DEUTSCHE BUNDESBANK
EUROSYSTEM

Linking Deutsche Bundesbank Company Data using Machine-Learning-Based Classification

Technical Report 2017-01

- This report refers to RDSC Company Linkage Table Version 8 -

Christopher-Johannes Schild
Simone Schultz
Franco Wieser

Citation:
Schild, C.-J., Schultz, S. and F. Wieser (2017). Linking Deutsche Bundesbank Company Data using Machine-Learning-Based Classification. Technical Report 2017-01, Deutsche Bundesbank Research Data and Service Centre.

Content

Abstract

We present a method of automatically linking several data sets on companies based on supervised machine learning. We employ this method to perform a record linkage of several company data sets used for research purposes at Deutsche Bundesbank. The record linkage process involves comprehensive data pre-processing, blocking / indexing, construction of comparison features, training and testing of a supervised match classification model as well as post-processing to produce a company identifier mapping table for all internal and public company identifiers found in the data. The evaluation of our linkage method shows that the process yields sufficiently precise results and a sufficiently high coverage / recall to make full automation of company data linkage feasible for typical use cases in research and analytics.

1 Introduction

Within Deutsche Bundesbank, various departments independently collect microdata on companies for different analytic and reporting purposes, data which often refers to the same real-world entities. Due to the still insufficient use of common internal or public identifiers, most of the data cannot be linked through unique common IDs.[1] However, especially since the financial crisis, the need for the integration of these separate financial microdata has increased. In this paper, we describe the results of a pilot project to automatically link Bundesbank company data using alternative variables, such as names, addresses and other variables. The produced ID-linkage table enables researchers to easily link our company data for example to enrich data on firms' foreign subsidiaries with firm-level balance sheet data.

When data sources do not have common unique identifying keys, other variables such as names and addresses have to be used to identify different representations of the same real-world unit. Alternative identifiers such as names are often erroneous, and usually it is not possible to fully standardize these variables. They therefore have to be compared using computationally costly similarity measures such as string distance metrics. However, since the number of comparisons depends quadratically on the number of units in the datasets, total computation costs can be very large. The number of comparisons therefore has to be kept under control, which usually requires domain specific indexing / blocking rules. In the presence of training data, an automatic classifier such as a supervised machine learning classification algorithm can then be used to predict the match probability for a list of match candidate pairs.[2]

We first briefly describe the different Bundesbank data sources that enter the record linkage in section 2. The third section describes data cleaning and standardization steps.[3] In the fourth section, we construct suitable comparison features, most importantly from firm name and address information, but also from other variables such as economic sector classification, legal form information and balance sheet positions. In the fifth section, we demonstrate our strategy of limiting the match candidate search space through blocking / indexing. In the sixth section, we construct a "ground-truth" subset of known matches and non-matches in order to be able to train and validate our classification model based on this ground-truth data. In the seventh section we present the process of selecting and cali-

[1] Currently there are efforts in place to establish more widely used unique common company identifier keys that will help to identify more firm entities as previously the case, such as the „LEI" of the Global LEI Foundation or the ECB RIAD-ID. Once the use of these common identifiers is widely established, newly incoming data on companies that enters organizations such as the Bundesbank will have such an ID attached to them, which will make it relatively easy to correctly assign this incoming information to the right database entry, if the key of these companies has already entered the database. It will however not solve the linkage problem for those companies that were already in the databases before the respective key was used. It will also not solve the linkage problem for those firms that simply are not assigned such a key for example because a relevant legal requirement is not met or they decide not to register for such a key. As of March 2017, less than 50,000 legal entities in Germany have been assigned a legal entity identifier (LEI).

[2] An ideal automatic match process of this kind finds all different representations of every entity for which at least one representation exists in the data, and assigns a common identifier value to all of these representations, while never assigning the same ID value to two representations that in fact do not refer to the same entity.

[3] Data processing is done in SAS, except for the machine learning based classification which is done using the Python Scikit-learn library (Pedregosa et al. (2011)).

brating a supervised classification model which uses the derived comparison features to make a match prediction for each match candidate pair. In the eighth section we evaluate the linkage process and the resulting linked dataset using an “unseen” hold-out data set of known matches and non-matches that was not used for training the classification model. The ninth section describes rules of post-processing the model’s match predictions to produce a final ID matching table. Finally in the tenth section, we attempt to manually evaluate the final match result using a random subsample of the final matching table.

2 Data on Companies

Seven data sources on non-financial companies enter the record linkage:

- AWMUS - Foreign Statistics Master Data on Companies[4]
- BAKIS-N - Bank Supervision Master Data on Borrowers[5]
- EGR - Data from the DESTATIS company register on European company groups[6]
- USTAN (JALYS) - Balance Sheet Data[7]
- KUSY - Deutsche Bundesbank Kundenssystematik
- DAFNE - Bureau Van Dijk / Dafne - Balance Sheet Data[8]
- HOPPE - Hoppenstedt / Bisnode - Balance Sheet Data[9]

In this version (Version 8) of the linkage, only recent years of available master data are used. The latter two datasets are balance sheet data from external data providers. They are acquired by the Bundesbank in order to complement balance sheet information collected by the Bundesbank in USTAN / Jalys. The EGR (“Eurogroups Register”) data is company data from Eurostat and the German Statistical Office “DESTATIS” that entered the linkage process due to the requirement to link the foreign direct investment information contained in AWMUS, which is a database that contains master data and analytical data on companies required to hand in reports on foreign payments and foreign direct investments. BAKIS-N as part of the Bundesbank's prudential information system contains data on borrowers for large credits. Finally the “KUSY” is a well maintained reference dataset of the largest German corporations provided by Deutsche Bundesbank.

3 Data Cleaning and Harmonization

At the variable level, all variables present in the different databases, not only master data, but also analytical variables such as balance sheet information, were interpreted, mapped to the variables of the other databases, and theoretically evaluated with respect to their potential to contribute information about the identity of the firm (“discriminative power”), alone or in combination with other variables. If a variable was potentially informative for contributing to identifying a firm and available for more than

[4] See Schild et al. (2015) and Lipponer (2011). MiDi and SITS are integrated into AWMUS.
[5] See Schmieder (2006).
[6] http://ec.europa.eu/eurostat/statistics-explained/index.php/EuroGroups_register
[7] See Stoess (2001).
[8] See https://dafne.bvdinfo.com.
[9] See http://www.bisnode.de. As a first step, only the most recent master data was used for the linkage.

one dataset, it was included in the process. Those variables were then standardized with respect to the variable name and variable label as well as with respect to the value level, if feasible. Standardization on the value level consisted in standardization of value meanings (codelists) for categorical variables and similar scales and units for continuous variables.

Firm names in the Bundesbank databases originate from either paper or electronic forms submitted to the Bundesbank. Their quality depends on a number of different factors, such as the design of the data entering interface and the frequency and quality of manual or automatic cross-checks with other data sources. Errors common to firm data are present also in the Bundesbank firm data, most notably non-harmonized abbreviations as well as uninformative insertions of name components of different kinds, and typing errors (single letter insertions, deletions etc.). For the firm name fields, data cleaning involves removing known variation in different correct notations, such as standardizing the German word „Gesellschaft“ to its most common abbreviation „Ges“ and “&”, “+”, “und”, “and” etc to “UND”. It also involved replacing German Umlauts „ä, ö, ü“ by their common non-Umlaut replacements „ae“, „oe“, „ue“ as well as capitalizing.

Next to the legal form information that could be extracted from the firm name field (see the section on derived field comparison features in the appendix), most databases also included a coded variable for the legal form. The codelists for this original legal form information differ vastly, however, and could only be harmonized to very coarse categories, with some values remaining non-assignable even to the defined common, coarse standard. For other string variables next to the firm name (for example the city and the street name), field cleaning was restricted to truncating after the 6th letter. Balance sheet data that was used for constructing comparison features was standardized to be denominated in 1,000€. External firm IDs present in more than one dataset were standardized if they followed different conventions in the different databases (such as the case for legal form abbreviations in the trade register number). For some variables, values could be imputed from other variables. For example, this was the case for the *Euro Group Register ID*, the “LEID”, which in the case of German firms can be constructed by combining the country code, a local trade register court ID and the local trade register firm ID.[10] Likewise, a synthetic ID was generated by concatenating the trade register ID with the postal code (“trade register / postal code-ID”).[11]

Other variables that were included in at least two datasets and were potentially informative (and therefore had to be harmonized) including the founding year or exact founding date of the firm, insolvency dates, the number of employees as well as telephone contact numbers.[12]

[10] The LEID number is composed of the two-letter country code, the internal register code assigned to the national register by the Euro Group Register (referred to as the national identification system code or NIS code) and the legal unit’s national identification (national ID) number, as assigned by this same national register.
[11] This generic ID served only to generate additional training data with respect to (quasi-)true positive matches, whenever it was not possible to generate a Euro Groups Register LEID. Combining the trade register ID with the postal code is by design not suitable to derive ground truth data about true non-matches (since there are likely too many falsely mismatching postal codes in the set of true matches).
[12] Economic sector information was not yet included (planned for the next revision).

Finally, a "raw data ID" was assigned based on unique (over all datasets) combinations of the cleaned firm name (i.e. after umlauts were exchanged and after capitalization but before legal form features were extracted), the city, and the postal code, in order to refer to different original representations in the data throughout the linkage process.

4 Field Comparison Features

The most important field to distinguish the different firm entities is the firm name field. According to IHK Köln (2012), the company name can be "any family name, term or any freely chosen name and may consist of several words […] has to comprise the legal form of the company […] may include company slogans […] has to be suitable to 'identify the registering trademan' and has to have 'discriminatory power'." Discriminatory power can be achieved by adding further name components, such as adding a geographical component to the company name.

Extraction of legal form information

The most important feature generated from the cleaned firm name is the *legal form information* encapsulated in the cleaned firm name. To detect the many different ways to spell legal form information, a large set of regular expressions was developed, repeatedly tested and improved until more than 90% of the legal forms were detected correctly. The feature derivation then involved mostly extracting the legal form information from the original firm name, saving it in a separate variable, and generating another version of the cleaned firm without the legal form patterns. The following categories were extracted: "GmbH", "AG", "SE", "KG", "OHG", "UG", "GbR", "e.V.", "e.G.", "KGaA", "V.a.G.", the most frequent foreign legal forms present in the data, such as the UK "Ltd." or the Dutch "BV", as well as most legally possible combinational constructs (shown for the GmbH) "GmbH & Co. KG", "GmbH & Co. KGaA", "GmbH & Co. OHG" (similarly for "AG", for its European equivalent "SE", "UG" as well as for the most frequent foreign forms). These detailed legal form categories that were previously unavailable in any of the datasets could subsequently also be used for analytical purposes. The different regular expressions to detect and extract these legal forms are presented in the appendix.

The main comparisons features that we derive from the firm name are based on a Levenshtein distance, a "Generalized Edit Distance" and a name-token based Soft-IDF measure.[13] All three measures were used to calculate distances between firm names using the original, non-standardized firm name, the standardized firm name without legal form information as well as the standardized firm name up to the position where the detected legal form information begins within the firm name.

One problem for firm name based data linkage is the often vastly differing informational value of a name's single components. For example, consider the following three (fictional) firm names:

1. QP**B** Immobilien Verwaltung und Vertrieb Gesellschaft mbH

[13] For an overview on string comparison algorithms see Cohen et al. (2003) and Christen (2012a), pp. 103-105. The "Generalized Edit Distance" was used as implemented in SAS by the function "COMGED". This measure punishes for insertions, deletions, replacements differently based on a cost function that was derived from best practice experiences in general data matching tasks. Towards the beginning of the firm name it is less tolerant to changes than towards the end of the firm name. The measure "Soft-IDF" is described below.

2. **Y**P**P** Immobilien Verwaltung & Vertrieb GmbH
3. QP**B** Immobilien Verw. und Vertrieb Ges. mbH

After standardization and legal form extraction, these names are changed to:

1. „**Q**P**B** IMMOBILIEN VERW**ALTUNG** UND VERTRIEB“ (name) and „GMBH“ (legal)
2. „**Y**P**P** IMMOBILIEN VERW**ALTUNG** UND VERTRIEB“ (name) and „GMBH“ (legal)
3. „**Q**P**B** IMMOBILIEN VERW UND VERTRIEB“ (name) and „GMBH“ (legal)

Between 1 and 2, a simple Levenshtein edit distance (counting the minimum number of character edits required to transform one string into the other) would count two required edits (both in the first token). Between 1 and 3, Levenshtein distance evaluates to 8. While a human classifier has no trouble deciding that 1 and 3 are very likely a match and that 1 and 2 as well as 2 and 3 are very likely not a match (since “VERW” is very likely an abbreviation of “VERWALTUNG”), a Levenshtein metric would rank a match between 1 and 2 highest.[14] One problem with most string comparison algorithms such as the Levenshtein distance is that they usually do not incorporate the possibility that different tokens may have a very different amount of discriminative power.[15]

To deal with this challenge, we employ the measure Soft-IDF, i.e. Soft Inverse Document Frequency. This measure first applies a “soft” comparison (meaning some error tolerant string comparison[16] between single tokens) of each token with all tokens of the other firm name, and weighs the calculated token-level scores with the inverse (log) frequency with which the respective tokens appear in the entire universe of firm names found in all datasets (inverse document frequency). With respect to the examples above, the tokens 2 to 5 then all enter with a very low weight, since the tokens “IMMOBILIEN”, “VERWALTUNG”, “VERW” (as a very common abbreviation of “Verwaltung”) as well as “VERTRIEB” are all very frequent tokens in the universe of firm names. The first components of the above examples are however very rare, and therefore enter the SoftIDF similarity measure with a large weight.[17]

To avoid losing a potential match candidate when one token is missing or an additional one is added, blocks of token combinations have been generated. For that, the name tokens with no more than 5 characters have been combined among each other, starting with token 1 and 2, then 1 and 3, …, 2 and 3, … (of the previously described tokens), and so on. Here the order has not been changed (i.e. there is no combination with leading token 2 followed by token 1). These blocks have not only be used for the indexing procedure, but they also served – besides the sole tokens described in the previous paragraph – as a base for a second Soft-IDF. This captures especially cases with few tokens where an important but frequent token can be then found in combination with a less frequent but discriminating token. Again, a “soft” string comparison has been employed.

[14] The three tokens „Immobilien Verwaltung und Vertrieb“ translate to „Real Estate Management and Sales“.
[15] Levenshtein (1966), Christen (2012a), pp. 103-105.
[16] here: truncation to the first five characters.
[17] Cohen et al. (2003).

A few balance sheet items were available not only in the three balance sheet datasets USTAN/Jalys, Hoppenstedt and Dafne, but also for some of the companies in the MiDi:

- Finanzanlagen *(financial assets)*
- Gezeichnetes Kapital *(subscribed capital)*
- Umsatzerlöse *(revenues)*

For each of these 3 items, a comparison feature was generated as the percentage difference between two records:[18]

$$pc = \frac{|n_i - n_j|}{\max(|n_i - n_j|)} \cdot 100,$$ [19]

where n denotes a balance sheet item found in two datasets i and j. The same procedure was used for the *number of employees*, which was available for USTAN/Jalys, Dafne, AWMUS/MiDi and the Euro Groups Register Data. For the codified legal form (harmonized to a coarse common classification), founding years and insolvency dates, binary indicators of exact identity have been generated. Similar values for external identifiers have not been used as features; from these, we generated "ground-truth" data for training the classifier and for final validation (see the section on the derivation of ground-truth data). Table 1 provides an overview for the availability of variables suitable for comparison for each dataset as well as the derived comparison features that entered the process.

Table 2: Comparison Features[20]

	USTAN /JALYS	**DAFNE**	**HOPPE**	**AWMUS /MiDi**	**BAKIS-N**	**EGR**	**KUSY**	***Comparison feature(s)derived***
Firm Name Features	+	+	+	+	+	+	+	*[see section "Name Features"]*
Codified Legal Form	+	+	+	+	+	+	+	*identical 0/1*
Postal Code	(+)	+	+	+	+	+		*identical 0/1*
City	+	+	+	+	+	+	+	*first 6 letters 0/1*
Street	(+)		+	(+)		+		*first 6 letters 0/1*
Telephone Contact	(+)		+	(+)				*first 7 digits 0/1*
Founding Year	+	+	+			+		*identical 0/1*
Insolvency Date		(+)			(+)	+		*identical 0/1*
Nr of Employees	+	+		+		+		*< 10% difference 0/1*
Consolidated Balance Sheet Indicator	+	+	+					*If this indicator is 0 then balance sheet are compared*
Financial Assets	+	+	+	(+)				*< 5% difference 0/1*
Subscribed Capital	+	+	+	(+)			+	*< 5% difference 0/1*
Revenues	+	+	+	(+)				*< 5% difference 0/1*

[18] See Christen (2015).
[19] Christen (2012a), p. 121.
[20] Brackets indicate a very large share of missing values.

5 Indexing

A complete classifier compares every representation found in the datasets with every other representation in the datasets, since it is not known ex ante, which pairs can be ruled out to be a match. For s datasets with n representations each, this would however result in a total number of N = (n * n) * (s-1) comparisons; the number of comparisons increases quadratically with n. In our case with 6 datasets and about 200.000 unique representations in each (see the frequency counts in the last section), this yields about 200.000.000.000 theoretical possible matching pairs. It would be practically impossible to apply computationally costly comparisons on these.[21]

In order to apply costly comparisons at all, we therefore have to limit the search space for matching pairs. We then compare only a set of (presumably) most likely matching pairs ("match candidate pairs") with each other (indexing / blocking). This consists in applying inexpensive exact pre-filters such as only comparing units within the same postal code area. In order to reduce the number of costly comparisons sufficiently while at the same time blocking out as few true matches as possible, both controlling block size and choosing the least erroneous blocking filters is essential. Since there are no exact filters available (even the postal code may be erroneous and subject to change), we are forced to overlay different blocking approaches.

We chose the following blocking strategies (meaning that record pairs become part of the set of matching candidates if either of these conditions is met):

1. One common, sufficiently rare[22] name token (first 5 letters) AND the Generalized Edit Distance (GED) of the cleaned name without legal form is in the upper 90th percentile of all matching pairs with at least one common name token (first 5 letters).
2. One common, frequent name token (first 5 letters) that is sufficiently rare in combination with 1 (to 5) of the postal code digits AND the GED of the cleaned name without legal form is in the upper 90th percentile of all matching pairs with at least one common name token (first 5 letters).
3. One common, frequent name token (first 5 letters) that is sufficiently rare in combination with 1 (to 5) first letters of the city name AND the GED of the cleaned name without legal form is in the upper 90th percentile of all matching pairs with at least one common name token (first 5 letters).
4. One common and rare pair of two name tokens (first 5 letters of each token).
5. One common and frequent pair of two name tokens (first 5 letters of each token) that is sufficiently rare in combination with 1 (to 5) of the postal code digits.
6. One common and frequent pair of two name tokens (first 5 letters of each token) that is sufficiently rare in combination with 1 (to 5) first letters of the city name.
7. Identical telephone contact numbers.

[21] Christen (2012a), p. 69-70. For an overview on indexing see Christen (2012b).

[22] The maximum size of a block for the current linkage version is 30, which leads to a maximum number of 30 x 30 = 900 comparisons per block. Note that due to the blocking strategy every representation is usually included in several blocks. It is assured that every representation is member of at least one block.

8. Both records have at least one common value for an external ID.

This procedure reduces the number of comparisons from roughly N = 200,000,000,000 to about C = 10 million candidate pairs. This leads to a reduction ratio of $RR = 1 - C/N$ = 99.995%, i.e. for all following steps, we limit our search for matching pairs to about 0.005% of all theoretically possible matching pairs.[23]

6 "Ground-Truth"-Data for Model Training and Testing

To train a classifier, we need a subset of matching pairs for which it is known whether these pairs really constitute a match or not ("ground-truth sample"). In order to allow the classifier to discover as many general relations between feature value identities or similarities and the match status of a match candidate pair as possible, this subset has to be as large and as representative of the universe of potential matches as possible. To test the prediction quality of a classifier, another subset of such pre-existing knowledge is needed. This subset has to be kept separately and should not be used to train the classifier up to the point of validation.[24]

While certainty about the true match status of any given match candidate pair is never achievable (we know from anecdotic evidence that external IDs in firm data are regularly erroneous for example due to being outdated), we can actually only extract a set of "quasi ground truth". This quasi ground truth can be used for model training on the condition that we calibrate our model so that it is insensitive to outliers.

We pursue a twofold strategy to extract quasi-ground truth data, using

1. Common external IDs
2. Quasi identical balance sheets

As *first* method for finding *quasi certain matches*, all common external ID variables (see Table 2) have been extracted from the data sources. For this only the universe of "cleaned firm name" / "city" / "postal code" - combinations ("raw data IDs", see the section on data cleaning) over all datasets has been considered. All raw data-IDs were then matched by the common IDs, which generated a list of raw data-ID pairs known to be very likely matches.

Similarly, using the same approach, a list of *quasi certain non-matches* have been generated. It is much easier to find true non-matches than true matches (there are naturally vastly more non-matches in the 10 million match candidate raw data-ID pairs). Therefore, non-matches were only declared when, next to a mismatch of at least one external ID, there was additionally at least one other mismatch between both entries in either a second external ID or the founding year of the firm. This reduces the risk of classifying a pair as a certain non-match due to an erroneous or outdated external ID in either of the two entries.

[23] The immediate restrictions to increasing the set of candidate pairs through less restrictive blocking are lack of RAM and processing speed / number of processors.
[24] Christen (2012a), p. 34.

Table 3: Identifiers found in the Raw Data[25]

Data Set / ID	USTAN / Jalys	DAFNE	HOPPE	AWMUS / MiDi	BAKIS-N	EGR	KUSY
Awmus-ID	.	.	.	+	.	.	.
Bakis-ID	(+)	.	.	.	+	.	.
Ustan-ID	+	.	.	.	.		.
Dafne-ID	(+)	+		(+)	.	+	.
Hoppe-ID	(+)		+	(+)	.	+	.
ISIN	.	(+)	(+)	(+)	.	.	.
EGR-LEID	.	+	+	+	.[26]	+	+
DestatisNr	.	.	.	(+)	.	+	.
LEI	.	.	.	.	.	(+)	.
TradeRegNr	.	+	+	+	+	+	+

The *second* method to generate ground truth knowledge consists in comparing balance sheets for the datasets that include comprehensive balance sheet data: USTAN, Hoppenstedt and Dafne. For this, we compared a subset of the balance sheet items that overall showed the fewest missing values and that were not before used for feature generation (see the section on feature generation):

- Sachanlagen *(fixed assets)*
- Umlaufvermögen *(current assets)*
- Vorräte *(inventories)*
- Sonstige Forderungen und Vermögensgegenstände *(other receivables and assets)*
- Kassenbestand *(cash)*
- Aktiver Rechnungsabgrenzungsposten *(prepaid expenses and deferred charges)*
- Eigenkapital *(equity)*
- Ergebnis der gewöhnlichen Geschäftstätigkeit *(profit of common business operations)*
- Betriebsergebnis *(operating income)*
- Rohergebnis *(gross profit)*
- Jahresüberschuss/-fehlbetrag *(net profit / loss)*

For each of these 11 items, the percentage difference between two potentially matching records (for identical balance sheet reference years)[27] was calculated as a percentage difference as described in the section “field comparison”. Whenever at least balance sheet 8 items were non-zero and non-missing in both datasets and their percentage difference was less than 3% on average, the pair was declared a *quasi match*. Whenever at least 8 items were non-zero and non-missing in both datasets and their percentage difference was more than 50% on average, it was declared a *quasi non match*. This information was then added to the “ground-truth” match candidate raw data-ID table.

[25] Brackets indicate a very large share of missing values.
[26] Information on the court district, which is a prerequisite to construct the EGG-LEID from the trade register number, was not available when this version (v8) of the linkage was run. It will however be available for the next version.
[27] And only for unconsolidated balance sheet information, i.e. excluding corporate group balance sheets.

From the balance sheet data, we generate a total of 57,859 quasi-certain matches between different "raw data-IDs" (original name / city / postal code - combinations) and 389,611 quasi-certain non-matches.

Using the external ID information and the information from balance sheet comparisons together, we end up with a total of 680,915 unique pairs of raw data-IDs for which we have quasi knowledge about the true match status. A random subsample of 25% (170,228) of these pairs were immediately reserved for later validation purposes (ultimate hold-out set), to avoid the early "contamination" of all available validation data by repeated human-machine interactive model calibration. The remaining 510,687 pairs were again split into a random 75% (383,015) training set and a 25% (127,672) validation set available in the course of the current linkage process. Of each of these subsamples, a share of about 28% constitutes matches.

In relation to the total number of about 10,000,000 match candidate raw data-ID pairs defined during the indexing step, this means that on average we have about 3.8 training pairs per 100 match candidate pairs, of which 2.8 are known negatives and about 1 is a known positive.

7 Model Selection

We construct two types of estimation samples: one is made up of the entire set of matching candidate raw name / city / postal code combinations (roughly 10 million matching candidate raw data-IDs, see above), and one consists of the matching candidate pairs of a particular bilateral relation between two datasets. The latter type of sample has fewer matching candidates / observations and is fed with an enhanced set of features (shorter target vector, shorter training and validation vectors, and a shorter but wider feature matrix). With this second type of samples, we attempt to use every feature that is available for a particular bilateral comparison of two datasets.

Classification based on Firm Names

While the rules of firm name construction do not require a firm name to be unique Germany-wide but only locally, we know from other sources that more than 90% of all firm names in Germany are in fact unique Germany-wide. We therefore consider it worthwhile to first attempt to classify the list of match candidate pairs solely by using the features that we extracted from the firm name. Another reason to do this is that the variable "firm name" is available in all 6 datasets without any missing values (this is not true for any other variable). This enables us to estimate a pooled model using all available match candidates, with a correspondingly large sample size.

Classification based on additional features

For all 15 bilateral comparisons between the 6 datasets, we then include the scores calculated by the purely name-based classifier as features, and complement it with the other features available for the particular bilateral relation. For each of the 15 bilateral relations, we have a different set of common variables and therefore a different set of comparison features (see the corresponding table in the section on feature generation).

The goal is always to calibrate the classification models such as to capture the general dependencies between comparison features and the true match status.[28] We automatize parameter choice using a randomized parameter search.[29]

We try to avoid the following common pitfalls of classification model selection:

1. Assuming linear relations when there likely are important non-linear relations
2. Not adapting the loss function to the use case
3. Using too many features with respect to the training sample size
4. In effect training on the test sample through repeated "trial-and-error" model calibration ("human-loop overfitting")
5. Failing to make the model robust against outliers

Regarding the first problem of erroneously assuming linear relations: we consider it likely that there are not only linear relationships between our features and our outcome (in particularly regarding the string comparison features). Therefore we rely on non-linear models such as tree-based models.

With respect to the second issue of not adapting the loss function, we choose to let the models maximize the F1-Score,

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

which is the harmonized mean of the precision measure and the recall measure, since we are both interested in a large precision and a large recall score (for these measures see the section on evaluation below), and their optimal weighting is generally determinable since it depends on the specific analytical use of the data.

For the third problem of using too many features: our training data set, at least for the pooled model, is quite large (see the section "Generating "Ground-Truth" Data for Model Calibration and Validation"). For the bilateral samples, however, the problem of sample size vs number of features becomes relevant, since for these samples there are a) more features and b) less training data. We try to deal with this by combining (merging) features that have a similar meaning (such as city name and postal code features both signal location), by dropping low-variance features and by letting models choose an optimal subset of the available features (see below: "randomized parameter grid search").

To avoid the fourth potential problem of overfitting by repeatedly adapting the model after testing on the same test data over and over again we take several complementing countermeasures:[30] First, we set aside 25% of our ground-truth data for final evaluation, not to be used until the correction loop of this report is completed. Second, we select our model parameters using k-fold cross-validation, which chooses a different subset of the training data for training and model testing over k tries. Third, instead of choosing the parameters of the models manually, we rely on automatic "hyperparameter"-tuning,

[28] All machine learning calculation steps are done in Python, using the package "scikit-learn", see Pedregosa et al. (2011).
[29] Bergstra et al. (2012).
[30] For best practices in model selection, particularly with respect to Python machine learning libraries see Pedregosa (2011).

which consists in defining a broad parameter search space which the models use to search for the best parameters ("parameter grid search"). Finally, since comprehensive grid searches are computationally expensive and since it has been shown theoretically and empirically that, given a fixed computational budget, randomly chosen parameter trials tend to yield better models than manual parameter selection we randomize parameter searches ("randomized parameter grid search").[31]

To counter the fifth problem of outlier robustness, which is essentially overfitting, we use model ensembling based on random sample sub-models, which includes a) a random forest model, which is in essence an average of several decision tree classification models, which are trained on different random subsamples of the training data (with replacement) and b) a gradient boosting classifier as another sub-model, which is a sequence of gradient descent models based on random subsamples with replacement.[32]

8 Test Data Based Evaluation of the Classification Model

We evaluate our predictor model using a random 25% holdout set of the data on known match / non-match pairs of raw data-IDs that was not fed to the classifier for training and which do not constitute exact matches (i.e. no pairs of identical "raw data"-IDs). This means that this hold-out set constitutes „unseen data" to the model and to the entire human - machine interactive process of calibrating the model up to this point. The hold-out set comprises 28,663 known matches and 74,651 known non-matches. We compute predictions for the pairs in the hold-out set and compare these predictions with their true match / non-match status. This leads to four possible outcomes:

- The pair is a non-match and correctly classified as a non-match („true negative", TN)
- The pair is a non-match and incorrectly classified as a match („false positive", FP)
- The pair is a match and correctly classified as a match („true positive", TP)
- The pair is a match and incorrectly classified as a non-match („false negative", FN).[33]

Two measures are used for evaluation:

Precision is defined as the fraction of true positives over all pairs classified as matches by the classifier, i.e.: TP / (TP+FP). Put differently, it is the share of the classified matches (TP+FP), that are in fact matches (TP).

Recall is defined as the fraction of true positives over all known true matches, i.e. TP / (TP+FN), or: the share of the known true matches (TP+FN) that the classifier classified correctly (TP).

Since each classified pair is assigned a matching likelihood by the classifier, we can trade precision against recall by changing the likelihood threshold above which a pair is classified as a match. Depending on our relative preferences for precision and recall, which depends on the analytical question, it may either be more desirable to include a rather large share of true matches in the analysis, at the

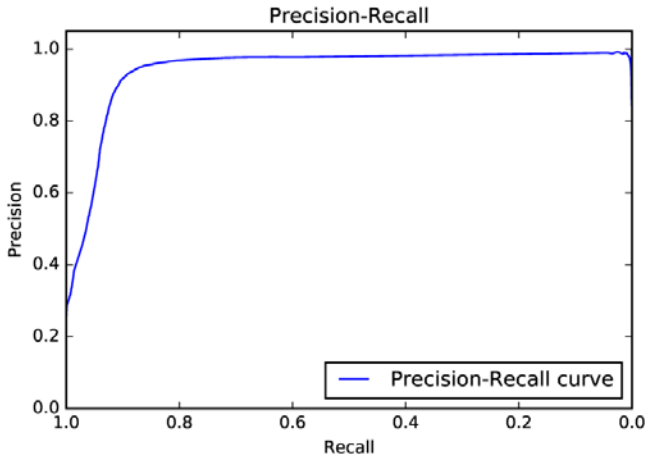
[31] Bergstra, J. and Bengio, Y. (2012).
[32] A gradient boosting classifier is essentially a sequential set of decision trees that are each trained to avoid the classification mistakes of their predecessor.
[33] Christen et al. (2007).

expense of a correspondingly large share of false positives (high recall / low precision), or it may be more desirable to include a rather low share of false positives, at the expense of missing a relatively large number of true matches.

8.1 Evaluation of the Pooled Name-Based Classification

For the purely name-based classification step, as described in the previous section, the entire universe of potential matching pairs, drawing from all 7 datasets, is classified using only the firm name and all features derived from the firm name. The precision / recall trade-off for this purely name-based classifier is described by figure 1.

Figure 1: Precision / Recall Curve for the Pooled Name-Based Model

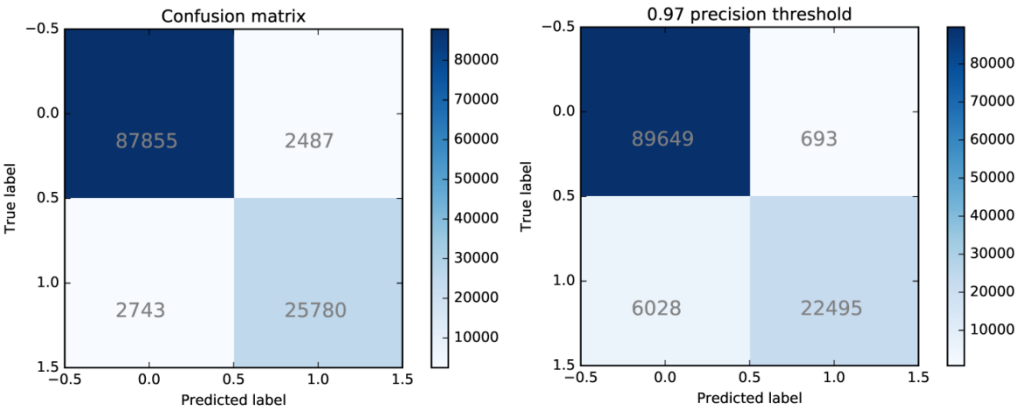


We assume two use cases: one „balanced“ case, in which precision and recall are about equally important, and a „high precision“-case, which focusses on achieving some minimum precision level, which we chose to be 97%.[34]

For the „balanced“ case we define the classification threshold as the classification rule of the underlying logistic model, which classifies a pair as a match when the estimated likelihood of a match is larger than for a non-match, i.e. when it is above 0.5. This tends to produce similar precision and recall scores (depending on the distributions of the likelihood scores for both true matches and true non-matches). For this evaluation scenario, the number of cases in each of the 4 classes (TN, FP, TP, FN) is shown in the left picture in Figure 2.

[34] This choice is arbitrary aside from the restriction that we want the minimum precision in this scenario to be higher than the precision to be expected in the balanced case, while being still low enough to yield an acceptable recall value (i.e. the recall should be at least around 75%).

Figure 2: Confusion Matrix for the Pooled Name-Based Model[35]



From these frequencies, we calculate precision (precision = TP / (TP+FP) = 25780 / 28267) of 91.2%. This means that when our classifier declares a matching candidate pair as a match whenever it considers it more likely to be a match than a non-match, judging by its performance on the unseen hold-out test data, it can be expected that 91,2% of all pairs that it classifies as matches are in fact really matches. From the above frequencies, we can also calculate the recall score (recall = TP / (TP+FN) = 25780 / 28523) of 90.4%. This means that when our classifier declares a matching candidate pair as a match whenever it considers it more likely to be a match than a non-match, the classifier managed to detect 90,4% of all true matches present in the unseen hold-out test set.

For the “focus on precision”-evaluation we assume a minimum precision requirement of 97% (see the right picture in Figure 3). This can be achieved by raising the bar of the required likelihood threshold, thus shifting some false positives to the group of true negatives (upper right to upper left corner of the confusion matrix), but at the cost of also shifting some true positives to the group of false negatives (lower right to lower left corner), until the desired precision is met. This generates the confusion matrix shown in the right picture in Figure 3. This generates a recall of (recall = TP / (TP+FN) = 22495 / 28532) 78.9%. This means that raising precision from 91.2% to 97% comes of the cost of reducing the recall score from 90.4% to 78.9%, which means missing out on about an additional 11 percent of all true matches.

8.2 Evaluation of the bilateral Classification Models

For the bilateral classification samples, we have more features, but less training and testing data (i.e. a wider but shorter feature matrix, shorter target vector, shorter training and testing vectors). Depending on the datasets involved in the bilateral relation, we have different comparison features available: when the set of non-missing comparison features is large, such as when both datasets contain balance sheet information or many cases with common external identifiers, prediction results tend to be better and vice versa.[36] For brevity, we present evaluation results not for all bilateral relations but (for

[35] Counts include only inexact match candidate pairs of raw data-IDs for which the true match status is known. That implies that the table excludes exact matches.
[36] Master data quality differences may also play a role.

illustration purposes) only for two relations; one with an above average prediction quality (USTAN-to-Hoppenstedt) and one with a below-average prediction quality (USTAN-to-AWMUS).

For the USTAN-to-Hoppenstedt relation, for a balanced precision / recall-requirement, we get a rather favourable combination of precision of 1955 / 2023 = 96.6% and a recall of 1955 / 2019 = 96.8% (see next section). For the USTAN-to-AWMUS relation, for a balanced precision / recall-requirement, presumably due to fewer training cases and a more narrow feature matrix, we get a less favourable combination of precision = 89.1% and recall = 95.1%. The following table shows the recall values for each bilateral relation between the six datasets on the basis of a "focus on precision" classification threshold.

Table 4: Recall values for a 97% target precision-level

	EGR	BAKIS	AWMUS	DAFNE	HOPPE
BAKIS	90.5%				
AWMUS	95.7%	95.9%			
DAFNE	96.6%	98.2%	83.6%		
HOPPE	94.7%	91.3%	90.6%	96.2%	
USTAN	95.3%	94.0%	87.0%	97.5%	96.7%

The weighted average recall score (weighted with the number of classified matches in each bilateral relation) for all bilateral models is 93,5%. This means that for a precision level of 97%, over all bilateral relations, the bilateral classification model is able to detect 93,5% of all test pair matches.

It is important to note that this recall score only refers to the probabilistic model classification, this means it refers to those match candidate pairs that we were not able to match exactly on name and addresses or by external IDs. Moreover, for some of the match candidate pairs that entered probabilistic classification, whether they were classified as matches by the model or not, we in fact know the true match status, which can be different from the model prediction. Those are exactly the "ground truth" pairs that we derived from comparing external IDs, and that we used for training, validation and testing. We use this knowledge in our final match classification, overriding model classification for these ground truth pairs, whenever there is a contradiction with the models' prediction. This means that the overall match classification error, measured as the share of all false positive assignments (based on either probabilistic classification, IDs or exact name and address-based assignment) over all assignments, turns out to be smaller than the pure probabilistic model classification error, which is based on the share of false positive model assignments over all model assignments.

9 Final Match Classification Rules ("Postprocessing")

In order to make a final decision for each match candidate pair on whether it should be classified as a match or not, we do not only use the model prediction, but we make use of all available information derived from model prediction, common IDs and exact name and address matches. This "postprocessing" consists of two main parts, 1: bilaterally consolidating matching rules and 2: matches by transitive closure.

9.1 Bilateral consolidation of matching rules

Final match classification rests on three main indicators or (sets of) match rules applied to each bilateral match candidate table:

Rule Set 1: External Identifiers

Rule Set 2: Exact agreement on name and address

Rule Set 3: Match prediction model score

The information gathered from these indicators has to be consolidated in order to make a final match classification decision. The first rule set actually only consists of one rule: match candidate pairs for which a common identical external ID can be found in the data are always classified as matches. The second set likewise only consists of one rule: match candidates that are exactly identical based on the original non-processed firm name and the original non-processed address are classified as matches. If there are contradictions between the first two rules, then the first rule beats the second rule. The third set of rules is based on the probabilistic score from the machine learning model and is applied only when the first two rules yield no result. It consists of broadly three sub-rules:

Rule 3a: Sorting out match candidate pairs below the chosen classification threshold[37]

Rule 3b: Sorting out match candidate pairs in which one of two candidates has a better match elsewhere ("not a mutual best match"-rule)

Rule 3c: Finding "second best" mutual matches

Rule 3a is a simple line-by-line classification rule, that states that only those match candidate pairs should be further considered that have a model score above some pre-defined score threshold. Rule 3b looks across all lines of the match candidate tables (including match candidates that have already been classified based on the first two rules) to see if this particular line is not only above the threshold but also mutual best match. For example, it is possible that either of both match candidates has already been matched to a different candidate by either rule 1 or rule 2 or there exists another match candidate for which the model score is higher.

[37] The classification threshold is chosen during model selection (see the previous section) and in our case was chosen to be 97%.

Table 5: Mutual best match rules

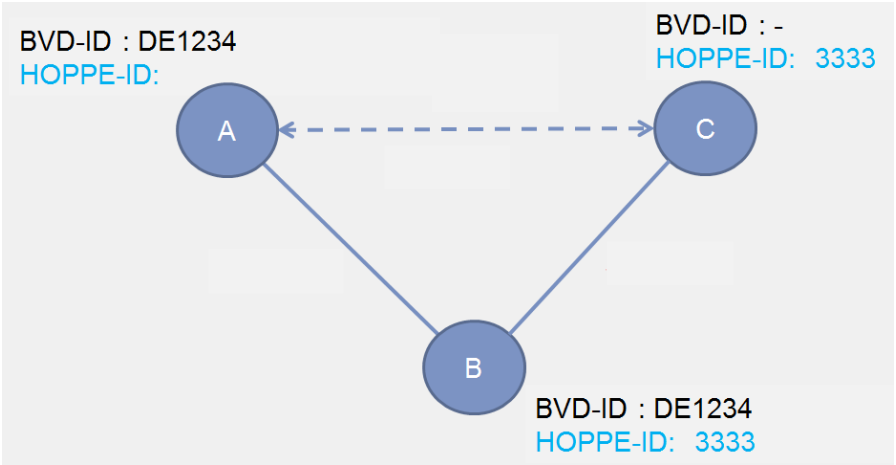
		Rule 3a	**Rule 3b**		**Rule 3c**
ID from Data A	ID from Data B	Model Score A - B	Above classification threshold? (ex.: 0.5)	Mutual best match?	**Final classification** (best or second best mutual match)
10	**3**	**0.91**	**1**	**1**	**1**
10	4	0.02	0	0	0
21	**6**	**0.79**	**1**	**0**	**1**
21	3	0.85	1	0	0
33	4	0.04	0	0	0

Such a case is illustrated in the table above. Here, candidate ID_A = 21 is best matched to ID_B = 3, however, ID_B = 3 has a better match to ID_A = 10. These two (ID_A = 10 and ID_B = 3) are mutual best matches, ruling out the pair (ID_A = 21 and ID_B = 3). Rule 3c then states that for those candidates that to not have a mutual best counterpart, it should be checked whether there is another, "second best" match candidate, for which the score is also above the threshold (in the example above this leads to ID_A = 21 being matched to ID_B = 6).

9.2 Multilateral consolidation ("transitive closure")

The second main postprocessing step takes a wider view in looking at all bilateral match candidate tables together. This is done by constructing left joins of all bilateral tables, one for each dataset. For example, starting with dataset A, such a "multilateral table" for this dataset is generated by left joining not only all match candidates of dataset B but also all match candidates with respect to dataset C. This enables us to apply rules of transitive closure by common identifiers. An example for this is provided by the figure below.

Figure 3: Transitive Closure by Common Identifiers



In this example, even though it was not possible to construct a direct id-based link between a particular match candidate from dataset A and another match candidate from dataset C bilaterally, it is possible to infer a link between these two units through a third dataset C. Such indirect ID-linkage rules are ranked higher than the matching rules based on the probabilistic score explained above. This means that in case of contradictions, they override rule set 3, but not rule sets 1 and 2.

10 Evaluation of the Final Match Classification

Since the final match classification is affected by our post-processing rules, and since the evaluation based on test data is limited by the assumption that the available test data is representative, we want to take a closer look at the quality of our final match result. To do this, we 1) briefly investigate the effects of our post-processing rules on the final match quality and 2) manually evaluate a random subsample from our final match classification table.

10.1 Effects of Post Processing on the Final Match Result

Final match classification is based on consolidating all derived match indicators from external ID comparisons, exact correspondence of name and address, the probabilistic model scores, as well as mutual best match rules and transitive closure rule (see previous section). First of all, it is important to note that due to the fact that there is a lot of information on external IDs in the data, final classification in the end rests to a large extent on external IDs. Since another large share of matches can be assigned by exact name and address comparisons, it is less than half of all match classifications (see the table below) for which in the end we have to rely on the probabilistic model. This means that the overall match precision value, as compared to the test data-based precision estimates for the probabilistic assignments, which was set to be 97% based on the test data, can be expected to be higher than that. In addition, the combined effect of these post-processing rules described in the previous section can be expected to further drive up precision.

Table 4: The Effect of Post-Processing on the Share of Probabilistic Matches

	Before post-processing	After post-processing	Difference
ID matches and exact matches	52.9%	61.5%	+8.6
matches based on the probabilistic algorithm	47.1%	38.5%	-8.6

The above table compares the share of matches derived from ID comparisons and exact name / address comparisons as a share of all match classifications, before and after post-processing. As the table shows, the precision/recall values of about 97%/93% previously calculated based on test data ultimately only refer to just about 38.5% of the match results. This leads us to expect the overall preci-

sion (referring to all match classifications, not only the probabilistic match classifications) to be closer to 99% than to 97%. Once all the record linkage processing stages are complete, the percentage of defined exact matches increases from 52.9% to 61.5%. This is particularly attributable to the transitive conclusions which are possible because the variety of master data sources with different characteristics permits indirect linkages. As indirectly identified IDs overwrite the model score based matches, indirect ID conclusions lower the percentage of probabilistically identified match pairs and increase precision/lower the error rate. However, the low error rate evolved during the manual quality checks (see next section) can also be attributed to the adjusting components in post-processing. They resolve contradictions and test for reciprocal best matches. The post-processing output can be found in the results column of the "Difference" column in Table 1; by applying the mutual best match rule and transitive closure, the share of match classifications that rely on the probabilistic model can be reduced by about 8.6%. To subject the process to an empirical examination and to verify our expectation that post-processing significantly increases precision, the final match results were subsequently checked manually based on random samples from the final match classification table.

10.2 Manual evaluation of the final match result

To manually evaluate the final matching result, a random sample of 1,000 entities was drawn from the overall 161,054 entities which could be linked to at least two input datasets. These 1,000 random match classifications were then cross-checked with respect to a set of basic match-relevant characteristics contained in the original master datasets that entered the record linkage process, such as original firm name, postcode, location, industry, founding year and additional IDs from all seven raw data sources. No incorrect matches were found among the 1,000 matched entities.

Second, another random sample of 100 matches was subjected to an even more intensive manual examination. Here, the final linkage table was not only checked using master data attributes from the source data, but also using at least one of the following online register databases for external verification: Registerportal Länder[38], Bisnode Konzernstrukturen online[39], Hoppenstedt Auskunfts-CD Großunternehmen – 2/2016[40] and Hoppenstedt Auskunfts-CD Mittelständische Unternehmen – 2/2016[41]. On balance, it was possible to verify 78% of the entity matches using the external register databases. This indicates that the Bundesbank source data are not always sufficiently up to date. It illustrates how valuable it would be to have access to company histories that would allow for the systematic evaluation of matters such as liquidations, takeovers, registered office relocations, changes of name and reporting threshold implications. Following an analysis of all available information - external registers and internal master data information, it was established that the sample of 100 entities had also been 100% correctly matched by the matching procedure.

[38] https://www.handelsregister.de
[39] https://www.hoppenstedt-firmendatenbank.de
[40] https://www.hoppenstedt-firmendatenbank.de
[41] https://www.hoppenstedt-firmendatenbank.de

The match result, with its current fairly restrictive settings, consequently delivers a quite high precision of probably above 99%, as shown above. This is due to the considerable power of the prediction model, the chosen minimum precision of 97% for the probabilistic matches, the fact that more than half of all match classification can be made based on exact name and address agreement and on external IDs and finally a presumed positive effect of post-processing on match precision. The number of match candidates found ("recall") is not likely to have decreased to the same extent: Post-processing only eliminates matches which cannot be matches on the basis of robust facts, eg contradictory IDs or non-reciprocal best matches. This increases precision, but does not reduce the recall quality measure. Moreover, transitive closure is likely to have a positive overall effect both on precision and on recall.

Nonetheless, an update to the match algorithm (Version 9) aims to extend coverage still further by enabling the algorithm to also process lower threshold values. To include entities with a lower probabilistic score result than 90% in the list of the first match candidates will presumably enhance the match recognition rate. It will then be a case of analysing on the one hand the extent to which this leads to a deterioration of match prediction accuracy. On the other hand it has to be evaluated to which extent more matches can be detected via this approach.

11 Conclusion

Given the data available for this record linkage, particularly due to the availability of external identifiers for a large subset of the data entries and the resulting abundance of training and testing data, conditions for a supervised classification approach were generally very good. Since a considerable number of entries did not have an external identifier, supervised classification was also worthwhile. We can also conclude that comprehensive pre-processing, such as legal form pattern extraction and standardization, is an almost indispensable step, at least with the classification algorithms available for this project. With respect to our blocking / indexing strategy, we are confident that our method is comprehensive enough to leave only few true matching pairs filtered out of the linkage process, while limiting calculation time and the size of the match candidate tables to a manageable amount. The probabilistic matching model yields precision / recall combinations of around 96%. Together with the available ID information and the exact name / address matches this yields an overall error rate in the final match classification of less than 1%. Because of this high precision result, we conclude that in the course of the coming updates of our record linkage process, we should also calculate and evaluate alternative matching tables that are based on with lower primary matching probability thresholds in order to evaluate how far we can increase the final recall while keeping final precision sufficiently low. Overall, we believe that our overall modelling strategy leads to a good predictive power (as documented by the precision and recall scores) especially when taking into account the typical data quality issues found in company master data. Also, we think that the choice of minimum precision of 97% constitutes a good compromise between accepting mismatches and loosing true matches. Given the presented precision / recall scores, semi-automatic matching results may for most analytical applications turn out to be sufficient even without additional manual classification. Finally, the presented method is sufficiently general to be applied to future matching projects that attempt to link micro data on legal entities.

12 Appendix

Table 6: Detected Legal Form Patterns and -Classification

Legal Form Pattern Captured (Abbr.)	Regular Expression(s)[42]	Legal Form Group Classification (Code)
GmbH	regexGMBH	3
Ltd.	regexLTD	3
BV	regexBV	3
UG	regexUG	3
e.G.	regexEG	4
AG & Co. KG	regexAG + regexUCO + regexKG	5
AG & Co. KGaA	regexAG + regexUCO + regexKGAA	5
AG & Co. OHG	regexAG + regexUCO + regexOHG	5
BV & Co. KG	regexBV + regexUCO + regexKG	5
BV & Co. KGaA	regexBV + regexUCO + regexKGAA	5
BV & Co. OHG	regexBV + regexUCO + regexOHG	5
GmbH & Co. KG	regexGMBH + regexUCO + regexKG	5
GmbH & Co. KGaA	regexGMBH + regexUCO + regexKGAA	5
GmbH & Co. OHG	regexGMBH + regexUCO + regexOHG	5
Ltd. & Co. KG	regexLTD + regexUCO + regexKG	5
Ltd. & Co. KGaA	regexLTD + regexUCO + regexKGAA	5
Ltd. & Co. OHG	regexLTD + regexUCO + regexOHG	5
SE & Co. KG	regexSE + regexUCO + regexKG	5
SE & Co. KGaA	regexSE + regexUCO + regexKGAA	5
SE & Co. OHG	regexSE + regexUCO + regexOHG	5
UG & Co. KG	regexUG + regexUCO + regexKG	5
UG & Co. KGaA	regexUG + regexUCO + regexKGAA	5
UG & Co. OHG	regexUG + regexUCO + regexOHG	5
KG	regexKG	6
KGaA	regexKGAA	6
OHG	regexOHG	7
AG	regexAG	10
SE	regexSE	10
e.V.	regexEV	11
GbR	regexGBR	11
V.a.G.	regexVAG	11

[42] A legal form pattern is detected if and only if all regular expressions listed are detected, in the order they are listed. For the actual regular expressions see table “Regular Expressions for Detecting Legal Forms”.

Table 7: Codelist for Legal Form Groups

Legal Form Group (Code)	Legal Form Group (Description)
3	Limited liability company
4	Cooperative
5	Limited (or general) partnership with a (public) limited company as general partner
6	Limited partnership
7	General partnership
10	Public limited company
11	Other

Table 8: (Perl) Regular Expressions for Legal Form Pattern Detection

Handle	Regular Expression (Perl)
regexBV	`( B\.?V\.?( | ?&|,|-|$))|((B|b)(esloten|ESLOTEN) (V|v)((ennootschap|ENNOOTSCHAP) (MET|met) (beperkte|BEPERKTE) (A|a)(ansprakelijkheid|ANSPRAKELIJKHEID))?)`
regexEV	`( |,|\.|^)(E|e)\.?(V|v)( |,|\.|$)|( |,|\.|^)(E|e)(INGETRAGENER|ingetragener) (V|v)(EREIN|erein)`
regexVAG	`( |,|\.|^)(V|v)\.?(A|a)\.?(G)( |,|\.|$)`
regexEG	`(G|g)(ENOSSENSCHAFT|genossenschaft)|( |,|\.|^)(E|e)(INGETRAGENE|ingetragene) (G|g)(EN|en)\.?(OSSENSCHAFT|ossenschaft)?|( |,|\.|^)(E|e)\.(G|g)( |,|\.|$)\|eG( \|,\|\.\|$)`
regexGBR	`(G|g)?(E|e)?(S|s)?\.?(ELL|ell)?\.?(SCH|sch)?\.?(AFT|aft)? (B|b)(Ü|ü|UE|ue)(RGERLICHEN|rgerlichen) (R|r)(ECHTS|echts)|( |,|\.|^)(Gbr|GbR|GBR)( |,|$)|( |,|\.)(Inh\.?(aber)?(in)?|INH\.?(ABER)?(IN)?)`
regexUCO	`(&|\+| UND | und | (U|u)(\.| )) ?(C|c)(O|o|IE|Ie|ie)( |,|-|\.|$|\))`
regexAG	`(( |-)A(G|g)( | ?&|,|-|$))|A(C|c|K|k)(TIEN|tien)-?(GES|ges)\.?(ELLSCHAFT|ellschaft|ELLSCH|ellsch)?\.?`
regexSE	`(( |-)SE( | ?&|,|-|$))|S(OCIETAS|ocietas) ?E(UROP|europ)`
regexUG	`((( |-)U((G|G-)|g)( |,|$))|(U(NTERN|ntern)\.?-?(EHMER|ehmer)?`

	?(GES|ges)\.?(ELLSCHAFT|ellschaft|ELLSCH|ellsch)?\.?))(\(?haftungsbeschr\.?(ä|ae)?(nkt)?\)?)?
regexLTD	(U(K|k))?(|,|\.|^)(LTD|Ltd)(-| |,|\.|$)|UK (Limited|LIMITED)
reg-exGMBH	(G|g)?(E|e)?(S|s)?\.?(ELL|ell)?\.?(SCH|sch)?\.?(AFT|aft)? ?(M|m)\.?(IT|it)? ?(B|b)(ESCHR|eschr)?\.?(Ä|ä)?(AE?|ae?)?(NKTER|nkter)? ?(H|h)(|,|\.|$|AFTUNG|aftung)
regexKG	((|-)(K|k)(G|g)(|&|,|-|\.|$|aA))|K(O|o)(M|m)(M|m)(A|a)(N|n)(D|d)(I|i)(T|t)\.?(G|g)?\.?(E|e)?(S|s)?\.?(E|e)?(L|l)?(L|l)?\.?(S|s)?(C|c)?(H|h)?\.?(A|a)?(F|f)?(T|t)?
regexOHG	(O\.?H\.?G\.?)(|,|$)|(O|o)(FF|ff)\.? ?(ENE|ene)?\.? ?(H|h)(ANDELS|andels)(|-)? ?(G|g)\.?(E|e)(S|s)\.?(E|e)?(L|l)?(L|l)?\.?(S|s)?(C|c)?(H|h)?\.?(A|a)?(F|f)?(T|t)?
regexKGAA	((|-)(K|k)(G|g)(a|A)A)|K(O|o)(M|m)(M|m)(A|a)(N|n)(D|d)(I|i)(T|t).*((A|a)uf A(ktien|KTIEN))

References

Bergstra, J. and Y. Bengio (2012). “Random Search for Hyper-Parameter Optimization”, *The Journal of Machine Learning Research*, 13, 281-305.

Biewen, E., S. Blank and S. Lohner (2013). „Microdatabase: Statistics on international trade in services. Technical Report, Deutsche Bundesbank.

Christen, P. (2012a). *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Springer Science & Business Media.

Christen, P. (2012b). “A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication”, *IEEE Transactions on Knowledge and Data Engineering*, 24(9), 1537-1555.

Christen, P. and K. Goiser (2007). “Quality and Complexity Measures for Data Linkage and Deduplication”, *Studies in Computational Intelligence (SCI)*, 43, pp. 127-151.

Cohen, W., P. Ravikumar and S. Fienberg (2003). “A Comparison of String Metrics for Matching Names and Records”. In *Kdd workshop on data cleaning and object consolidation*, Volume 3, pp. 73-78.

Fellegi, I. P. and A. B. Sunter (1969). “A Theory for Record Linkage”, *Journal of the American Statistical Association* 64(328), 1183-1210.

Levenshtein, V. I. (1966). “Binary Codes Capable of Correcting Deletions, Insertions, and Reversals”, *Soviet Physics – Doklady* 10(8): 707-710.

Lipponer, A. (2011). “Microdatabase direct investment - MiDi. A brief guide”. Technical documentation, Deutsche Bundesbank.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay (2011). “Scikit.learn: Machine Learning in Python. *Journal of Machine Learning Research,* 12:2825-2830.

Raschka, S. (2015): *Python Machine Learning*. Packt Publishing.

Schild, C.-J. and F. Walter (2015). “Microdatabase Direct Investment 1999-2013”. Technical report, Deutsche Bundesbank.

Schmieder, C. (2006). “The Deutsche Bundesbank’s Large Credit Database (BAKIS-M and MiMiK)”. *Schmollers Jahrbuch*, 126(4):653-663.

Stoess, E. (2001). “Deutsche Bundesbank’s Corporate Balance Sheet Statistics and Areas of Application.” *Schmollers Jahrbuch: Zeitschrift fuer Wirtschafts- and Sozialwissenschaften (Journal of Applied Social Science Studies)*, 121:131-137.