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and applications in the analysis of historical documents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 Informative *k*-Itemsets
7. Experiments
8. Conclusions and future work

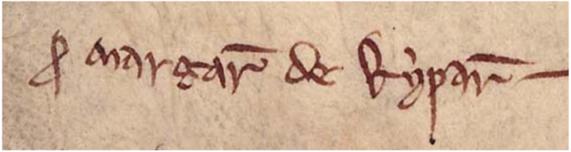
Traces Through Time (1) – Context

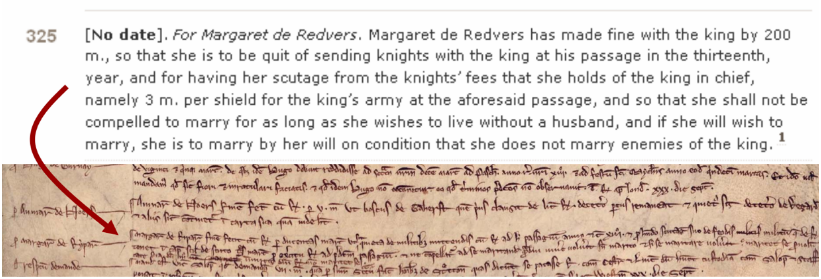
- The National Archives stores millions of documents.
- Many documents have been converted to a digital format.
    - Automatic: Optical Character Recognition (OCR).
    - Manual: transcribed by hand.
- Connecting pieces of information regarding people is mostly done manually.
- Automating this process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

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 (Arno Knobbe, Kleanthi Georgala, me)
- 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.

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

Record Examples – Fine Rolls of King Henry III

“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.”[1]

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

[1] *Fine Rolls Henry III: 18 HENRY III (28 October 1233–27 October 1234).* http://www.finerollshenry3.org.uk/content/calendar/roll_033.html [Accessed: 28-07-2015]. 2015.

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.
  - $\boxed{\text{Joh}}$n
  - J$\boxed{\text{ohn}}$
- String similarity between words defined as the similarity between their respective multisets of *q*-grams.[2]

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

[2] Esko Ukkonen. “Approximate string-matching with q-grams and maximal matches.” In: *Theoretical Computer Science* 92.1 (1992), pp. 191–211.

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.[2]

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

[2] Ukkonen, "Approximate string-matching with q-grams and maximal matches."

Record Field Comparison (3) – More Metrics

- Many different string similarity functions exist.
  - Edit distance[3]: uses number of transformation steps.
  - Soundex[4]: 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.

[4] Vladimir I Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals." In: *Soviet physics doklady*. Vol. 10. 8. 1966, pp. 707–710.

[4] *The Soundex indexing system.* http://www.archives.gov/research/census/soundex.html [Accessed: 28-07-2015]. 2015.

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.

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

- 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.

Universiteit Leiden

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) – An 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

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:

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

Contextual Information (1)

- The previously described procedure makes use of information that is relatively easy to obtain.
- Fields often have missing values 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 Informative k-Itemsets*[5]for this.

[5] Arno J. Knobbe and Eric K. Y. Ho. "Maximally Informative K-itemsets and Their Efficient Discovery." In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '06. ACM, 2006, pp. 237–244. ISBN: 1-59593-339-5.

Maximally Informative *k*-Itemsets – Definitions

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.

Maximally Informative *k*-Itemsets – Definitions

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.

Maximally Informative *k*-Itemsets – An Example

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 Informative k-Itemsets – Definitions

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.

Universiteit Leiden

Maximally Informative k-Itemsets – Definitions

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.

Universiteit Leiden

Maximally Informative k-Itemsets – Algorithm

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

[5] Knobbe and Ho, “Maximally Informative K-itemsets and Their Efficient Discovery.”

Maximally Informative *k*-Itemsets – Usage

- Per definition, Mikis consist of items that are uncorrelated.
- Roughly stated: items that occur frequently together are captured in the Miki by only one of the items.
- This makes the items within a Miki a good candidate for modelling ‘topics’ of text segments.
- For each item in the Miki, we introduce a binary feature to the records, using the probability of each in the ‘lookup step‘ of the classifier.

Experiments – Dataset

- For validation of the system, we used manually annotated data from The Gascon Rolls project.
- The Gascon Rolls are records that were drawn up by the English royal administration of Aquitaine-Gascony (south-western France) between 1273 and 1468.
- Contain grants of land, oaths of treaties and other important documents.
- The dataset consists of so-called *calendars*, which are summaries of the actual rolls.

Experiments (1) – Dataset

Table: Overview of the size of the Gascon Rolls dataset.

	Size
Number of rolls	66
Number of membranes	943
Number of sections	1153
Number of occurrences	30426
Sum of file sizes	27 MB

Experiments (2) – Occurrence Extraction

- The dataset contains 25836 annotated occurrences.
- A parser based on a context-free grammar was able to find and segment 22206 occurrences.
- Out of these 1684 were not annotated and removed from the dataset resulting in 20522 segmented records.

Experiments (3) – Occurrences

Field	Description
id	As provided in the source document.
forename	
article	Word(s) between the forename and surname.
surname	
provenance	Origin of a person.
title	Used to address a person.
role	Ascribed to person.
regnal_number	Used to distinguish monarchs.
fileId	Filename
sectionId	Section identifier.
pos	Start position in section.
endpos	End position in section.
words	Words and their frequencies in section.
orig	Original text.

Experiments (4) – Mikis

- To reduce the number of items considered, several unpromising items were removed.
  1. *Infrequent tokens*: any tokens that appear less than 5 times are discarded.
  2. *Frequent tokens*: all tokens appearing in a fixed list of 319 stop words were ignored.
  3. Tokens that are part of *occurrences*, such as first names, were ignored.

Experiments (5) – Evaluation

Precision

Precision, E_p, is the fraction of pairs that are correctly classified as true matches, i.e.,

$$E_p = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FP}|}$$

Recall

Recall, E_r, is the fraction of true matches that are detected by the system, i.e.,

$$E_r = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}|}$$

Experiments (5) – Evaluation

Precision

Precision, E_p, is the fraction of pairs that are correctly classified as true matches, i.e.,

$$E_p = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FP}|}$$

Recall

Recall, E_r, is the fraction of true matches that are detected by the system, i.e.,

$$E_r = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}|}$$

Experiments (6) – Evaluation

F-measure

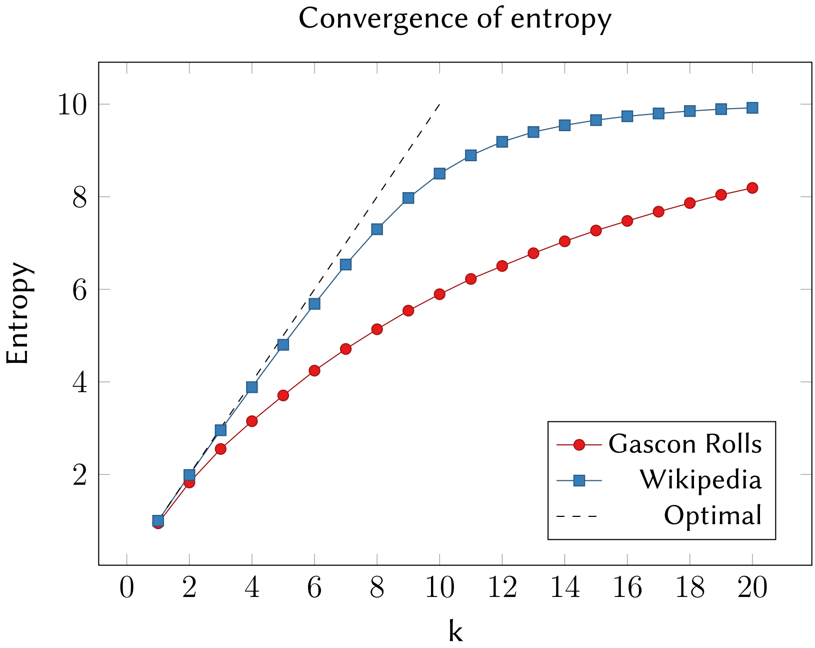
The general *F-measure* measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision in the following way:[6]

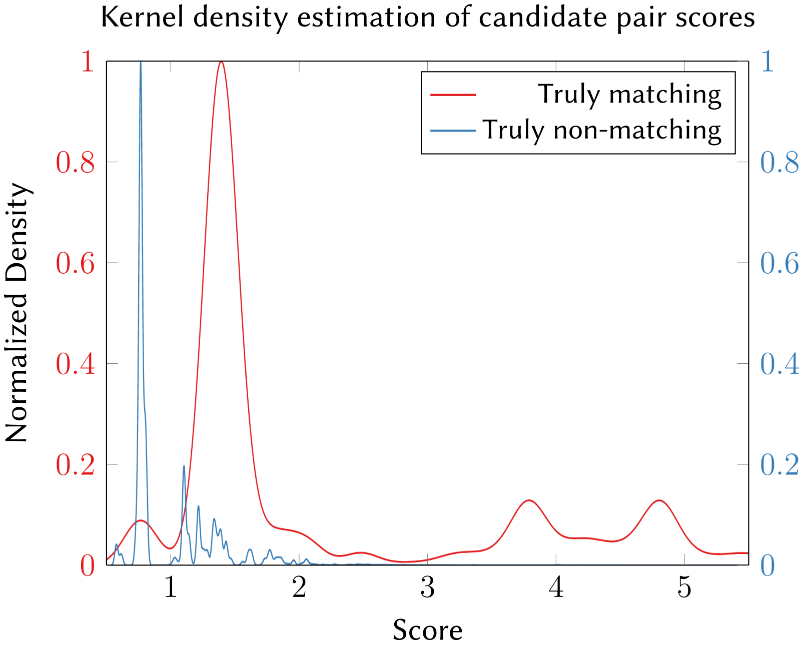
$$F_\beta = (1 + \beta^2) \cdot \frac{2E_pE_r}{(\beta \cdot E_p) + E_r}$$

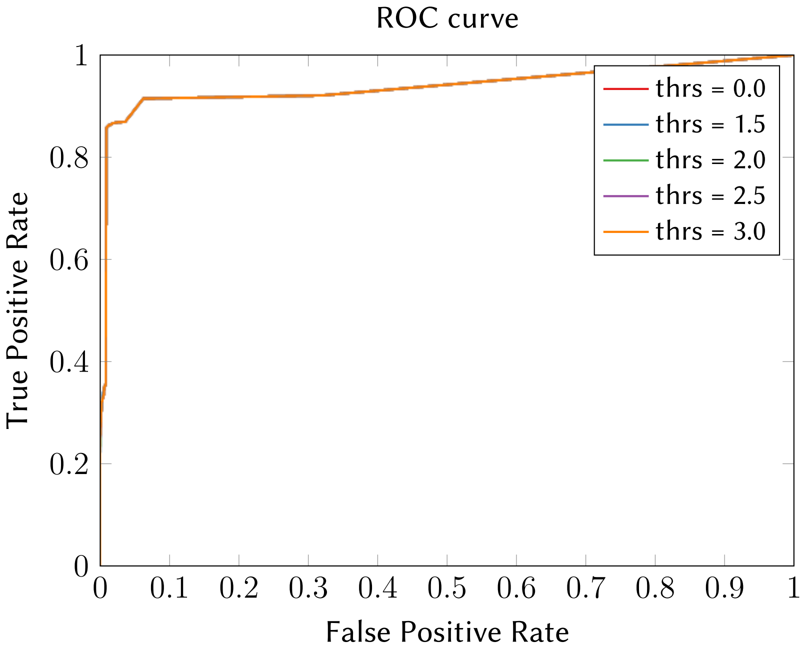
- Details are not important: the F-measure is a means of combining precision and recall in one measure.

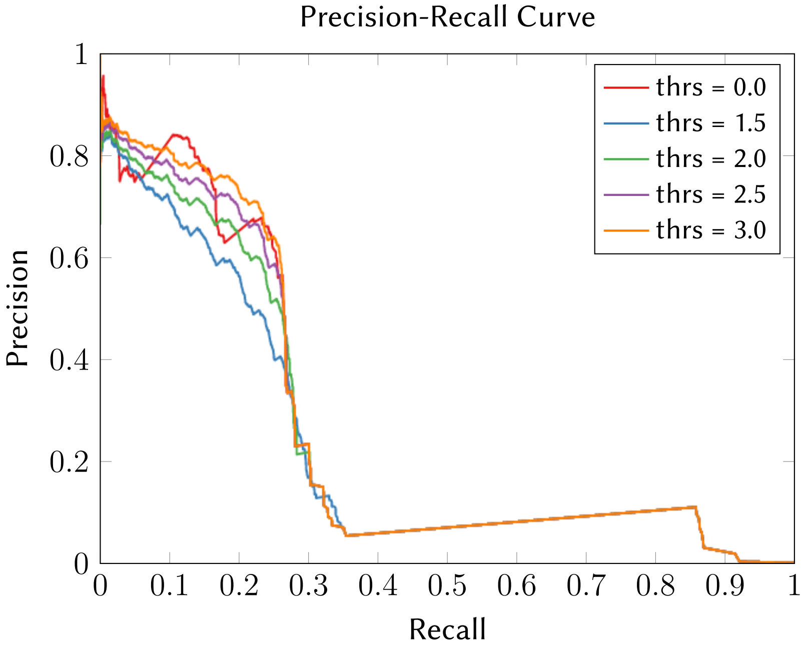
[6] C. J. Van Rijsbergen. *Information Retrieval.* 2nd. Newton, MA, USA: Butterworth-Heinemann, 1979. ISBN: 0408709294.

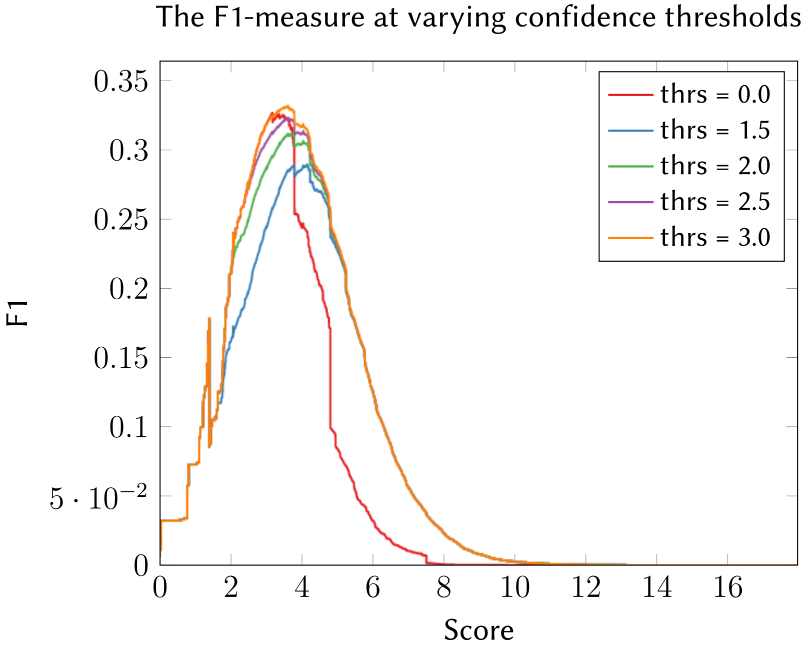
k	Word	Cum. Entr.	p
1	king	0.94	0.36
2	letters	1.83	0.30
3	order	2.55	0.29
4	service	3.15	0.16
5	bordeaux	3.71	0.17
6	gascony	4.24	0.15
7	duchy	4.71	0.14
8	ordered	5.14	0.12
9	granted	5.54	0.12
10	grant	5.89	0.11


Convergence of entropy
Entropy
k
0
2
4
6
8
10
12
14
16
18
20
2
4
6
8
10
Gascon Rolls
Wikipedia
Optimal


Kernel density estimation of candidate pair scores
Normalized Density
Score
Truly matching
Truly non-matching
0
0.2
0.4
0.6
0.8
1
1
2
3
4
5
0
0.2
0.4
0.6
0.8
1


ROC curve
True Positive Rate
False Positive Rate
0
0.2
0.4
0.6
0.8
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
thrs = 0.0
thrs = 1.5
thrs = 2.0
thrs = 2.5
thrs = 3.0


Precision-Recall Curve
Precision
Recall
0
0.2
0.4
0.6
0.8
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
thrs = 0.0
thrs = 1.5
thrs = 2.0
thrs = 2.5
thrs = 3.0


The F1-measure at varying confidence thresholds
F1
Score
0.35
0.3
0.25
0.2
0.15
0.1
$5 \cdot 10^{-2}$
0
0
2
4
6
8
10
12
14
16
thrs = 0.0
thrs = 1.5
thrs = 2.0
thrs = 2.5
thrs = 3.0

Conclusions (1) – Occurrences

- The grammar proved to do a reasonable job at extracting occurrences.
- Writing the grammar is a lot of work and dataset specific.
- Occurrence extraction was not the main focus of this project and should be done with more sophisticated methods.

Conclusions (2) – Mikis

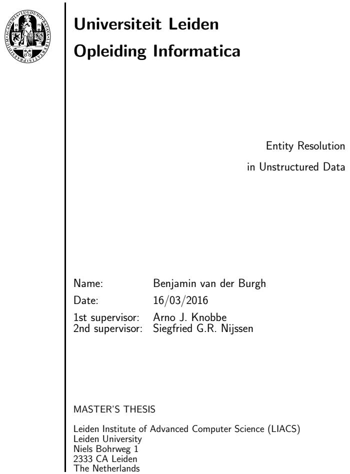
- Mikis have not shown to have a significant impact on the performance of the record linker.
- There can be many reasons for this.
  1. Topic modelling does not give the kind of textual information we want.
  2. Mikis are unable to model topics appropriately.
  3. The usage of Mikis was incorrectly handled in the confidence score computation.
- More work is needed to be conclusive.

Conclusions (3) – Record Linkage

- The linker was able to retrieve about 30% of the truly positive pairs with reasonable precision.
- The remaining part was hard to distinguish from the negative pairs, based on the confidence score.
- Candidate pairs are now compared in isolation.
- It might be worthwhile to research methods that consider pairs in a relational setting.

Fin

Universiteit Leiden



Universiteit Leiden

Opleiding Informatica

Entity Resolution
in Unstructured Data

Name:	Benjamin van der Burgh
Date:	16/03/2016
1st supervisor:	Arno J. Knobbe
2nd supervisor:	Siegfried G.R. Nijssen

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

LaTeX source available at
https://github.com/benjaminvdb/master_thesis

LaTeXTips

- Use the *booktabs* package for pretty tables.
- Employ a model-view style for displaying data:
  1. Store data in separate file, e.g., CSV (the model).
  2. Style the table or plot in a separate file (the view).
  3. Include these in your LaTeXto keep your text separated from graphics.
  4. Easy to include newer version of results (just replace data file!).
- Plot all your data using LaTeXwith the *tikz* package.