Linking historical ship records to newspaper archives

THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

ARTIFICIAL INTELLIGENCE

specializing in

KNOWLEDGE TECHNOLOGY AND INTELLIGENT INTERNET APPLICATIONS

Author:
Andrea Cristina Bravo-Balado

Supervisor:
Victor de Boer
Second reader:
Niels Ockeloen

Department of Artificial Intelligence
Faculty of Sciences
VU University Amsterdam
Amsterdam, Netherlands

July 2014

Linking historical ship records to newspaper archives

Andrea C. Bravo-Balado
Vrije Universiteit Amsterdam
Faculty of Sciences
Amsterdam, the Netherlands
a.c.bravobalado@student.vu.nl

Abstract

Linking historical datasets and making them available for the Web has increasingly become a subject of research in the field of digital humanities. In this master project, we focused on discovering links between ships from a dataset of Dutch maritime events and newspaper articles from historical archives. We have taken a two-stage approach: first, an heuristic-based method for record linkage and then machine-learning algorithms for article classification to be used for filtering in combination with domain features. Evaluation of the linking method has shown that certain domain features were indicative of mentions of ships in newspapers. Moreover, the classifier methods scored near perfect precision in predicting ship related articles.

Keywords

Text classification, machine learning, record linkage, entity linkage, historical research, digital humanities, digital history.

1 INTRODUCTION

The Digital Humanities are a rising area of research concerning the intersection of different disciplines of the humanities with information technologies. More specifically, digital history, as the name suggests, is dedicated to the improvement of the study and preservation of history. Particularly in the area of historical research, it is common for historians to collect data from archives for their own research questions, often manually. Recently, many historical archives have been digitized but the problem of the data being scattered in different repositories and formats persists, driving researchers in inter-disciplinary settings to focus their efforts on linking historical datasets for enrichment and availability on the Web, as shown in the survey [18] and the works in [2].

However, the problem of aligning and linking datasets is not trivial. There are many challenges regarding disambiguation of entities, particularly on unstructured datasets and on the field of historical research, where things change radically over time. Using domain knowledge as well as machine learning text classification approaches, we are able to link different datasets for enrichment and help ease collaboration among historical researchers.

In the Netherlands, history is intimately related to the maritime activity and research in this field is very active. The maritime history has been essential in the development of economic, social and cultural aspects of Dutch society. As such an important sector, it has been well documented by shipping companies, governments, newspapers and other institutions. Interesting data, such as shipping movements and ship crew members has been made available and current efforts focus on linking these different datasets using Linked Data techniques and principles. Our work is part of the Dutch Ships and Sailors Linked Data Cloud project [5], sharing the domain and one of the datasets but with a different goal.

In this project we assume that, given the importance of maritime activity in every day life in the XIX and XX centuries, announcements on the departures and arrivals of ships or mentions of accidents or other events, can be found in newspapers. Enriching historical ship records with links to newspaper archives is significant for the digital history community since it connects two datasets that would have otherwise required extensive annotating work and man hours to align.

The challenges on record linkage motivate our work and lead us to our research question:

- How can we automatically enrich a dataset of historical ships by effectively linking instances with historical newspaper archives?

In order to answer this question, we need to ask ourselves the following:

1. To what extent can we use domain knowledge from historical ships to identify ship instances on newspaper archives for this record linking task?
2. To what extent can we use a text classification approach to label relevant documents?
3. How can we combine these approaches to effectively link both datasets?

The remainder of this thesis is organized as follows: first, we describe previous work in which we build upon in Section 2. In Section 3.1, we describe the MDB dataset, containing information about Dutch ships and sailors and in Section 3.2, the description of the newspaper archives provided by the National Library of the Netherlands can be found.

In Section 4, we give details about our implementation. In Section 5.1, we present our approach of using domain features as filters for record linkage. Following, Section 5.2 is focused on our text classification approach and the details of data transformation and classifiers settings. Then, a combination of both approaches is explained in Section 5.3. Finally, we give some final remarks on Section 6, followed by Section 7, where we conclude our work.

2 RELATED WORK

This master project fits into several areas of research in the realm of artificial intelligence, such as record linkage, information retrieval, information extraction and machine learning. Additionally, previous work on the field of historical research has been carried out based on Linked Data and Semantic Web technologies.

In order to link datasets, there are many approaches found in the literature. For instance, the idea of using domain knowledge for entity linkage is not new. In [15], the task of entity linkage is focused on linking named entities extracted from unstructured text to the entities on a knowledge base. Although our approach is essentially the opposite, where we use a structured dataset (instead of a knowledge base) to find entities in the unstructured text and the domain is different, this area of research is related to ours. Even though the domain is different, the most similar to our goals is the research done in [21], where the authors present a system to disambiguate entity mentions in texts and link them to a knowledge base, the main difference being our use of a database in place of a knowledge base. Furthermore, in [9], in addition to domain knowledge, the authors take an information retrieval approach for linking entities. In [13], the authors use TD-IDF weighting scheme to create a term vector and use standard scores of precision and recall for information retrieval to evaluate their experiments of linking articles from different encyclopedic sources. Even though our work builds on top of some of these techniques found in the literature, we have not found works on the domain of historical ships and sailors.

Furthermore, finding and linking relevant newspaper articles has been done in [3] using a vector space model with a similarity function. The main difference with our research, besides the domain, is that the authors intend to link similar and current archives, while ours are essentially different and historical. Moreover, linking relevant newspaper articles from the Dutch National Library archives has been done in [11] and [14], albeit on a different domain, linking parliamentary and political debates with media outlets. Similarly, although their approach is by means of a semantic model and topic modeling, as well as using named entities for ranking, our experimental setup and evaluation procedure is based on [12].

Machine learning text classification has also been used for entity disambiguation and linkage. In [27], the authors experiment with text classification methods for literary study. Yu goes into detail about the importance of preprocessing and choice of classifiers, in which we have based some of our work. Moreover, in [25] the authors present information extraction as a classification problem to be solved using machine learning algorithms, such as Support Vector Machines (SVM) and Naive Bayes (NB), among others in order to extract information related to natural disasters from newspaper articles in Spanish.

Our work is part of recent efforts into linking historical datasets in the Netherlands, namely the works on linking datasets from German occupied Dutch society in [4], historical census data in [17] and Linked Data for cultural heritage in [6].

3 DATASET DESCRIPTION

3.1 MDB

The *Noordelijke Monsterollen Databases* (en: Northern muster rolls databases) is a dataset containing official lists of crew members, known as muster rolls, for ship companies from the three northern provinces of the Netherlands (Groningen, Friesland and Drenthe). The data was curated from mustering archives by historian Jurjen Leinenga and extends over the years 1803 and 1937. The MDB dataset was provided on CSV format and imported into a PostgreSQL database, resulting in two tables: one table for Ships with 16974 rows and one table for Crew, with 76941 rows. However, more preprocessing was required for the data to be usable for the experiments in this project.

First, an intermediate table for compiling the relevant information from the tables Ships and Crew was created. The criteria used was that the identifier from the MDB coincided in both tables Ships and Crew and that the year was the same in both tables. Additionally, only the relevant information about the captain of the ship and no other crew members was kept. This table resulted in 11930 instances because items without a ship name from the original collection were left out.

In the MDB collection, names of ships are not unique and may appear several times. Since we wanted each instance to be as unique as possible, we decided to group ships sharing the same name, the last name of the captain and the type of ship by means of the *SQL DISTINCT ON* operation. Additionally, every mentioned year has been grouped together with the respective ship name. According to domain experts, these three features can help decide whether two or more ships in a text are referring to the same ship. A new table was created by this operation, resulting in 5604 ship instances. For example, the ship named "Anna Christina" is featured 3 times in the MDB in the years 1878, 1909 and 1910 (See Table 1). Hence, only one record of the ship is kept, grouping the relevant information (See Table 2). Following, there is a simple sorting of the years array from lowest to highest. This is done so that a year interval can be determined by picking the lowest and the highest years. For instance, following the same example, for the ship Anna Christina, the year interval will be: 1878, 1910. The resulting table was used in the acquisition of the following dataset. The queries used for these operations can be found on Appendix A.

MDB Example Data			
ID	MDBid	Year	Ship Name
5858	1878-106	1878	Anna Christina
9271	1909-11	1909	Anna Christina
9410	1910-26	1910	Anna Christina

Table 1: Example of instances before preprocessing

MDB Example Data				
ID	Ship Name	Count	Ship IDs	Ship Years
231	Anna Christina	3	5858 9271 9410	1878 1909 1910

Table 2: Example of instances after preprocessing

3.2 National Library of the Netherlands newspaper archives (KB)

The historical newspaper archives belong to the *Koninklijke Bibliotheek* (en: National Library of the Netherlands). The data contains text and images of newspaper articles from 1618 to 1995 in the Dutch language. It is important to note that the newspaper archives are not limited to the maritime domain. The text of the articles have been obtained by means of OCR (Optical Character Recognition), which has the advantage of making the text fully searchable but the disadvantage of the noisiness of the text. The metadata of the articles is available in XML format. Given that querying the National Library directly is costly and time consuming, a local dataset needed to be built. The interface for querying the KB collections is based on a Java implementation of the SRU (Search/Retrieve via URL) standard XML-based protocol for search queries, known as JSRU.
The first step to build our own dataset, was to compose an URL for each instance on the preprocessed MDB dataset. The name of the ship was used as the query search term. Given that the protocol needs a full date for querying, rather than only a year, the year interval was coupled with the beginning and end of the calendar year (01/01, 31/12) to create the start and end date of our search. Furthermore, we subtracted 5 years to the bottom year and added 5 years to the top, in order to broaden our dataset, opening up to the possibility of finding mentions of ships besides known instances. For the sake of convenience, it has been chosen to retrieve the top 100 results. The reasoning behind this is that queries with more results would be less likely to be ship mentions. The built URLs are stored in the database and then used to query the KB.
The next step was to parse the response from the KB. The response is a list of candidate articles in XML format. Once each XML file has been retrieved and stored, it is parsed so that every candidate result, including important metadata, is stored. Search queries that returned empty candidates lists were left out. In the XML structure, there is an identifier (URI) which points to the OCR text of the newspaper section.
In the final preprocessing step, the OCR text URIs are used to retrieve and store the article text for every instance.
This process will serve as the baseline experiment, which will be explained in Section 5.1.2

4 IMPLEMENTATION

In this section, we will explain the technical details of our implementation. All the data preprocessing and experiments explained in this thesis have been performed using scripts coded on Java SE 6, which are available at the BitBucket repository for this project [1]. Furthermore, all data has been stored in a relational database on PostgreSQL 9 and we have used Hibernate ORM (Object/Relational Mapping) for data persistence via JDBC.
We begin with the process of data conversion from CSV to SQL using a parser library for Java known as OpenCSV. Once the MDB data was converted and stored in the database, we generated the URLs for querying the KB, as explained in Section 5.1.1. Then, we proceeded to harvest information from the KB in order to build the local dataset of newspaper articles we needed. This was possible through the JSRU protocol implemented at the KB. The next step was to parse the XML response from the KB. For this, we have implemented a native Java parser that produces DOM object trees from XML documents. Afterwards, we selected the nodes to be kept and stored them in the database. This process was run in batches given the volume of data and limited resources. The algorithms used for the baseline experiment, as well as experiments 1, 2, 3, 6 and 7 were written as SQL queries, which allowed us to select subsets for evaluation. Furthermore, initial exploration for Experiments 4 and 5 was done using the stand-alone version of WEKA [2]. However, after getting acquainted with the dataset, we decided to use WEKA's library on our own Java code for the task of labeling unseen data. Our code for the classifier is based on that of Gómez-Hidalgo [3]. We have also implemented a script for the creation of training and testing ARFF files (compatible with WEKA), from information on the database. Newly labeled data was stored in plain text files on disk and subsequently updated in the database.

5 EXPERIMENTS

In the following subsections, we will focus on the experiments performed in order to answer our research questions. We have divided our experiments in three stages. First, the experiments where domain knowledge features are used as filters for record linkage. Second, the experiments using machine learning techniques for text classification and finally, a combination of both techniques for a final algorithm. Given the exploratory nature of this research, where each experi-

[1] https://bitbucket.org/andreacbravob/master-project

[2] http://www.cs.waikato.ac.nz/ml/weka/

[3] https://github.com/jmgomezh/tmweka/tree/master/FilteredClassifier

ment gave us insight to design the next, we have organized them in chronological order.

5.1 DOMAIN FEATURE FILTERING EXPERIMENTS

5.1.1 Approach and methodology

For these experiments, our approach is to identify domain knowledge features that are suitable for distinguishing different ship instances and thus work well for filtering candidate links. More specifically, our method is to restrict the number of candidate links by means of the SQL *WHERE* clause. More details for every experiment are provided next.

5.1.2 Baseline: Name of the ship and date restriction

This first experiment serves as both a baseline algorithm as well as the data acquisition process for building our local dataset. As such, it will serve as the basis for all subsequent experiments. As explained in Section 3.2, the dataset has been drawn using two domain features in the query: the name of the ship and a year interval.
On the one hand, ship names are hardly unique. Ship names are usually female names and geographical locations as well as abstract concepts such as friendship, hope and faith. Although this feature on its own is not helpful for disambiguation, it allows us to ensure that the text for each candidate link we import into the dataset contains at least one mention of the ship name. On the other hand, the year interval allows us to control the search scope, i.e. only considering mentions of ship names within the years mentioned in the dataset. Once more, given that ship names are not unique, it is more likely that ships mentioned more than a few years apart (e.g. more than 30 years) refer to different ship instances.

5.1.3 Experiment 1: Captain's last name

For this experiment, we wanted to test a particular feature that domain experts have indicated can help on the disambiguation of candidate links from newspaper archives: the last name of the captain. This is mainly due to the fact that the captain of a ship is unlikely to change over time, unless the ship gets lost or destroyed. Note that this time, we are not performing queries directly on the KB newspaper archives but rather on the local dataset we have built, explained in Section 3.2. This means that we are querying text that contains at least a ship name and the dates are within a possible range for given instance.
We use the *like* operator to compare the last name of the captain for every instance found in our dataset with the text of the articles on the candidate links. The *like* expression "returns true if the string matches the supplied pattern" [4]. For this query, the pattern is represented by *text like '%'* || *captainlastname* || *'%'*. The complete SQL query for this algorithm can be found in Listing 1.

[4] http://www.postgresql.org/docs/9.2/static/functions-matching.html

```
SELECT distinctcandidates.id AS "C_ID",
distinctships.monsterrollen AS "M_ID",
distinctships.shipname AS "Ship_Name",
distinctcandidates.pdfurl AS "Article_URL",
distinctcandidates.texturl AS "Text_URL",
distinctcandidates.texttitle AS "Text_Title",
distinctcandidates.text AS "Text",
distinctcandidates.texttype AS "Text_Type",
distinctships.shiptype AS "Ship_Type",
distinctships.shipsizes AS "Ship_Sizes",
distinctships.captainlastname AS "Captain_last_name",
distinctships.captainfirstname AS "Captain_names"
FROM distinctcandidates, distinctships
WHERE
distinctcandidates.baselinedsid=distinctships.id
AND text like '%' || captainlastname || '%';
```

Listing 1: SQL query Experiment 1

5.1.4 Experiment 2: Year restriction

Recall that the instances on the local dataset already contain at least one mention of a ship name and that the date used to query the KB was extended to five years prior and five years after the year interval featured on the MDB database. For Experiment 2, we wanted to make sure that the year of publication of the candidate links from the newspaper archives was within the original MDB year interval for each ship instance. The reasoning behind it is that ship instances that have already been found on historical records on given years, are less likely to be mentioned on newspapers on years too far apart given that the average useful lifespan of a ship, according to domain experts, is about 30 years.
We use the PostgreSQL *extract* function for dates to obtain the year of publication and operators are used to compare it with the year of the ship. The SQL query can be found under Listing 2

```
SELECT
distinctcandidates.id AS "C_ID",
distinctships.monsterrollen AS "M_ID",
distinctships.shipname AS "Ship_Name",
distinctcandidates.pdfurl AS "Article_URL",
distinctcandidates.texturl AS "Text_URL",
distinctcandidates.texttitle AS "Text_Title",
distinctcandidates.text AS "Text",
distinctcandidates.texttype AS "Text_Type",
distinctships.shiptype AS "Ship_Type",
distinctships.shipsizes AS "Ship_Sizes",
distinctships.captainlastname AS "Captain_last_name",
distinctships.captainfirstname AS "Captain_names"
FROM distinctcandidates, distinctships
WHERE
distinctcandidates.baselinedsid=distinctships.id
AND extract(year from distinctcandidates.textdate)
>= distinctships.bottomyear
AND extract(year from distinctcandidates.textdate)
<= distinctships.topyear;
```

Listing 2: SQL query Experiment 2

5.1.5 Experiment 3: Combining captain's last name and year restriction

For this final domain feature experiment, we have decided to combine the last name of the captain along with the year restriction in order to test our hypothesis that the latter could be useful along with a suitable feature. Our theory is that these two features combined would boost the precision obtained on Experiment 1. The SQL query relies on a composed *WHERE* clause to combine both domain features. The query can be found on Listing 3.

```
SELECT
distinctcandidates.id AS "C_ID",
distinctships.monsterrollen AS "M_ID",
distinctships.shipname AS "Ship_Name",
distinctcandidates.pdfurl AS "Article_URL",
distinctcandidates.texturl AS "Text_URL",
distinctcandidates.texttitle AS "Text_Title",
distinctcandidates.text AS "Text",
distinctcandidates.texttype AS "Text_Type",
distinctships.shiptype AS "Ship_Type",
distinctships.shipsizes AS "Ship_Sizes",
distinctships.captainlastname AS "Captain_last_name",
distinctships.captainfirstname AS "Captain_names"
FROM distinctcandidates, distinctships
WHERE
distinctcandidates.baselinedsid=distinctships.id
AND text like '%' || captainlastname || '%'
AND extract(year from distinctcandidates.textdate)
>= distinctships.bottomyear
AND extract(year from distinctcandidates.textdate)
<= distinctships.topyear;
```

Listing 3: SQL query Experiment 3

5.1.6 Evaluation

In order to assess these experiments, we have decided to perform a manual evaluation by means of a survey. Given the size of the dataset, it is unfeasible to manually assess every instance. Therefore, we have randomly selected a subset of 50 instances to be included in the survey for every experiment. In [24], it is stated that a sample of 50 instances is enough to extrapolate the evaluation to the rest of the dataset, according to their experiments for a similar problem. The SQL queries used to retrieve the samples are the same as listed on Listings 1, 2 and 3, except adding the SQL clause *ORDER BY RANDOM() LIMIT 50;*.

For every instance, we have included domain knowledge to assist the rater in the evaluation process. The background knowledge includes data from both datasets, including the name of the ship and its type, the last name of the captain and an array of possible first names from the MDB dataset and the type of text, e.g. article, advertisement, obituary, etc., the title and text content from the KB dataset.

In this case, the evaluation criteria is based on a 5-point Likert scale, ranging from strong disagreement to strong agreement to answer the question of whether the newspaper text shall or shall not be linked to the given ship. The 5-point Likert scale is used for manual evaluation and the calculation of mean and standard deviation. For precision, recall and F1 score calculations, this scale is transformed into binary, where values 1, 2 and 3 are considered non-relevant items (label 0) and values 4 and 5 are considered relevant (label 1). A sample of the survey can be found under Appendix B.

For the initial experiment, the evaluation has been performed by historian and domain expert Jurjen Leinenga (rater C), as well as supervisor Victor de Boer (rater B) and myself (rater A).

In order to validate both our evaluation method and our instrument, we have chosen to measure the inter-rater agreement for each pair of raters by means of the weighted Cohen's Kappa coefficient (κ). Also, distance measures have been used as weights for this calculation.

The results of the evaluations can be extrapolated for the rest of the dataset and thus we are able to calculate precision and an approximate recall for each experiment.

5.1.7 Results

In this section we present the results for the domain feature filtering experiments. After the preprocessing phase, the resulting table contains 413863 candidate links, corresponding to 5078 ship instances, which conforms our local dataset and was used in all subsequent experiments in this project. Recall that we have performed manual evaluations of random subsets of 50 instances for each algorithm performed by three raters. We calculated a weighted Cohen's Kappa coefficient and the results are found in Table 3. Among the many interpretations of this measure found in the literature, we have chosen the one described by Viera et al. [26]. Using this interpretation, we can say that between raters A and B as well as raters A and C there is substantial agreement, whereas the degree of agreement between B and C is moderate.

Weighted Cohen's Kappa		
Raters	κ	**Interpretation**
A-B	0.76	Substantial agreement
B-C	0.58	Moderate agreement
A-C	0.62	Substantial agreement

Table 3: Inter-rater agreement for pairs of raters

For the assessment of our experiments, we have chosen to calculate the standard precision and F_1 scores, as well as an approximate recall. Precision (1) is the fraction of retrieved documents that are relevant while the F_1 score (2) is a weighted harmonic mean of precision and recall [16].

$$Precision = \frac{\text{\# relevant items retrieved}}{\text{\# retrieved items}} \tag{1}$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2}$$

In order to calculate the recall score, given the difficulty to assess the actual number of relevant results in the newspaper archives we propose an approximation, based on the estimated number of correct links retrieved by an algorithm divided by the estimated number of correct instances in the dataset, for which we take the baseline experiment, i.e. our most inclusive algorithm. The approximate recall (3) is calculated by:

$$Recall_x = \frac{\text{retrieved\_items}_{\text{x}} \cdot \text{precision}_{\text{x}}}{\text{retrieved\_items}_{\text{baseline}} \cdot \text{precision}_{\text{baseline}}} \quad (3)$$

The results of the experiments are found on Table 7. Overall, the highest precision score is 0.96, given by the algorithm version of Experiment 3, where 48 out of 50 instances were considered relevant. Furthermore, the algorithm version with the best F_1 score is that of Experiment 1.

5.1.8 Discussion

In this section, we will discuss some of our findings. The first experiment has served to establish a baseline, as well as the construction of the dataset for the rest of this project.
Considering that ship names are hardly unique, even when searching within time constraints, we were surprised to find correct candidate links. Our precision shows that we can use domain knowledge to retrieve related results from newspaper archives. Recall that the newspaper archives are not specialized in ships or maritime activities. We believe this is important because even using a rather small sample, we were able to find, albeit a few, relevant links. This lead us to think that we could try other domain knowledge features to find a better indicator of an article mentioning ships.
One of our main concerns regarding the data coming from the KB newspaper archives was the quality of the text obtained by means of OCR. Even when the retrieved XML files are semi-structured, the article texts themselves are unstructured. However, given these results it is possible to determine that OCR text is of sufficient quality for the algorithms to work.
As for the MDB data, misspellings of ship or crew member names are a common problem. In this case, we have considered different spellings as different instances, because dealing with misspellings slightly derails from our research questions.
Furthermore, the results for the inter-rater agreement coefficient help us validate the instrument and show us that raters A and B are qualified to evaluate samples, even lacking the domain experience that rater C has. It also lead us to decide that in the case of rater C being unavailable, rater A would perform the evaluations, with supervision by rater B in case of doubts.
Moreover, in Experiment 1, we have found that the last name of the captain of a ship appears to be a good indicator for candidate link selection. This has also helped us gain more insight on the way ship instances are featured on the newspaper archives. By analyzing the texts, we have noticed that it is common to find the name of the ship along with the last name of the captain (either before or after), a port name and a date at the beginning of the sentence. An example can be found on Table 4.

Common ship instance	
Ship Name	Grietina
Captain's last name	Sprik
Text title	BINNENGEKOMEN
Text	NEW-YORK, 6 Aug.; Albert, Meijer, Bremen. UITGEZEILD. KROON3TAD, 16 Aug.; Thorbecke, Witting, Kopcnh. - **Grietina, Sprik**, Bergen. NEW-YORK, 8 Aug.; zeilkl.: Ceres, Meuldijk, Antwerpen. RIO JANEIR'), 19 Julij; Maria, Lindquist, Abo. BUENUS-AYRES, 2 Julij; Industrie, Muller, Montevideo.

Table 4: Example of an instance where the last name of the captain appears along with the ship name.

For Experiment 2, on the one hand, we have found that using the years of publication and appearance on the MDB dataset do not yield successful results in terms of precision. Even when the ship name appears in the text, only limiting the links to those of the years we have knowledge of is not enough for record linkage. However, we believe that this domain feature could be used either as a preprocessing step or in combination with more suitable domain features. On the other hand, this experiment yielded new insight into the domain at hand. Another domain feature that we were considering to use for disambiguation was the ship type, even though not all instances contain a ship type in the MDB dataset. However, by analyzing the texts of the candidate links for this evaluation, particularly the ones labeled as irrelevant, we could see that the ship type is not commonly mentioned on newspapers and when it is mentioned, it usually is incorrect or incomplete. This is likely due to the poor knowledge of ships that journalists or people in charge of writing such articles had. This lead us to decide that the ship type is not a domain feature suitable for our research, being discarded for future experiments.
Furthermore, we were able to prove our hypothesis that precision could be improved by combining domain features in one algorithm version. It is important to note that this version of the algorithm was evaluated by 2 raters, obtaining the same precision score of 0.96.
Finally, the mean indicates the average of the ratings of the articles. For the baseline experiment as well as Experiment 2, the instances were labeled on average as 2.37 and 2.58, respectively, indicating that most instances were incorrect. On the other hand, Experiments 1 and 3 scored a mean of 4.62 and 4.80, respectively, suggesting that most instances were correct.

It could be argued that the recall of these algorithms is low. However, we decided to concentrate on obtaining high precision scores because in general, historians would prefer to have fewer but accurate links than the opposite. Improving recall could be the subject of future work.
In general, the results obtained by these experiments are satisfactory and have helped us gain more insight about the domain. These results have also given us an indication of the extent to which we can use domain knowledge for record linkage, thus answering our first research question. We have also obtained valuable manual labels that will be used in the following experiments.

5.2 TEXT CLASSIFICATION EXPERIMENTS

In this section, we describe our experiments on text classification using machine learning algorithms. These experiments are focused on exploring the structure of newspaper articles in order to train a classifier that would be able to predict labels for unseen data. The main difference with previous experiments is that the text classification process is not intended for nominal record linkage on its own. Furthermore, the only distinction between Experiments 4 and 5 is the choice of classifier: Naive Bayes and Support Vector Machine (SVM) with Sequential Minimal Optimization (SMO), respectively.
In the literature, Naive Bayes has been described as a simple and effective classification method [16] and even, as the best classifier for document classification [23]. On the other hand, SVM has been found to be very well suited for text categorization [10], especially with Sequential Minimal Optimization (SMO) [1]. Additionally, both classifiers have been used in previous works similar to this project [27]. Finally, in [22], the authors define these classifiers as the best choice for text processing tasks, reaffirming our selection.
In the following subsections, the approach and details on the data transformation procedure are explained.

5.2.1 Approach and methodology

For these experiments, our method focuses on using off-the-shelf supervised learning algorithms incorporated on our code, to train and evaluate a classifier model and then, using the classifier to predict labels for the remaining unseen dataset. According to Gómez-Hidalgo [8], the process of text classification involves two main steps:

- Representing the text dataset to enable learning and to train a classifier on it and,
- Using the classifier to predict text labels of new, unseen instances.

First, we have decided to use the 200 labeled samples (121 positive and 79 negative instances), obtained during the manual evaluation of the previous experiments, for the sake of convenience. From these samples, we have chosen only the text of the newspaper article and we have converted the 5-point Likert scale label into binary, where labels 1, 2 and 3 are transformed into 0 (negative examples) and labels 4 and 5 into 1 (positive examples). This training data has been compiled in an ARFF file, that can be found on Appendix C. The next step, was to choose and set multiple filters for data transformation. We make use of a multi-filter in order to apply all the chosen filters at once. In the following subsections we go into detail about these filters and their options, given that the results derived from the default settings did not suit our needs.

5.2.1.1 String to word vector We have chosen the StringToWordVector (STWV) [5] filter in order to represent the newspaper texts as feature vectors. For this experiment, we implemented a bag-of-words model, where the frequency of occurrence of each term is used as a feature, ignoring their order in the document [16]. For this model, we would need a term weighting strategy and the STWV filter allows us to enable the option of calculating *(tf-idf)* (6), short for term frequency-inverse document frequency, which considers the frequency of a term on a document but at the same time, attenuates the effect of terms that occur too often in the collection [16]. Typically, the *(tf-idf)* weight is composed by two terms: (a) term frequency *(tf)*, which measures how frequently a term occurs in a document and is normalized using the document length and can be calculated as a logarithmically scaled frequency (4) and (b) inverse document frequency *(idf)*, which measures the importance of the term (5). Also, we have set the minimum term frequency to 2, to ignore words that appear only once in our small training set. Furthermore, we have set to 300 the number of words to be kept in the string vector.

$$\mathrm{tf}_{t,d} = \log(1 + f_{t,d}) \tag{4}$$

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}. \tag{5}$$

$$\text{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t. \tag{6}$$

Additionally, this filter allows us to transform all tokens into lowercase, since we are interested on the words independently of their case. We have decided this has more value for our problem than keeping capitalization, which could be useful for a different kind of problem, e.g. for the detection and extraction of geographical and/or proper names.
Moreover, we have decided not to use stemming because the Dutch stemming algorithms available were not suitable to work directly with our implementation. However, we did make use of the alphabetic tokenizer available in the WEKA library, which allowed us to parse special characters from the text seamlessly.

[5] `http://weka.sourceforge.net/doc.dev/weka/filters/unsupervised/attribute/StringToWordVector.html`

Finally, we wanted to ignore common words, such as function words, for which we have used a stop words list, including numbers and months of the year, based on the stopword lists [6]. The complete list can be found under Appendix D.

5.2.1.2 Remove by name The remove by name filter, removes attributes based on a regular expression matched against their names [7]. This allows us to remove short words that were not ignored by the previous process. We have used the regular expression $(\backslash b \backslash w 1,3 \backslash b)$.

5.2.1.3 Attribute selection In order to use this filter, we first had to transform the features into nominal values, using the NumericToNominal [8] filter. The AttributeSelection is a supervised filter that can be used to select attributes and allows for different search and evaluation methods [9]. For this experiment, we have chosen as evaluation method, InfoGainAttributeEval, which must be used along with a Ranker search algorithm in WEKA.
In WEKA, InfoGainAttributeEval evaluates the worth of an attribute by measuring the information gain with respect to the class, using the information entropy of the class [10]. In a classification context, information gain (7) is a measure of how common a feature is in a given class in comparison to all other classes. A word that occurs mostly on positive examples and fewer times on negative ones, is considered high information [19].
The search method known as Ranker, simply ranks attributes by their individual evaluations [11]. In combination, both methods allow us to obtain a ranked list of string attributes obtained from the text of our training data.

$$\text{InfoGain}_{class,attribute} = \text{H}_{class} - \text{H}_{class|attribute} \quad (7)$$

5.2.2 Training and testing

As recommended by WEKA documentation, the classifier is defined and evaluated but not yet trained. The evaluation is performed using both Naive Bayes and SMO classifiers and consists of a 10-fold cross validation using training data.
Once the Naive Bayes and SMO classifiers have been evaluated, they can be used for learning. The *buildClassifier* class, as the name states, generates a classifier and then, the learned model is stored on disk simply by serializing the classifier Java object. Using our own Java script, we have generated several text files with all unlabeled instances from our dataset, to be used as the test set. The test set contains 413663 instances, divided into 5 files for easier management. The instances keep the same order as in the database, so that the newly labeled instances can be updated accordingly.
As for the testing process itself, first the model and the test set are loaded into the program and every instance on the text file is converted into an actual instance using WEKA's classes *FastVector* and *Instances* subsequently. Then, the *classifyInstance* method returns a prediction, which is then stored in the instance object and subsequently on a text file. Finally, all new labels are imported into a new table for each classifier in the database to make it possible to associate labels to ship instances afterwards.

5.2.3 Evaluation

Given that the evaluation performed by the classifier was based only on training data, we have decided to perform a manual evaluation of 50 random instances with classifier labels, similarly to how we evaluated the previous experiments, so that results would be comparable.
The manual evaluation was performed by means of a survey, presenting to the rater the article text and the categorization given by the classifier in question. The rating task consisted of deciding whether the newspaper text mentions or does not mention a ship or ships. Moreover, instead of a 5-point Likert scale, we have used a binary scale, where 0 means that there is no mention of a ship or ships in the text and 1 means the opposite, that there is a mention of a ship or ships in the text. A sample of this survey can be found on Appendix E.
Once more, the samples have been drawn using SQL queries, found on Listing 4, for each classifier.

```
SELECT id AS "Candidate_ID",
text AS "Text",
label AS "Label_given_by_classifier"
FROM classifier_table
ORDER BY RANDOM()
LIMIT(50);
```

Listing 4: SQL query for Experiment 4 sample

5.2.4 Results

5.2.4.1 Experiment 4 By means of manual evaluation for the labels given by the Naive Bayes classifier, we are able to obtain a precision of 1 and an approximate recall of 0.42. Furthermore, 26 out of 50 instances are relevant and there were no false positives, which means that the classifier did not label incorrect instances as positive. The F1 Score for this experiment is 0.59. The confusion matrix for this experiments is found on Table 5 and the results on Table 7.
By analyzing the labeled data, we can see that about 1/4 of the dataset has been labeled as positive, leaving the remaining 3/4 of the data as negative instances (Figure 1).

[6] Stopwords lists are available at https://code.google.com/p/stop-words/ and at http://www.damienvanholten.com/blog/dutch-stop-words/

[7] http://weka.sourceforge.net/doc.dev/weka/filters/unsupervised/attribute/RemoveByName.html

[8] http://weka.sourceforge.net/doc.dev/weka/filters/unsupervised/attribute/NumericToNominal.html

[9] http://weka.sourceforge.net/doc.dev/weka/filters/supervised/attribute/AttributeSelection.html

[10] http://weka.sourceforge.net/doc.dev/weka/attributeSelection/InfoGainAttributeEval.html

[11] http://weka.sourceforge.net/doc.dev/weka/attributeSelection/Ranker.html

Naive Bayes classifier			
		Predicted class	
		Negative (0)	**Positive (1)**
Actual Class	**Negative (0)**	24	15
	Positive (1)	0	11
Precision: 1		**Relevant documents:** 26	
Approximate recall: 0.42		**F1 Score:** 0.59	

Table 5: Confusion matrix for Experiment 4

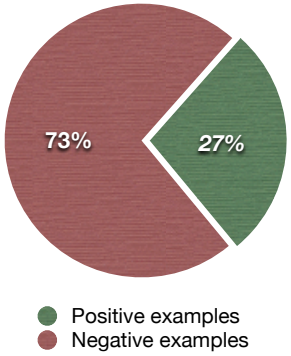

Figure 1: Portion of labeled instances in the dataset by Naive Bayes text classification

5.2.4.2 Experiment 5 The manual evaluation for the labels given by the SMO classifier, results in a precision of 1 and an approximate recall of 0.45. Moreover, 33 out of 50 instances are relevant and, as with the previous experiment, there were no false positives, meaning the classifier has not labeled relevant instances as irrelevant. The F1 Score for this experiment is 0.63. The confusion matrix for this experiment is found on Table 6 and the results on Table 7. In this experiment, 35% of instances are labeled as positive while 65% is labeled as negative, as shown on Figure 2.

SMO classifier			
		Predicted class	
		Negative (0)	**Positive (1)**
Actual Class	**Negative (0)**	17	18
	Positive (1)	0	15
Precision: 1		**Relevant documents:** 33	
Approximate recall: 0.45		**F1 Score:** 0.63	

Table 6: Confusion matrix for Experiment 5

5.2.5 Discussion

In these text classification experiments, there are many details to be discussed. First of all, it could be argued that our

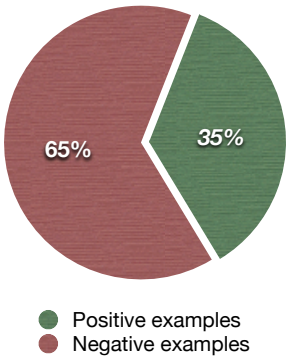

Figure 2: Portion of labeled instances in the dataset by SMO text classification

training set only has 200 instances. While this is true, we believe that the amount of time and man power needed to produce more labeled examples is beyond our resources for this project. Still, our results show that text classification in this domain is an useful approach, regardless of the size of the training set.

As part of our thought process during this project, we had considered compiling a list of terms associated with maritime activity in order to use them as features. However, by analyzing the feature vectors for both classifier models, we have found that the vectors consist mostly of port names (places) and female names, indicating that maritime activity terms are not mentioned on newspaper texts, at least not on common ship mentions.

Finally, our choices for data transformation and classifier configuration seem to be appropriate for the task at hand.

5.3 COMBINING DOMAIN FEATURE FILTERING AND MACHINE LEARNING CLASSIFIERS LABELS

The final version of our algorithm consists of combining both techniques from the previous experiments. More specifically, we have combined our most successful algorithm (See Section 5.1.5) with the labels assigned by the classifiers as an additional feature (See Section 5.2). More details are provided in the following subsections.

5.3.1 Approach and methodology

For these experiments, our method is to restrict the number of candidate links by means of the SQL WHERE clause, as done on Experiments 1-3, explained on Section 5.1.1. As on Experiment 3, the last name of the captain and the restriction of the year of publication of the article are the domain features chosen for filtering. Additionally, we have selected the label provided by the classifiers from Experiments 4 and 5 for positive examples to further filter the candidate links, in hopes of improving our previous results. The only distinction between Experiment 6 and 7 is the choice of label used: labels given by a Naive Bayes Classifier and labels given by SMO classifier, respectively.

5.3.1.1 Experiments 6 and 7 For Experiment 6, the most successful algorithm from previous experiments (Experiment 3 with a precision score of 0.96) is combined with positive labels given by the Naive Bayes classifier. The complete SQL query can be found on Listing 5. Similarly, the algorithm is combined with positive labels given by the SMO classifier for Experiment 7, found on Listing 6.

```
SELECT
distinctcandidates.id AS "C_ID",
distinctships.monsterrollen AS "M_ID",
distinctships.shipname AS "Ship_Name",
distinctcandidates.pdfurl AS "Article_URL",
distinctcandidates.texturl AS "Text_URL",
distinctcandidates.texttitle AS "Text_Title",
distinctcandidates.text AS "Text",
distinctcandidates.texttype AS "Text_Type",
distinctships.shiptype AS "Ship_Type",
distinctships.shipsizes AS "Ship_Sizes",
distinctships.captainlastname AS "Captain_last_name",
distinctships.captainfirstname AS "Captain_names"
FROM distinctcandidates, distinctships
WHERE
distinctcandidates.baselinedsid=distinctships.id
AND text like '%' || captainlastname || '%'
AND extract(year from distinctcandidates.textdate)
>= distinctships.bottomyear
AND extract(year from distinctcandidates.textdate)
<= distinctships.topyear
AND naivebayeseval = 1;
```

Listing 5: SQL query Experiment 6

```
SELECT
distinctcandidates.id AS "C_ID",
distinctships.monsterrollen AS "M_ID",
distinctships.shipname AS "Ship_Name",
distinctcandidates.pdfurl AS "Article_URL",
distinctcandidates.texturl AS "Text_URL",
distinctcandidates.texttitle AS "Text_Title",
distinctcandidates.text AS "Text",
distinctcandidates.texttype AS "Text_Type",
distinctships.shiptype AS "Ship_Type",
distinctships.shipsizes AS "Ship_Sizes",
distinctships.captainlastname AS "Captain_last_name",
distinctships.captainfirstname AS "Captain_names"
FROM distinctcandidates, distinctships
WHERE
distinctcandidates.baselinedsid=distinctships.id
AND text like '%' || captainlastname || '%'
AND extract(year from distinctcandidates.textdate)
>= distinctships.bottomyear
AND extract(year from distinctcandidates.textdate)
<= distinctships.topyear
AND smoeval = 1;
```

Listing 6: SQL query Experiment 7

5.3.2 Evaluation

The evaluation process followed the same procedure as in Section 5.1.6. We have selected a 50 random instance subset by limiting the algorithm using *ORDER BY RANDOM() LIMIT 50;* and evaluated it manually using a 5-point Likert scale ranging from strong disagreement to strong agreement to decide whether the newspaper text shall or shall not be linked to the given ship. A sample of the survey used can be found on Appendix F.

5.3.3 Results

The results for these experiments can be found on Table 7. Experiments 6 and 7 resulted in a precision score of 0.94 and a similar approximate recall, of 0.09 and 0.10, respectively. These low recall scores affect the F1 scores, which are consequently low as well. This is mainly due to the restrictive nature of these algorithms, as evidenced by the number of retrieved links, being the lowest of all the experiments performed for this project.

5.3.4 Discussion

The main goal of these experiments was to further improve precision on the record linkage task at hand. However, the difference between the precision scores of Experiment 3 and Experiments 6-7 is only of 0.02. Therefore, it is not possible to say that the former outperforms the latter in terms of precision, especially considering the weaknesses of manual evaluation. It could be argued that given further evaluation, the precision scores could be equal. Nevertheless, if other measures are considered, such as the approximate recall, it is clear that these algorithm versions do not yield the best results. The restrictive nature of adding more features to the query causes loss of recall given that less links are retrieved. We believe that a more sensible use of the classifier labels, e.g. as a filter before querying instead of it being part of the query, could result in better scores. Still, more testing would be needed in order to decide whether combining both techniques is a suitable approach for this task.

6 FINAL DISCUSSION

In this section, we will report on other interesting findings. Recall we did not use the article titles in the training data for the classifiers. However, after updating the labels in the database, in Figure 3 and Figure 4, it is possible to see a distribution of article titles associated with each of the labels. Although there is some overlapping, it is noticeable that some newspaper sections seem more likely to be associated to ships than others, e.g. titles like "Advertentie" and "Familiebericht" seem to be indicative of texts unrelated to ships while titles like "ZEETIJDINGEN.", "Vreemde Havens." and "Buitenl. Havens." are the opposite. However, it seems likely that given the size of our training set, some article types were not included, resulting in them being classified as false negatives, e.g. titles like "ZEILKLAAR", "Schepen liggende ter reede" and "Carga-Lijsten. Amsterdam" would suggest relation to maritime activity and have been associated to negative examples by the classifier. These findings offer new insight into the domain and could be used as a feature for a different algorithm, e.g. for topic discovery. We believe it could also give an indication of the possible structures within the text articles given the

Results for all experiments						
Algorithm	**Precision**	**Approximate recall**	**F1 Score**	**Links retrieved**	**Mean**(λ)	(σ)
Baseline (Average)	0.23	1	0.37	413863	2.37	1.35
1: Captain's last name	0.90	0.40	0.56	51925	4.62	0.88
2: Year restriction	0.28	0.19	0.23	79113	2.58	1.58
3: Combination of 1 and 2	0.96	0.13	0.23	16037	4.80	0.49
4: Naive Bayes text classifier *	1	0.42	0.59	413663	0.22	0.42
5: SMO text classifier *	1	0.45	0.63	413663	0.3	0.46
6: Combination of 3 and 4	0.94	0.09	0.17	11356	4.82	0.72
7: Combination of 3 and 5	0.94	0.10	0.18	12215	4.84	0.51
KB results in the dataset: 413863						
* Experiments 4 and 5 have been evaluated using a binary scale instead of 5-point Likert scale						

Table 7: Results for all experiments, including precision, approximate recall and F1 scores, numbers of links retrieved, mean (λ) and standard deviation (σ)

distinctions between titles.

Finally, we would like to discuss how the approach of this thesis can be generalized to other dataset linking problems. We believe that a generic approach for record linkage on historical datasets is not feasible, since domain knowledge is needed to identify instances of one specific dataset in a more generic one. However, following an approach similar to ours would assist in linking instances effectively. Some guidelines that should be considered are:

1. Identify domain knowledge features that would allow for instance identification in generic text. For this step, it is important to study both datasets and elicit input from domain experts.
2. Generate queries using the identified features in order to search entities in one dataset on the other. This would retrieve candidate links that should be evaluated to determine if the links are of sufficient quality to be used in practice.
3. Consider using data from the generic dataset to train a machine learned model to perform text classification and annotation. This also helps in identifying new features that can be used for disambiguation.
4. Use the labels in combination with domain knowledge features to retrieve new candidate links and evaluate their precision.
5. Choose the links of the best performing algorithm version and enrich the specific dataset.

7 CONCLUSIONS

In this thesis, we have successfully enriched a dataset of historical Dutch ships by linking its instances to corresponding mentions on newspaper archives. More specifically, we have explored different strategies for record linkage and used both domain knowledge features and machine learning algorithms to link a dataset of historical Dutch ships with their mentions in newspaper archives provided by the National library of the Netherlands.

In order to answer our first research question: How can we automatically enrich a dataset of historical ships by effectively linking instances with historical newspaper archives?, we consider several intermediate questions.

First of all, research question #1 proposes the subject of exploring to what extent we can use domain knowledge for the task of record linkage. We have shown in Section 5.1, that some features taken from domain knowledge, such as the last name of the captain combined with time restrictions, are suitable filters for identifying ship mentions on unstructured text articles. Using domain knowledge features for heuristic rules in this case is highly accurate but depends on the specific domain and requires expertise and time for creation and maintenance of the rules. Our findings on the domain are valuable for other projects with the same or similar background of historical maritime activity.

As for research question #2, we have shown in Section 5.2, that it is possible to use machine learning algorithms for text classification to label instances with a good precision. Even though this approach is not suitable for nominal record linkage, the labels given by trained classifiers are a good feature to be considered for the identification of ship instances in the text. The biggest advantage of using machine learning for this problem is the independence of the domain. However, the main drawback is the need for manual annotation in case of supervised learning algorithms.

Moreover, in Section 5.3, we combine both approaches to answer research question #3. We have come to the conclusion that more testing is needed in order to consider this approach a success.

Overall, we believe that our algorithm versions scoring high precision values provide very valuable links for historians and history enthusiasts that would have otherwise needed many hours of manual search and/or classification by experts. This is important for the accessibility of historic datasets on the Web as well as the preservation of them through time.

Finally, we believe that converting all data into a standard format for the Web, such as RDF, is the best way to facilitate its further use. Even though it has not been included as part

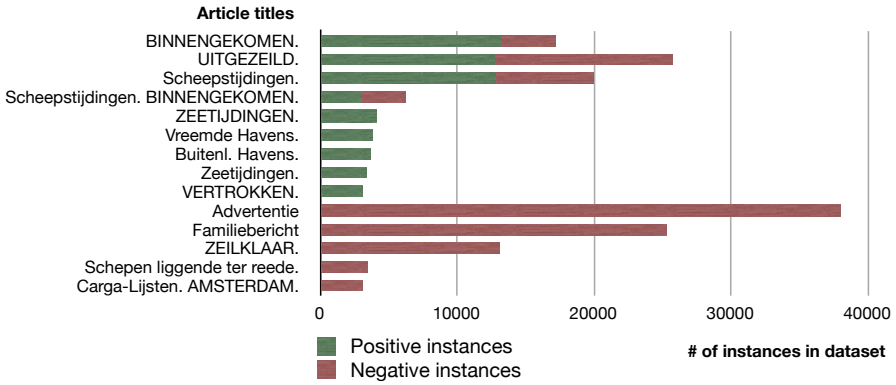

Figure 3: Article titles associated to positive and negative text instances labeled by a Naive Bayes classifier

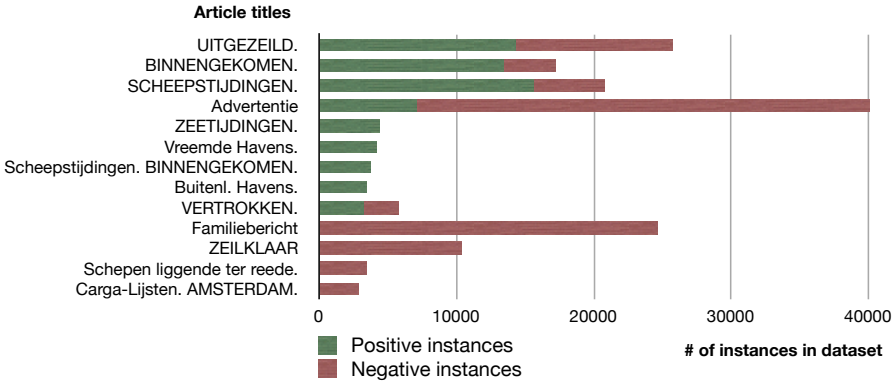

Figure 4: Article titles associated to positive and negative text instances labeled by an SMO classifier

of this thesis, the addition of 16037 links derived from the algorithm described on Experiment 3 has been done as part of the Dutch Ships and Sailors Linked Data by De Boer et al. [5] and it is available at `http://dutchshipsandsailors.nl/`. We have chosen the algorithm version from Experiment 3 given its high F1 Score in comparison to that of Experiments 6 or 7, in addition to the number of links retrieved (16037), compared to 11356 retrieved in Experiment 6 and 12215 retrieved by Experiment 7.

8 FUTURE WORK

In this section, we would like to comment on some ideas for future work.

First, the method we have used to preprocess the datasets could be improved, for example, by using a more expressive language than SQL or by assigning weights to the features we used to identify distinct ships.

Second, we believe that unsupervised learning is also a possibility for text classification tasks. Naturally, there are many other classifier algorithms available in WEKA that could be used for this task, such as the PART algorithm proposed by Frank et al. [7] or decision trees like J48 by Quinlan [20], as well as other machine learning tools available. Additionally, although we experimented with many of the options of the StringToWordVector filter until we found a suitable combination, there could still be a combination of settings that might improve the outcome.

Moreover, it would be interesting to implement a Named Entity Recognition engine to detect female names and/or geographical names for better ship disambiguation. Given the time factor in the dataset, in which geographical places have changed names and associations, we would suggest the use of the Dutch Place Names dataset [12]. Similarly, we believe that newspaper sections and article titles can be used for topic discovery. Additionally, the problem on this project was tackled using an exploratory approach, i.e. testing sensible features and techniques and designing other experiments based on results. After careful examination, we have noticed that executing our experiments in a different order, i.e. using text classification as a filter in the beginning in order to ignore unrelated texts and then using domain features for record linkage, might be a suitable method as well.

[12] `http://www.iisg.nl/hsn//`

9 ACKNOWLEDGMENTS

First and foremost, I would like to express my endless gratitude to supervisor Victor de Boer for the constant assistance and guidance throughout the realization of this project. We would also like to thank Niels Ockeloen for being the second reader of this thesis. Also, many thanks to Jurjen Leinenga for providing the data, as well as important domain knowledge and for his evaluation effort. Furthermore, we are also grateful to the nice people at the KB. Astrid, thank you for encouraging me all the way.
I cannot miss the opportunity to thank my family for supporting me in every sense. ¡Gracias papi, mami, Adri y Ale! Last but not least, I don't think I would be writing this without your support and encouragement. Thank you, Bernardo.

References

[1] AL-SHARGABI, B., AL-ROMIMAH, W., AND OLAYAH, F. A comparative study for arabic text classification algorithms based on stop words elimination. In *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications* (New York, NY, USA, 2011), ISWSA '11, ACM, pp. 11:1–11:5.

[2] BOONSTRA, O., BREURE, L., AND DOORN, P. Past, present and future of historical information science. *Historical Social Research / Historische Sozialforschung 29*, 2 (2004).

[3] BRON, M., HUURNINK, B., AND DE RIJKE, M. Linking archives using document enrichment and term selection. In *Proceedings of the 15th International Conference on Theory and Practice of Digital Libraries: Research and Advanced Technology for Digital Libraries* (Berlin, Heidelberg, 2011), TPDL'11, Springer-Verlag, pp. 360–371.

[4] DE BOER, V., VAN DOORNIK, J., BUITINCK, L., MARX, M., VEKEN, T., AND RIBBENS, K. Linking the kingdom: Enriched access to a historiographical text. In *Proceedings of the Seventh International Conference on Knowledge Capture* (New York, NY, USA, 2013), K-CAP '13, ACM, pp. 17–24.

[5] DE BOER, V., VAN ROSSUM, M., LEINENGA, J., AND HOEKSTRA, R. Dutch ships and sailors linked data. Manuscript submitted for publication., 2014.

[6] DE BOER, V., WIELEMAKER, J., VAN GENT, J., HILDEBRAND, M., ISAAC, A., VAN OSSENBRUGGEN, J., AND SCHREIBER, G. Supporting linked data production for cultural heritage institutes: The amsterdam museum case study. In *The Semantic Web: Research and Applications*, E. Simperl, P. Cimiano, A. Polleres, O. Corcho, and V. Presutti, Eds., vol. 7295 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 733–747.

[7] FRANK, E., AND WITTEN, I. H. Generating accurate rule sets without global optimization. In *Fifteenth International Conference on Machine Learning* (1998), J. Shavlik, Ed., Morgan Kaufmann, pp. 144–151.

[8] GÓMEZ-HIDALGO, J. M. A simple text classifier in java with weka. `http://jmgomezhidalgo.blogspot.nl/2013/04/a-simple-text-classifier-in-java-with.html`, 2013. Retrieved on March 2014.

[9] GOTTIPATI, S., AND JIANG, J. Linking entities to a knowledge base with query expansion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Stroudsburg, PA, USA, 2011), EMNLP '11, Association for Computational Linguistics, pp. 804–813.

[10] JOACHIMS, T. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning* (London, UK, UK, 1998), ECML '98, Springer-Verlag, pp. 137–142.

[11] JURIC, D., HOLLINK, L., AND HOUBEN, G. Bringing parliamentary debates to the semantic web. In *Proceedings of the workshop on Detection, Representation and Exploitation of Events in the Semantic Web (DeRIVE 2012)* (12 November 2012 2012 (to appear)).

[12] JURIC, D., HOLLINK, L., AND HOUBEN, G.-J. Discovering links between political debates and media. In *ICWE* (2013), F. Daniel, P. Dolog, and Q. Li, Eds., vol. 7977 of *Lecture Notes in Computer Science*, Springer, pp. 367–375.

[13] KERN, R., AND GRANITZER, M. German encyclopedia alignment based on information retrieval techniques. In *Proceedings of the 14th European Conference on Research and Advanced Technology for Digital Libraries* (Berlin, Heidelberg, 2010), ECDL'10, Springer-Verlag, pp. 315–326.

[14] KLEPPE, M., HOLLINK, L., KEMMAN, M., JURIC, D., BEUNDERS, H., BLOM, J., OOMEN, J., AND HOUBEN, G. Polimedia: Analysing media coverage of political debates by automatically generated links to radio and newspaper items. In *LinkedUp Veni Competition 2013, Proceedings of the LinkedUp Veni Competition on Linked and Open Data for Education* (2014), vol. 1124 of *CEUR Workshop Proceedings*, CEUR Workshop Proceedings, pp. 1–6.

[15] LV, Y., MOON, T., KOLARI, P., ZHENG, Z., WANG, X., AND CHANG, Y. Learning to model relatedness for news recommendation. In *Proceedings of the 20th International Conference on World Wide Web* (New York, NY, USA, 2011), WWW '11, ACM, pp. 57–66.

[16] MANNING, C. D., RAGHAVAN, P., AND SCHÃIJTZE, H. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[17] MEROÑO-PEÑUELA, A., ASHKPOUR, A., RIETVELD, L., AND HOEKSTRA, R. Linked humanities data: The next frontier? a case-study in historical census data. In *Proceedings of the 2nd International Workshop on Linked Science 2012* (2012), vol. 951.

[18] MEROÑO-PEÑUELA, A., ASHKPOUR, A., VAN ERP, M., MANDEMAKERS, K., BREURE, L., SCHARN-

HORST, A., SCHLOBACH, S., AND VAN HARMELEN, F. Semantic technologies for historical research: A survey. *Semantic Web Journal* (2014), 588–1795.

[19] PERKINS, J. Text classification for sentiment analysis. eliminate low information features. streamhacker.com, 2010. Retrieved on July 2014.

[20] QUINLAN, R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[21] RAO, D., MCNAMEE, P., AND DREDZE, M. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, Multilingual Information Extraction and Summarization*, T. Poibeau, H. Saggion, J. Piskorski, and R. Yangarber, Eds., Theory and Applications of Natural Language Processing. Springer Berlin Heidelberg, 2013, pp. 93–115.

[22] SEBASTIANI, F. Machine learning in automated text categorization. *ACM Comput. Surv. 34*, 1 (Mar. 2002), 1–47.

[23] S.L. TING, W.H. IP, A. H. T. Is naïve bayes a good classifier for document classification? *International Journal of Software Engineering and Its Applications 5*, 3 (2011), 37–46.

[24] STASIU, R., HEUSER, C., AND DA SILVA, R. Estimating recall and precision for vague queries in databases. In *Advanced Information Systems Engineering*, O. Pastor and J. a. Falcão e Cunha, Eds., vol. 3520 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005, pp. 187–200.

[25] TÉLLEZ-VALERO, A., MONTES-Y GÓMEZ, M., AND VILLASEÑOR PINEDA, L. A Machine Learning Approach to Information Extraction. 2005, pp. 539–547.

[26] VIERA, A. J., AND GARRETT, J. M. Understanding Interobserver Agreement: The Kappa Statistic. *Family Medicine 37*, 5 (May 2005), 360–363.

[27] YU, B. An evaluation of text classification methods for literary study. *LLC 23*, 3 (2008), 327–343.

Appendix

A SQL operations for preprocessing the MDB dataset

The MDB dataset needed preprocessing before it could be useful for the purposes of this project, as explained on Section 3.1. The SQL operations performed for the preprocessing task can be found on Listing 7.

```
* Generating an intermediate table, compiling the relevant information from the tables Ships and Crew

SELECT ships.mid,
ships.id AS "shipsID",
crew.id AS "crewID",
ships.date,
ships.year,
shipname,
shiptype,
shipsize,
lastname, firstname, rank
INTO shipstypecaptain
FROM ships, crew
WHERE ships.mid=crew.mid
AND ships.date=crew."date"
AND ships."year"=crew."year"
AND rank = 'kapitein'
ORDER BY ships.id

* Then, generating the distinctships table, parting from the shipstypecaptain table.

SELECT DISTINCT
shipname,
count(*),
array_to_string(array_agg("mid"), ';', '*') AS "monsterrollen",
array_to_string(array_agg("shipsID"), ';', '*') AS "ships",
array_to_string(array_agg("crewID"), ';', '*') AS "crew",
min(year) AS "bottomyear",
max(year) AS "topyear",
shiptype,
array_to_string(array_agg("shipsize"), ';', '*') AS "shipsizes",
lastname AS "captainlastname",
string_agg("firstname", ';') AS "captainfirstname"
INTO distinctships
FROM "public".shipstypecaptain
GROUP BY shipname, shiptype, lastname
ORDER BY shipname

* Adding an id column to the table distinctships

CREATE SEQUENCE bds_seq;
ALTER TABLE distinctships ADD COLUMN id integer default nextval('bds_seq');

* Add subsequent columns

ALTER TABLE distinctships ADD COLUMN requesturl character varying;
ALTER TABLE distinctships ADD COLUMN resultxml character varying(200000);
ALTER TABLE distinctships ADD COLUMN numberofresults integer;
```

Listing 7: SQL query for MDB dataset preprocessing

B Sample of the evaluation survey for baseline and Experiments 1-3

Sample of the survey for the evaluation of the baseline experiment, as well as Experiments 1-3, explained in Section 5.1.6. The survey contained 50 items divided in 10 pages of 5 items each. It would take a rater around 30 minutes to complete. The surveys were managed using Google Drive Forms. The surveys for baseline, Experiment 1, Experiment 2 and Experiment 3 are available online.

Record linkage evaluation form

Welcome and thank you for your time.
This project consists of linking ship records from the Nordelijke Monsterrollen collection (which contains data of ships and ship movements from the northern regions of the Netherlands from 1803 to 1937) with unstructured noisy text, obtained by means of OCR (Optical Character Recognition) from the Koninklijke Bibliotheek newspaper collection.
The texts from the newspaper have been drawn using the following domain knowledge: ship name, the year a ship's record appears in the
Nordelijke Monsterrollen collection plus and minus 5 years. Additionally, we have provided for each instance: the ship type, the Captain's last name and possible first name (including different spellings).
Your task is to decide whether the newspaper text shall or shall not be linked to the given ship. You can use the given information about each ship as well as background knowledge. You may also leave comments on the rationale of your decision for each ship. Please, remember that the text is noisy and there might be misspellings.

Choose
1: If you are completely sure it is not the right link. i.e. it is not about ships.
2: If it is unlikely the given ship and the text ship are the same ship.
3: If it is possible the given ship and the text ship are the same ship but you are not really sure.
4: If it is possible the given ship and the text ship are the same ship.
5: If you are completely sure there is a link between the given ship and the text ship, i.e. the ship is mentioned in the text.

There are a total of 50 items, divided in 10 pages of 5 items each. It should take you about 30 min. to complete.

* Required

1. **1 .- Ship Name: Elsje | Ship Type: kof | Captain's last name: Tap | Captain's first names: Albert K.;Albert K.;Albert K. | Text Type: advertentie** *

   [Advertentie] MANUFACTUREN. Met 1 November kan geplaatst worden een BEDIENDE, van de P. G. In Overijssel of Gelderland in bovengenoemde betrekking geweest zijnde verdienen de voorkeur. Brieven franco Lett. A. Z., brj den Boekhandelaar JAC. VAN DER MEER, te Deventer. (24570) c. g. withüys, Romancen, Verhalen, Vertellingen. Deze Bundel Poëzy bevat uitmunten de Stukken voor de voordracht, als: Huibert van Eyken.—Het Verloren Kind.—De Vrouwe van Stavoren.— Bart en Elsje. — De Kranke. — De Meineerl, euz. Prijs ƒ1.80; rijk geb. ƒ2.25. Uitgave U. J. VAN KESTEREN, Amsterdam. (24556)

   Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

C Sample of the training set for Experiments 4 and 5 in ARFF format

For Experiments 4 and 5, a labeled set was needed to train the classifiers for text classification, as described in Section 5.2.1. The labeled examples were obtained through the evaluation of the baseline as well as Experiments 1-3. The training set contained 200 instances. A sample, featuring 4 instances of the training set can be found below.

```
@relation textEvaluation

@attribute evaluation {1,0}
@attribute text String

@data
1,"Amsterdam aang. 3 Nov. (IJkade) Danae, S, Patras. Stukgoed. Cargad. Hudig, Veder & Co. (Rietlanden) Dallon,
    B, Newcastle. Steenkolen. Cargad. Hudig, Veder & Co. 4 (Houthaven) Sigurd, S. Kroonstad. Hout, Cargad,
    Ruys & Co, (H.kade) Progress. S. Burryport. Steenkolen. Cargad. De Wed. Jan Salm & Meijer, Rynstroom, Ě,
    Huil. Steenkolen eu Stukgoed. Cargad. Holl. Stmbt.-Maatschij. 5 Koster, S. Newcastle. Steenkolen en
    Stukfo. d, Cargad. Hudig. 'eder & Co. IJ stroom, S. Londen. Stukgoed. Carg. Holl. Slmbt.-Maatsehij.
    Eapwing, S, Londen. t3tuks;oed. Carg. Nobel & Holtzapffel. (Houth.) Drammen, S. Christiania. Hout en
    IJzer. Cargad. B. J. van Hengel. Urnuiden 5 Nov. Wind Z.W.,hai _l: wind aangekomen 4 Progress, S.
    Burryport Sigurd, S. Petersburg Rijnstroom, S. . Huil Hermann, S. Archangel 5 Koster, S. Newcastle
    Drammen, S.Christiania IJstroom, S. Londen Lapwing, S, id. Hornfels, S. Riga Emden, S. Goole Orion, S.
    Hamburg vertrokken 4 Waterland, S. Amble Milo, S. Bristol Alster, S, Hamburg Tem, S. Londen P/ulo, S.
    Malaga BŔta, S. West-Hartlepool De loodskotters zijn binnen; de stoomloodsboot doet loodsdien. t.
    Oostmalioru aang. 3 Nov. Morgenster, S.domons, Elmshorn Morgenster, Salomous, id, VJle aang. 4 N'ov.
    Konfid, Schuitema, Helsingborg Herzogm Ingeborg, Schuitema, id. 5 Swan, S. Huil vertrokken 5 Teal, S.
    Londen Delfzijl aang. 3 Nov. Hunze IX, S. Hamburg Delfzyl, De Boer,Nordeu Goede Verwaehtiny, Leertouwer,
    id. Antilope I, Jonker, Emden Eemstroom, Engelsman, id, 'fiere Gebroeders, Matroos, id, Ebenhaezer, Mlnke
    , Bmden vertrokken 2 Albach 2, Bos, Leer Albach 12, Weber, id. Catharina-, v. (1. Wal, Termunterzjjl
    Charloltenburg, Kruize, Nieuweschans Broederlromc, Smit, EmdenConfianee, Voorde wind, id. DanJcbaarheid,
    v. d. Werf, id. 3 Eecrdina, Groenhagen, Tormunterzijl Pelrolea, Coop, Nordham De Hoop, Houwing, Norden
    Zeemeeuw, Voordewind, id. Sieberdina, Davids, id. Goede Verwachting. Houwiug, Emden Broedertrouw, Smook,
    id, Appingcdam, ?. ãysÈiels, id. Ebenhaezer, De Boer, Aurich Vijf Gebroetiers, Kajuiter, Borkum Twee
    Gezusters, Wolthuis, id. Cr'SKSliayen, S. Blyth (Verbetering) Hellevoetsluis 5 Nov. aangekomen Hedvnng, S
    . Archangel met hout naar Dordt. __aa__iulN 5 Nov. Wind Z.W., harde wind: aangekomen 3 StelŢa, S. Danzig
    Jeannctte T. oer_>_Ç_>!, S. Hamburg 4 Rio Retzlaff, S. Huelva, Unita, S. Narvik Sheafield, S. Newcastle
    Saturn, S. Hamburg Elbing 11, S. Memel Lion, S. Sunderland Ella, S. id. Vienna, S. Harwich Batavier IV, S
    . Londen Blanchland, S. Newc&stle Outensland, B, Poti Hermina, S. Kilig's Lynn Import, S. Londen Clacton,
     S. Harwich Sylvia, S. Patras Vmea, S, Wyborg Albia, S, Bilbao Clio, S. Aqnilas Swift, B. ;Hull
    Gelderland, S. Newcastle Xicolas, S. Grimsby Stav.ete.sj, S. id. Oberhauscn, S. Hamburg Arie/, S.
    Hudikswall 5 Amiral VHermite, 8. Duinkerken Circe, S. Caen Camil/e, S. id. Eh-e, S. Hamburg Ohio, S.
    Baltimore Dresden, S. Harwich Motala, S. HernŽsand Btarting, S. Londen Wharfe, S. Goole Rotterdam. S.
    Graugemouth Ac7a, S, Sulina Çer iroifceii 3 Mannheim, S. Petersburg _4. >/ Scfiejfer, S. Havre Rhein, S.
    Hamburg Rein/eld, S. id. Hebbic, S. Goole Automaat, S. Londen TheŮry, S. Southampton Bittern, S. id, C'
    o_s_ Jiwttii-icos, S.Carditf Grele Cords, S. Goole Amsterdam, S. Harwich 4 Swallow, S. Huil Xorthenden, S
    . Grimsby 2lieodora, S. King's Lynn Apollo, S. Bristol Glenmore. S. Middlesbro iVhimbrell, S. Liverpool
    Woodcock, S. Londen Ariadne, S, Hamburg Lerla, S, Danzig Gibraltar, S, Lissabon Lord Doicnshire, S.
    Cardiff Bucuresli, S. Ibrail Jabiru, S. Liverpool ___nlu_;_s S, Wilhelmshaven Yarmouth, S. Harwich Pena
    Caelillo, S. Santandi r Vienna, S. Harwich Vlissingen 5 Nov. Wind Z.W,, harde wind; gepasseerd Golha, S.
    Gothenburg Markomannia, S. Hamburg beid- u. Antwerpen ts uit zee teruggekeerd Ciudad diAmberes, sleepboot
    , BraziliŚ Ternenzen aang. 4 Nov. Rtver Lagan, S Londen 5 Jet, S. Newcastle Carl Menzell, S. Riga
    vertrokken 8 Atiw Sehetdt, S. Londen Z/ruEa, S, Goeie"
0,"KONINGSBERGEN, H. J. Hazewinkel: Arendina Harmina: 2825 schepels Graauwe Erwten, Order."
1,"TEXEL, HEDEN Vrijdag ochtend, 16 Sept.; binnengekomen: Aidina Anna Susanna, Schenk, Nickerie. - Anna en
    Arnoldina,van Wijk, Saramacca. LONDEN, HEDEN Vrydag 16 Sept. Consols op tijd 95} a}; Gren. Uitg. 8j a J;
    Buenos-Ayres 64 a 66; Spanje 1 pCts. 22* a } ; Dito j Certif. 5} a }. Overige onveranderd , maar vast. Dc
     beurs vertoont neiging tot ver. j dere rijzing. mm -f"
0,"HULL, Enchantress, T. Farr. Ñ 5525 stav. en 1897 boss. IJzer, E. S. de Jonge. HULL, Ocean Queen , C. Hardy.
     - 633 boss. IJzer, E. S. de Jonge. STOCKHOLM , Albertina, Lever. Ñ 2615 staven en 96 boss. IJzer, E. S.
    de Jonge. STOCKHOLM, Cycloop, Takes. Ñ 485 stav. en 6 boss. IJzer, E. S. de Jonge."
```

Listing 8: Sample of the training set in ARFF format

D Stopwords list used for Experiments 4 and 5

For Experiments 4 and 5, a stopword list was needed in order to ignore function and other unnecessary (for our purposes) words, as explained in Section 5.2.1.1. Below is the list we have used for our experiments.

```
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, aan, acht, af, al, alle, alles, als, altijd, andere, april, augustus,
b, ben, bij,
c, d, daar, dan, dat, de, december, der, deze, die, dit, doch, doen, door, drie, dus,
e, een, eens, en, er,
f, februari,
g, ge, geen, geweest,
h, haar, had, heb, hebben, heeft, hem, het, hier, hij, hoe, hun,
i, iemand, iets, ik, in, is,
j, ja, januari, je, juni, juli,
k, kan, kon, kunnen,
l,
m, maar, maart, me, mei, meer, men, met, mij, mijn, moet,
n, na, naar, negen, niet, niets, nog, november, nu,
o, of, oktober, om, omdat, ons, ook, op, over,
p,
q,
r, reeds,
s, september,
t, te, tegen, tien, toch, toen, tot, twee,
u, uit, uw,
v, van, veel, vier, vijf, voor,
w, want, waren, was, wat, we, wel, werd, wezen, wie, wij, wil, worden,
x,
y,
z, zal, ze, zei, zes, zelf, zeven, zich, zij, zijn, zo, zonder, zou
```

Listing 9: Stopwords list

E Sample of the evaluation survey for Experiments 4 and 5

As explained in Section 5.2.3, Experiments 4 and 5 were manually evaluated by means of a survey which contained 50 items divided in 10 pages of 5 items each. It would take a rater around 30 minutes to complete. The surveys were managed using Google Drive Forms. The surveys for Experiment 4 and Experiment 5 are available online.

Text evaluation form

Welcome and thank you for your time.
This project consists of linking ship records from the Nordelijke Monsterrollen collection (which contains data of ships and ship movements from the northern regions of the Netherlands from 1803 to 1937) with unstructured noisy text, obtained by means of OCR (Optical Character Recognition) from the Koninklijke Bibliotheek newspaper collection.
Your task is to decide whether the newspaper text mentions or does not mention a ship or ships.
Please, remember that the text is noisy and there might be misspellings.

Choose
0: If there is no mention of a ship or ships in the text.
1: If there is a mention of a ship or ships in the text.

There are a total of 50 items, divided in 10 pages of 5 items each. It should take you about 30 min. to complete.

* Required

1. **1)** *
   VLISSINGEN den 29 october. Gisteren en heden zijn, voor Antwerpen bestemd, op onze reede aangekomen: De Sirene, kapt. J. G. Kruger, van Pillau, met weedasch en lijnzaad; Freude Broder, kapt. B. Petersen , van Riga, met raapzaad; Catharina, kapt. H.Heeren, van Carolinenseel, met garst; de Jonge Frederik, kapt. C. Stuhl, van Rusterseel, met haardasch; Agneta Maria, kapt. H. A. Molm, van Nyborg, met raapzaad; Marianne Pauline, kapt. A. Mahlman, van Busem, met garst; Margaretha, kapt. C. Stehr, van Hamburg , met haver. Nog is alhier ter reede gekomen Anna Adelkeid, kapt. G. J. Wesjeiing,, van Bergen naar Leuven gedestineerd, met stokvisch. Ook zijn sedert den 26 dezer van deze reede naar zee gezeild : Van Vlissingen, Linne von Udewalle , kapt. A. Lundberg, naar Cadix. Van Brussel, Lisetta Engelina, kapt. H. L. Rotgers, naar Papenburg, metsteen; Panama, kapt. B. Freeman, naar Newcastle, en the Billom, kapt. J. Bogardns, naar Roehelle, beide met ballast; de Jonge Johanna , kapt, J. Verbruggen , naar Londen , met boomschors , en de Eendragt, kapt. G. Frantzen , naar Keulen , met stukgoederen. Van Antwerpen, la Caroline , kapt. L. Jouet, op avontuur, met ballast; Carolina, kapt. E.Janssen, naar Carolinenseel, met stukgoederen;; Regina, kapt. O. L. Ketelbotter, naar Koppenhagen, met ballast; VF.s-perance, kapt. A. van Geyt., naar Londen, met boomschors, Helena, kapt. A. J. Ricke, paar Embden, met steen; de Jonge Johanna, kapt. t> J. Ricke, naar Yarmouth, met boomschors; Harriet c? Jane, kapt. . A. Hoeve, naar Arbroath , met vlas; Johanna , \\a\\pi. S. Evers, en de Stad tingen, kapt. Th. Schipman , beide naar Bordeaux ; Josephine, kapt. F. Rustèr, en Industrie, kapt. H. L. Rehbock, beide op avontuur en alle vier met ballast; John & Catharina, kapt. H. Ord , naar Huil, met vlas; Commerce, fcapt. A. Carpels, naar Londen, met boomschors; Wenskabe, kapt. A. Land, naar Noorwegen, en Gude Wennrr, kapt. H.M. , Mortensen, naar Mortnezer, beide met ballast; de Lodewyk, kapt. A. E. van Dijck, naar Pennray ,en Dispath, kapt.T. Jackson , naar Yarmouth, , beide met boomschors; die Hofnung, kapt. A. H. Scheepman, naar Embden, en de twee Gebroeders, kapt. T. Sonnichsen , naar Cuxliaven, beidej met stukgoederen.; Waarborg, kapt. N. Jansen, naar Newcastle, met ballast; de Vrouw Hendrika, kapt. S. Gelsenia ; de Vrouw Gezina , kapt. J. H. Bischop; de Vrouw Gebina, kapt, M. D. Gerdes ; die Hofnnng, -kapt. J. D. Ihider; de Herstelling , kapt. L. E. Gust; de Kleine David, kapt. j. H. Jansen ; de drie Gebroeders, \'kapt. E. Alberts, en de Vrouw Catharina, kapt. J. G. Juister, alle acht naar Embdenr met ballast. VERE den 25 october. Heden zijn gezeild de Engelsche brikschepen Janet en Bilbao., kapiteins J. Elliot en W.Roberson, beide van Middelburg naar Sunderland, met ballast.
   Mark only one oval.
   - ◯ 0
   - ◯ 1

F Sample of the evaluation survey for Experiments 6 and 7

As explained in Section 5.3.2, Experiments 6 and 7 were manually evaluated by means of a survey which contained 50 items divided in 10 pages of 5 items each. It would take a rater around 30 minutes to complete. The surveys were managed using Google Drive Forms. The surveys for Experiment 6 and Experiment 7 are available online.

Record linkage evaluation form

Welcome and thank you for your time.
This project consists of linking ship records from the Nordelijke Monsterrollen collection (which contains data of ships and ship movements from the northern regions of the Netherlands from 1803 to 1937) with unstructured noisy text, obtained by means of OCR (Optical Character Recognition) from the Koninklijke Bibliotheek newspaper collection.
The texts from the newspaper have been drawn using the following domain knowledge: ship name, the year a ship's record appears in the
Nordelijke Monsterrollen collection plus and minus 5 years. Additionally, we have provided for each instance: the ship type, the Captain's last name and possible first name (including different spellings).
Your task is to decide whether the newspaper text shall or shall not be linked to the given ship. You can use the given information about each ship as well as background knowledge. You may also leave comments on the rationale of your decision for each ship. Please, remember that the text is noisy and there might be misspellings.

Choose
1: If you are completely sure it is not the right link. i.e. it is not about ships.
2: If it is unlikely the given ship and the text ship are the same ship.
3: If it is possible the given ship and the text ship are the same ship but you are not really sure.
4: If it is possible the given ship and the text ship are the same ship.
5: If you are completely sure there is a link between the given ship and the text ship, i.e. the ship is mentioned in the text.

There are a total of 50 items, divided in 10 pages of 5 items each. It should take you about 30 min. to complete.

* Required

1. **1 .- Ship Name: Rensina | Ship Type: NULL | Captain's last name: Mulder | Captain's first names: Abraham Klasens;Abraham Klaassens | Text Type: artikel** *

   [NOT FOUND] TEXEL , 4 April, West;' T. B. Muister , Christina , Huil. - D. de Jong, de jonge Clement, id. VLIE, 3 April, West; B. Molenaar, Vr. Stientje , de Oostzee. - H. T. Bieze , Anna , Koningsb - J. J. Gocsens , Oudewerf, Stavanger. -J. E. Ebeling , Antonie , Drobach. -R. C. de Groot, Eendragt, Ostcrrisoer' -T. H. de Jong, Argo. Noorwegen - A. K. Mulder, Rensina , Hamb. - W. D. Dekker, Alida, op Avontuur. - K. A. Tap, Maria Beertha , id. - A J. Verloe, jonge Jaeob, id. -J. N. van Duinen , Alkanua Elisabclh , id. - t\ 11, Fokkïus, Gcsjua Calhariu» Brons, id. HELVOETSLUIS , 4 April, W. _. W. ; T. B. Center, het V, mouwen , Steltin. BATAVIA, 6 Dcc; Reiniersen , Formosa, Rot. — 9 Dec. Mugge, de Zwijger, Dordt.— (zeilklaar) 12 Dcc, Veening, Prins Hfcudrik, Amst. - Sipkes , 3 Vrienden, id. _.$£- _»-»(t-<-__£È TRIEST; 24 Maart, (zeilklaar): Classen , Freija , Rott./™*^

   Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree