Collocation Extractor

Table of Contents

1 BASIC INFORMATION

1.1 Tool name

The tool is called **Collocation Extractor**.

1.2 Overview and purpose of the tool

Collocation Extractor identifies and extracts collocations along with their contexts of occurence from a given preprocessed text. The tool is a stand-alone application developed in C#.

1.3 A short description of the algorithm

The algorithm and different studies on its performance have been described in several papers: Ştefănescu et al. (2006), Todiraşcu et al. (2007), Ştefănescu et al. (2008), Todiraşcu et al. (2009), Ştefănescu (2010).

In this approach, we considered a collocation to be an expression formed by 2 principal content words which satisfy the following constraints:

– the distance between them is relatively constant;

– they appear together more often than expected by chance: Log-Likelihood.

Looking at this definition, one can notice, that from a strict linguistic point of view, such a construction can be seen as a strong co-occurrence, rather than a collocation.

The first component of our solution is based on a method developed by Smadja (1993). This uses the average and the standard deviation computed on distances between words to identify pairs of words that regularly appear together at the same distance, a fact which is considered to be the manifestation of a certain relation between those words. Collocations can be found by looking for such pairs for which standard deviation is small.

In order to find certain types of collocations, the application allows for POS (Part Of Speech) filtering, computing the standard deviation only for pairs having certain POS-es within a user-defined window of non-functional words. It stores all the pairs for which standard deviation is smaller than a user-defined threshold. According to Manning and Schütze (1999), a good value for this threshold is 1.5. This method can identify good candidates for multi-word expressions but not good enough. *Collocation Extractor* further filters out some of the pairs in order to keep only those composed by words which appear together more often than expected by chance. This is done by computing the Log-Likelihood scores for all the above obtained pairs and keeping only those above a user-defined threshold.

This tool is language-independent and can be also used for finding multi-word terminological expressions (Ştefănescu, 2012).

2 TECHNICAL INFORMATION

2.1 Software dependencies and system requirements

This tool requires Windows machines with Microsoft .Net Framework 3.5 installed and 2Gb of RAM.

2.2 Installation

This tool requires Microsoft .Net Framework 3.5. No other installation is required.

2.3 Execution instructions

Run the executable *Collocation Extractor.exe* and select the menu entry '*Collocations*'.

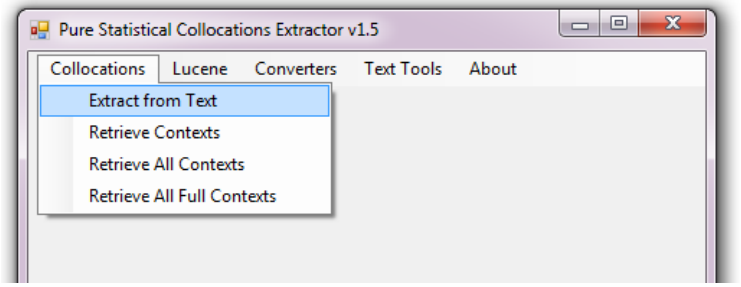

Figure 1: *Collocation Extractor* menu options

In order to extract the collocations from a text, the user needs to select the menu item '*Extract from Text*'. A configuration window will appear, allowing the user to set the parameters of collocation extraction.

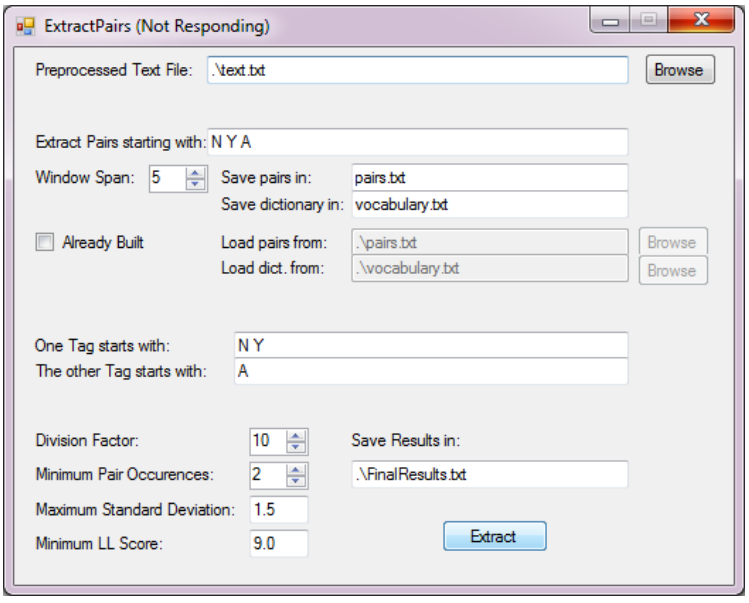

Figure 2: Configuration window for collocations extraction

The user is required to input the path to the preprocessed text file and to define the parameters of the extraction process:

- the POS-es of the words the application should take into account (*Extract Pairs starting with*). In the example given in Fig. 2, the user intends to look only at nouns (all words having the POS-tag starting with 'N' and 'Y' (proper nouns)) and adjectives (all words having the POS-tag starting with 'A'). Obviously, this field should be set according to the tagset used for pre-processing the text.
- the *Window Span* of content words which is going to be considered (e.g.: a window span of 5 (see Fig. 2) means a window of 11 content words having the current considered content word in the middle);
- the intermediary file names which will contain all word pairs and vocabulary satisfying the user constraints. If these files already exist, the user has the possibility of checking the box '*Already Build*' and give the path to these files;
- the POS tags a pair should be formed of (*One / The other Tag starts with*). In the example given in Fig. 2, the user searches for noun(N, Y)-adjective(A) pairs. It practice, it does not matter which one is first, since negative distances are also considered.
- division factor (default value is 10) refers to the number of parts in which the application divides the problem in order to consume less memory and still be efficient. This is similar to MapReduce algorithm, yet this is not parallelized. It allows us to correctly determine the frequencies of the existing pairs without consuming too much memory. The division factor should be set depending on the size of the input file. The larger the file, the higher the division factor;
- the minimum number of occurrences for a pair in order to be taken into account (default value is 2);
- the maximum standard deviation allowed for a pair (default value is 1.5);
- the minimum Log-Likelihood pair allowed for a pair (default value is 9).

This step finds the word pairs which define the collocations we are looking for. In order to find the real collocations, one needs to extract the occurring expressions formed by these principal words. In order to do this fast, the application can be used to index the sentences of the text as documents by selecting the menu item '*Index text*' (see Fig. 3).

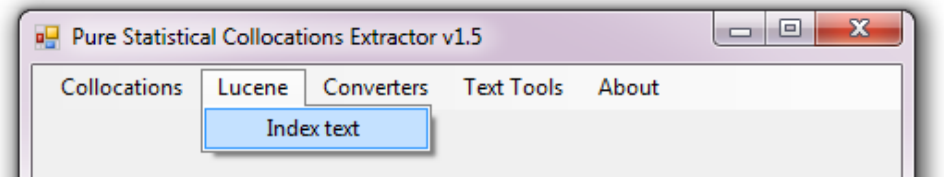

Figure 3: Indexing the input text with Lucene

After the index is created, the user can fully retrieve the collocational expressions and the general context in which they occur, by accessing the other options available in the menu entry '*Collocations*' (see Fig. 1).

Selecting the menu item '*Retrieve Contexts*' allows the user to get the context dependent data within the user interface. In this new window, the user is required to '*Load Data From*' the file containing the final results (*FinalResults.txt* in our example) and then double click a collocation from the left-side panel (see Fig. 4).

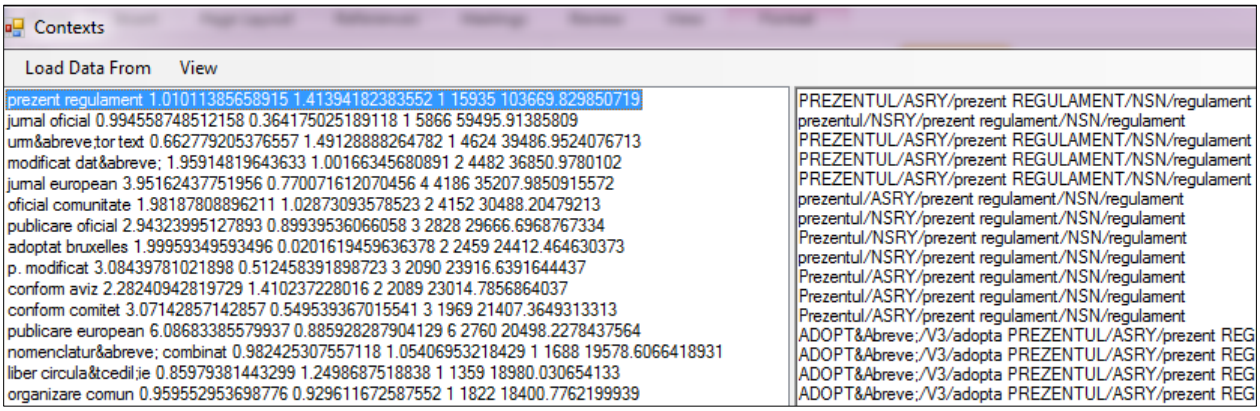

Figure 4: User interface allowing the user to see the occurences of the collocations in text

Selecting the menu item '*Expressions only*' from the menu entry '*View*' will compact all the data in the right panel into unique expressions and their frequencies (see Fig. 5).

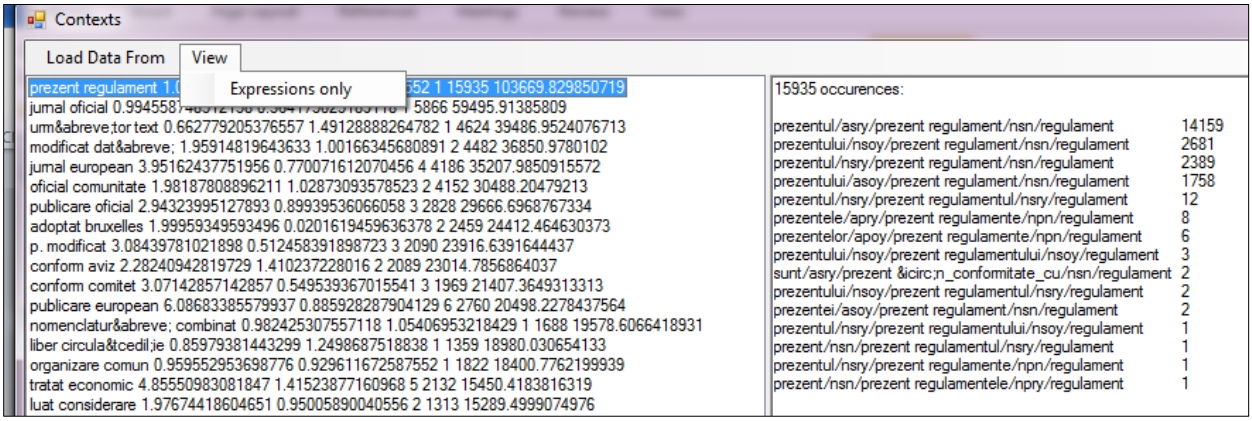

Figure 5: Compacted collocations

The user has also the option to save this data on disk by selecting the menu item '*Retrieve All Contexts*' from '*Collocations*' (see Fig. 1) (the user is required to '*Load Data From*' the file containing the final results (*FinalResults.txt* in our example)). The output would be similar to that in Fig 5. The application allows the user to also save the entire sentences containing the collocations through the option '*Retrieve All Full Contexts*' (the user is required to '*Load Data From*' the file containing the final results (*FinalResults.txt* in our example)).

In case the input file does not have the correct encoding, the application offers several possibilities of conversion (see Fig. 6) which were implemented due to practical reasons. Some other options are still under construction (see the menu entry '*Text Tools*').

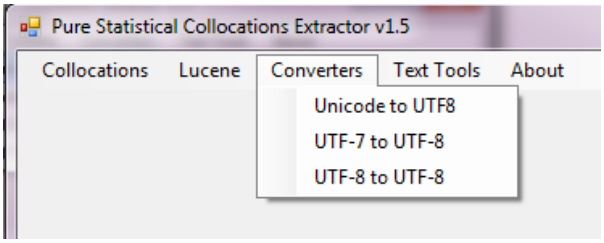

Figure 6: Options for encoding conversions of text

2.4 Input / Output data formats

The input should be a pre-processed text file of the following format (see Fig. 7):

word_form <tab> *POS-tag* <tab> *lemma*

```
investit&abreve;        ASN     investit
de      S       de
Tratat  NSN     tratat
.       PERIOD  .
Agen&tcedil;ia  NSRY    agen&tcedil;ie
este    V3      fi
reglementat&abreve;     ASN     reglementat
de      S       de
dispozi&tcedil;iile     NPRY    dispozi&tcedil;ie
Tratatului      NSOY    tratat
&scedil;i       CR      &scedil;i
ale     TP      al
prezentului     NSOY    prezent
statut  NSN     statut
.       PERIOD  .
```

Figure 7: Format of the input text

The output depends on the selected menu item:

Extract from Text:

The output file is a list of collocations ordered according to the LL score (see Fig. 8):

<word_1> <word_2> <avg> <st_dev> <round(avg)> <freq> <LL>

where:

<word_1> <word_2> are the principal words forming the collocation;
<avg> is the average distance at which the two words occur in text;
< st_dev> is the standard deviation from the average for the two words;
<round(avg)> is the actual distance used for this collocation. It is computed using *round* math function, since we need an integer for distance and *avg* is usually not an integer;

<*freq*> is the frequency of the pair;
<*LL*> is the Log Likelihood score for that pair.

```
jurnal oficial 0.994558748512158 0.364175025189118 1 5866 59495.91385809
urm&abreve;tor text 0.662779205376557 1.49128888264782 1 4624 39486.9524076713
modificat dat&abreve; 1.95914819643633 1.00166345680891 2 4482 36850.9780102
jurnal european 3.95162437751956 0.770071612070456 4 4186 35207.9850915572
oficial comunitate 1.98187808896211 1.02873093578523 2 4152 30488.20479213
publicare oficial 2.94323995127893 0.89939536066058 3 2828 29666.6968767334
adoptat bruxelles 1.99959349593496 0.0201619459636378 2 2459 24412.464630373
p. modificat 3.08439781021898 0.512458391898723 3 2090 23916.6391644437
conform aviz 2.28240942819729 1.410237228016 2 2089 23014.7856864037
conform comitet 3.07142857142857 0.549539367015541 3 1969 21407.3649313313
publicare european 6.08683385579937 0.885928287904129 6 2760 20498.2278437564
nomenclatur&abreve; combinat 0.982425307557118 1.05406953218429 1 1688 19578.6066418931
liber circula&tcedil;ie 0.85979381443299 1.2498687518838 1 1359 18980.030654133
organizare comun 0.959552953698776 0.929611672587552 1 1822 18400.7762199939
tratat economic 4.85550983081847 1.41523877160968 5 2132 15450.4183816319
luat considerare 1.97674418604651 0.95005890040556 2 1313 15289.4999074976
reglementare administrativ 2.07810499359795 0.998548334796929 2 1281 14589.118712215
```

Figure 8: Collocations list given as output

Retrieve All Contexts

The output is the same list as above, but each collocation is followed by the list of its real unique occurrences in the given text, along with their corresponding frequencies (see Fig. 9).

```
1. prezent regulament 1.01011385658915 1.41394182383552 1 15935 103669.829850719
--------------------------------------------------
prezentul/asry/prezent regulament/nsn/regulament        14159
prezentului/nsoy/prezent regulament/nsn/regulament      2681
prezentul/nsry/prezent regulament/nsn/regulament        2389
prezentului/asoy/prezent regulament/nsn/regulament      1758
prezentul/nsry/prezent regulamentul/nsry/regulament     12
prezentele/apry/prezent regulamente/npn/regulament      8
prezentelor/apoy/prezent regulamente/npn/regulament     6
prezentului/nsoy/prezent regulamentului/nsoy/regulament 3
sunt/asry/prezent &icirc;n_conformitate_cu/nsn/regulament       2
prezentului/nsoy/prezent regulamentul/nsry/regulament   2
prezentei/asoy/prezent regulament/nsn/regulament        2
prezentul/nsry/prezent regulamentului/nsoy/regulament   1
prezent/nsn/prezent regulamentul/nsry/regulament        1
prezentul/nsry/prezent regulamente/npn/regulament       1
prezent/nsn/prezent regulamentele/npry/regulament       1
##################################################
2. jurnal oficial 0.994558748512158 0.364175025189118 1 5866 59495.91385809
--------------------------------------------------
jurnalul/nsry/jurnal oficial/asn/oficial        5257
jurnal/nsn/jurnal oficial/asn/oficial   520
jurnalului/nsoy/jurnal oficial/asn/oficial      78
jurnale/npn/jurnal oficiale/apn/oficial 9
jurnalele/npry/jurnal lor/ps/lui oficiale/apn/oficial   2
jurnalele/npry/jurnal oficiale/apn/oficial      2
##################################################
3. urm&abreve;tor text 0.662779205376557 1.49128888264782 1 4624 39486.9524076713
--------------------------------------------------
urm&abreve;torul/asry/urm&abreve;tor text/nsn/text      4597
urm&abreve;toarele/apry/urm&abreve;tor texte/npn/text   26
urm&abreve;torului/asoy/urm&abreve;tor text/nsn/text    1
##################################################
4. modificat dat&abreve; 1.95914819643633 1.00166345680891 2 4482 36850.9780102
--------------------------------------------------
modificat&abreve;/asn/modificat ultima/m/ultima dat&abreve;/nsrn/dat&abreve;    3447
modificat/asn/modificat ultima/m/ultima dat&abreve;/nsrn/dat&abreve;    944
modificate/apn/modificat ultima/m/ultima dat&abreve;/nsrn/dat&abreve;   57
modificat&abreve;/asn/modificat ultima/m/ultima data/nsry/dat&abreve;   7
modificat&abreve;/asn/modificat la/s/la data/nsry/dat&abreve;   5
modificate/apn/modificat la/s/la data/nsry/dat&abreve;  3
modificat&abreve;/asn/modificat ultim&abreve;/m/ultim&abreve;
dat&abreve;/nsrn/dat&abreve;    3
modificat/asn/modificat la/s/la data/nsry/dat&abreve;   2
modificat/asn/modificat ultima/m/ultima data/nsry/dat&abreve;   2
modificat&abreve;/asn/modificat &icirc;nainte_de/s/&icirc;nainte_de
data/nsry/dat&abreve;   2
```

Figure 9: '*Retrieve All Contexts*' output

Retrieve All Full Contexts

The output resembles that of the '*Retrieve All Contexts*' but in this case each collocation is followed by the list of the unique sentences in the given text which contain that collocation, along with their corresponding frequencies (see Fig. 10).

Figure 10: '*Retrieve All Full Contexts*' output

Another output is the Lucene index which is constructed when selected the command '*Lucene*'-> '*Index text*' (see Fig. 3). This will create a folder named '*LuceneIndex*' in the current folder, containing the Lucene index for the input text.

2.5 Integration with external tools

Collocation Extractor is fully self-contained.

3 CONTENT INFORMATION

This application can be tested by using the example provided in the '`test/`' folder. The user should run the executable *Collocation Extractor.exe* and then go through the above explained procedures (see Section 2.3) using as input the file '`input.txt`' in the same folder.

3.1 Test input files

See the testing kit in the '`test/`' folder. The input file is '`input.txt`' which contains a preprocessed Romanian 346.9 Mb text from the JRC-Acquis corpus (Steinberger et al., 2006).

3.2 Output files

One may obtain the following output files:

- `FinalResults.txt` – file containing the collocation list extracted from the given pre-processed text (see Section 2.4 and Fig. 8);
- `vocabulary.txt` – file containing all lemmas (with their corresponding POS tags) in the given text and their frequencies. This file is used for generating the `FinalResults.txt` file;
- `pairs.txt` – file containing all pairs extracting from the given text according to user preferences / constraints defined as in Fig. 2. On each line it contains a word pair, the distance between the words and the POS tags of the two. This file is used for generating the FinalResults.txt file;
- `log.txt` – contains the running times for different stages of the extraction process;
- `LuceneIndex` – folder which contains the Lucene index constructed as in Fig. 3;
- `Contexts.txt` – file containing the data described in Section 2.4 and Fig. 9;
- `FullContexts.txt` – file containing the data described in Section 2.4 and Fig. 10;

3.3 Running times

In order to report the running times for this tool, we run it on a 64bit 12-core Intel(R) Core(TM) i7 CPU 980 @ 3.33GHz and 16 GB of RAM.

For the example given in '`test/`' folder we obtained the following timings (see `log.txt` file in '`reference_results/`' folder):

- '*Extract from Text*' completed in 4:44 minutes;
- Lucene index completed in 1:01 minutes;
- '*Retrieve All Contexts*' completed in 5:11 minutes;
- '*Retrieve All Contexts*' completed in 5:15 minutes;

4 ADMINISTRATIVE INFORMATION

4.1 Contact

For further information, please contact Dan ŞTEFĂNESCU (http://www.racai.ro/~danstef/; danstef@racai.ro).

5 REFERENCES

Manning C., Schütze H. (1999). Foundations of Statistical Natural Language Processing, MIT Press, Cambridge.

Ştefănescu, D. (2010). Intelligent Information Mining from Multilingual Corpora. PhD thesis (in Romanian). Romanian Academy, Bucharest.

Ştefănescu, D. (2012). Mining for Term Translations in Comparable Corpora. In Proceedings of the 5th Workshop on Building and Using Comparable Corpora (BUCC 2012), Istanbul, Turkey.

Ştefănescu, D., Tufiş, D., Irimia, E. (2006). Automatic Identification and Extraction of Collocations from Texts. In Proceedings of the 2nd Romanian Workshop for Linguistic Tools and Resources Volume, 3 Nov. 2006, Bucharest, Romania (in Romanian).

Ştefănescu, D., Ceauşu, A., Ion, R., Todiraşcu, A., Heid, U., Gledhill, C., Rousselot, F. (2008). Extraction de collocations monolingues et bilingues: application a la traduction. In Proceedings of the Latin Union Conference, 28-29 Feb. 2008, Bucharest, Romania (in French). ISBN 978-9-291220-37-3.

Smadja F. (1993). Retrieving Collocations from Text: Xtract.Computational Linguistics 19, pp. 143-175.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D., Varga, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC), pp. 2142--2147. Genoa, Italy, 24-26 May 2006.

Todiraşcu, A., Gledhill, C., Ştefănescu, D. (2007). Extracting Collocations in Context: the case of Romanian VN constructions. In Proceedings of RANLP 2007, 27-29 Sep. 2007, Borovets, Bulgaria.

Todiraşcu, A., Gledhill, C., Ştefănescu, D. (2009). Extracting Collocations in Contexts. In Human Language Technology. Challenges of the Information Society, Lecture Notes in Computer Science Series, Springer Berlin / Heidelberg. ISSN: 0302-9743 (Print) 1611-3349 (Online), Volume 5603/2009, pp. 336-349, 2009. ISBN 978-3-642-04234-8.