

A DOUBLE METAPHONE ENCODING FOR APPROXIMATE NAME SEARCHING AND MATCHING IN BANGLA

Naushad UzZaman

Center for Research on Bangla Language Processing
BRAC University
Dhaka, Bangladesh
naushad@bracuniversity.ac.bd

Mumit Khan

Center for Research on Bangla Language Processing
BRAC University
Dhaka, Bangladesh
mumit@bracuniversity.ac.bd

ABSTRACT

Almost any word can be a Bangali name, and the name in turn is often spelled in many different ways, all of which are considered correct and interchangeable. The reason for the spelling complication is two-fold: (1) there is a large gap between the script and pronunciation in Bangla, largely attributed to the large scale Sanskritization process that started in the 12th century and continued throughout the middle ages, and (2) typical Bangla names have very different origins, from the indigenous names derived primarily from Sanskrit, to the imported Muslim names from Persian and Arabic, Christian names from Portuguese, and even the names from popular Western TV soap-operas. However, there is always a large degree of phonetic similarity in the spelling variants of a name, which is the key to searching and matching names in records. We present a Double Metaphone encoding for Bangla names, taking into account the various spelling and phonetic rules in use, which can be used by applications to search for and match names. We encode the spelling variants of a large number of names found in the literature to demonstrate that the encoding does indeed show that the variants of a name are equivalent. A name searching algorithm may employ various figures of merit to narrow the list of possibilities when searching for similar names; we demonstrate one such figure of merit using name encoding and edit distance that has shown good promise.

Keyword: Name Searching, Name Encoding, Phonetic Encoding, Double Metaphone Encoding, Bangla, Bengali

1. INTRODUCTION

Names are quite often spelled in a variety of different ways, with all variants considered equivalent. This creates a challenge when searching for and matching names in databases, and linking records among different data sources. The situation is quite complex in Bangla because of its archaic and complex orthographic rules, arising in part from the large gap between the script and pronunciation in Bangla. The Bangla language had gone through a vigorous process of Sanskritization during the 12th century, continuing throughout the middle ages, and this process in large part contributed to this gap. In addition, non-indigenous Bangla names are often derived from a variety of different origins – from Sanskrit, Perso-Arabic languages, Portuguese, and other Western languages. Most of the imported names have gone through at least one significant change in both spelling and pronunciation from the original, and have evolved as names with multiple equivalent spellings in both Bangla and English. However, the spelling variants of most of these names

have one thing in common – phonetic similarity – a feature that can be used to match these names with each other. For example, মুরতোজা /murtoʃa/ and মরতুজা /morʃa/ are common spelling variants of the same name. The similarity of the two names will be obvious to any native Bangla speaker because of the phonetic similarity along with some knowledge of Bangla name-spelling rules, but may be difficult for an algorithm because of the two-character mismatches in two different positions. One solution is to encode the names using a phonetic encoding that encapsulates Bangla orthographic rules along with the peculiarities of the name-spelling rules, and then match the resulting encoded versions. We propose a Double Metaphone encoding that is capable of matching most of the common names in all spelling variants, and in addition, providing the correct suggestion in case of a misspelled name, where the spelling error is a phonetic one.

While there are well-established phonetic similarity encodings and algorithms available for English and other Western languages [1-3], similar work for Bangla, despite it being the 4th largest language by population, is still in its infancy. Most of the recent efforts in Bangla phonetic similarity algorithm are based on Soundex [4-5], which cannot encode the sound of complex Bangla words; the Double Metaphone encoding in [6], tailored for spelling checking application, encapsulates the entire range of orthographic rules, including those involving the large repertoire of consonant clusters in Bangla. We base our proposed name encoding on [6], and extend it to support the name-spelling peculiarities in Bangla. We can use this encoding to match similar sounding names in a database, and then use other metrics to rank the match (or the suggestion in the case of a spelling checker). The rules in the encoding are derived from a large number of names found in the literature [7-9].

2. BANGLA NAME ENCODING FOR A NAME SEARCHING APPLICATION

Table 1 details the proposed name encoding for Bangla, followed by the rationale for the various mapping rules. Since any word in Bangla can be name, a fair number of the rules are inherited from the spelling encoding described in [6], and so we describe the rationale for only those that are specifically for names. As in [6], we assume that the Bangla text is encoded using Unicode Normalization Form C (NFC) [10]. The dashed circle in the glyph for some of the letters is a placeholder for the consonant (or consonant cluster) that the diacritic is attached to. The consonant clusters are displayed as conjuncts in the Bangla script.

Table 1. Bangla Name encoding table

No	Letter	Name	Unicode	Code	Context	Example
1	্	SIGN VIRAMA Hasant	\u09CD	Not Coded		আব্দুল (abdul)
2	ঁ	CANDRABINDU	\u0981	Not Coded		চাঁদনী (chādni)
3	অ	A	\u0985	Not Coded		
4	আ	AA	\u0986	Not Coded		
5	া	SIGN AA	\u09BE	Not Coded		
6	ই	I	\u0987	Not Coded		
7	ঈ	II	\u0988	Not Coded		
8	ি	SIGN I	\u09BF	Not Coded		
9	ী	SIGN II	\u09C0	Not Coded		
10	উ	U	\u0989	Not Coded		
11	ঊ	UU	\u098A	Not Coded		
12	ূ	SIGN U	\u09C1	Not Coded		
13	ু	SIGN UU	\u09C2	Not Coded		
14	ও	O	\u0993	Not Coded		
15	়ো	SIGN O	\u09CB	Not Coded		
16	এ	E	\u098F	Not Coded		
17	ে	SIGN E	\u09C7	Not Coded		
18	ঐ	AI	\u0990	Not Coded		
19	ৈ	SIGN AI	\u09C8	Not Coded		
20	ঔ	AU	\u0994	Not Coded		
21	ৌ	SIGN AU	\u09CC	Not Coded		
22	ক	KA	\u0995	“k”		
23	খ	KHA	\u0996	“k”		
24	ক্ষ		\u0995 \u09CD \u09B7	“k”	@ the beginning	ক্ষত (khaṭo)
25			\u0995 \u0995 \u09CD \u09B7	“kk”	@ middle/end	দক্ষ (ḍokkha)
26	গ	GA	\u0997	“g”		
27	ঘ	GHA	\u0998	“g”		
28	ঙ	NGA	\u0999	“ng”		বাঙলা (baṅla)
29	ং	ANUSVARA	\u0982	“ng”		বাংলা (baṅla)
30	চ	CA	\u099A	“s”		
31	ছ	CHA	\u099B	“s”		
32	শ	SHA	\u09B6	“s”		শাদমান (ṣadman)
33	স	SA	\u09B8	“s”		সামীন (ṣamin)
34	ষ	SSA	\u09B7	“s”		
35	য	YA as phalaa	x\u09CD\u09AF	Not Coded	@ the beginning as YA phalaa	শ্যমা (ṣama)
36			...xy \u09CD z \u09CD \u09AF	Not Coded	@ middle/end with conjuncts	সক্ষ্যা (ṣondhya)
37			...xy \u09CD \u09AF	Doubles: yy	@ middle/end	সত্যজিত (ṣatṭojit)
38	য	YA	\u09AF	“j”		
39	জ	JA	\u099C	“j”		
40	ঝ	JHA	\u099D	“j”		
41	ঞ	NYA	\u099E \u099A	“n”	Before CA	অঞ্চল (aṅcol)
42			\u099E \u099B	“n”	Before CHA	বাঞ্ছা (bancha)
43			\u099E \u099C	“n”	Before JA	মঞ্জু (maṅḡu)
44			\u099E \u099D	“n”	Before JHA	যঞ্ছা (yaṅḡa)
45			\u099A \u099E	“n”	After CA	যাচ্ছা (yaḡa)

No	Letter	Name	Unicode	Code	Context	Example
46			\u099E \u0985 \u099E\u0987	<i>Not Coded</i>	Before A I	মিঞা (miã)
47			\u099C \u09CD \u099E	“ge”	@ the beginning after JA	জ্ঞাত (gæʈa)
48			... \u099C \u09CD \u099E	“gg”	@ middle/end after JA	বিজ্ঞান (biggæn)
49			\u099E \u09CD	“n”	With hasant	নঞ (non)
50	ট	TTA	\u099F	“T”		
51	ঠ	TTHA	\u09A0	“T”		
52	ড	DDA	\u09A1	“D”		
53	ঢ	DDHA	\u09A2	“D”		
54	ঋ	VOCALIC R	\u098B	“ri”	@ the beginning	ঋতু (riṭu)
55			x\u098B	“ri” xri	@ middle/end	বিকৃত (bikkriṭo) বিকৃত (bikriṭo)
56	র	RA as <i>phalaa</i>	x\u09CD \u09B0	“r”	@ the beginning	প্রকাশ (prokaʃ)
57			...x\u09CD \u09B0	“r”	@ middle/end	রাত্রি (raṭtri) রাত্রি (raṭri)
58	র	RA	\u09B0	“r”		
59	ড়	RRA	\u09DC	“r”		
60	ঢ়	DDHA	\u09A2	“r”		
61	ন	NA	\u09A8	“n”		
62	ণ	NNA	\u09A3	“n”		
63	ত	TA	\u09A4	“t”		
64	থ	THA	\u09A5	“t”		
65	দ	DA	\u09A6	“d”		
66	ধ	DHA	\u09A7	“d”		
67	প	PA	\u09AA	“p”		
68	ফ	PHA	\u09AB	“p”		
69	ব	BA as <i>phalaa</i>	x\u09CD \u09AC y...	<i>Not Coded</i>	@ the beginning	স্বপ্না (ʃɔpna)
70			...x\u09CD y \u09CD \u09AC	<i>Not Coded</i>	BA <i>phalaa</i> with conjuncts	তত্ত্ব (toṭṭɔ)
71			... \u09AC \u09CD \u09AC	“bb”	After BA as conjuncts	তিব্বত (tibboṭ)
72			... \u09AE \u09CD \u09AC	“mb”	After MA as conjuncts	লম্ব (lombo)
73			... \u0997 \u09CD \u09AC	“gb”	After GA as conjuncts	দিগ্বিদিক (ḍigbiḍik)
74			\u0989 \u09A6 \u09CD \u09AC	“udb”	After Ud- (U DA BA...)	উদ্বৈগ (udbeg)
75			...y \u09CD \u09AC	Doubles: yy	@ middle/end	বিশজিত্ব (biʃʃɔṭit)
76	ব	BA	\u09AC	“b”		
77	ভ	BHA	\u09AD	“b”		
78	ম	MA as <i>phalaa</i>	x\u09CD \u09AE...	<i>Not Coded</i>	@ the beginning	স্মরণ (ʃɔron)
79			...x\u09CD y \u09CD \u09AE	<i>Not Coded</i>	MA <i>phalaa</i> with conjuncts	স্মৃক্ষ (ʃuḱkho)
80			... \u0995 \u09CD \u09AE	“km”	After KA as conjuncts	রুক্মিনী (rukmini)
81			... \u0997 \u09CD \u09AE	“gm”	After GA as conjuncts	যুগ্ম (jugmɔ)
82			... \u0999 \u09CD \u09AE	“ngm”	After NGA as conjuncts	বান্ধম (baṅmoi)
83			... \u099F \u09CD \u09AE	“tm”	After TTA as conjuncts	কুট্মল (kutmol)
84			... \u09A3 \u09CD \u09AE	“nm”	After NNA as conjuncts	মৃণ্ম (mrinmɔḍe)
85			... \u09A8 \u09CD \u09AE	“nm”	After NA as conjuncts	জন্ম (ʃɔnmo)
86			... \u09AE \u09CD \u09AE	“mm”	After MA as conjuncts	রুম্মান (rumman)
87			... \u09B2 \u09CD \u09AE	“lm”	After LA as conjuncts	গুল্ম (gulmo)
88			... \u09B6 \u09CD \u09AE	“sm”	@ middle/end with SHA	কাশ্মীর (kaʃmir)
89			... \u09B7 \u09CD \u09AE	“sm”	@ middle/end with SSA	কুম্মাভ (kuʃmandɔ)

No	Letter	Name	Unicode	Code	Context	Example
90			... \u09B8 \u09CD \u09AE	“sm”	@ middle/end with SA	সুমিতা (ʃuʃmiʃa)
91			...y \u09CD \u09AE	Doubles: yy	@ middle/end otherwise	রশ্মি (raʃʃe)
92	ম	MA	\u09AE	“m”		
93	য়	YYA	\u09DF	<i>Not Coded</i>		মিয়া (mia), সায়েম (saiem)
94	ল	LA	\u09B2	“l”		
95	হ	HA	\u09B9 \u09CD \u098B	“ri”	HA with Vocalic R	হৃদয় (rhiidoi)
96			\u09B9 \u09CD \u09B0	“r”	HA with R as <i>phalaa</i>	হ্রদ (rod)
97			\u09B9 \u09CD \u09A8	“nn”	HA with NA	পূর্বাঙ্ক (purbannā)
98			\u09B9 \u09CD \u09A3	“nn”	HA with NNA	প্রাঙ্ক (prannfi)
99			\u09B9 \u09CD \u09AE	“mm”	HA with MA	ব্রঙ্কা (brommfia)
100			\u09B9 \u09CD \u09AF	“jj”	HA with YA as <i>phalaa</i>	উহ্য (uhfi)
101			\u09B9 \u09CD \u09B2...	“l”	HA with LA @ beginning	ফ্লাদ (lfiad)
102			... \u09B9 \u09CD \u09B2	“ll”	HA with LA @ middle/end	আফ্লাদ (allfiad)
103			\u09B9 \u09CD \u09AC	“h” “o”	HA with BA	আফ্বান (aovan) আফ্বান (afiofian)
104	হ	HA	\u09B9	<i>Not Coded</i>	Otherwise	
105	ঃ	Visarga	One to one Transformations	Encode using rest of the rules after transformation		মোঃ → মোহাম্মদ (mohammød)
106			x\u0983 y...	Doubles: yy	@ the middle	দুঃসময় (duʃʃomø)
107			x\u0983	“h”	@ the end strlen == 1 2	উঃ (ufi), বাঃ (bafi)
108			x\u0983	<i>Not Coded</i>	Otherwise @ the end	পুনঃ (puno)

There are a total of 108 transformations in the encoding, which includes the vowels, consonants, and conjuncts in all different contexts and a few one-to-one transformations in No. 105, which will be expanded as more data is available.

3. RATIONALE FOR BANGLA NAME ENCODING RULES

The transformation or rules described in Table 1 were derived from a large set of names in the literature [8-10], which include both common and uncommon names, and of different origins. We describe the rationale for the name-encoding transformations below.

Transformations 1, 2: Reason why SIGN VIRAMA (Hasant) and CANDRABINDU are to be *Not Coded* can be found in [4].

Transformations 3 – 21: In our encoding, vowels are *Not Coded*. This is to account for pronunciation differences from person to person, or region to region, where the differences are due to vowels.

The following is an example of a name which is spelled (and pronounced) differently by native speakers:

মরতুজা /mørʃuʃa/, মুরতোজা /murʃoʃa/, মরতোজা /mørʃoʃa/, মোরতুজা /mørʃuʃa/

In our encoding, all of these variants are encoded as “mrtj”, and can be matched against each other regardless of spelling variation. Table 2 shows a few more such examples justifying the decision to mark vowels as *Not Coded*.

Table 2. Example of vowels encoding

Similarly pronounced names	Encoding
নাইম /naim/, নঈম /noim/	“nm”
নাহলীন /nahleen/, নেহলীন /nehleen/	“nlñ” ¹
নওশাদ /noʃad/, নাওসাদ /naoʃad/	“nsd”
সুমিন /ʃumin/, সোমেন /ʃomen/	“smñ”
রাসেদ /raʃed/, রশিদ /raʃid/	“rsd”
মুস্তোফা /mustofa/, মোস্তফা /mostofa/	“mstp”

Transformations 22-29: Names are just words, so the rationale is the same as for a spelling checker [6].

Transformations 30-34: In encodings designed for spelling checkers [4, 6], শ (/s/, /ʃ/), স (/s/, /ʃ/), ষ (/ʃ/) are encoded the same as they are very close in pronunciation; similarly for চ /c/ and ছ /ch/. However, in case of name encoding, we encode all 5 of these letters to the same code. The reason is that in Bangla, the sound /s/ is expressed using স (/s/, /ʃ/), but sometimes also with ছ /ch/. Our solution is to encode স (/s/, /ʃ/) and ছ /ch/ the same

¹Rationale for হ to be Not Coded is according to Transformation

way. Since these two letters belonged to two different groups, we combine the two groups and use the same code.

Example: The name /salam/ is usually written as সালাম /ʃalam/, but often also as ছালাম /chalam/. সালাম /ʃalam/ is phonetically more appropriate as স sounds like /s/ and /ʃ/; to make matters worse, even if /salam/ is written as ছালাম /chalam/, it is still pronounced as /salam/. Following are few more examples of names where স (/s/, /ʃ/) and ছ /ch/ are both pronounced as /s/, to justify the decision to make স (/s/, /ʃ/) and ছ /ch/ in the same group.

Table 3. Example of স and ছ

Name with pronunciation (according to rules)	Both Locally pronounced as	Encoding
বাসেত /baʃeʈ/, বাছেত /bacheʈ/	/baset/	“bst”
মুকসিত /mukʃit/, মুকছিত /mukchit/	/muksit/	“mkt”
নামিস /nafɪʃ/, নামিছ /nafich/	/nafis/	“nfs”
হাসিনা /haʃina/, হাছিনা /hachina/	/hasina/	“sn” ²

Transformation 35: At the beginning of a word, and if the word is অ-কারাত্ত /ɔ/ or আ-কারাত্ত /a/, it is pronounced as /æ/ and if there is a ই or উ after য phalaa, then it is pronounced as এ /e/. Both of these were encoded to “e” in [6]. But in case of names, vowels are *Not Coded*. So, it is *Not Coded*.

Example: শ্যামা /ʃæma/ and শেমা /ʃema/ are both encoded as “sm”, which are similar sounding.

Transformations 36-92: Names are just words, so the rationale is the same as for a spelling checker [6].

Transformation 93: In names, য় is almost silent; it mainly gets the sound of attached vowel and sometimes causes nasalization. So, it is *Not Coded*.

Example: মিয়া /miã/ → “m”, সায়েম /saiẽm/ → “sm”, সারিয়া sariã/ → “saria”.

Transformations 94-103: Names are just words, so the rationale is the same as for a spelling checker [7].

Transformation 104: In names, হ is usually silent or almost silent. So, it is *Not Coded*.

Table 4. Example of হ

Names With হ	Names Without হ	Encoding
যাহরা /jaɦra/	যারা /jara/	“jr”
নাবিলাহ /nabilah/	নাবিলা /nabila/	“nbl”
তাহমিনাহ /taɦminah/	তামিনা /tamina/	“tmn”
ফাহমিদা /faɦmiða/	ফামিদা /famida/	“pmd”

Transformation 105: The equivalent of the English “.” in name abbreviations and titles is Bangla ং, e.g., মোহাম্মদ /mohammad/. Since these are often ad-hoc, one-to-one transformations are used before encoding process. This set of transformations will of course be expanded as more new cases

²Rationale for হ to be *Not Coded* is according to Transformation

come in use. So, to encode মোঃ we will first transform it to মোহাম্মদ /mohammad/ before the final encoding /mohammad/.

Table 5. One to one transformation of ং

Short cut	Elaborated form	Encoding
মোঃ	মোহাম্মদ /mohammad/	“mmmD”
ডঃ	ডক্টর /dɔktor/	“DkTr”
ডাঃ	ডাক্টর /dactar/	“DkTr”
এডঃ	এডভোকেট /advokæt/	“DbkT”

Table 5 lists just a few of the very common – there is quite a large number in use, and new cases do get added to the colloquial use over time.

Transformations 106-108: Names are just words, so the rationale is the same as for a spelling checker [6].

4. APPLICATION TO NAME SEARCHING AND FIGURE OF MERIT

One important application of the proposed name encoding is in searching for names in databases. A naïve approach is to search for the encoded string in the database, which may return a large number of names, many of which are not considered equivalent to the name being searched for. The encoding removes all the vowels and the letters marked as *Not Coded*, so the encoded string is typically much shorter than the original name. Since many other names may map to this shorter encoded string, the match returns many irrelevant names in addition to the “equivalent” ones. To avoid this problem, other figures of merit must be used to narrow this list to include only the desired set, and to rank the resulting set in order of relevance [11]. We propose one such figure of merit (FOM) that uses a weighted sum of the orthographic and phonetic edit-distances to exclude dissimilar names from the query result. We outline to steps to search for a name মরতুজা /mɔrtuʃa/ below. Table 6 shows a pre-encoded list of names to search, with various columns that are computed during the various steps.

- 1) Encode the name to search for: মরতুজা /mɔrtuʃa/ → mrtj.
- 2) Compute the Levenshtein edit-distance [12] (column ED) between the candidate name and each of the names from list.
- 3) Compute the edit distance score (column EDscr) between the two strings s1 and s2 from ED: EDscr = (maxLen(s1, s2)-ED)/maxLen(s1, s2).
- 4) Compute the phonetic edit-distance (column PED), using the encoded versions.
- 5) Compute the phonetic edit distance score (PEDscr) from PED: PEDscr = (maxLen(s1, s2)-ED)/maxLen(s1, s2).
- 6) The figure of merit (FOM) is the weighted sum of PEDscr and Edscr, with PEDscr as the dominant factor: (PEDscr + Edscr/10)/1.1 and value ranges from 0 to 1.

Table 6: Generating suggestions for names using name encoding and other trivial methods

Names	Encoding	E D	EDscr	P E D	PE Dscr	FOM
জুমিন						
/ʃumin/	“smn”	6	0	4	0	0

রশিদ /rɔʃid/	"rsd"	5	0.167	4	0	0.02
মুস্তোফা /mustofa/	"mstp"	5	0.375	2	0.5	0.49
বাহেত /bachet/	"bst"	6	0	3	0.25	0.23
মুকসিত /mukʃit/	"mkst"	5	0.167	3	0.25	0.24
মরতুজা /mɔrʃuʃa/	"mrtj"	0	1	0	1	1
মুরতোজা /murʃoʃa/	"mrtj"	2	0.714	0	1	0.97
মরতোজা /mɔrʃoʃa/	"mrtj"	1	0.833	0	1	0.98
মোরতুজা /mɔrʃuʃa/	"mrtj"	1	0.857	0	1	0.99

We can use the FOM to rank the matches returned by the query, which in this case does correspond to the expected convention for the Bangla name মরতুজা /mɔrʃuʃa/. We expect that a name searching algorithm will need to tailor the figure of merit to the application domain.

5. CONCLUSION

We present a Double Metaphone encoding for Bangla, tailored for name searching and matching application. This encoding encapsulates the complex spelling rules for Bangla, and in addition, takes into account the special cases for names. Name searching and matching applications can use this encoding to provide a much smaller set of suggestions, which in turn can be ranked using other methods, such as string edit distance methods or other similarity measures.

6. ACKNOWLEDGMENT

This work has been partially supported by BRAC University, Southtech Limited and the PAN Localization Project (www.PANL10n.net) grant from the International Development Research Center, Ottawa, Canada, administered through Center for Research in Urdu Language Processing, National University of Computer and Emerging Sciences, Pakistan.

7. REFERENCE

- [1] The Soundex Algorithm, available online at http://www.archives.gov/research_room/genealogy/census/soundex.html.
- [2] Lawrence Phillips, Hanging on the Metaphone, *Computer Language*, 7(12), 1990.
- [3] Lawrence Phillips, The Double Metaphone Search Algorithm, *C/C++ Users Journal*, 18(6), June, 2000.
- [4] Naushad UzZaman and Mumit Khan, A Bangla Phonetic Encoding for Better Spelling Suggestion, *Proc. 7th International Conference on Computer and Information Technology*, Dhaka, December, 2004.
- [5] Md. Tamjidul Haque and M. Kaykobad, Use of Phonetic Similarity for Bangla Spell Checker, *Proc. 5th International Conference on Computer and Information Technology*, Dhaka, December, 2002.
- [6] Naushad UzZaman, Phonetic Encoding for Bangla and its Application to Spelling checker, Transliteration, Cross language information retrieval and Name searching, Undergraduate thesis (Computer Science), BRAC University, May 2005.
- [7] Sadikur Rahman, *Apnar shontaner prio naam*, Salahuddin Boi Ghar, Bangla Bazar, Dhaka, September, 2003.
- [8] Anis Ahmed, *Bissher shreshtho 110 monishi*, Dhaka, Bangladesh.
- [9] BANGLAPEDIA: National Encyclopedia of Bangladesh, Dhaka, Bangladesh, 2003.
- [10] The Unicode Consortium, *The Unicode Standard, Version 4.0*, Addison-Wesley, 2003.
- [11] NameX Technology, available online at <http://www.imagepartners.co.uk/Thesaurus/AboutNameX.htm>.
- [12] Levenshtein edit distance algorithm, available online at <http://www.nist.gov/dads/HTML/Levenshtein.html>.