Bloothooft, G. (1994). 'Corpus-Based Name Standardization', History and Computing, 6.3, 153-167.

CORPUS-BASED NAME STANDARDIZATION

Gerrit Bloothooft

Summary
A method is described to standardize nominal data on the basis of a combination of rules and a probabilistic similarity measure. Onomastic corpora are used to estimate the probability of spelling variations automatically. These corpora are also the basis for finding the most likely standard for a name not encountered before.

Abstract
Variation in the spelling of names has various origins, many of which many are difficult to describe by rule. We present a method that uses both rules and a similarity measure of a probabilistic nature, and which can make use of existing onomastic corpora. Rules first convert an unknown name to a semi-phonemic form. Then a selection is made of possible candidates in the onomastic corpus. For this set, the similarity to the unknown name is computed and a decision procedure chooses the best candidate. If no specific onomastic corpus is available, the method provides a tool for a clustering of similar names. The method is demonstrated on a corpus of 49.193 first names from 18th century parish registers, in the availability of a Dutch corpus with 22.579 variants of 4.482 base forms of first names.

I Introduction

Name standardization is an essential element in automatic nominal record linkage procedures in historical analysis for which solutions have been sought in rule-based methods[1], similarity measures[2], or a combination of both[3]. Although these methods have been quite successful, there are also several limitations. Major questions concern (1) whether the set of rules can ever be complete, (2) whether the number of rules does not proliferate to an extent that many rules describe rare exceptions, (3) how rules can model writing, reading, and typing errors, (4) how to improve the simple foundation for similarity measures based on counts, and, most important, (5) how to utilize knowledge from onomastic sources and previously standardized nominal data.

The position taken in this paper is that at least part of the variation in names can be described as being of a statistical nature. There is a certain probability that a name is written in some way, while some variants are more likely than others. The introduction of probability in name standardization creates new challenges, new problems, and new solutions. A very important feature of a probabilistic approach is that we can learn automatically from standardizations that have been made earlier. This allows us to incorporate knowledge that has been gathered in onomastic research and knowledge that can be derived from truly linked records from previous historical research.

A probabilistic approach does not exclude the application of rules. Some processes are far better described by rules, while others, for instance typing errors, are easier described in probabilistic terms. Therefore, one of the major questions is to find the most effective balance between rules and statistics.

In this paper, we will first discuss the issues of the standard of a name and the characteristics of corpora that can be used for standardization. Then we describe the first, rule-based steps that can be made: partial grapheme-to-phoneme conversion and (optional) removal of name endings (especially for first names). We subsequently propose the diphone[4] as a basic unit for probabilistic computations and show how the probability that different diphones occur in variants of the same name can be estimated. This results in a similarity measure that gives the probability that two names are variants of each other.

If we already have a corpus of variants of names and we encounter a new, yet unseen name, the problem arises how to find the best standard for this name. We will show how to reduce the search space in our corpus in an efficient way into a small subset of likely name candidates. Finally, we give a decision procedure how to find the most likely standard.

The entire procedure is illustrated using an electronic version of the Dictionary of Dutch First Names (22.579 variants), and is tested on an independent corpus of 49.193 first names from 18th century parish registers from the city of Goes.

II Standards

If we want to utilize large corpora of names in the process of record linkage we need to introduce standards for names. These standards are not always available and their definition may depend on the intended application. The issue is what choice to make for the standard form of a given name. Onomastic experts may prefer the etymological base name, but for the purpose of record linkage, there is more freedom in choosing the standard as long as this standard is representative for a group of variants. The standard itself is not essential: a standard may even be some sort of code, see the typology given by Nygaard5. For record linkage, in order not to miss a link, one may prefer a standard that generalizes considerably. For instance, Koen and Koenraad are different base names at the etymological level, but it is wise to bring them under one standard for the purpose of record linkage. On the other hand, it may occur that we would like to split up variants of a single base name into two or more groups. In general, there is a tendency that (groups of) variants of a single base name can get an own identity and are not used interchangeable anymore after some time, such as present day forms Tineke and Catalijne, which are both variants of Catharina. Depending on the historical period addressed in record linkage, a base name may generalize too much. Essentially we have to pose the question: is it likely that, in some period and region, one person is named as variant one (Tineke) in one record, and as variant two (Catalijne) in another? Of course, the best information to answer this type of question comes from truly linked records. For corpus-based name standardization it would be ideal if already linked nominal data could be collected and made available for the further optimizing of standards.

We need standards because this is the way to accumulate knowledge on variants. Once we have seen a variant and have determined the related standard, we can add this knowledge to our corpus. If we

encounter the name again, a simple look-up procedure will suffice to find the corresponding standard. Name standardization is then converted into the problem of predicting the standard for a yet unseen name. In this process we may use all information that is already available in our corpus. That is, we may use all knowledge of variants already seen. For instance, if we know that Trijntje is a variant of Catharina, the yet unseen variant Trien is easily (automatically) recognized as being similar to Trijntje and thus can be linked to Catharina.

III The Corpus

III.1 Primary information

Basically, a corpus for name standardization should contain fields for the (1) original spelling of a name, (2) the corresponding standard (or standards), (3) the gender if appropriate, and (4) frequency of occurrence of the name. Other useful information, such as (5) the region(s) (country) where a name has been found, and (6) the interval in time when the name has been used, can also be included. In this paper, however, we will limit ourselves to the first three data types.

III.2 A corpus of first names

The procedures, outlined in this paper will be illustrated with results obtained on a large corpus of Dutch first names. This does not imply any limitation of the generality of the procedures, but this corpus happened to be the only one available for Dutch that fulfilled the necessary requirements.

The corpus consists of 22.579 different Dutch first names (10.611 male and 11.968 female) that ever occurred in written sources. The corpus originates from the Dictionary of Dutch First Names[6] (20.823 names) with additional first names from a 88.000 names sample of the census of 1947 (1682 extra first names[7]) and from a project on 19th century social-economic developments in the Meierij region (74 extra first names). We will refer to this corpus as our main corpus.

Each of these first names is linked to a base name[8]. The set of base names consists of 4482 names (2189 male and 2293 female). This set is gender dependent, which means that a base name may occur twice, both as a male and a female name[9].

The frequency of occurrence of first names from the sample of the census of 1947 has been added as an indicative figure. Frequency data were available for the 4736 different first names in this 84.000 names sample.

IV The rule-based phase

In general, the first steps toward standardization of a name include a conversion towards a more phonemic form and some morphological analyses concerning prefixes, suffixes, compositions etcetera. We believe that these first steps are best performed by rules. However, these rules should be robust and not very sensitive to context. If it is not possible to develop robust rules, the description of variation should be left to the probabilistic phase of the standardization.

IV.1 Grapheme-to-phoneme conversion

It may seem a good step to convert a name into its phonemic form[10]. The reason for this is that we may hope that the phonemic form results in reduced variation among names. There are, however, two major concerns to address: (1) The first is that the process should be very robust to uncommon spellings and spelling errors. Otherwise, the conversion itself may generate new errors that are hard to recover. Recent research on modern spelled names[11] shows the difficulty in automatically realizing phonemically transcribed forms. The difficulties will be even larger for historical names. (2) The second concern is the question what we really gain by this conversion: Is our expectation that variation reduces in a phonemic form of a name really valid?

We have chosen for a partial transformation, leading to a semi-phonemic form of a name. The process consists of a grapheme-to-grapheme and a grapheme-to-phoneme conversion for major spelling variation [in Dutch the rules affect the spelling for c (including ck and ch), d, h, j, ph, q, x, z, and create new symbols for diphthongs]. Only 67 rules are needed in this process. We did not try to convert vowels, because the distinction between long, short, or reduced vowels is hard to make reliably. We simply kept

the original spelling. In general, the approach proved to be robust. A problem could still arise with typing errors, for instance Racehl instead of Rachel, prohibiting the conversion of ch to X (Rachel -> RAXEL). Table 1 gives some example of the semi-phonemic conversion of first names.

Table 1: *Semi-phonemic conversion of some first names*

original	semi-phonemic
Adelheid	ADELHYD
Christiaan	KRISTIAAN
Aenke	ÆNKEN
Gualtherus	WALTERUS
Daphne	DAFNE
Elbrich	ELBRIG
Zander	SANDER
Aschwin	ASWIN
Dominique	DOMINIK
Annechien	ANNEXIN
Oscar	OSKAR
Theun	TŐN
Calixte	KALIKSTE
Francois	FRANSŌS
Eckart	EKKART
Tsjetske	TJETSKE
Rijcklof	RYKLOF

For all first names and their base names in our corpus of first names, the semi-phonemic form was computed. First names with equal semi-phonemic form were considered similar and merged. The same was done for the base names. For the first names this resulted in a reduction from 22.579 to 18.999 names (a reduction of 16%). For base names, the reduction was 26 names only, from 4482 to 4456 semi-phonemic base names. This small reduction in the number of base names indicates that these forms are dissimilar at the pronunciation level, as has to be expected. Examples of base names that have the same semi-phonemic form: Rut and Ruth -> RUT, Riemer and Reimer -> RYMER.

IV.2 Removal of name endings

After the partial grapheme-to-phoneme conversion, further reduction of variation in names can be realized by removing name endings, and a few cases schwa elision (Illebregt->ILBERGT). For first name endings, the reductions may consist of various types of diminutives (Jantje->JAN), latinized name endings (Albertus->ALBERT); for patronymics, the reference to 'son of' or 'daughter of' (Dirkse->DIRK); for surnames, the same as for patronymics (if appropriate, Benjaminsen->BENJAMIN) and a few diminutives (Maatjes->MAAT) and latinized endings (Appelius->APPEL). In surnames, reduction of name endings should be applied very carefully, because there is a much greater danger in removing essential information than with first names. This process was carried out by 69 rules. We call the resulting name the kernel of the name. This kernel has no onomastic interpretation, but should be considered as the part of a name that is most characteristic of a name. Both the semi-phonemic form and kernel of a name were used in further procedures. Table 2 gives some examples.

Table 2: *Example of kernels of some first names and patronymics*

original	*kernel*
patronymics	
Pietersd(r)	Pieter
Pieters	Pieter
Pietersdogter	Pieter
Klase	Klas
Klasen(s)	Klas
Dirkse	Dirk
Janszoon	Jan
Hagen	Hag
various name endings	
Fransis	Fran
Frank	Fran
Wilma	Wilm
Ella	El
Kristel	Krist
Aaltine	Aalt
Gertjen	Gert
Abeltje	Abelt
Katinka	Katin
Aartsje	Aart
Abeke	Ab
latinized forms	
Petrus	Peter
Emilius	Emil
Fransiscus	Fran
Albertus	Albert
Aristoteles	Aristotel
Alegondis	Alegond
schwa epenthesis	
Illebregt	Ilbregt
Fedderik	Fedrik

The removal of name endings in our corpus of first names resulted in a further reduction from 18.999 to 10.377 different names (5634 male and 4743 female), which is 46 % of the original corpus. The implication is that variation in names can be reduced to about a half by rules (more precisely, our rules and this databases and under the assumption that the results are correct). Again, for base names the reduction was much less, from 4469 to 4021 names (2041 male and 1980 female), 90 % of the original set. There is still quite a gap between the 10.377 variant kernels and the 4021 base name kernels that could not be bridged by rules.

V The probabilistic phase

If it were possible to create a set of rules that could convert any name into the corresponding standard, the problem of name standardization would be solved. The conversions discussed above are a step towards the reduction of variation in names but do not pretend to capture all variation. The claim we want to make here is that it is impossible to design a complete set of rules, due to unpredictable spelling variations and all kinds of errors: we need probabilistic procedures to overcome this limitation.

In case of unpredictable uncertainties in spelling, the issue of standardization of a name becomes the problem of finding the most likely standard for this name. This introduces the need for a measure of similarity of two names. The most likely standard for an unknown name is the standard that, from all available standards, has the highest probability. Three major questions arise: (1) how do we quantify the similarity of two names in a probabilistic sense, (2) how can we obtain these probabilities from a training corpus, and (3) how do we find the name with the highest probability in an efficient way.

V.1 Basic units

If we compare two names, we first have to define the unit by which this comparison is carried out. Because the original spelling was already converted to a semi-phonemic spelling, names can be thought of as sequences of phones (single elements), diphones (two adjacent elements), or triphones (three subsequent elements). Let us consider the first name ALBERT (# indicates a start/end symbol):

phones: #, A, L, B, R, T, #
diphones: #A, AL, LB, BE, ER, RT, T#
triphones: #AL, ALB, LBE, BER, ERT, RT#

According to our experience, a phone is too general and a triphone too specific to be used, whereas the diphone gives the best results.

In comparing ALBERT and the variant ELBERT, we have to consider the following diphone sequences:

#E, EL, LB, BE, ER, RT, T#
#A, AL, LB, BE, ER, RT, T#

and we need to know the probability that the diphones #A and AL in ALBERT are changed into #E and EL in ELBERT. In this case it is more or less trivial that we should consider the diphone interchange of #A with #E and of AL with EL. In other cases, with inserted characters or heavily changed spelling, it is not immediately obvious which diphones to compare. We will come to that later. What we need anyhow, however, is the probability that a diphone interchange is made. These probabilities can be estimated automatically on the basis of a training corpus.

V.2 Estimation of diphone interchange probabilities

In a training corpus, we have names and their corresponding standards. Names with the same standard are considered to be variants of each other. If we consider the Albert / Elbert example once again, we need to know the probability that #A is changed into #E. This probability is estimated by counting the number of names that start with #A that have at least one variant that starts with #E, and dividing this number by the total number of names that start with #A. Since we do not want to arrive at different measures comparing Albert with Elbert or Elbert with Albert, we also estimate the probability of the reverse interchange between #E and #A. We take the highest value of both probabilities as a final estimate for both the interchange of #A and #E, and for the interchange of #E and #A. This procedure can be applied automatically for all diphone interchanges observed in the corpus[12].

We arbitrarily used a default minimum value for the diphone interchange probability of 0.10, which means that any interchange always got a minimum probability of 0.10. This default value was used because of the limited amount of training data (22.579 first names), in which many diphone interchanges were rarely observed, and reliable estimates consequently could not be made.

In Table 3, the diphone interchange probabilities derived from 22.579 Dutch first names, with a frequency of occurrence of the interchange of at least 50 are given. We used some special symbols here. The symbol = indicates a single phone, for instance =A. The pair =A,AA implies that one name has a single phone A where the other name has the diphone AA. The symbol * is a wild card, in this case the first element of the diphone does not matter; such a pair can be interpreted as a phone interchange12. Diphthong symbols are explained as original spelling sequence between square brackets. The total number of diphones pairs with an interchange probability above 0.10 (or above wild card probability) was 652, which is very small relative to the theoretical maximum of about 500.00013. This indicates that spelling variation with a reasonable frequency of occurrence is quite limited.

Table 3: *Diphone interchange probabilities derived from 22.579 Dutch first names, with a frequency of occurrence of the interchange of at least 50. The diphone pairs are ordered according to the right element of the first diphone: the vowels A, E, I, O and U, and consonants.*

diphone pair		prob	diphone pair		prob
#A,	#Æ [ae,ea]	.79	WI,	WY [ie,ei,ij,y]	.69
AA,	=A	.73	SI,	SY [ie,ei,ij,y]	.59
#A,	#Å [ao,oa]	.65	RI,	RY [ie,ei,ij,y]	.58
*A,	*Æ [ae,ea]	.63	*I,	*Y [ie,ei,ij,y]	.56
AA,	=Æ [ae,ea]	.44			
#A,	#Â [ai,ay]	.41	OO,	=O	.68
*A,	*Â [ai,ay]	.41	*O,	*Å [oa,ao]	.49
*A,	*Å [ao,oa]	.37	*O,	*Û [oe]	.30
BA,	BE	.35	*O,	*Ä [au,ou]	.25
=A,	AD	.31	*O,	*U	.15
GA,	GE	.31			
*A,	*E	.28	UU,	=U	.75
*A,	*Ä [au,ou]	.14	*U,	*Ý [ui,uy]	.66
#A,	=#	.14	*U,	*Û [oe]	.45
EE,	=E	.70	*B,	*P	.12
GE,	GÆ [ae,ea]	.53	*D,	*T	.18
*E,	*Æ [ae,ea]	.47	*F,	*V	.14
KE,	KO	.26	#H,	=#	.13
*E,	*Â [ai,ay]	.26	*G,	*X [ch]	.28
*E,	*Y [ie,ei,ij,y]	.24	IK,	IG	.15
ME,	MI	.22	*K,	*X [ch]	.34
RE,	RI	.20	*K,	*T	.11
RE,	ER	.19	LL,	=L	.42
*E,	*I	.18	MM,	=M	.51
*E,	*O	.15	NN,	=N	.49
BE,	BR	.15	SS,	=S	.47
LE,	EL	.14	TT,	=T	.53
#E,	=#	.11	*V,	*W	.14
				default	.10

It is remarkable that most of the diphone pairs concern vowels, especially A and E. This relates to the earlier choice not to convert vowels into a semi-phonemic form. The data in table 3 nicely demonstrate that vowels can be easily interchanged (with appropriate probability). Many consonant diphone pairs can be interpreted as phone pairs (because of the symbols = and *). Some pairs have an interesting relation to the phonemic distinction voiced-unvoiced: (*B,*P),(*D,*T), and (*V,*F). Several consonant geminates can easily reduce to a single consonant, shown by high probabilities for (MM,=M), for instance. The X in the semi-phonemic code stands for an original ch spelling that could not be resolved between k, g and sj. That ambiguity returns here as probabilities for (*K,*X) and (*G,*X). The (#H,=#) pair models a h-deletion in the beginning of a name, which is a common phenomenon in the province of Zeeland.

V.3 The best match between two names

So far, we did not yet give a general solution to the problem of how to find the best way to match two names. For Albert and Elbert this may be obvious, for Petronella and Pieterneel this is not. In any match, the total probability that names are variants of each other is given by the product of all diphone interchanges. For the best match, this total probability should have the maximum value.

To find this maximum, the technique of dynamic programming was used[14]. This technique efficiently finds the best match between two strings[15]. Consider the example given in table 4, which compares the name Piter with the misspelled form Ptier. The semi-phonemic forms of the names are PITER and PTYR, respectively.

Table 4. *String matching of PITER and PTYR with dynamic programming.*

		#	P	I	T	E	R
#	|	**100**	16				
P	|	16	**100**	**41**			
T	|		41	16	**41**		
Y	|			26	16	**14**	
R	i				7	6	**14**

The table values give the cumulative probability (in percentage) from the starting point (#,#). The bold numbers show the path (or warp) that gives the maximum value at the end, at (R,R). The best warp shows that from #,# we best go, of course, to (P,P). Then we follow the diphone pair (=P,PI) (this models the insertion of I), but we are punished for this with a reduction of the total probability to 41%. After (T,T) we have an diphone interchange (TY,TE) and this reduces the total probability further to 14%. With this probability we arrive at (R,R).

We could have tried to follow other paths, but this would always have led to a lower total probability. The non-bold values in the example give the other intermediate cumulative probabilities that are considered by the algorithm, but which were rejected because they resulted in a lower total probability at the end. Note that the algorithm, for reasons of efficiency, does not investigate all possible paths.

What interpretation should we assign to the total probability values that result from a comparison of two names? As a rule of the thumb, values higher than 50% indicate very similar names, while for values lower than 10% there is little chance that names are variants.

V.4 A final measure of similarity

For computational reasons, it is easier to use the logarithm of a probability, log(prob), rather than the probability itself, because the product of probabilities is then transformed into a summation of log(prob) values. However, because longer names involve more diphone pairs, there is a tendency that the total probability is lower for comparisons among longer names. Because we will use fixed thresholds we have to compensate for this. In the final measure of similarity, we therefore included an extra (empirical) multiplication factor, which is related to the length of the names:

3 / (max_length - 1)

where max_length is the maximum of the two lengths of the names under comparison. The factor is applied only if max_length is larger than 4. We used the result as the final measure of similarity between two names.

VI How do we find the best standard?

Now that we have determined a way to quantify the similarity between two names, the problem of how to find the best standard for an unknown name out of a large corpus still has to be solved. The procedure followed here was that we first split off the subset of names in our corpus that were rather similar to our unknown name, that is, the subset of names for which the measure of similarity exceeded a certain threshold. Once this subset was obtained, the next problem is the decision procedure that gives the best candidate.

VI.1 Reducing the search space

An essential factor in a successful name standardization using large corpora is to find a way to reduce the number of possible name candidates as quickly and efficiently as possible. It is totally inefficient to compare an unknown name with more than 22.579 names from the general corpus. An alphabetical search is inadequate, because spelling differences may occur in the beginning of the name. We therefore devised the following procedure to reduce the search space.

The semi-phonemic version of a name[16] is reduced further to one or two name skeletons. In such a skeleton, one full syllable is retained and all consonants, but no other vowels. Adjacent equal characters in the semi-phonemic form are reduced to one. In the first skeleton, the first syllable is retained, in the second skeleton the next following syllable that does not contain the vowel e. Examples:

JAKOB -> JAKB, JKOB
BERENDINA -> BERNDN, BRNDIN
GERRARD -> GERRD, GRARD
ALEXANDER -> ALKSNDR, LKSANDR

A second or subsequent syllable with the vowel E is discarded because of the likelihood that the E involves a schwa. A schwa is not a very stable element in a name: a schwa cannot bear stress, can be easily reduced, and may therefore be absent in variants.

The argument for making a name skeleton is that it is likely that variants of the name have at least a part of at least one of the two skeletons in common. Consider for instance some variants of Herman:

HERMAN -> HERMN, HRMAN
HARREMAN -> HARMN, HRMAN
ARMAN -> ARMN, RMAN
HARMEN -> HARMN

On the level of the name skeleton, these examples have three or more characters in common, but these common characters may not be consecutive. Therefore, we add a start symbol # to the skeleton and perform a rotation (cyclic permutation) of the name skeletons:

HERMN -> #HERMN, HERMN#, ERMN#H, RMN#HE, MN#HER, N#HERM
HRMAN -> #HRMAN, HRMAN#, RMAN#H, MAN#HR, AN#HRM, N#HRMA
HARMN -> #HARMN, HARMN#, ARMN#H, RMN#HA, MN#HAR, N#HARM
ARMN -> #ARMN, ARMN#, RMN#A, MN#AR, N#ARM
RMAN -> #RMAN, RMAN#, MAN#R, AN#RM, N#RMA

We see, for instance, that the skeleton harmn of Harmen has a best match, from left to right, in the version RMN#HA with the skeleton version RMN#HE of Herman: the first five characters are equal.

In practice, we create a database with all rotated versions of skeletons of a set of known names. Typically, this database contains about eight times as many records as the original database. The database with rotated skeletons is indexed in alphabetical order.

If we have an unknown name, we first make the rotated skeletons of this name. We then search the database with known rotated versions of skeletons for existing skeletons that have most characters in common with one of the (rotated) skeletons of the unknown name. The corresponding names are the best candidates for a further investigation of similarity.

The search for comparable skeletons should be limited, because it is, for instance, of no use to know the names that have two characters in common in a skeleton of length of eight. In general, the search is terminated if the length of the search string is less than n-1 (n is the length of the skeleton of the unknown name). This may depend, however, on the number of candidates found, and n itself. For n>6, n-2 may be tried also.

Typically, the resulting search space reduction varies between a few names to about 30-50 names, with an estimated average between 10 and 20 names. Compared to more than 22.579 surnames this is a reduction with a factor 1000, that is realized in a very efficient way.

VI.2 The decision procedure

Using the search space reduction technique described before, we obtain a subset of names that are candidates for being related to the unknown name. For all these names we compute the similarity measure and reject names that are below a certain threshold. The similarity measure is computed for both the kernels of the names and the semi-phonemic forms. The maximum value is used. In most cases this is the similarity between kernels of names. If, for some cases, kernels are reduced too far (for instance because of the effect of typing errors on the application of rules), better results are obtained with the semi-phonemic form.

There are several options left to choose the best candidate. We can simple take the candidate with the highest similarity measure. This is sometimes the best decision to make, but may not be appropriate if the best similarity is not very high. In such a case we can take into account that, for instance, several

candidates have the same standard. We chose to sum the similarity measures over all candidates with the same standard. The standard with the largest summed similarity is the winner. This procedure promotes the standard that already has several variants.

VI.3 Reconsidering base names as standards

Before applying the given procedure for standardization on a test set of first names, we first have to reconsider the appropriateness of the base names. Initially, we can use these base names as standards, but we should take into account that base names may be quite similar and that, with an application for record linkage in mind, we had better be not too specific. Although, by definition, base names cannot be merged from the onomastic point of view, we subjected the base names themselves to the normalization procedure to see which base names clustered.

For instance, a combination of the base names Dankaart, Dankwart and Dankmar into one standard Dankaart is a practical option. In such a case we do not have to worry whether Dankwart has been a misspelling or misreading of Dankaart. And if, for instance, Dankaart and Dankwart do occur both in the same dataset and denote different individuals, we often have additional information, such as surnames, to distinguish between persons of these first names.

The normalization of base names themselves requires a few iterations. As the initial standard of a name in the general database we take the base name itself. We then analyze all base names, in the order of frequency of occurrence, and derive a standard for each base name. Often this may be the base name itself, sometimes it may be a different name. After this phase we replace all standards with the new ones and repeat the entire procedure. After a few iterations, the solution is stable.

In order to decide whether different base names can be merged into one standard, it is of great importance to use all the variation in first names, as present in the main database, that relate to a single base name. The base name Antonius, for instance, has variants around Teunis. The latter variants may confuse to (variants of) the base name Tunne. It turned out, therefore, that Tunne and Antonius merged into a single standard although this might not have been expected on the basis of the names themselves. In this case, it is of interest that the Dictionary of Dutch First Names indeed says that Tunne may be related to Teunis, and that cross-references are made to both Antonius and Tunne for a number of variants.

Although the automatic procedure resulted in a very acceptable set of standards, it is necessary to consider the result manually and critically. It is well possible that some obvious clustering had been missed or that some base names had been merged that should have been left separated. The latter was the case for base names with a high frequency of occurrence, which, preferably, should not merge. The first case relates, for instance, to names from which one name is an abbreviated version of the other. Koen and Koenraad, Bert and Adelbert are some examples that were not brought together by the automatic procedure. In such cases, manual adaptations were required. In the present report, however, no manual adaptations were applied in order to show the results of the automatic procedure itself.

The procedure reduced 4456 semi-phonemic base names to 3483 standards (1834 male and 1644 female), a reduction of 22 %.

In Table 5, an example is given for the search for the best standard for Ptier (typing error of Piter). The given scores in Table 5 are similarity scores and do not reflect probabilities. The lower the number, the better the similarity. The score is based on a comparison with the name given in italics (semi-phonemic form or the kernel). Names with scores higher than 19 are not considered in the final decision, which clearly is PETRUS. Note that Piter itself is not present in the subset. A score 0 between the kernels PTYR and PYTR results from a modification of the algorithm that allows for an interchange of adjacent characters without penalty.

Table 5: *Results of the search for the best standard for Ptier (typing error of Piter).*

original	semi-phonemic	kernel	score	standard	base name
Ptier	PTYR	PTYR		PETRUS	Petrus
Pietro	PYTRO	PYTR	0	PETRUS	Petrus
Petrie	PETRY	PETR	9	PETRUS	Petrus
Pieter	PYTER	PYTER	11	PETRUS	Petrus
Piero	PYRO	PYR	12	PETRUS	Petrus
Pier	PYR	PYR	12	PETRUS	Petrus
Pierre	PYRRE	PYR	12	PETRUS	Petrus
Piere	PYRE	PYR	12	PETRUS	Petrus
Pytrik	PYTRIK	PYTRIK	14	PETRUS	Petrus
Pieterdienus	PYTERDINUS	PYTERT	16	PETRUS	Petrus
Pitter	PITTER	PITTER	16	PETRUS	Petrus
Pieterjan	PYTERJAN	PYTERJAN	21	PETRUS	Petrus
Peterus	PETERUS	PETER	25	PETRUS	Petrus
Peeting	PEETING	PEET	27	PETRUS	Petrus
Thierri	TYRRI	TYR	27	DYDERIK	Diederik
Thierry	TYRRY	TYR	27	DYDERIK	Diederik
Petri	PETRI	PETR	27	PETRUS	Petrus
Petronius	PETRONIUS	PETRON	27	PETRONIUS	Petronius
Petrus	PETRUS	PETER	27	PETRUS	Petrus
Patrick	PATRIK	PATRIK	29	PATRISIUS	Patricius
Pitrik	PITRIK	PITRIK	29	PETRUS	Petrus

VII Putting it all in place

The practical procedure to relate first names of some historical data set to standards is as follows: We first make a semi-phonemic form for the new first name. This name is the basis for all further comparisons. We then perform a lookup in the main corpus of names to determine whether the name is already known with the same gender. If so, we directly know the corresponding standard. If not, we search for the name but with the opposite gender. Again, if found, we accept the corresponding standard. If this search also fails, we compose a set of similar names with the same gender out of our corpus and choose the standard that is the most prominent of this set, in the way outlined above. If there is not any similar name, the unknown name may be essentially new to our corpus. We have the choice to add the name automatically (both as original, base name and standard), or to make a separate manual decision on this (and to ask advice from an onomastic expert).

VIII A test of the procedure

As a test of our standardization procedures we used a completely independent data set which consisted of first names from the 18th century parish registers (complete) of the city of Goes in Zeeland17. In total, there were 49.193 first names (tokens), comprising 2171 different first names (types), 947 male and 1224 female. A base name was added to each name by means of a semi-automatic procedure. The results were checked and corrected manually by Doreen Gerritzen, editor of the Dictionary of Dutch First Names. 11 first names could not be assigned to a base name by the expert, three names probably were surnames and 17 names were just initials. These names were excluded from the test, leaving 2140 first names with a total token frequency of 49.145. A total of 398 base names were needed to describe the first names (191 male and 207 female). This implies that, on the average, more than five variants existed for each base name in the Goes data set. At the level of standards, the 398 base names were slightly reduced into 365 standards.

The test involved an automatic determination of the standard of each first name on the basis of the original spelling. Results were compared to the reference standard given by the expert. The first, somewhat surprising, finding was that only 921 out of the 2140 names were directly found in our main database of 22.579 first names (459 male and 462 female). This implies that for 57.0ÿ% of the first names additional standardization procedures were required. This illustrates the necessity to have standardization procedures, even if a large database of first names is already available. With respect to frequency of occurrence, the results were less dramatic: the 921 known first names already included 91.0ÿ% of all name tokens.

The initial step in the standardization procedure was the conversion of the names to their semi-phonemic form. At this level 372 variants (17.0ÿ%) became identical to an already existing form. After application of the full standardization procedure, 1923 out of the 2140 different first names had the correct standard, an improvement from 43.0ÿ% (lookup result) to 89.9 %. The number of correct tokens increased from 91.0ÿ% to 98.2 %. The results are actually even better if we have a closer look at the deviating names.

Eight abbreviations were not correctly assigned to the right standard (although 34 others were, which is a good result). This is not surprising, because abbreviations are not yet part of our database. A limitation that can easily be overcome. For 69 first names the result apparently did not match because we did not adapt the result of the automatic distribution of base names into standards (Koen and Koenraad were not merged into one standard, for instance). Again, this is a simple adaptation that will lead to considerable additional improvement, especially since some names involved had a high frequency of occurrence.

If we add both mentioned categories as 'correct' results, we could end up with 2006 correct first names (93.7 %) with a token percentage of 99.4 %. A further inspection of the remaining 123 errors revealed that several of these related to first name variants that had never been encountered before, and were not recognized as such. The final step in the procedure is, of course, to add the new names and the corresponding, corrected, standards to the full database. This reduces the probability that closely related errors will occur again.

VIII.1 Typing errors

A part of the variation in first name spelling originates from ordinary typing errors. Sometimes these can be easily recognized (Lqurens in stead of Laurens, q and a are closely grouped together on a QWERTY key board). For other names it may be impossible to see whether the spelling is original or the result of a typing error. This is a serious problem for onomastic research on large corpora. A frequency of occurrence that is higher than one may indicate a genuine spelling variant, although, on the other hand, interchange of adjacent characters may also occur frequently in the same name (Corenlis vs Cornelis, frequency 8).

We found the following categories of typing errors (number of first name types involved between square brackets):
- interchange of adjacent characters [27] (Corenlis vs Cornelis, Jsoijna vs Josijna)
- missing character [19] (Fancois vs Francois, Anthni vs Anthoni, Hatarina vs Chatarina)
- mistyping (often with neighboring key on keyboard) [9] (Gendrik vs Hendrik)
- repeated characters [5] (Mmaria vs Maria, Annna vs Anna)
- insertion of (transposed) character [4] (Katrharina vs Katharina)

Out of the total of 64 different typing errors of this kind (3ÿ% of all first name types), 61 were automatically interpreted correctly.

VIII.2 Example

In Table 6, we give an example of variants of the base name Catharina as found in the Goes corpus. All names but one were correctly assigned to that standard. The exception was the abbreviation Cath., that was given the standard Cato. This immediately shows the problem of the choice of standards, since Cato is also derived (in French) from Catharina, but is considered in the Dictionary as an independent base name. The table makes a distinction between names directly found in the main corpus (KNOWN) and names that are new to that corpus and for which the standardization procedure had to be applied (NEW). Typing errors are marked with /t, possible typing errors with /t?. It should be noted that many other variants could have arisen from typing errors (missing of inserted characters), but the resulting name is phonologically quite acceptable, so there is no argument for an interpretation as reading error or typing error. Many close variants of Catharina, however, seem to be due to typing errors, while variants of Cathalina are under represented in the main corpus. Variants of Kaatje, Trijntje, and Katolina would never have been assigned to Catharina if no examples of their standardization were already present in the main corpus. It is uncertain, however, whether it would be better to apply Catharina, Cathalina, Kaatje, Trijntje, and Katolina as independent standards. As long as we do not have enough historical evidence of the absence of alternate use of variants out of these subsets (in some region and/or period), we better keep the very general standard Catharina.

Table 6: *All variants of Catharina as found in the Goes corpus with their frequency of occurrence and differentiated to names already known in the main corpus and those that are new to that corpus. (/t indicates a likely typo, /t? a possible typo)*

KNOWN		NEW		KNOWN		NEW		KNOWN		NEW	
name	**freq**	**name**	**freq**	**name**	**freq**	**name**	**freq**	**name**	**freq**	**name**	**freq**
Caatrina	1			Caatlijntje	2			Ka	1	Caatie	5
Catrina	61	Catrijna	15	Kaatlijntje	1			Katje	2	Kaatie	2
Katrina	57	Catrijnna	1	Catalina	5	Catalintje	8	Kaatje	18		
								Caatje	26		
Cathrina	6	Cathrijna	2			Catalijntie	3				
								Caetje	1		
Catarina	71	Catarijna	5			Catalijntje	26				
Katarina	8	Katarijna	3			Catatlijntie /t	1	Trientje	1		
								Trijntie	2		
Catharina	604	Catharijna	137	Cathalina	2	Cathalijna	3	Trijntje	6		
								Thrijntje	2		
Katharina	50	Katharijna	2	Cathalijntje	2	Cathalijntie	2				
								Trinje	1		
		Catharjna /t	1	Catelina	4			Tina	2	Tijna	1
		Cathaijna /t	1	Catelintie	1						
		Cathaina /t	1	Catelintje	1	Katolina	1	Catolina	4		
		Cattharina /t?	1			Katelijna	1	Catolijna	1		
		Katrharina /t	1			Katelijntie	1	Catolijntje	1		
		Cathariona /t?	1			Catelijntje	10				
		Cath.	1			Katelijntje	2				
		Chatarijna /t?	1			Cateleintje	1				
		Chaterina /t?	1			Cathelijntie	1				
		Hatarina /t	1			Cathelijntje	4				
						Cattelina	1				
						Kattelina	1				
						Katlina	1				
						Catlintie	2				
						Katlintje	1				
						Catlijna	2				
						Catlijntie	7				
						Katlijntie	1				
						Catlijntje	4				
						Katlijntje	4				
						Chatalijntje /t?	1				

IX Standardization without a corpus

If no corpora are available with names and their standards, one may attempt to cluster names in a data set in such a way that true variants group into a single cluster. For Dutch, this problem exists for surnames because, although we have a corpus with 144.000 surnames, no standards for these names are available. A possible procedure for this is as follows[18]. We first order the names according to their frequency of occurrence. Starting with the most frequent name, we select names that are highly similar to that name (probability>0.50). For all these names we apply the most frequent name as initial estimate of a standard. We then proceed with the next most frequent name, and so on. In the end all names have an initial estimate of a standard and we can subsequently apply the general procedure as the next iteration. We take a name, select the set of most likely candidates, and decide what standard within the group of candidates is the best estimate for our name[19]. One or two iteration rounds are sufficient for a stable solution.

As an example, we followed this procedure for the Goes corpus, without using our large main corpus. The result was a correct clustering of 1809 out of the 2140 names (84.6%), including 96.5% of all tokens. Although this is about 10% less (in types) than what we expect to realize using the basic corpus, it is a very acceptable result. It also shows, however, the benefit of the main corpus.

Differences in using a basic corpus or not are shown in Table 7. This table gives the variants of the base name Hieronymus in the Goes corpus. If no basic corpus is used, the set is divided into three clusters, with the most frequent name in a cluster as the standard: Hieronijmus, Jeloen, and Jeronimus. With no other information available, these names are considered different by the algorithm. With the help of our basic corpus, all variants except Jeloen are recognized as variants of Hieronymus, although only Jeroen and Jeronimus are actually present in that corpus. Jeloen, which may be a writing or reading error of Jeroen, is associated to Jelle.

Table 7: *Standards given to variants of Hieronymus out of the Goes corpus, with and without using the main corpus of first names.*

name	frequency	using corpus	without corpus
Hieronijmus	3	Hieronymus	Hieronijmus
Jeronimus	2	Hieronymus	Jeronimus
Jeroen	2	Hieronymus	Jeronimus
Hieronimus	1	Hieronymus	Hieronijmus
Jeronymus	1	Hieronymus	Jeronimus
Jeloen	1	Jelle	Jeloen

X CONCLUSIONS

We have shown the principles of a method that combines both rules and statistics in the process of name standardization using a large corpus of previously standardized variants. Although the method is illustrated for Dutch first names, it has a general applicability, even if no large corpora with variants and standards are available. Several aspects of the given solutions may need further refinement, but the present results are already very promising. The important feature is the possibility to reuse available onomastic and historical information in making decisions on name standardization.

NOTES

[1] For examples of a rule-based approach to name standardization see for instance L.Nygaard, 'Name Standardization in Record Linkage: An Improved Algorithmic Strategy', History and Computing 4 (1992), pp. 63-74, and Alhaug, G. and G. Thorvaldson, 'The problem of name variants; how the historian can help the anthroponymist', Proc. of the 18th Int. Congress on Onomastics, (Trier, 1993).
[2] Algorithms using similarity measures based on simple string matching are described by Gloria Guth, 'Surname Spellings and Computerized Record Linkage', Historical Methods Newsletter 11 (1976), pp. 10-19, and M. Olsen, 'Theory and Applications of Inexact Pattern Matching: A discussion of the PF474 String Co-Processor', Computers and the Humanities 22 (1988), pp. 203-213.
[3] Both rules and similarity measures are applied by G. Bouchard, and C. Pouyez, 'Name Variations and Computerized Record Linkage', Historical Methods 13 (1980), pp. 119-125.
[4] The diphone is a combination of two speech sounds (phonemes). A phoneme is the smallest speech sound that has a linguistic meaning.
[5] Lars Nygaard, 'Name Standardization', pp.63-65.
[6] Schaar, J. van der, D. Gerritzen, and J.B. Berns, Spectrum voornamenboek (Spectrum, Utrecht 1992).
[7] Courtesy to Doreen Gerritzen for making the Dictionary and its additions available and her expert help and encouragements during the whole project.
[8] Names additional to the Dictionary of Dutch First Names have been checked by its editor, Doreen Gerritzen.
[9] Some details: 1199 male base names only generated male first names, while 817 male base names generated both male and female first names. For female base names this was 1249 (only female names) and 62 names (both female and male names), respectively. 84 names are explicitly mentioned in the Dictionary of First Names uni-gender base names, for 99 names no base name was given. These figures depend strongly on the way of presentation of a base name, however.
[10] The phonemic form of a name is a string of phonemes that explain the pronunciation of the name. The graphemic form is the written form. The relation between both forms is usually complex. Depending on context the same grapheme may have various phonemic realizations, and reversely.
[11] Pronunciation of modern names and addresses in European languages are the subject of the European Union project ONOMASTICA that takes place within the frame work of the Language Research and Engineering programme.
[12] In addition to computing diphone interchange probabilities, we also compute phone interchange probabilities by generalizing diphone interchange probabilities with respect to the left character. Phone interchange probabilities are given, for instance, by (*A,*E). If a certain diphone interchange probability (MA,GE) is not known (because too little training data exist), we consider the phone interchange probability (*A,*E). If this value is also unknown we apply the default probability of 0.10. Introducing phone interchange probabilities has the implication that diphone interchanges with equal right character get a probability 1.0, because these interchanges generalize to a form like (*A,*A) which has, by definition, probability 1.0. This implies that our method is only sensitive to the left context of a phone. Full context-sensitivity can be obtained by triphones, but in many cases there will be too little training data to train them properly.

[13] On the basis of 33 phonemes, there will be a maximum of 33x33 = 1.000 diphones. The maximum number of possible diphone pairs therefore is 1.000.000. If we neglect the order of the diphones in a pair this number is reduced to 500.000.
[14] In dynamic programming, we maximize the cumulative probability C in cell (i,j) by computing the effect of coming along three possible paths to the cell (in the example: from right above, from the left, or from oblique above) and taking the maximum. In formula:

$$C(i,j) = \max \begin{cases} C(i-1,j) + \log\{P(di-1,i,dj,j)\} \\ C(i-1,j-1) + \log\{P(di-1,i,dj-1,j)\} \\ C(i,j-1) + \log\{P(di,i,dj-1,j)\} \end{cases}$$

We use log{P(dx,dy)}, which is the logarithm of the interchange probability between diphones dx and dy, to allow for summation. In practice, the computations are limited to some range around the diagonal of the matrix. It is also possible to add weighting factors to contributions from different directions.
[15] See also V.P. Concepcion and D.P. D'Amato, 'A string matching algorithm for assessing the results of an OCR process', Proc. of the 7th Int. Congress of the Association for History and Computing 1992, (Bologna, 1994) pp. 694-701, for an application in optical character reading
16 It is also possible to use the kernel of a name as the basis for the skeleton. In our experience, however, the use of the semi-phonemic form gave somewhat better results with respect to the reduced search space.
[17] Courtesy to Willem de Vries for making these data available.
[18] A prerequisite for the procedure without a corpus is that diphone interchange probabilities are known. We derived these probabilities from the large main corpus of first names. As a very coarse first estimate, these probabilities may be used for other Germanic languages as well, although language-dependent re-estimations should be applied as soon as training data become available.
[19] Because we select names within a single database, the name under investigation is always part of the selection. Because a given name is 100% similar to itself, the similarity of all names with the same kernel was artificially reduced to 30%, in order to avoid unwanted self-attraction.

The author

Gerrit Bloothooft
Department of Computer & Humanities
Achter de Dom 22-24,3512 JP Utrecht
The Netherlands
phone: +31.30.536042
fax: +31.30.536000
email: bloothooft@let.ruu.nl

Gerrit Bloothooft (Alkmaar, The Netherlands 1952) is a staff member of the Department of Computer & Humanities of Utrecht University, The Netherlands. He took his Masters in Technical Physics, his PhD on 'Spectrum and Timbre of the Singing Voice', and is presently responsible for the curriculum specialization in Speech Technology. He transferred his knowledge of automatic speech recognition techniques to the field of name standardization and historical record linkage.