Entity Resolution in Unstructured Data

and applications in the analysis of historical documents

Benjamin van der Burgh

March 15th 2016

Supervisors: Dr. Arno Knobbe
Dr. Siegfried Nijssen

Overview

1. Goals of the Traces Through Time project
2. Format problem description
3. Record extraction
4. Comparison of record fields
5. Candidate pair classification
6. Maximally k-informative itemsets
7. Experiments
8. Conclusions and future work

Universiteit Leiden

Traces Through Time (1) – Context

- The National Archives stores millions of documents.
- Many documents have been converted to a digital format.
  - Automatic: Optical Character Recognition (OCR).
  - Manual: transcribed by hand.
- Connecting pieces of information regarding people is mostly done manually.
- Automating this process allows for studying people in all layers of society, not just the aristocracy.

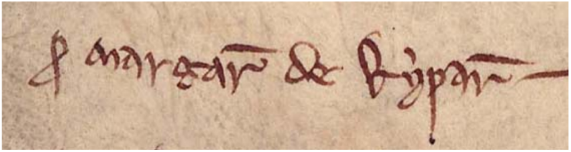
“ *No matter what he does, every person on earth plays a central role in the history of the world. And normally he doesn’t know it.* ”

Paulo Coelho (The Alchemist)

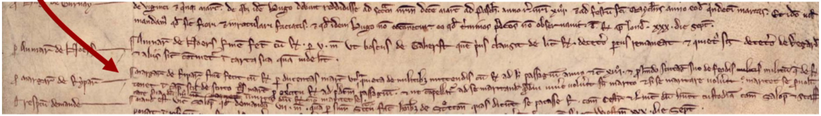
Traces Through Time (2) – Goals

- Develop a methodology to identify and trace individuals across large and diverse historical datasets.
- Look particularly at 'fuzzy' data
    - Aliases: Will, William
    - Incomplete data: John (only a name)
    - Spelling variations: Owen, Eoghan
    - (OCR) Errors: Wihiam (William)

Margaret de Redvers



325 [**No date**]. *For Margaret de Redvers*. Margaret de Redvers has made fine with the king by 200 m., so that she is to be quit of sending knights with the king at his passage in the thirteenth, year, and for having her scutage from the knights' fees that she holds of the king in chief, namely 3 m. per shield for the king's army at the aforesaid passage, and so that she shall not be compelled to marry for as long as she wishes to live without a husband, and if she will wish to marry, she is to marry by her will on condition that she does not marry enemies of the king. [1]



Traces Through Time (3) – Collaboration

- The project set out as a collaboration between several institutes:
  - The National Archives
  - Institute of Historical Research
  - Brighton University
  - Leiden University
- Brighton University worked on *Natural Language Processing*.
- Our job was to perform record linkage on the extracted references delivered by Brighton University.

Problem Definition (1)

Record

A record r is a tuple of m attributes, each having a certain domain, that describes an entity, i.e., $r \in A_1 \times A2 \times \cdots \times A_m$.

- We assume that records are descriptions of people.
- Records are potentially ambiguous: they can describe more than one person.

Problem Definition (2)

Record Linkage

Given a set $\mathcal{R}$ of records, determine which of these records refer to the same entity.

- Record linkage is a binary classification problem.
- Record pairs are classified as matching or non-matching.
- The set of entities is usually unknown.
- Even with expert knowledge, it is hard to determine the match status of a record pair.

Record Extraction

- Instead of waiting for input from Brighton University, a simple context-free grammar was written in order to extract occurences.
- First names and articles (of, de la, etc.) were used as anchor points in the text.
- Capitalization, punctuation and ordering define the class of surrounding words.

$$\{\text{first name}\}\ \{\text{article}\}\ \{\text{capitalized word}\}$$
$$\downarrow$$
$$\{\text{first name}\}\ \{\text{article}\}\ \{\text{last name}\}$$

Record Examples

Concerning the corn of Roger of Hyde. Order to the sheriff of Oxfordshire to make the king's advantage without delay, by the view of law-worthy men, from all of the corn of Roger of Hyde, knight, in Hyde, who is with the Earl Marshal, and to put in gage etc. all those who he will find threshing that corn and intermeddling with the land of the same Roger without warrant, to be before the king at his command to answer for it.

Title	First name	Article	Last name	Role
	Roger	of	Hyde	
sherrif		of	Oxfordshire	
	Roger	of	Hyde	knight
	Roger			

Record Field Comparison (1)

- Records are compared on a per-field basis.
- Fields can be of many different types, but we assume strings.
- Many different ways of computing distances between strings exist.
- To give an impression we will have a look at one particular approach.

Record Field Comparison (2) – *Q*-gram similarity

- A *q*-gram is a sequence of *q* characters.
- To compute the *q*-grams of a word, move a sliding window over the word.
  - $\boxed{\text{Joh}}\text{n}$
  - $\text{J}\boxed{\text{ohn}}$
- String similarity between words defined as the similarity between their respective multisets of *q*-grams.

$$\mathrm{sim}_{\mathrm{jaccard}}(\sigma_1, \sigma_2) = \frac{c_{\mathrm{common}}}{c_1 + c_2 - c_{\mathrm{common}}}$$

Universiteit Leiden

Record Field Comparison (2) – *Q*-gram similarity

- A *q*-gram is a sequence of *q* characters.
- To compute the *q*-grams of a word, move a sliding window over the word.
  - $\boxed{\text{Joh}}\text{n}$
  - $\text{J}\boxed{\text{ohn}}$
- String similarity between words defined as the similarity between their respective multisets of *q*-grams.

$$\mathrm{sim}_{\mathrm{jaccard}}(\sigma_1, \sigma_2) = \frac{c_{\mathrm{common}}}{c_1 + c_2 - c_{\mathrm{common}}}$$

Record Field Comparison (3)

- Many different string similarity functions exist.
  - Edit distance: uses number of transformation steps.
  - Soundex: phonetic similarity.
- Similarity values can often be converted to distances, e.g., $\mathrm{dist}(\sigma) = 1 - \mathrm{sim}(\sigma)$.
- Distance function chosen depending on the content.

Candidate Pair Classification (1) – Distances

- A distance function is defined for every field.
- The classifier first uses these functions to map a record pair to an array of distance values.

$$\mathrm{map}_{\mathrm{dist}}(\boldsymbol{r}_1, \boldsymbol{r}_2) \rightarrow (d_1, d_2, \ldots, d_n) \qquad \text{with } |\boldsymbol{r_1}| = |\boldsymbol{r_2}| = n$$

- Distance values are thresholded to obtain a binary value: fields are either equivalent or nonequivalent.
- If one or both values are missing, the fields are considered equivalent.

Candidate Pair Classification (1) – Distances

- A distance function is defined for every field.
- The classifier first uses these functions to map a record pair to an array of distance values.

$$\mathrm{map}_{\mathrm{dist}}(\boldsymbol{r}_1, \boldsymbol{r}_2) \rightarrow (d_1, d_2, \ldots, d_n) \qquad \text{with } |\boldsymbol{r_1}| = |\boldsymbol{r_2}| = n$$

- Distance values are thresholded to obtain a binary value: fields are either equivalent or nonequivalent.
- If one or both values are missing, the fields are considered equivalent.

Candidate Pair Classification (2) – Probabilities

- If a record field pair is equivalent, we look up the prior probability of a person having that property, e.g., in a census.
- If such information is unavailable, we can compute the prior probability from the data.
- Equivalent, but are not unequal values, are treated as an equivalence class and their probabilities are summed.
- Using the data itself introduces a bias towards 'famous people', i.e., people that occur often.

Candidate Pair Classification (3) – Example

	First name	*Article*	*Last name*
p	0.182	0.917	0.00214
r_1	John	de	Engelfield
	$0.0 \updownarrow$		$0.13 \updownarrow$
r_2	John		Englefield
p	0.182		0.00321
	First name	*Article*	*Last name*

	First name	*Article*	*Last name*
p'	0.182		0.0535
E_q	1	1	1

Universiteit Leiden

Candidate Pair Classification (4) – Confidence

- The last step of classification is to aggregate the probabilities in a confidence score.
- Record pairs with nonequivalent fields are not considered for linking.
- Assume independence of fields, e.g., *First Name = John* does not affect the probability of *Last Name = Williams.*
- The confidence score is computed as the sum of log probabilities:

$$\mathrm{conf}(\boldsymbol{p}) = \sum_{i=0}^{|\boldsymbol{p}|} \log p_i$$

Contextual Information (1)

- The previously described procedure makes use of information that is relatively easy to obtain.
- Fields are often empty and the confidence score is therefore low.
- We may be able to exploit the fact that references occur within a certain context.

Contextual Information (2) – An Example

"A letter from the Secretary to Mr. Carkesse, desiring him to move the Commissioners of the Customs, that their Officers in the Out Ports may give this Board an Account of the quantities of Salt that is necessary and used in curing several species of Fish, was agreed and ordered to be sent."

"Ordered that Mr. Carkesse be desired to let this Board have on Tuesday next, if possible, the Account of Fish exported, which was desired the 17th of the last month."

Contextual Information (3) – Observations

- Many stop words occur that are probably not informative.
- There are a few interesting words: Customs, Fish, Salt.
- Individual words might be indicative of the topic discussed.
- We need of means of extracting these words from the data.
- We propose *Maximally k-Informative Itemsets* for this.

Maximally k-Informative Itemsets

Joint entropy

Suppose that $X = \{x_1, \ldots, x_k\}$ is an itemset, and $B = (b_1, \ldots, b_k) \in \{0, 1\}^k$ is a tuple of binary values. The *joint entropy* of X is defined as

$$H(X) = - \sum_{B \in \{0,1\}^k} p\left(x_1 = b_1, \ldots, x_k = b_k\right) \lg p\left(x_1 = b_1, \ldots, x_k = b_k\right)$$

- Presence and absence of items are treated equally.
- The maximum achievable entropy of an itemset of size k is k and has $P(X) = 0.5$.

Universiteit Leiden

Maximally k-Informative Itemsets

Joint entropy

Suppose that $X = \{x_1, \ldots, x_k\}$ is an itemset, and $B = (b_1, \ldots, b_k) \in \{0,1\}^k$ is a tuple of binary values. The *joint entropy* of X is defined as

$$H(X) = -\sum_{B \in \{0,1\}^k} p\left(x_1 = b_1, \ldots, x_k = b_k\right) \lg p\left(x_1 = b_1, \ldots, x_k = b_k\right)$$

- Presence and absence of items are treated equally.
- The maximum achievable entropy of an itemset of size k is k and has $P(X) = 0.5$.

Maximally k-Informative Itemsets

A	B	C	D
1	1	1	0
1	1	0	0
1	1	1	0
1	0	0	0
0	1	1	0
0	0	0	1
0	0	1	1
0	0	0	1

I	H
A	1.00
B	1.00
C	1.00
D	0.95

I_1	I_2	H
A	B	1.81
A	C	2.00
A	D	1.41
B	B	1.81
B	D	1.41
C	D	1.91

Maximally k-Informative Itemsets

Maximally informative k-itemset

Suppose that I is a collection of n items. An itemset $X \subseteq I$ of cardinality k is a *maximally informative k-itemset*, iff for all itemsets $Y \subseteq I$ of cardinality k,

$$H(Y) \leq H(X)$$

- There are many itemsets that can be a Miki: $\binom{n}{k}$.
- Knobbe et al. proposed several algorithms for finding exact and approximate Mikis.

Maximally k-Informative Itemsets

Maximally informative k-itemset

Suppose that I is a collection of n items. An itemset $X \subseteq I$ of cardinality k is a *maximally informative k-itemset*, iff for all itemsets $Y \subseteq I$ of cardinality k,

$$H(Y) \leq H(X)$$

- There are many itemsets that can be a Miki: $\binom{n}{k}$.
- Knobbe et al. proposed several algorithms for finding exact and approximate Mikis.

Maximally k-Informative Itemsets

function ForwardSelection(k, n)
    $X := \varnothing$
    for $i := 1$ ***to*** k **do**
        $h_{\max} := 0$
        for $j := 1$ ***to*** n **do**
            $h := \mathrm{JointEntropy}(X \cup \{j\})$
            if $j \notin X$ **and** $h \geq h_{\max}$ **then**
                $h := h_{\max}$
                $m := j$
                $X := X \cup \{m\}$
    return X

Maximally k-Informative Itemsets

Universiteit Leiden