# Tool [Name the tool you are describing]

DEACC: Lexical Dictionary Extractor from Comparable Corpora

# Overview and purpose of the tool

The purpose of the tool is to extract bilingual lexical dictionaries (word-to-word) from comparable corpora. The corpus does not have to be aligned at any level (document, paragraph, etc.)

The method implemented in this tool is introduced by (Rapp, 1999). The application basically counts word co-occurrences between unknown words in the comparable corpora and known words from a Moses extracted general domain translation table (which from now on will be referred to as the base lexicon). We adapted the algorithm to work with polysemous entries in the translation table (very frequent situation which is not treated in the standard approach).

As the purpose of this tool (and of all the other tools in the project) is to extract from comparable corpora data that would enrich the information already available from parallel corpora, it seems reasonable to focus on the open class (versus closed class) words. Obviously, this approach reduces the space and time necessities. Moreover, the closed class words we decided to ignore (pronouns, prepositions, conjunctions, articles, auxiliary verbs) do not behave according to any semantic pattern (they are too vague); therefore, they are not useful in an approach that is based on the tendency of some words to occur in the same semantic context as other words. Because in many languages, the auxiliary verbs can also be main verbs, frequently basic concepts in the language (see “be” or “have” in English), and most often the POS-taggers don’t discriminate correctly between the two roles, we decided to eliminate their main verb occurrences also. For this purpose, the user is asked to provide a list of all these types with all their forms in the language of interest other than English.

**A short description of the algorithm:**

Firstly on the corpus of the source language and secondly on the corpus of the target language, a co-occurrence matrix is computed, whose rows are all word types occurring in the corpus and whose columns are words in that corpus appearing in the base lexicon. Initially, the co-occurrence matrix contains the co-occurrence frequencies.

The next step is to replace all these frequencies with the log-likelihood scores. In the end, a similarity computation is done between all the vectors in the source matrix and all the vectors in the target matrix.

For a specific source vector, the first ten target vectors with the highest similarities are considered to be the possible translations of the corresponding source-language word. The similarity score can be used in a Moses type decoder to select the most probable translation of the word in a specific context.

The measure used to compute the similarity score is DiceMin (see (Gamallo, 2008) for a discussion about the efficiency of several similarity metrics combined with two weighting schemes: simple occurrences and log likelihood).

# Software dependencies and system requirements

The aligner is implemented in the programming language C#, under the .NET Framework 2.0. It requires the following settings to run:

* .NET Framework 2.0.
* 2+ GB RAM (4 GB preferred)
* On a multi-processing system, the computing of the similarity score can be divided on different processors at the user’s request, by a parameter in the configuration file of the application.

# Installation

The application does not require any installation.

# Execution instructions

Given that the user machine has .NET Framework 2.0 installed, the application can be run as an executable file both under windows and linux platforms.

The .exe file must be placed in an working folder, containing two subfolders: “source corpus” and “target corpus” and two other files: the “base lexicon” and a configuration file named: “cooc.cfg”

The “source corpus” and “target corpus” folders will contain one or more documents, named after the rule: "\*\_corpus.txt". The text in the documents should be in the format: word\_form|lemma|POS;

The base lexicon is in the format: source\_form|target\_form.

The cooc.cfg file, reproduced below, is self-explanatory; the configuration file might be subject to further modifications during the next month, in case the application suffers any changes.

// 1. if the user's machine has multiple processors, the application can apply a function that splits the time consuming problem of //computing the vector similarities and runs it in parallel.

\*multithreading:yes/no (default=no)

//2.to avoid overloading the memory, the application gives the user the opportunity to decide how many of the source/target //vectors are loaded in the memory at a specific moment; it avoids overloading the memory but can bring an important time delay;

//this parameter is activated only for "multithreading:yes"

//default value: 0; if the parameter's value is bigger than the number of vectors in the matrix, its use becomes obsolete.

\*loading:int(default=0)

//3.the minimal frequency in the corpus of the words the user wants to find translation equivalents for: being based on word //countings, the method is sensitive to the frequency of the words. the bigger frequency, the better performance. this parameter //should be at least bigger than 3 and should take into account the corpus dimension.

\*frequency:int(default=3)

//4.the user can specify the length of the text window in which coocurences are counted

\*window:int(default=5)

//5.asking for the loglikelyhood of a coocurence to be bigger than a certain threshold, the user can reduce the space and time costs

\*ll:int(default=3)

//6.the user is asked to introduce a list of all the auxiliary/modal verbs for the source language, with all their morphological //variants, separated by white space

\*sourceamverblist:string (default=is are be will shall may can etc.)

//7.the user is asked to introduce a list of all the auxiliary/modal verbs for the source language, with all their morphological //variants, separated by white space

\*targetamverblist:string (default=este sunt suntem sunteţi fi poate pot putem puteţi etc.)

// 8.the user can decide if he/she allows to the application to cross the boundaries between the parts of speech (e.g. to translate a //noun as a verb)

\*crossPOS:yes/no(default:no)

//9.the user has to provide a list of all the open class POS labels (i.e. labels for common nouns, proper nouns, adjective, adverbs //and main verbs) of the source language

\*sPOSlist:string(default=nc np a r vm)

//10.the user has to provide a list of all the open class POS labels (i.e. labels for common nouns, proper nouns, adjective, adverbs and main verbs) of the target language

\*tPOSlist:string(default=nc np a r vm)

//11.the user can decide if a cognet score (Levenstein Distance) will be take into account in computing the vector similiarities for proper nouns

\*LD:yes/no(default=yes)

multithreading:yes

loading:5000

frequency:10

window:5

ll:3

sourceamverblist:am is are was were been beeing had has have be will would shall should may might must can could need

targetamverblist:este sunt suntem va voi vor vom fi pot putea puteam puteaţi

crossPOS:no

sPOSlist:nc np a r vm

tPOSlist:nc np a r vm

LD:yes

# Input/output data formats

## Input data formats

The “source corpus” and “target corpus” folders will contain one or more UTF8 documents, named after the rule: "\*\_corpus.txt". The text in the documents should be in the format: *word\_form|lemma|POS*;

The base lexicon is in the format: *source\_wordform|target\_wordform*.

The cooc.cfg file, reproduced below, is self-explanatory; the configuration file might be subject to further modifications during the next month, in case the application suffers any changes. It will be provided to the user together with the .exe file.

The base lexicon and configuration file must also be UTF8.

## Output data format

UTF-8 dictionary in the format: <source\_lexical^POS>|<target\_candidate1^POS><score>#<target\_candidate2^POS><score>…#<target\_candidate10^POS> <score>

# Integration with external tools

The application does not need any external tool.

# Useful references

Gamallo P. (2008) "Evaluating two different methods for the task of extracting bilingual lexicons from comparable corpora", In Proceedings of LREC 2008 Workshop on Comparable Corpora, Marrakech, Marroco, pp. 19-26. ISBN: 2-9517408-4-0.

Reinhard Rapp. 1999. Automatic Identification of Word Translations from Unrelated English and German Cor- pora. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99), pages 519-526, college Park, Maryland, USA.