

Construction and Performance of a Language Recognizer

Deliverable 8 of the *Modus Operandi* project
carried out by the JRC for the
European Anti-Fraud Office OLAF

Administrative arrangement 14408-98-10

September 1999

Johan Hagman

**Joint Research Centre of the European Commission
Institute for Systems, Informatics and Safety (ISIS)
Risk Management and Decision Support Unit (G3)
Anti-Fraud Information Management (AIM)**

T.P. 361
21020 Ispra (VA), Italy
tel / fax + 39-0332-785646 / 9098
johan.hagman@jrc.it
<http://www.jrc.cec.eu.int/jrc>

Construction and Performance of a Language Recognizer

1	Introduction	1
2	Preparation of training material	1
2.1	Definition of the alphabet	1
2.2	Creation of bigram matrixes	2
3	The running recognizer	2
3.1	Loading the input files	2
3.2	Language-representing data types	3
3.3	Graphotactic language characteristics	3
3.4	Transliteration	5
3.5	Recognition algorithms	6
3.6	Presenting the results	8
3.7	Language recognition using different word context width	10
3.8	The LR generating input to other programs	12
4	Operating the LR	12

Note that in our processing, the feature upper/lower case is not considered. All characters are converted to either all upper or all lower case. It is quite intuitive, knowing the typical statistical distribution of the diacritic characters in these languages, that by simply scanning such a letter count, one could make a good guess of what language has been examined. But even if we were disposed of diacritics, we would nevertheless have quite some information left indicating particularities of the single languages; if there are plenty of 'k' we would go for a Germanic language and are there also many 'w' we could most likely exclude from this group the Nordic languages. On the expense of 'k' we find higher rates of 'c' and 'q' in the Latin languages and in that subgroup we may use the presence of 'j' and 'x' to lower the possibility for the text of being Italian but raising it for French. The letter 'y' would also lower the probability regarding both Italian and Portuguese but equally raise the odds for both Spanish and French, and so forth.

2.2 Creation of bigram matrixes

Letter statistics tell a lot, but better still are sequence tables showing how letters follow each other. If we talk about statistics of *letter pairs* in a language, we refer to the *bigrams* of that language. A text (written in the same language or not) can be even more specifically characterized than that; one could calculate and describe it with its *trigrams* as well, thus showing in a three-dimensional table the frequencies of the sequences of three characters. Note that a separator (a common one will do, representing the space character and all other non-alphabetical characters) must also be included in such a bi- or trigram as it will give as valuable information as any of the alphabetical characters – namely which of the latter are common in word startings and endings.

A special, independently working program was built for the creation of bigram tables based on whatever text supplied by the user in simple text format. That program was fed with typical samples of each of the ten languages in question in order to create a 'character pair', or a bigram matrix for each language. In Figure 2 we see the top left part of one for German. This matrix is defined on the 56 characters of Figure 1 plus the general separator character just described above, in the figure denoted as '_'. This table was based on 250,111 character pairs which is not particularly big a text sample but the bigram does not change much once a representative enough quantity of text is analyzed. Having huge and too varied model texts might even bring some undesired oddities into the bigram matrix via proper names and formatting codes which leave strange looking letter traces after the conversion from formatted text to simple text format. The partition we see below constitutes only 1.5% of the whole matrix.

German	57	250111								
	_	a	b	c	d	e	f	.	.	.
_	0	1624	2095	978	6103	2331	635	.	.	.
a	242	2073	59	266	505	15	351	.	.	.
b	86	176	140	2	0	2004	0	.	.	.
c	41	118	1	6	2	205	0	.	.	.
d	2000	528	59	0	115	7883	1	.	.	.
e	9262	87	583	715	1035	2235	528	.	.	.
f	237	27	9	1	264	211	174	.	.	.
g	2152	347	8	4	204	3890	19	.	.	.
.
.
.

Figure 2 Top left corner of a bigram matrix for German. The character '_' denotes 'space' or other delimiters/separators. Thus: 1624 words in this sample start with an 'a' and there is only one word containing the sequence '~fc~' but not a single one containing '~cf~'.

Just commenting on the three most frequent sequences found in Figure 2: many German words in the sample end in ~e (9,262); the letter d is often followed by e, forming ~de~ (7,883), and many words start with d~ (6,103). The second and especially the third of these sequences occur in the frequent German definite articles, standing in various case forms.

3 The running recognizer

3.1 Loading the input files

Starting up, the LR first reads a file defining the alphabet. It simply consists of what is shown by Figure 1 in a little simple ASCII file. Secondly the language matrices are loaded. Currently they are all contained in one file, following each other. Even this file is of simple text format. Although being easily extendable to more languages, this LR was implemented to be immediately and solely applied on non-Greek EU texts. Therefore, and for the sake of efficiency, the number of languages has not yet been parametrized, i.e. the program is currently hard-

3.4 Transliteration

To be considered when building language recognizers is that where only the “English letters” are provided for, writers of different languages make up for this restriction in different ways. The acute and grave accents are often substituted by an apostroph, or, more elegantly, by a single quotation mark after the vowel, for example: both é and è become either e' or e'. The problem with this is that the characters ' and ’ just as often are used for quotations, so they are not reliable neither for describing nor recognizing languages – unless a much more complex algorithm is worked out which actually analyses the use of these characters in some clever way. In the LR implementation outlined here, they are simply fused with other non-alphabetical characters into the general delimiter ‘ ’ or ‘ _’. More seldom do we encounter other diacritic characters transliterated into combinations of a letter plus some other character, as e.g. ä→a: ê→e^ ð→o~ or ç→c, and neither are these solutions recognized by our LR as diacritic characters. French orthography actually recommends text written all in upper case *not* to have any accents, so maybe writers of French would feel freer to simply drop them all even when writing normal-case text, if restricted to the English alphabet. Writers of German and the Nordic languages have to opt for one of two alternatives concerning the vowels: either skipping the diacritic signs (ääöøü→aaouu) or to transliterate them into the combination of vowels of which they were historically created (â→aa ä→ae ö→oe ø→oe and ü→ue), sometimes with funny results: the Finnish surname Jääaro→Jaeaearo for instance. The Danish ligature æ is simply released to ae and the German ß is transliterated into ss (as it is written in e.g. Swiss German since long).

Again, whether non-English EU texts are written with unlimited access to diacritics or not is more than a marginal curiosity when it comes to automatic language identification. To estimate the inaccuracy of e.g. a Finnish text with the umlauts ä/ö simply ignored a/o, we may in Figure 3 *subtract* the *information* given by the ä and ö rows and columns and *add* the same amount as *disinformation* to the a and o rows and columns, respectively. In case these umlauts ä/ö were transliterated into ae/oe, we would also add twice as much distortion to the single row/column e as to either one of the rows and columns a and o and this would certainly have some impact on the accuracy of the LR. But to what extent? In a series of experiments we ran the LR, trying to recognize one sample each of the EU languages plus one extra, transliterated version of each the two central EU languages German and French.. The result of this test is shown in Figure 6.

seen as →	FIN	SWE	DAN	GER	DUT	ENG	FRE	ITA	SPA	POR
FIN	88.4	0.0	1.2	0.0	0.0	4.9	1.7	2.6	0.0	1.2
SWE	4.3	63.0	8.6	3.3	6.4	1.7	2.9	6.4	1.9	1.4
DAN	2.5	9.9	57.6	9.5	5.9	5.4	1.8	4.3	0.7	2.5
GER	1.6	5.1	5.5	75.8	4.9	1.6	1.2	0.8	0.4	2.9
GER	5.8	5.3	4.6	56.5	12.5	3.6	3.4	2.2	1.5	4.5
DUT	1.5	9.2	2.9	4.8	71.5	4.8	4.6	3.5	0.4	4.6
ENG	1.9	1.7	4.9	5.3	2.8	60.9	7.0	3.4	5.3	6.8
FRE	2.3	0.3	3.9	5.4	7.9	3.7	58.8	4.2	8.8	3.6
FRE	2.9	1.9	3.8	4.4	9.0	7.7	49.2	4.4	8.3	8.3
ITA	1.6	2.5	2.1	2.1	2.7	1.8	3.1	68.8	9.5	5.8
SPA	1.3	1.5	0.4	0.9	13.8	3.4	6.2	4.6	53.5	14.3
POR	0.2	0.6	1.9	0.9	4.9	1.1	5.8	7.0	6.6	70.9

Figure 6 Result of experiment where the LR was fed with one small text each of the ten languages plus one transliterated version each of German and French. The numbers are percentages of total match points assigned to each text in the sample and they sum up to 100.0 for each row. I.e. in the DUT/ SWE cell 9.2 means that the Dutch text was estimated to be Swedish by a probability of 9.2 %. Note e.g. how transliterated German became more Dutch-like and transliterated French English-like.

Figure 6 gives an idea of how unique the graphotax of each language is (although based on tiny test samples); we see in the diagonal that the Finnish letter order has more unique features than does e.g. the Spanish text which in this test also looked Portuguese (less surprisingly) and Dutch (which is more surprisingly – maybe the text contained many Dutch proper names). Well, what is the performance of this LR? This has not yet been formally measured. If we define it as the value of the champion candidate for each text divided by the sum of that value plus the second best candidate, we would have $88.4 / (88.4 + 4.9) = 95\%$ for Finnish; $49.2 / (49.2 + 9.0) = 84\%$ for French without accents; and just $56.5 / (56.5 + 12.5) = 82\%$ for German without diacritic signs. Thus, the language identification seems to be fairly robust here in spite of lacking diacritics; the champion candidate is always the desired one and the margin to the second best candidate is quite reassuring.

When this LR was later put to work, sorting about 30,000 pieces of text according to the language, the French texts in this collection of texts written in all EU languages were of special importance and they were also varying as for the usage of diacritic characters. For that application it was motivated to supply the LR with both matrices for French referred to in Figure 7 in order to optimize the identification of both versions of French texts.

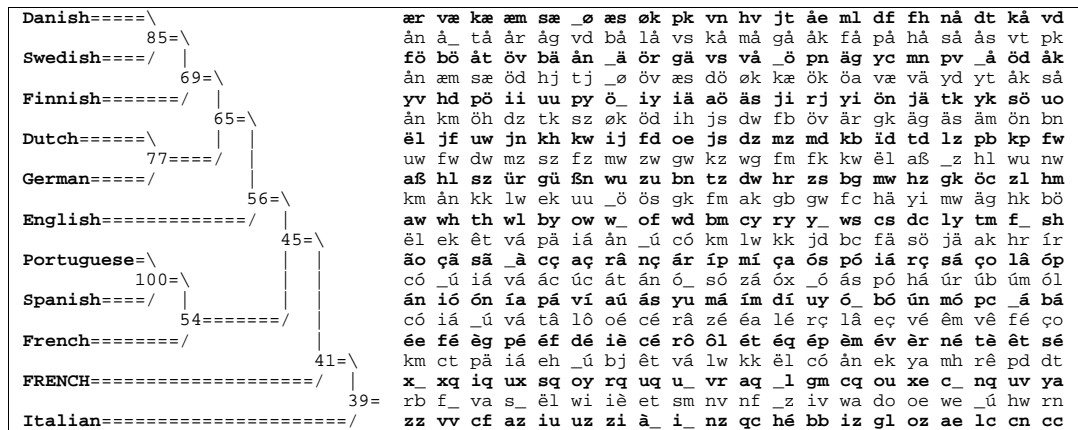


Figure 7 The dendrogram of Figure 5 re-calculated with two bigram matrices for French, i.e. the ordinary one (**French**) with diacritics and another (**FRENCH**) based on the same training texts where all diacritic characters were substituted with the respective non-diacritics. We note that there are still many typical French sequences (e.g. those containing 'x' and 'q') which characterize the texts even without diacritics. Note also that the presence of this additional matrix changes the values of the average matrix used in extracting language-peculiarities, which explains why the values in the nodes are slightly altered.

3.5 Recognition algorithms

When the real processing starts, the text to be analyzed is read, character by character, and the relative frequency for each character pair passed is checked for each language in the respective matrix. There is a counter for each language which is incremented proportionally as to how the bigram just read is representative for that language. For instance: reading the sequence ~ée~, the SWE counter is increased somewhat as this bigram is present in fairly common words like 'idéer' ('ideas') and 'muséer' (plural of 'museum'), but the FRE counter will be increased considerably more since this sequence is so typical of, and highly frequent in French.

Let us follow an example; we feed the text shown in Figure 8 into the LR and in Figure 9 we see how counters, for each language, are incremented, letter by letter, as the text is scanned. There are two counters for each language; those within [...] which show the probability distribution in percentage of the latest bigram read, and those within [...] which show the percentage of the accumulated probability distribution of all bigrams read up to that point.

Romanian police announced on 8 March that they are holding two individuals for attempting to sell stolen radioactive material, according to press reports. A police spokesman announced that the two had in their possession 82 kg of radioactive material including low enriched uranium. Officials also found reportedly secret documents stolen from the Research and Design Center for Radioactive Metals.

Figure 8 A news text in English (taken from the web) used for testing the LR.

So, if it was equally common to have words starting with r~ in all ten languages, all these counters would have started with the value 10 on the first line in Figure 9. This is however not the case, as this initial bigram apparently is about 60% more common in Swedish than in the average of these languages, and about 80% less common in Finnish with respect to the same average. But, as we read the next character pair (i.e. ~r0~), the probability of this word being Finnish gains strength as this bigram is statistically more Finnish than anything else; in fact, this is also 60% more common than average but in favour of Finnish, and ~r0~ seen as a Danish bigram is twice as rare as average within this ten languages. In the [...] separated columns to the right in Figure 9 we see how the accumulated values change throughout the words – the champion language after the first word separator (here: the space character) is given to the right of the [...] columns and, as we can see, the first word actually ends up being considered Finnish after a tough fight with Swedish, Italian, and Portuguese, all three of which scored 12 before the space character was encountered. After the second word, Italian has taken the lead and the field behind it is very even, with Finnish, English, French, Spanish and Portuguese all having their accumulated probability scoring 11. The language indicators within [...] to the extreme right tell which language is the champion language of the entire paragraph as read so up to that point. This leading position could be calculated in at least two ways, either by considering the current accumulated value within [...], or by simply counting the number of times each single language has been a word champion up to that point. The former way is applied in this example. This race reminds a little of those funfair games where a mechanized horse race field have its horses moved triggered by players gaining points by throwing balls at concentric circles until somebody wins.

	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por		
r	[2	16	7	6	6	12	12	13	11	15]	2	16	7	6	6	12	12	13	11	15]		
o	[16	13	5	4	8	10	7	12	13	12]	9	14	6	5	7	11	10	13	12	14]		
m	[9	20	12	4	7	11	9	8	8	12]	9	16	8	5	7	11	10	11	11	13]		
a	[14	6	6	6	8	10	11	11	12	17]	10	14	8	5	7	10	10	11	11	14]		
n	[16	10	8	6	19	10	10	8	7	6]	11	13	8	5	10	10	10	10	10	12]		
i	[20	14	10	7	6	8	5	13	8	9]	13	13	8	6	9	10	9	11	10	12]		
a	[13	7	2	1	2	10	4	21	18	21]	13	12	7	5	8	10	8	12	11	13]		
n	[16	10	8	6	19	10	10	8	7	6]	13	12	7	5	9	10	9	12	11	12]		
p	[22	9	6	17	20	9	6	3	9	0]	14	12	7	6	10	10	8	11	10	11]	Fin	{Fin}
o	[10	5	6	2	3	10	17	15	16	16]	14	11	7	6	10	10	9	11	11	11]		
l	[6	5	5	2	2	12	19	16	18	15]	13	10	7	6	9	10	10	12	12	12]		
i	[16	15	11	9	6	6	4	18	7	8]	13	11	7	6	9	10	10	12	11	11]		
c	[21	7	10	10	12	6	8	15	5	6]	14	11	7	6	9	9	9	12	11	11]		
e	[0	0	1	22	4	15	10	17	18	14]	13	10	7	7	9	10	9	13	11	11]		
a	[0	0	5	1	3	20	32	12	12	15]	12	9	7	7	8	11	11	13	11	11]		
n	[1	4	11	7	14	12	16	14	11	10]	11	9	7	7	9	11	11	13	11	11]	Ita	{Ita}
n	[4	7	13	9	6	15	10	12	10	15]	11	9	7	7	9	11	11	13	11	12]		
n	[16	10	8	6	19	10	10	8	7	6]	11	9	7	7	9	11	11	12	11	11]		
n	[20	8	13	14	7	5	19	13	0	0]	12	9	8	7	9	11	11	12	11	11]		
o	[5	16	3	3	5	8	6	27	9	18]	11	9	7	7	9	10	11	13	10	11]		
u	[13	0	0	0	9	27	40	0	0	10]	11	9	7	7	9	11	13	13	10	11]		
n	[15	9	5	25	3	9	9	9	10	6]	12	9	7	8	9	11	12	12	10	11]		
c	[0	0	2	0	6	20	20	13	21	17]	11	8	7	7	8	12	13	12	10	11]		
e	[0	0	5	1	3	20	32	12	12	15]	11	8	7	7	8	12	13	12	11	11]		
d	[4	9	31	6	12	23	0	4	5	3]	10	8	8	7	8	12	13	12	10	11]		
o	[0	5	13	14	18	35	8	3	4	0]	10	8	8	7	9	13	13	12	10	10]	Eng	{Eng}
n	[13	13	14	2	13	20	5	4	4	11]	10	8	8	7	9	13	12	11	10	10]		
o	[12	12	5	6	5	14	17	15	9	6]	10	8	8	7	9	13	13	12	10	10]		
m	[22	9	6	17	20	9	6	3	9	0]	11	8	8	7	9	13	12	11	10	10]	Fin	{Eng}
a	[10	12	14	9	9	10	9	8	10	9]	11	8	8	7	9	13	12	11	10	10]		
r	[14	6	6	6	8	10	11	11	12	17]	11	8	8	7	9	13	12	11	10	10]		
c	[5	23	8	5	6	11	7	12	14	9]	10	9	8	7	9	13	12	11	10	10]		
h	[0	0	1	10	2	13	17	14	24	19]	10	9	8	7	9	13	12	11	10	10]		
c	[0	16	0	45	13	7	4	10	3	1]	10	9	8	8	9	13	12	11	10	10]		
t	[0	49	0	26	2	23	0	0	0	0]	10	10	7	9	9	13	12	11	10	10]	Ger	{Eng}
h	[16	5	9	2	8	37	5	6	5	6]	10	10	7	9	9	14	11	11	10	10]		
a	[0	0	1	2	1	95	1	0	0	0]	9	9	7	9	9	16	11	10	10	9]		
t	[4	13	10	23	10	17	5	6	9	3]	9	10	7	9	9	16	11	10	10	9]		
a	[6	13	11	7	9	16	12	18	3	5]	9	10	7	9	9	16	11	11	9	9]		
t	[9	14	20	12	17	11	17	0	0	0]	9	10	8	9	9	16	11	10	9	9]	Eng	{Eng}
t	[16	5	9	2	8	37	5	6	5	6]	9	10	8	9	9	16	11	10	9	9]		
h	[0	0	1	2	1	95	1	0	0	0]	9	9	8	9	9	18	11	10	9	9]		
e	[1	5	6	15	19	44	1	7	1	1]	9	9	8	9	9	19	11	10	9	9]		
y	[0	0	4	0	0	95	0	0	1	0]	9	9	8	9	9	21	10	10	8	8]		
a	[0	0	0	0	0	70	1	0	28	0]	9	9	7	8	8	22	10	9	9	8]	Eng	{Eng}
r	[4	7	13	9	6	15	10	12	10	15]	9	9	7	8	8	21	10	9	9	8]		
e	[5	23	8	5	6	11	7	12	14	9]	8	9	7	8	8	21	10	10	9	8]		
h	[0	9	13	10	9	13	13	12	10	11]	8	9	8	8	8	21	10	10	9	8]	Eng	{Eng}
o	[1	4	11	7	14	12	16	14	11	10]	8	9	8	8	8	21	10	10	9	8]		
l	[5	15	17	11	30	9	2	5	6	1]	8	9	8	8	9	21	10	10	9	8]		
l	[8	4	12	5	19	20	4	1	10	17]	8	9	8	8	9	21	10	9	9	8]		
d	[16	15	11	9	6	6	4	18	7	8]	8	9	8	8	9	20	10	10	9	8]		
i	[0	20	37	6	26	9	0	0	0	1]	8	9	9	8	9	20	10	9	9	8]		
n	[1	4	5	21	13	6	6	26	9	8]	8	9	8	9	9	20	10	10	9	8]		
n	[15	12	9	11	13	14	6	9	6	6]	8	9	8	9	10	20	10	10	9	8]		
g	[0	17	15	25	26	13	1	1	1	1]	8	9	9	9	10	20	9	10	9	8]		
t	[0	9	29	19	26	13	1	1	1	1]	8	9	9	9	10	20	9	9	9	8]	Dut	{Eng}
w	[16	5	9	2	8	37	5	6	5	6]	8	9	9	9	10	20	9	9	8	8]		
o	[0	0	1	24	43	31	0	0	0	0]	8	9	9	9	11	20	9	9	8	8]		
o	[0	0	0	20	57	22	0	0	0	0]	8	9	9	9	11	20	9	9	8	8]	Eng	{Eng}
i	[2	0	1	1	0	7	0	29	20	40]	8	9	9	9	11	20	9	9	8	8]		
n	[3	14	12	9	9	17	6	16	7	7]	7	9	9	9	11	20	9	9	8	8]		
d	[15	12	9	11	13	14	6	9	6	6]	8	9	9	9	11	20	9	9	8	8]		
i	[0	10	23	21	14	14	4	5	4	5]	7	9	9	9	11	20	9	9	8	8]		
v	[1	4	5	21	13	6	6	26	9	8]	7	9	9	10	11	19	9	10	8	8]		
i	[2	9	14	3	3	12	11	17	14	15]	7	9	9	10	11	19	9	10	8	8]		
d	[7	17	18	1	4	9	10	15	8	12]	7	9	9	9	11	19	9	10	8	8]		
u	[11	7	8	2	15	6	4	5	20	23]	7	9	9	9	11	19	8	10	9	8]		
a	[5	0	5	10	10	13	29	6	11	11]	7	9	9	9	11	19	9	10	9	8]		
a	[3	0	2	2	6	10	8															

	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
r	[2	16	7	6	6	12	12	13	11	15]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
o	[16	13	5	4	8	10	7	12	13	12]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
m	[9	20	12	4	7	11	9	8	8	12]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
a	[14	6	6	6	6	8	10	11	11	12 17]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
n	[16	10	8	6	19	10	10	8	7	6]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
i	[20	14	10	7	6	8	5	13	8	9]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
a	[13	7	2	1	2	10	4	21	18	21]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
n	[16	10	8	6	19	10	10	8	7	6]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
p	[22	9	6	17	20	9	6	3	9	0]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
o	[10	5	6	2	3	10	17	15	16	16]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
l	[6	5	5	2	2	12	19	16	18	15]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
i	[16	15	11	9	6	6	4	18	7	8]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
c	[21	7	10	10	12	6	8	15	5	6]	Fin	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por
e	[0	(0)	1	22	4	15	10	17	18	14]	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por	
a	[0	(0)	5	1	3	20	32	12	12	15]	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por	
n	[1	4	11	7	14	12	16	14	11	10]	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por	
n	[4	7	13	9	6	15	10	12	10	15]	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por	
n	[16	10	8	6	19	10	10	8	7	6]	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por	
n	[20	8	13	14	7	5	19	13	0	0]	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por	
o	[5	16	3	3	5	8	6	27	9	18]	Swe	Dan	Ger	Dut	Eng	Fre	Ita	Spa	Por	
u	[13	0	0	0	9	27	40	0	0	10]	Dut	Eng	Fre	Ita	Spa	Por				
n	[15	9	5	25	3	9	9	9	10	6]	Dut	Eng	Fre	Ita	Spa	Por				
c	[0	0	2	0	6	20	20	13	21	17]	Dut	Eng	Fre	Ita	Spa	Por				
e	[0	0	5	1	3	20	32	12	12	15]	Dut	Eng	Fre	Ita	Spa	Por				
d	[4	9	31	6	12	23	0	4	5	3]	Dut	Eng	Fre	Ita	Spa	Por				
o	[0	5	13	14	18	35	8	3	4	0]	Dut	Eng	Fre	Ita	Spa	Por				
n	[13	13	14	2	13	20	5	4	4	11]	Dut	Eng	Fre	Ita	Spa	Por				
n	[12	12	5	6	5	14	17	15	9	6]	Dut	Eng	Fre	Ita	Spa	Por				
m	[22	9	6	17	20	9	6	3	9	0]	Dut	Eng	Fre	Ita	Spa	Por				
a	[10	12	14	9	9	10	9	8	10	9]	Dut	Eng	Fre	Ita	Spa	Por				
r	[14	6	6	6	8	10	11	11	12	17]	Dut	Eng	Fre	Ita	Spa	Por				
c	[5	23	8	5	6	11	7	12	14	9]	Dut	Eng	Fre	Ita	Spa	Por				
h	[0	0	1	10	2	13	17	14	24	19]	Dut	Eng	Fre	Ita	Spa	Por				
t	[0	16	0	45	13	7	4	10	3	1]	Dut	Eng	Fre	Ita	Spa	Por				
t	[0	49	0	26	2	23	0	0	0	0]	Dut	Eng	Fre	Ita	Spa	Por				
h	[16	5	9	2	8	37	5	6	5	6]	Dut	Eng	Fre	Ita	Spa	Por				
a	[0	0	1	2	1	95	1	0	0	0]	Dut	Eng	Fre	Ita	Spa	Por				
t	[4	13	10	23	10	17	5	6	9	3]	Dut	Eng	Fre	Ita	Spa	Por				
a	[6	13	11	7	9	16	12	18	3	5]	Dut	Eng	Fre	Ita	Spa	Por				
t	[9	14	20	12	17	11	17	0	0	0]	Dut	Eng	Fre	Ita	Spa	Por				
t	[16	5	9	2	8	37	5	6	5	6]	Dut	Eng	Fre	Ita	Spa	Por				
h	[0	0	1	2	1	95	1	0	0	0]	Dut	Eng	Fre	Ita	Spa	Por				
e	[1	5	6	15	19	44	1	7	1	11]	Dut	Eng	Fre	Ita	Spa	Por				
y	[0	0	4	0	0	95	0	0	1	0]	Eng									
	[0	0	0	0	0	70	1	0	28	0]	Eng									

Figure 10 Same type of log as shown in Fig. 9 but whereas the *sums* of the probabilities were considered there, in this algorithm the *products* are considered. This has the effect that as soon as a probability of 0 is encountered, that language is brutally and instantly ruled out. This first happens to Portuguese here as its matrix does not report on any word ending in ~n. "(0)" for SWE where FIN is ruled out means *close* to 0 but not *equal* to 0.

In Figure 10 we see what may happen when a tougher algorithm is applied, i.e. one considering the accumulated *products* of the probabilities instead of the *sums*. This algorithm is very efficient where the texts to be analyzed are very purely language-typical (i.e. not containing atypically spelt foreign names nor loan words) and when one wants to rule out quickly less relevant candidates. If the text is not of this nature however, this algorithm may be quite a risky one, although in this example it applies fortunately with success.

3.6 Presenting the results

The LR result is presented to one of two categories of readers: man or machine. In developing the program and evaluating the alternative algorithms, the programmer+linguist (in this case the one and same person) needs an instant, clear overview of the effects of the chosen algorithm. We have already seen Figures 9 and 10 which, as a matter of fact, are already an abstraction of a yet more detailed type of log file generated that contains e.g. even the decimals of the of dynamic probability values. That output is one way of presenting the result.

The next step makes more use of human's excellent capacity of processing images, i.e. in this step some *data visualization* is applied. Figures 11 though 13 below show the output of three other news reports (also found publicly available on Internet in the spring of 1999), processed the way described above relating to Figure 9, i.e. considering the dynamic sums of the respective language probabilities. The concept of a *paragraph* must be defined (here we are operating with units separated by two 'hard line breaks' but this definition must be text-sensitive). The moment a paragraph delimiter is reached, the champion language of that paragraph is calculated in accordance with some stipulated criteria, hinted above. Thereafter the paragraph is written to a file in HTML code by which the text and its background can be coloured as to represent the languages. This is a very perceivable way of visualizing exactly what parts of a text was recognized as belonging to what language. In more detail: each language is associated to one text colour and one background colour – inspiration was drawn from national flags when doing this – and once the dominant language is identified, that whole paragraph is coloured in the respective text/background colour combination except for the words not belonging to that dominant language according to the word-by-word assignment marked to the right in Figure 9. So, in Figure 11 all non-English words are marked with respective language code. The space characters and all other alphabetical text except

capital letter initials are written in the colours of the major language. Numerals and other non-alphabetical characters are reproduced in a language-neutral colour on the same background as that of the major language.

Looking briefly at Figure 11 (preferably in colour), one might wonder why e.g. ‘material’ is thought of as Italian – Italian polysyllables typically ending with a vowel! Well, presumably this is due to the very high frequency of articles and preposition+article compounds ending in ~l in masculine gender settings; an Italian text has loads of *il del dell’ al all’ nel nell’ sul sull’* for example (where the apostrophes are ignored as commented under section 3.4, above). What is more: ~gn~ is the Italian way of writing the phoneme corresponding to what in Spanish is written as ~ñ~ and in Portuguese as ~nh~ and this probably explains the assignment of Italian to the proper name ‘Ignalina’. The ending sequence ~el in ‘fuel’ coincides with the common Spanish masculine article ‘el’ and the ending ~nt coincides with the many French verbs in 3rd person plural.

Lithuanian authorities announced that they had arrested seven people and seized nearly 100 kg of radioactive material, according to press reports. The material, believed to be uranium, will undergo further tests to ascertain its makeup and origin. It was emitting 14,000 microroentgens per hour. Some reports stated that the material was a component of a nuclear fuel assembly which has been missing from the nearby Ignalina nuclear power plant for several years. The Ignalina plant manager claims that the seized material is not nuclear fuel or equipment used at his facility.

ENGLISH

Figure 11 A text recognized as English in spite of several words having less typically English character sequences.

In Figure 12 where the text is correctly recognized as French, ‘mercredi’ is believed to be Italian due to the highly frequent Italian noun+plural and adjective+plural ending ~i. The sequence ~ch~ is apparently more common in German where we find it in both ~ch~ and ~sch~ than in French where only the former is seen and therefore the assumption that ‘chercher’ and ‘marchandise’ are German. In this example we also note that ‘français’ is ironically not recognised as typically “français” but as Portuguese and this is no wonder since the pairs ~ça~ ~ai~ and ~s are quite often found in that language. The same goes for ‘quais’ in this text.

Mercredi matin, les douaniers de Saint-Brieuc ont mis la main sur un trafic de cigarettes de contrebande de marque «L.M» en provenance d'Europe de l'Est. Sur l'un des quais du port du Légué où le navire russe le «Ladoga 101», devait débarquer sa cargaison de phosphate, attendait une camionnette avec à son bord deux ressortissants français. Ils étaient venus chercher la marchandise. Les douaniers ont alors interpellé les deux hommes ainsi que le commandant russe du cargo. Ils ont découvert deux gros cartons remplis d'une centaine de cartouches de cigarettes. L'équivalent de 30.000 F.

FRANÇAIS

Figure 12 This text was analyzed as being more French than anything else.

The text in Figure 13, finally, has a clear German predominance with Dutch as its most frequent alternative interpretation. The short words thought to be Dutch actually do have similar correspondents in that language and 'Tabakwaren' and 'Kaffee' may curious enough be words once imported, in some form, into German from Dutch which in northern Europe was second only to English in importance during historical colonial and trade activities.

Künftig müssen in Rumänien auch gemeinnützige Stiftungen und Vereine für die Einfuhr von Alkoholika, Tabakwaren, Kaffee und Kraftfahrzeugen Zoll bezahlen. Einen entsprechenden Beschluß hat die Regierung in Bukarest am 06.11.1997 gefaßt.
DEUTSCH

Figure 13 Many strongly typical German letter combinations in this text suggest in what language it is written.

3.7 Language recognition using different word context width

Hitherto we have favoured a LR identification heuristics assigning a probable language to each word of a paragraph and then assigning a presumed language to that paragraph, as a whole, which could be the language assigned to most words in it. The language assignment to each word is based on the character transitions *into*, *within*, and *out of* that same word. In this section we will extend this word = language assignment a little to take even the immediate context of that word into consideration.

The idea is still simple; when deciding on a major language for a word and therefore evaluate the corresponding language probability decuple (which is the sum of the [...] annotations throughout that word as shown in Fig. 9), we let the probability decuple of the *previous* word or words influence as well, and so also the *following* one or more words. I.e. we use a 'window' of a certain number of words at the assignment of each word. The impact of surrounding words is set to decrease with distance and at current it is $1/(1+x)$ where x is the distance in words from the one in focus. Thus, inside a text the influence would be <33> <50> <100> <50> <33> percent, where 100 (%) is assigned to the word in focus and the window breadth is five words.

Since there are no words to consider beyond the beginning nor the end of the text, in order to keep the smoothing effect of the neighbouring words, those still available on the other side will make an extra contribution, albeit a little weakened. So if | denotes a paragraph delimiter, the respective influence, expressed as above, would be <33> <50> | <100> <50+33=83> <33+25=58> percent at the first word and, <33+25=58> <50> <100> <50> | <33> percent at the penultimate word of the text. Reading out this last graphically expressed formula says that the language to be assigned to the penultimate word of the text is given by taking 100% of the [...] annotations for that word and to this add the [...] annotations of both the last but two word and the very last word and assign an impact of 50% each to these, and finally add the [...] annotations of the last but three word with an impact of 58%.

In a series of experiments a different window breadth was tried out for the same text. The text was composed of small parts of an annual report of the European Commission written in all ten languages using the Latin alphabet. The language order reflects a geographic itinerary from Finland via UK and Italy to Portugal and thereby some sort of linguistic continuity is also created. The purpose of this was to make the transition between the languages more difficult; it is harder to identify the transition between two languages if they have similar graphotax – especially as both languages appear inside the window of analytic context.

Figures 14 through 16 show how the LR identified the ten language stripes, each enclosed within '[N:]', where N is the order number of that language. In the first of these figures the window contained only one word, thus working as <100> in the denotation used above. Nevertheless we again see that Finnish in virtue of its very special graphotax is already identified correctly except for one single word. Even Dutch, Italian and Portuguese are fairly well recognized when processed without attention to the context. In the next figure exactly that window of <33> <50> <100> <50> <33> just described was applied and the result has improved considerably. In fact, already a window of <50> <100> <50>, which we skip showing here to save space, cleans up the result quite a lot. In the last of these three figures, a seven words broad window was applied: <25> <33> <50> <100> <50> <33> <25> and, in addition, the square parentheses marking the language boundaries were set to be considered by the algorithm (actually by inserting the mark '–', which we have declared a paragraph delimiter). The result at this point is quite satisfying and at the very last step of the process, the dominant language is assigned to the whole of each paragraph – and in the cases we see here there are no doubts which are the respective dominant languages.



Figure 14 Language identification of ten consecutive texts without word context window.



Figure 15 Same process as in Figure 14 but after applying a context window of 5 words.



Figure 16 Same as above but using a window of seven words and marked paragraph boundaries.

3.8 The LR generating input to other programs

As mentioned in the introductory section, this LR is being developed to be used as a module in a larger “information placer mining” system. The first time we tried this LR on a tough data set were when we needed to sort all paragraphs of nearly 30,000 low-quality and poorly structured case reports. The purpose was to keep all French paragraphs and send off the English, German and Spanish parts to the translation service carried out by SYSTRAN in Luxembourg in order to have even these parts translated into French as this language was chosen in which to unify the data. The other six languages for which no translation service is yet provided were treated in another way to maximally extract recoverable information. So, the task of this LR module was to identify the language of each single paragraph of the 30,000 reports (with the ultimate purpose of clustering these texts with another module, described in [Hagman 99]). This task was far from trivial, though, as many of the paragraphs were language mixes. Figure 17 shows a tiny part of the output of this LR run, viz. three paragraphs. Each paragraph is preceded by a tag, an info sextuple telling ¹{how much better} ²{the dominant language} describes this paragraph than ³{the second best language} does, and then follow ⁴{one original report reference} and ⁵{another original report reference} and, finally, ⁶{an automatically generated control index number}.

```
<< 1.26 Eng Ger 165 9426 165 >>
ireland the irish investigation division has informed the commission services that thorough
enquiries indicate that it was a once off transaction between the firms [CENSORED], korea and
[CENSORED] in ireland, which was arranged by the firm [CENSORED] in the united kingdom. no
other importations of counterfeit [CENSORED] could be traced. diese mitteilung unterliegt arti-
kel 19 der verordnung (ewg) nr. 1468/81 betreffend die gegenseitige unterstuetzung mitteilung

<< 1.19 Eng Fre 13 9534 263 >>
[CENSORED] investigation division manchester royaume-uni

<< 1.00 Eng Fre 135 10204 385 >>
the other member states are requested to advise similar cases to the commission. cette communi-
cation est couverte par l'article 19 du _reglem_1468/81_ relatif a l'assistance mutuelle. com-
munication am /93 : envois de textiles d'origine chinoise presumée avec des certificats
d'origine du bangladesh. les autorités espagnoles ont signale a la commission l'information
```

Figure 17 Example of output from the LR of texts written in mixed languages. The first of these three texts is 26% more English-looking than German-looking. Confidential parts not shown here are marked [CENSORED].

Given the linguistically commingled nature of the input in the above case, we are quite satisfied with the result.

4 Operating the LR

As said above, this LR is still a prototype. In order to meet the varying format of both input and output data, all time and effort has been devoted to the operating functions and procedures, thereby postponing the creation of a user interface to the future. At present the program is run from inside its C code development environment but, of course, once the above-mentioned formats are set, a stand-alone executable may be created.

References

Barbas, T. and R. Steinberger (1999)

“New information infrastructures”, in *Institute for Systems, Informatics and Safety; Annual report 1998*, EUR 18721, ISBN 92-828-6645-9, European Commission 1999 (pp.25-26).

Hagman, J. (1999)

An Implemented Cluster Analyzer for Documents and their Indexing Terms,
Technical Note (November), European Commission, Joint Research Centre, ISIS/SAIA.