Knowledge-poor Approach to Constructing Word Frequency Lists, with Examples from Romance Languages

Mikhail Alexandrov
Center for Computing Research, National Polytechnic Institute (IPN), Mexico
dyner@cic.ipn.mx

Xavier Blanco
Department of French and Romance Philology, Autonomous University of Barcelona
Xavier.Blanco@uab.es

Alexander Gelbukh
Center for Computing Research, National Polytechnic Institute (IPN), Mexico;
Chung-Ang University, Seoul, Korea
gelbukh@gelbukh.com, www.Gelbukh.com

Pavel Makagonov
Mixteca University of Technology, Mexico
mpp2003@inbox.ru

Resumen: Las listas de palabras con sus frecuencias se usan ampliamente en muchos procedimientos de agrupamiento y categorización de textos. Usualmente para la compilación de tales listas se usan las aproximaciones basadas en morfología (como el *stemmer* de Porter) para unir las palabras con el mismo significado. Desafortunadamente, tales aproximaciones requieren de muchos recursos lingüísticos dependientes de lenguaje cuando se trabaja con datos multilingües y colecciones multitemáticas de documentos. En este artículo se proponen dos procedimientos basados en formulas empíricas de similitud entre palabras. Un simple ajuste de los parámetros de las fórmulas permita su adecuación a diferentes lenguajes europeos. Se demuestra la aplicación de las fórmulas con ejemplos reales del francés, italiano, portugués y español.
Palabras clave: *stemming*, indexación, métodos independientes de lenguaje, métodos estadísticos

Abstract: Word frequency lists extracted from documents are widely used in many procedures of text clustering and categorization. Usually for compilation of such lists morphological-based approaches (such as the Porter stemmer) to join the words having the same base meaning are used. However such an approach needs many language-dependent linguistic resources or knowledge when working with multilingual data and multi-thematic document collections. We suggest two procedures based on empirical formulae of word similarity. Simple adjustment of the parameters of the formulae allows tuning them to different European languages. We demonstrate the application of our formulae on real examples from French, Italian, Portuguese, and Spanish.
Keywords: stemming, indexing, language-independent methods, statistical methods

1 *Introduction*

The typical procedure for constructing a word frequency list consists in the following two steps:

1. All words are ordered alphabetically and their counts are set to 1.
2. Every word is substituted by its stem or lemma (normal dictionary form).
3. Equal strings are joined together and their counts are summed up.

By the normal form we mean singular for nouns, indefinite infinitive for verbs, etc. Such a transformation is called lemmatization. It relies on morphological rules and a large morphological dictionary. It provides practically ideal accuracy on known words and good accuracy on the words absent in the dictionary [Gelbukh, 2003; Gelbukh and Sidorov, 2003].

Instead of the normal form, the words can be truncated to their stems, which often reflect their invariant meaning; e.g., *sad*, *sadly*, *sadness*, *saddens*, *saddened* are truncated to the same stem *sad-*. This method is much simpler since it uses only lists of suffixes and suffix removal rules, without any dictionaries [Porter, 1980]. Its accuracy is lower than that of lemmatization. Still it relies on language-dependent resources.

Makagonov [2002] suggested an empirical formula for testing word similarity and demonstrated how to use such a formula for stemming in constructing word frequency lists. The formula itself was based on the number of the coincident letters in the initial parts of the two words and the number of non-coincident letters in their final parts. The frequency list is obtained by an iterative procedure, which compares word pairs and determines their common part.

Our experiments with Romance languages show that one of suggested empirical formulae provides 80%–90% accuracy (F-measure), with 2%–5% of false collatings and 20%–25% of failuries to collate similar words. This is rather acceptable in semi-automatic setting since the human expert can easily join the similar words after the grouping procedure [Alexandrov, 2004].

However, many important issues concerning application of the empirical formulae for stemming remain open. In this paper we compare the application of two different formulae and consider two iterative procedures on real texts in four Romance languages—French, Italian, Portuguese, and Spanish—in the domain widely discussed currently in the European Community: mortgage and crediting. To evaluate the accuracy of the results, we use characteristics from both mathematical statistics and information retrieval.

2 *The Algorithms*

2.1 Formulae for Testing Word Similarity

The empirical formulae used in the procedure of constructing word frequency list test a hypothesis about word similarity. We consider only the languages where the word base (the morphologically invariant part) is located at the beginning of the word. This is generally true for the European languages.

We consider two classes of formulae:

1. The formulae that do not take into account the position of the final part in a word pair;
2. The formulae that do take it into account.

The formulae of the first class can be represented in the following form:

$$n/s \le F(y), \quad F(y) = a + b_1 y + b_2 y^2 + \ldots + b_k y^k, \qquad (1)$$

where:

n:	the total number of *final* letters differing in the two words,
y:	the length of the common *initial* substring,
s:	the total number of letters in the two words,
$F(y)$:	the model function.

For example, for the words *sadly* and *sadness*, the maximal common initial substring is *sad-*, thus the differing final parts are *-ly* and *-ness* so that $n = 6$, $y = 3$, and $s = 12$.

When the inequality holds, the hypothesis about word similarity is accepted. Using the Inductive Method of Model Self-Organization (IMSOM) by Ivahnenko [1980] we found that the best formula (in the considered class) is lineal, with the following parameters:

French:	$n/s \le 0.481 - 0.024\,y$
Italian:	$n/s \le 0.571 - 0.035\,y$
Portuguese:	$n/s \le 0.528 - 0.029\,y$
Spanish:	$n/s \le 0.549 - 0.029\,y$

The formulae of the second class can be represented in the following form:

$$D(y,N) \le 0,$$

$$D(y, N) = \sum f(i)\,\delta(i), \quad i = y + 1, \ldots, N \qquad (2)$$

$$F(i) = a + b_1/i + b_2/i^2 + \ldots + b_k/i^k,$$

where:

y:	the length of the common *initial* substring,
N:	the length of the longest word in the list,
$\delta(i)$:	indicator of letters at the i-th position

We set $\delta(i) = 0$, 0.5, or 1 if i is greater than the lengths of both words, of the length of one of the two words, or less than, or equal to, the lengths of both words, respectively. For example, for the words *sadly* and *sadness*, the maximal common initial substring is *sad-*, so that $y = 3$, $\delta(4) = \delta(5) = 1$, $\delta(6) = \delta(7) = 0.5$, $\delta(8) = ... = \delta(15) = 0$ ($N = 15$).

When the inequality holds, the hypothesis about word similarity is accepted. Using IMSOM, we found that the best model is inverse lineal with the following parameters:

French:	$F(t) \leq -0.21 + 0.029 / t$
Italian:	$F(t) \leq -0.26 + 0.015 / t$
Portuguese:	$F(t) \leq -0.24 + 0.016 / t$
Spanish:	$F(t) \leq -0.27 + 0.033 / t$

These formulae can be considered an initial approximation for further tuning on other romance languages.

Testing word similarity one can commit two types of errors: join words that are not similar (false positive) or fail to join words which are similar (false negative). Obviously since the empirical formula is constructed on the basis of statistical regularities of a language, it leads to the both types of errors. Tuning the parameters we can control the balance between these two kinds of errors, but not to completely avoid them.

Our approach for testing word similarity is not applicable to irregular words: it is impossible to construct a simple formula that could detect the similarity between English irregular verbs such as *buy* and *bought*—these words have only one common letter in the initial part of the string. This situation sometimes occurs in the Romance languages considered in this paper.

2.2 Constructing Word Frequency List

We consider two procedures:

A1. Algorithms based on joining a pair of adjacent similar words, and

A2. Algorithms based on joining a group of similar words.

Each of the two algorithms can use any of the empirical formulae described above, or any other one. The two algorithms differ in their capability of word conflation, independently of the formulae used.

First of all, the text or a group of texts is transformed into a sequence of words. All stopwords (including short words) are eliminated from the list. Then with each word, a counter is associated and set to the initial value 1. Then all words are ordered alphabetically, equal strings are joined together, and their counts are summed up.

The two variants of the algorithm work as follows.

Algorithm A1 (pair-wise) From the beginning of the list, the first word similar (according to the formula) to the next one is looked for. The two words are replaced with one new "word"—their common initial part, with the counter set to the sum of the counters of the two original words. Then the procedure is repeated (from the same position in the list, since we know that there are no similar words above the current position), until no pair of similar words is found.

Algorithm A2 (chain-wise) From the beginning of the list, the first word similar to the next one is looked for. Then second word of the pair is compared with the next word, etc., thus building a maximal chain in which every two adjacent words are similar. The whole chain is replaced with one new "word"—their common initial part, with the counter set to the sum of the counters of all original words. Then the process continues from the next position in the list.

Example Suppose we have the following list of Spanish words ordered alphabetically, with the following counts: *transformación* (7), *transformado* (5), *transformamos* (7), *traducción* (6), *traductor* (7), *transporte* (11), *transportado* (2). The digits in brackets mean the number of word's repetitions.

The following table shows the first steps of the work of the pair-wise algorithm:

	Initial list		First iteration
→	***transformación* (7)**	→	***transforma-* (12)**
	***transformado* (5)**		***transformamos* (7)**
	transformamos (7)		*traducción* (6)
	traducción (6)		*traductor* (7)
	traductor (7)		*traduje* (4)
	traduje (4)		*transportado* (2)
	transportado (2)		*transporte* (11)
	transporte (11)		

	Second iteration		Third iteration
→	*transforma-* (19)	✓	*transforma-* (19)
	***traducción* (6)**	→	***traduc-* (13)**
	***traductor* (7)**		***traduje* (4)**
	traduje (4)		*transportado* (2)
	transportado (2)		*transporte* (11)
	transporte (11)		

The following table shows the work of the chain-wise algorithm:

	Initial list		First grouping
→	***transformación* (7)**	✓	*transforma*- (19)
	***transformado* (5)**	→	***traducción* (6)**
	***transformamos* (7)**		***traductor* (7)**
	traducción (6)		***traduje* (4)**
	traductor (7)		*transportado* (2)
	traduje (4)		*transporte* (11)
	transportado (2)		
	transporte (11)		

	Second grouping		Third grouping
✓	*transforma*- (19)	✓	*transforma*- (19)
✓	*tradu*- (17)	✓	*tradu*- (17)
→	***transportado* (2)**	✓	*transport*- (13)
	***transporte* (11)**	→	

2.3 Discussion

The suggested algorithms prove to be sensible not only to the parameters of the formula but to the size of the word list. Namely, the shorter the list the greater the probability of conflation. This especially affects short words.

For example, in a short vocabulary the probability for the following English words

bead
bear
beat

to occur together is high, and according to the formulae from [Makagonov, 2002], these words will be considered similar and conflated by the algorithm. In case of a large vocabulary these words would be separated:

bead
...etc...
beagle
beagling
...etc...
bear
...etc...
beast
beastliness
...etc...
beat

However, in this case many short words that ought to be conflated will be separated in the vocabulary by unrelated words:

cat
catalogue
cataplasm
catastrophe
catenary
cats

These examples were taken from our private communication with M. Porter, the author of the stemming algorithm [Porter, 1980].

The former situation is a problem. However, we assume that in the one domain so different short words do not frequently occur. As to the latter situation, we did observe such cases. However, this is not dangerous because these related words can be easily revealed manually, since in practice an expert usually checks the results of the work of the algorithm.

3 Experimental Results

The algorithms were checked on real document collections. The goals of these experiments were:

- To compare the quality of the formulae for a given language,
- To compare the quality of algorithms for a given language,
- To compare the quality of a given algorithm for different languages.

Document Collections We considered the documents on a popular theme: mortgage and crediting. This theme is narrow enough to provide a representative set of similar words. To reduce the number of comparisons we randomly selected some paragraphs from the document collections. Then we obtained alphabetical lists of words, exemplified in the following table:

French	**Italian**
absence	*accedere*
accepte	*acquistare*
accord	*acquisto*
accorde	*adatte*
accordee	*adatto*
...etc...	...etc...
Total: 456 words	Total: 567 words

Portuguese	**Spanish**
abrange	*abaratado*
abrangencia	*abogado*
abrangendo	*abogados*
acessoes	*acceder*
acima	*actual*
...etc...	...etc...
Total: 506 words	Total: 536 words

French (455 tests = 115 similar + 340 non-similar)

Formula + algorithm	(1) + *A*1	(2) + *A*1	(1) + *A*2	(2) + *A*2
Similar cases	104	88	126	111
Not similar cases	351	367	329	344
False alarms	7	3	18	10
Omissions	18	30	7	14
False positive P_p	2.0%	0.9%	5.3%	2.9%
False negative P_n	15.7%	26.1%	6.1%	12.2%
Total errors	17.7%	27.0%	**11.4%**	15.1%
Recall R	84.3%	73.9%	93.9%	87.8%
Precision P	93.3%	96.6%	85.7%	91.0%
F-measure	88.6%	83.7%	**89.6%**	89.4%

Italian (566 tests = 140 similar + 426 non-similar)

Formula + algorithm	(1) + *A*1	(2) + *A*1	(1) + *A*2	(2) + *A*2
Similar cases	149	120	193	166
Not similar cases	417	446	373	400
False alarms	39	19	65	45
Omissions	30	39	12	19
False positive P_p	9.2%	4.2%	15.3%	10.6%
False negative P_n	21.4%	27.9%	8.6%	13.6%
Total errors	30.6%	32.1%	**23.9%**	24.2%
Recall R	78.6%	72.1%	91.4%	86.4%
Precision P	73.8%	84.2%	66.8%	72.9%
F-measure	76.1%	77.7%	77.2%	**79.1%**

Portuguese (505 tests = 138 similar + 367 non-similar)

Formula + algorithm	(1) + *A*1	(2) + *A*1	(1) + *A*2	(2) + *A*2
Similar cases	136	117	159	141
Not similar cases	369	388	346	364
False alarms	15	9	27	17
Omissions	17	30	6	14
False positive P_p	4.1%	2.5%	7.4%	4.6%
False negative P_n	12.3%	21.7%	4.3%	10.1%
Total errors	16.4%	24.2%	**11.7%**	14.7%
Recall R	87.7%	78.3%	95.7%	89.9%
Precision P	89.0%	92.3%	83.0%	87.9%
F-measure	88.3%	84.7%	**88.9%**	**88.9%**

Spanish (535 tests = 165 similar + 370 non-similar)

Formula + algorithm	(1) + *A*1	(2) + *A*1	(1) + *A*2	(2) + *A*2
Similar cases	164	128	183	167
Not similar cases	371	407	352	368
False alarms	22	8	34	23
Omissions	23	45	16	21
False positive P_p	5.9%	2.2%	9.2%	6.2%
False negative P_n	13.9%	27.3%	9.7%	12.7%
Total errors	19.8%	29.5%	**18.9%**	**18.9%**
Recall R	86.1%	72.7%	90.3%	87.3%
Precision P	86.6%	93.8%	81.4%	86.2%
F-measure	86.3%	81.9%	85.6%	**86.7%**

Results In our experiments we examined the results of algorithm's work manually. The above tables show the result of automatic processing for the formulas (1) or (2) above in combination with the algorithms A1 or A2. The quality of the result is judged by a single-value parameter called F-measure [Baeza-Yates et al., 1999]:

$$F = 2\frac{1}{\dfrac{1}{recall} + \dfrac{1}{precision}}.$$

4 Conclusions

We have suggested knowledge-poor algorithms for constructing word frequency lists with stemming. Our empirical formulae do not require any morphological dictionaries of a given language. This is useful for working with multilingual databases and multi-thematic document collections. The accuracy of the suggested algorithms is very promising.

In the future we plan to construct several other empirical formulae and consider other algorithms. We plan to take into account some statistical regularities extracted from the text itself (we thank M. Porter for useful suggestions on such modifications).

Acknowledgements

The work was done under partial support of Mexican Government (CONACyT, SNI, CGPI-IPN). The third author is currently on Sabbatical leave at Chung-Ang University, Korea.

References

Alexandrov, M., X. Blanco, P. Makagonov (2004): Testing Word Similarity: Language Independent Approach with Examples from Romance. In: F. Meziane el al. (Eds.) "*Natural Language for Information Systems*", Springer, LNCS, N 2515 (to appear).

Baeza-Yates, R., B. Ribero-Neto (1999): *Modern Information Retrieval*. Addison Wesley.

Gelbukh, A. (2003): Exact and approximate prefis search under access locality requirements for morphological analysis and spelling correction. *Computación y Sistemas*, vol. 6, N 3, 2003, pp. 167–182.

Gelbukh, A., G. Sidorov (2003): Approach to construction of automatic morphological analysis systems for inflective languages with little effort. In: *Computational Linguistics and Intelligent Text Processing* (CICLing-2003), Lecture Notes in Computer Science N 2588, Springer, pp. 215–220.

Ivahnenko, A. (1980): *Manual on typical algorithms of modeling*. Tehnika Publ., Kiev (in Russian).

Makagonov, P., M. Alexandrov (2002): Empirical formula for testing word similarity and its application for constructing a word frequency list. In: A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing* (CICLing-2002), Lecture Notes in Computer Science, N 2276, Springer Verlag, pp. 425-432.

Porter, M. (1980): An algorithm for suffix stripping. *Program*, 14, pp. 130–137.