# Natural Language Processing

## *LEXISTERM* – THE PROGRAM FOR TERM SELECTION BY THE CRITERION OF SPECIFICITY

**Roque Lopez, Mikhail Alexandrov, Dennis Barreda, Javier Tejada**

***Abstract**: Term selection is one of the principal procedures in natural language processing. Existing advanced methods allow to construct multiword terms, to form hierarchy of related terms, etc. It provides a high quality of problem solutions where these terms are used. But almost always an expert needs a simple tool to glance a document corpus to reveal the most distinctive features. For this purpose we propose the simple program LexisTerm for one-word term selection based on a well-known criterion of term specificity. Speaking 'specificity' we mean the relation of term frequencies in a given document/corpus and in some gold standard as, for example, a National corpus of document. The program has two options, which give an opportunity. to select both specific terms in an individual document and specific terms for the whole corpus. In the paper we describe this program and demonstrate the results of its work on a real example. The program LexisTerm is free-share.*

***Keywords**: Natural language processing, term selection, indexing*

***ACM Classification Keywords**: I.2.7 Natural Language Processing*

***Conference topic**: Natural Language Processing, Speech Understanding*

## Introduction

Term selection has many applications in natural language processing (NLP). Its realization depends on the goal:

- whether we want to describe a contents of a domain or a document corpus (a)
- whether we want to classify or to cluster document set (b)

In case (a) selected terms should reflect some common properties of document set and in general each of such terms must have more or less equal relative frequency of its occurrence in documents (of course, each term has his own frequency distribution). In case (b) selected terms should have good distinctive properties and in general each of such terms must not have equal relative frequency of its occurrence in documents (of course, each term has his own frequency distribution).

The general approaches and algorithms for term selection are presented well in the well-known monographies [Baeza-Yates, 1999; Manning, 1999]. Researchers continue to consider special cases: key-phrases extraction for summarization [Schutz, 2008], indexing for clustering in narrow domains [Pinto, 2008], etc. Some authors propose to use a set of criteria related with cases (a) and (b) simultaneously. It allows to determine terms for description sub-topics in the framework of a topic reflected in a given document set [Makagonov, 2000]. It is a well-known that word collocations have a large informative and distinctive power. Just these collocations form so-called multiword terms [Yagunova, 2010]. But all these techniques are not simple. They often need complimentary information about word distribution in a corpus, correlation between words, etc.

In this paper we consider the simplest case: one-word term selection based on the principal of word specificity. The specificity of word is determined on the basis of its frequency in a given document or in a given corpus and in some standard corpus, which is considered as a gold standard. Such a criterion of term selection is well-known. In particularly, we could mention two works [Makagonov, 2000; Makagonov, 2004], where this criterion was used among the other ones for constructing domain-oriented vocabularies and for clustering super-short documents.

In section 2 we present the criterion of term selection in two forms of its realization. In section 3 we describe functions of the program. Section 4 contains the results of experiments. Section 5 includes conclusions.

## Criterion of specificity

### *2.1 General lexis and word specificity*

To simplify the further program description we give the following two local definitions:

Definition 1. *The general lexis is a frequency word list based on a given corpus of texts*

The given corpus means here any standard document set reflecting the lexical richness of a given language. Generally such a corpus contains in a certain proportion the documents taken from newspapers, scientific publications related with various domains, novels and stories. For example, it could be the British National corpus.

The general lexis can contain:

- unlemmatized or lemmatized word frequency list
- absolute and/or relative word frequencies

Unlemmatized word list contains all forms of words from a given standard corpus. Example: the words *move* and *moved* (English) are considered as the different ones with their own frequencies. When the general lexis is presented in unlemmatized form then user can use his document set without any transformation. Here words from a given document corpus are compared with words from the general lexis as they are.

Lemmatized word list contains lemmas of all words from a given standard corpus. In this case instead of words *move* and *moved* (English) the list contains one word *move.* Its frequency is the sum of frequencies for all forms of the verb *move*. User should take into account this circumstance by the following ways:

- To construct the word frequency list of a given document corpus and then to lemmatize all words from this list. The frequencies of words having the same lemma are summarized. But such a procedure needs special tools including morphological dictionaries for the language under consideration
- To substitute lemmatized word frequency list of a general lexis for stemmed word frequency list. For this all words from the lemmatized list are reduced to its stems, and the frequencies of words having the same stem are summarized. The same operation should be done with a given document corpus. Namely, it is necessary to construct the word frequency list of this corpus, then substitute all words for their stems, and then summarize frequencies of words having the same stem

Note. Stemming can be implemented by means of the well-known Porter's stemmers [Porter, 1980]

Let we have a word ***w.*** Let its relative frequency in a document is equal $f_D(\boldsymbol{w})$, in a document corpus $f_C\boldsymbol{w})$, and in the general lexis $f_L(\boldsymbol{w})$.

Definition 2. The level of specificity of a given word **w** in a given document corpus C is a number $K \geq 1$, which shows how much its frequency in the document corpus $f_C(\boldsymbol{w})$ exceeds its frequency in the general lexis $f_L(\boldsymbol{w})$:

$$K = f_C(\mathbf{w}) / f_L(\mathbf{w})$$

Definition 3. The level of specificity of a given word **w** in a given document D is a number $K \geq 1$, which shows how much its frequency in the document $f_D(\mathbf{w})$ exceeds its frequency in the general lexis $f_L(\mathbf{w})$:

$$K = f_D(\mathbf{w}) / f_L(\mathbf{w})$$

*2.2 Preprocessing general lexis*

LexisTerm having read the general lexis always completes two operations: search of duplicate words and normalization of word frequencies

1) Duplicate word analysis and 'black list'

Some words from the general lexis can have copies. Their frequencies can be equal or no. Such a situation reflects the cases when a word has several meanings. The total number of copies usually does not exceed ten or about. Lexis Term joins equal words and summarizes their frequencies.

This operation proves to be very useful for excluding undesirable words. Really, for this it is only necessary to add to the general lexis these words with large values of their frequencies. For example, these values can be done equal to maximum frequency in a given list (absolute or relative frequency). Therefore such words look like words from '*a black list*'

2) Normalization

LexisTerm normalizes all frequencies from the general lexis on their total sum. It means that the program always deals with *relative frequencies*.

If the general lexis contains absolute frequencies then such normalization is justified. If the general lexis contains relative frequencies then this normalization is unnecessary, but the program does not know in advance about it.

## Program description

*3.1 Modes of document processing*

Program LexisTerm has two modes for processing document set: corpus-based term selection and document-based term selection

1) Corpus-based term selection

In this mode the program determines word frequencies considering the entire corpus as one document. Therefore the output file contains all words whose total relative frequency (that is corpus relative frequency) exceeds their frequency in the general lexis in *K* times.

2) Document-based word selection

In this mode the program determines word frequency in each document separately. It collects all words in each document, whose document relative frequency exceeds their frequency in the general lexis in *K* times. Then all equal selected words are joined

*3.2 Data format*

1) Input data

- It is a document corpus where documents are presented in a textual form. All documents should be located in one directory.
- It is a general lexis, presented in a textual file. It should contains words with their frequencies. Each line should contain the word itself and its frequency. Other information in the line is ignored.

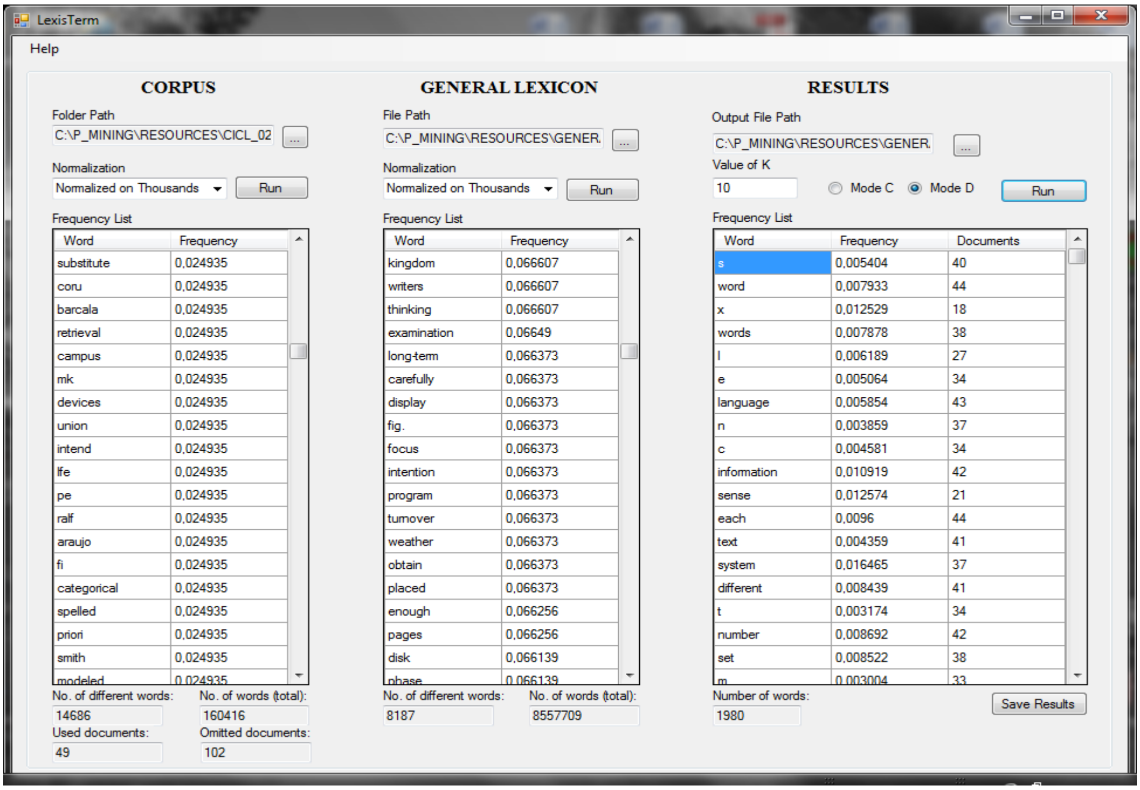
Note. One should use *dot* instead of *comma* to separate a fractional part of numbers.

2) Output data, results

It is a textual file, which contains the list of selected words with their relative frequencies and the number of documents where this word occurred.

There is a difference in the content of values calculated by the program:

- In case of corpus-based term selection the frequency of word means the relative frequency of this word in the entire corpus, which is considered as one document. The number of documents shows the number of documents, where this word occurs at least one time
- In case of document based term selection the frequency of word means the average relative frequency of this word in the documents where this word satisfies the criterion of word selection. This value does not take into account the document sizes, so it is not the weighted averaged value. The number of documents shows the number of documents, where the word satisfies the criterion of word selection. So, this value does not take into account other documents even they can contain this word.



CORPUS

Folder Path: C:\P_MINING\RESOURCES\CICL_02

Normalization: Normalized on Thousands

Frequency List

| Word | Frequency |
|---|---|
| substitute | 0,024935 |
| coru | 0,024935 |
| barcala | 0,024935 |
| retrieval | 0,024935 |
| campus | 0,024935 |
| mk | 0,024935 |
| devices | 0,024935 |
| union | 0,024935 |
| intend | 0,024935 |
| lfe | 0,024935 |
| pe | 0,024935 |
| ralf | 0,024935 |
| araujo | 0,024935 |
| fi | 0,024935 |
| categorical | 0,024935 |
| spelled | 0,024935 |
| priori | 0,024935 |
| smith | 0,024935 |
| modeled | 0,024935 |

No. of different words: 14686

No. of words (total): 160416

Used documents: 49

Omitted documents: 102

GENERAL LEXICON

File Path: C:\P_MINING\RESOURCES\GENER.

Normalization: Normalized on Thousands

Frequency List

| Word | Frequency |
|---|---|
| kingdom | 0,066607 |
| writers | 0,066607 |
| thinking | 0,066607 |
| examination | 0,06649 |
| long-term | 0,066373 |
| carefully | 0,066373 |
| display | 0,066373 |
| fig. | 0,066373 |
| focus | 0,066373 |
| intention | 0,066373 |
| program | 0,066373 |
| turnover | 0,066373 |
| weather | 0,066373 |
| obtain | 0,066373 |
| placed | 0,066373 |
| enough | 0,066256 |
| pages | 0,066256 |
| disk | 0,066139 |
| phase | 0,066139 |

No. of different words: 8187

No. of words (total): 8557709

RESULTS

Output File Path: C:\P_MINING\RESOURCES\GENER.

Value of K: 10

○ Mode C ● Mode D

Frequency List

| Word | Frequency | Documents |
|---|---|---|
| s | 0,005404 | 40 |
| word | 0,007933 | 44 |
| x | 0,012529 | 18 |
| words | 0,007878 | 38 |
| l | 0,006189 | 27 |
| e | 0,005064 | 34 |
| language | 0,005854 | 43 |
| n | 0,003859 | 37 |
| c | 0,004581 | 34 |
| information | 0,010919 | 42 |
| sense | 0,012574 | 21 |
| each | 0,0096 | 44 |
| text | 0,004359 | 41 |
| system | 0,016465 | 37 |
| different | 0,008439 | 41 |
| t | 0,003174 | 34 |
| number | 0,008692 | 42 |
| set | 0,008522 | 38 |
| m | 0,003004 | 33 |

Number of words: 1980

*Fig.1 Dialog box of the program*

*3.3 Interface*

The following controls are used to manage the program functionality

- The name of directory with document corpus, the name of file with the general lexis, and the name of file with results are indicated in text boxes located at the top part of the main dialog box.
- Parameter *K* is assigned in the corresponding text box located at the right top of the dialog box.
- The mode of term selection is assigned by radio buttons at the right top of the dialog box.

Information about number of documents, words and selected words are shown at the bottom part of the dialog box. If file with the general lexis has incorrect format then the corresponding message appears.

A user has possibility to see the following information in three windows:

- words from document set with their frequencies (absolute, relative, and scaled values)
- general lexis, that is words with their frequencies (absolute, relative, and scaled values)
- selected words with their relative frequencies and the number of documents, where these words occurred (output data are described in p.3 above)

The view of program dialog box is presented on Figure 1

## Experiments

*4.1 Document set*

We tested the program LexisTerm in our project related with Peruvian blogosphere. The purpose of the project was to reveal the relation of active part of the blogosphere to the notion 'terrorism'. The document set included 100 documents downloaded from the Internet. Table 1 shows the general characteristics of this document corpus.

The typical solution consisted in clustering users, events, etc. Naturally, such a procedure needed an attribute space, and LexisTerm prepared this space on the basis of selected terms.

*Table 1. Lexical resources of corpus*

| | |
|---|---|
| Number of documents | 100 |
| Number of words | 45294 |
| Number of different words | 12392 |

In our experiments we studied

- how parameter *K* (level of specificity) affects on term list
- difference between options C and D
- influence of stemming

We expected that:

- the number of selected terms would be reduced approximately according the logarithmic law with respect to *K* (it could be a consequence of Zipf low) in mode C
- option D always would give essentially longer term list than option C
- stemming would increase term list

Our experiments confirmed all these suppositions.

*4.2. Experiment with different values of K and different options C/D*

In this experiment we varied the threshold *K* and options C and D. Table 1 contains the description of document set, Table 2 shows the results of experiment, and Figure 2 demonstrates these results in graphic form. The table cells contain the number of selected terms

*Table 2. Number of words for different options and K-values*

| ***k*** | ***Mode C*** | ***Mode D*** |
|---|---|---|
| 2 | 7112 | 9973 |
| 5 | 4767 | 9936 |
| 10 | 3395 | 9851 |
| 20 | 2492 | 9510 |
| 50 | 1250 | 8367 |
| 100 | 1139 | 7014 |

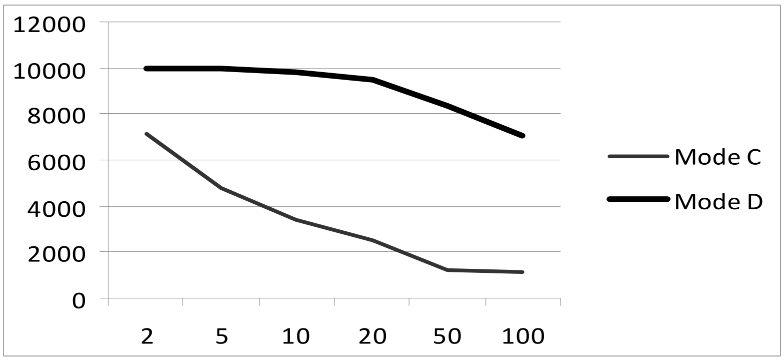


*Fig.2 Number of words for different options and K-values (graphical illustration)*

It is easy to see, that for mode C the central part of graphics is almost a straight line. It means we have here the logarithmic law having in view the logarithmic scale on axis X. Besides, one can see that mode D gives essentially longer list of terms than mode C.

*2.3. Experiment with stemming and hybrid scheme*

In this experiment we used only the mode C. First of all we completed stemming both for a given corpus and for the general lexis. The results are presented in Table 3.

*Table 3. Data about corpus and general lexis before and after stemming*

| | Corpus without stemming | General lexis without stemming | Corpus with stemming | General lexis with stemming |
|---|---|---|---|---|
| Total number of words | 45294 | 152558294 | 45294 | 152558294 |
| Number of different words | 12392 | 737799 | 8047 | 404659 |

Then we did two experiments: a) a pure experiment, when both a given corpus and a general lexis were taken after stemming and b) a mixed experiment, when terms were selected without stemming and then we applied stemming to the selected list of terms. The results are presented in Table 4, its graphical illustration is given on Figure 3

*Table 4. Number of words  for different schemes*

| *k* | ***Without stemming*** | ***Hybrid scheme*** | ***With stemming*** |
|---|---|---|---|
| 2 | 7112 | 4490 | 3216 |
| 5 | 4767 | 3376 | 2049 |
| 10 | 3395 | 2593 | 1515 |
| 20 | 2492 | 2016 | 1183 |
| 50 | 1250 | 1111 | 834 |
| 100 | 1139 | 1030 | 782 |

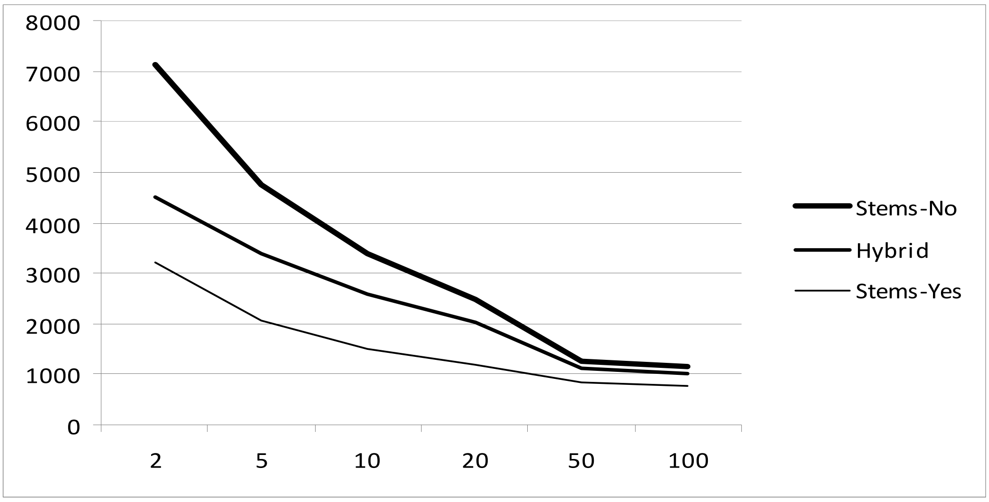


*Fig. 3. Number of words for different schemes (graphical illustration)*

One can see that stemming increases the list of selected terms. But given a large value of *K* the results become close from the point of view of the quantity of selected terms. Hybrid scheme is between both options.

## Conclusion

In the paper we introduced the notion 'term specificity' with respect to corpus and to individual documents. We developed the program LexisTerm, which implements term selection based on the introduced definitions. We demonstrated the program functionality on the real example. The results of experiments can be useful to evaluate how criterion parameters affect the list of selected terms.

## Bibliography

[Baeza-Yates, 1999] Baeza-Yates, R., Ribero-Neto, B. Modern Information Retrieval. Addison Wesley, 1999.

[Makagonov, 2000] Makagonov, P., Alexandrov, M., Sboychakov, K. A toolkit for development of the domain-oriented dictionaries for structuring document flows. In: Data Analysis, Classification, and Related Methods, Studies in classification, data analysis, and knowledge organization, Springer-Verlag, pp. 83–88, 2000

[Makagonov, 2004] Makagonov, P., Alexandrov, M., Gelbukh, A. *Clustering Abstracts instead of Full Texts*. In : "Text, Speech, Dialog", Springer, LNAI, N_3206, pp. 129-135, 2004

[Manning, 1999] Manning, D., Schutze, H. Foundations of statistical natural language processing. MIT Press, 1999.

[Pinto, 2008] Pinto, D., On clustering and evaluation of narrow domain short-text corpora. Doctoral Dissertation, Polytechnic University of Valencia, Spain, 2008.

[Porter, 1980] Porter, M. An algorithm for suffix stripping. Program, 14, pp. 130–137, 1980.

[Schutz, 2008] Schutz, A. Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical methods. Master Dissertation, National University of Ireland, Galway, 2008

[Yagunova, 2010] Yagunova, E., Pivovarova, L., The Nature of collocations in the Russian language. The Experience of Automatic Extraction and Classification of the Material of News Texts // Automatic Documentation and Mathematical Linguistics, 2010, Vol. 44, No. 3, pp. 164–175. © Allerton Press, Inc., 2010.

## Authors' Information

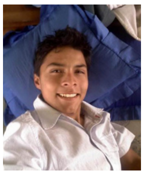
***Roque López** – Student of System Engeenering at San Agustin National University, calle Santa Catalina Nº 117 Arequipa, Peru; e-mail: rlopezc27@gmai.lcom*

*Major Fields of Scientific Research: natural language processing, text mining, social network analysis*

***Mikhail Alexandrov** – Professor, Academy of national economy and civil service under the President of Russia; Prosp. Vernadskogo 82, bld. 1, Moscow, 119571, Russia; fLexSem Research Group, Autonomous University of Barcelona, 08193 Bellaterra (Barcelona), Spain;*

*e-mail: MAlexandrov@mail.ru*

*Major Fields of Scientific Research: data mining, text mining, mathematical modeling*



***Dennis Barreda Morales** – Student of the School of Computer Science at San Pablo Catholic University, Arequipa-Perú; Researcher, Developer (Cátedra CONCYTEC), Arequipa, Peru;*

*e-mail: dennis.barreda@ucsp.edu.pe*

*Major Fields of Scientific Research: text mining, natural language processing*

***Javier Tejada Cárcamo** – Professor of Computer Science Department, San Pablo Catholic University; Research and Software Development Center of San Agustin National University (Cátedra Concytec); e-mail: jawitejada@hotmail.com*

*Major Fields of Scientific Research: natural language processing, word space models, business intelligence*