is indicated that the hidden Markov models are trained using a bootstrap technique. However, before the models can be trained to estimate the state transition probabilities and observation probabilities, one first has to specify a model. That is, one has to choose the number (and meaning) of states and what topology connects them. Is this done from scratch or does the software provide a generic master model. Is the model modified, perhaps in some iterative fashion, to add or delete states or transition path? Certainly choosing a good model is fundamental for successful parsing. We are given a "simplified model" for address parsing, which helps us understand and perhaps appraise the methodology, but not one for name parsing.

Having an idea of the name parsing model could help to understand how it handles different situations. While this may not have much to do with the final name standardizer model, Table 2 indicates just two fields: given names and surnames. Does this indicate that a given person can have more than one of each? Does one get more than one surname simply by having more than one name appear in the look-up table? There are some punctuation tokens in Table 3, but there is no indication how hyphenated names are handled. Do we get one or two (or more) surnames with "Reginald Alfred John Truscott-Jones"? What happens when the hyphen or either half is omitted? There is a token for one letter words (initials) but is there any distinction given to their position in the name word order? What happens with "I.A.L. Diamond" or "L. Frank Baum" or "C. Wright Mills" or "H.P.F. Swinnerton-Dyer" or conceivably "H. Peter F. Swinnerton-Dyer"? It is not clear what is meant by the token for name prefixes, but last names can contain space delimiters. Names like "de la Renta", "De Mille", "La Guardia", "Te Kanewa" should probably ultimately be classified as one name, while it may be less clear with "von Stoheim" or "Von Neumann". Sometimes these same names can appear without spaces, as in "Van Dyke" or "Vandyke", "van der Jagt" or "Vanderbilt".

Training

The paper describes the procedures for training the address and name standardization models, but it does not indicate why different methods were used for each. The address standardization is carried out by an iterative refinement: a small training set is used to determine initial model parameters which are used on a larger set whose misclassified elements are fed back to refine the model. Eventually records from another data set are used for further refinement. In general, a smaller amount of data is used to train a larger set of validation data. For the name standardization, a ten part cross validation procedure is used, where repeatedly 90% of the data is used to train the model to standardize the remaining 10%

of the data for validation.

Typographical Errors

The paper has two typographical errors:

1. On p.13 for the arithmetic expression for the HMM probability, the first factor should be 0.08, not 0.8.
2. On p.31, in Table 1, in record number 4, some of the street address has crept into the Locality column.

Conclusion

This review may seem like a litany of complaints, but they are mostly generated by a desire to understand better the methodology described in the paper. Except for the typos, the rest of the comments may be taken as implicit discretionary revision suggestions. If some comments indicate that I have not understood the paper at some points, perhaps the text can be revised for clarity in these areas. This is an interesting, innovative, and possibly very fruitful approach to the problem of standardization, and it merits communication and consideration.

Reference

V. Borkor, K. Deshmukh, S. Sarawagi. "Automatic Segmentation of Text into Structured Records". Electronic Proceedings of ACM SIGMOD Conference 2001: Santa Barbara, California, USA. New York, Association for Computing Machinery, 2001.

Competing interests:

None declared.