TICCLops Documentation

version 0.0.1

Martin Reynaert
Induction of Linguistic Knowledge Research Group
Tilburg centre for Cognition and Communication
Tilburg University

October 26, 2010

Contents

Chapter 1

Introduction

The spelling and OCR-error correction system Text-Induced Corpus Clean-up (TICCL) gradually developed by Martin Reynaert in prior projects at ILK is now TICCLops (TICCL online processing system), a fully operational web application and RESTful web service. This is thanks to the Computational Linguistics Application Mediator (CLAM), developed by Maarten van Gompel, also at ILK.

CLAM and TICCLops are the result of Call 1 in the CLARIN-NL programme, a Dutch initiative in the wider European CLARIN framework. The project TICCLops (CLARIN-NL/09-011) ran from February, 1st to July, 31st of 2010. The project was coordinated by Martin Reynaert. Maarten van Gompel was the scientific programmer. Martha van den Hoven, as student-assistant, did part of the work on the documentation.

Development Notes
TICCLops was designated the first step in the language technology tools workflow being developed within the CLARIN-NL project TTNWW. As a consequence, both TICCLops and to a larger extent CLAM are still under development. As such, this documentation is not definitive, but a work in progress.

Chapter 2

Functional description of TICCL

2.1 Global description of TICCL

TICCL (Text Induced Corpus Clean-up) is a system that is designed to search a corpus for all existing variants of (potentially) all words occurring in the corpus. This corpus can be one text, or several, in one or more directories, located on one or more machines. TICCL creates word frequency lists, listing for each word type how often the word occurs in the corpus. These frequencies of the normalized word forms are the sum of the frequencies of the actual surface forms found in the corpus. TICCL is a system that is useful to detect typographical errors (misprints) and OCR errors (optical character recognition) in texts. When books or other texts are scanned from paper by a machine, that turns these scans into digital text-files, errors occur. For instance, the letter combination 'in' can be read as 'm', and so the word 'regeering' is incorrectly read as 'regeermg'. TICCL can be used to detect these errors and suggest a correct form.

TICCL's main goal is to output a list of focus words together with their variants, or correction candidates. A focus word is the word form that is found in the input corpus. What are considered to be variants depends on the maximum difference, which is an input parameter of TICCL. This maximum edit distance is measured in terms of Levenshtein distance. If there is a one-letter difference in a word pair, this counts as 1 edit cost. It does not matter if the one letter is an extra letter in the variant (insertion), a missing letter in the variant (deletion), or a different letter (substitution). If two letters should change places, this counts as an edit cost of 2 (transposition).

We give some examples in Table 2.1.

Table 2.1: Edits

Correct word	Variant	Name	Cost
regering	regeering	insertion	1
regering	regerng	deletion	1
regering	regeriug	substitution	1
regering	regernig	transposition	2
regering	regermg	substitution + deletion	1 + 1

If TICCL is run with a maximum edit distance of 1, the first three variants given in the example will be paired up with the correction candidate regering , the others wont. The maximum possible edit distance that TICCL works with is 3. The bigger the edit distance used, the higher the number of false positives, meaning supposed variants of a focus word are retrieved, that are in reality not variants of the focus word, but different words. If the maximum edit distance is chosen too small, not all variants of a focus word are retrieved. A maximum edit distance of 2 usually gives the best results.

TICCL returns the number of types (distinct word forms) and tokens (total number of words in the corpus). Also the type/token ratio is returned. In a normal corpus, this ratio will be about 50%. However, if there are many OCR-errors in the texts, the ratio can be much higher.

It is also possible to evaluate how well TICCL is doing. In order to do that, a so-called gold standard evaluation file is needed, which contains the text of the corpus. In this file are all misrecognised words in the OCR-ed version of the book, lined up with their correct or canonical form. Any variant of a focus word returned by TICCL, which is also in the gold standard, is counted as a true positive. Any variant returned by TICCL that is not in the gold standard, is a false positive. Any variant not returned by TICCL, but that exists in the gold standard, is a false negative. In this way of measuring how well TICCL performs no account is taken of the correct words in the OCR for which TICCL does not report anything. For more in depth information about spelling correction evaluation, please refer to [Reynaert(2008)].

The user can influence the amount and type of words that TICCL tries to find variants for. It is possible to set criteria for word frequency and word length, so TICCL uses only words that match those criteria from the word frequency list it builds from the corpus.

TICCL assumes the corpus it is used on is not tokenized. TICCL uses its own tokenizing algorithm that is able to deal with common OCR-errors that mistakenly put punctuation inside words.

TICCL is in principle language independent, however, the resource files

that currently come with TICCL are only for Dutch.

2.2 Definitions

- corpus: the input texts that contain the words for which the variants in a given lexicon are looked up.

- lexicon: the lexicon TICCL uses as an exhaustive list of validated word forms. TICCL is released with several lexicons, see elsewhere for a description.

- canonical or frequent form: the form of a word as found in the validated word list (lexicon).

- focus word: the word form used to look up variants, taken from the corpus

- normalized variant: word forms as found in the corpus but normalized using a reduced alphabet.

- surface form: the word form exactly as it is found in the corpus, without normalization.

- character confusion: a possible confusion of one or more characters. In the earlier example 'regeermg' vs. 'regeering' the confusion is 'm' 'in'.

2.3 TICCLops' address

TICCLops is online at CLARIN Centre INL. TICCLops is available to the parties having been granted access and user rights at the following URL (password required):

http://ticclops.dev.inl.nl

2.4 Acknowledging TICCLops contribution

If TICCL has proven valuable to in your own research, please refer to the following publication:

[Reynaert(2010)]

2.5 TICCLops demonstrator

A demonstration of the TICCL system is available. This demonstration uses an 18th century Dutch book, Kort begrip der waereld-historie voor de jeugd by J.F. Martinet, printed in 1789.

This book has been scanned, using optical character recognition. The text of the book as it should be, without any errors – the so called gold standard, is also available. The gold standard for the OCR-process has been produced at INL in the framework of the European project IMPACT and was kindly put at our disposal. Within the TICCLops project we have produced a gold standard for OCR post-correction. The main difference is that an OCR gold standard faithfully reproduces what the OCR process should have recognized what is in fact on the page, the post-correction gold standard documents exactly what should have been on the page, disregarding the actual splitting of words at the end of lines or even the fact that printed books may suffer from typesetting errors.

The TICCLops demonstration can be started using the CLAM interface by surfing to the address given in Section 2.3. CLAM is a universal wrapper for Natural Language Processing tools, developed within the CLARIN-NL framework, that handles validation of the input, and the call to the actual tool, TICCL in this case. The output of TICCL is made available by CLAM for viewing or downloading. How to set up and start CLAM is described in Van Gompel (2010).

The next chapter describes the TICCLops environment as presented to the user.

Chapter 3

TICCLops online user environment: Step-by-step description

The following subsections describe the simple steps a user should take to be able to upload a corpus for correction by TICCLops and to be able to download the results after processing has been completed.

3.1 Step 1 Choose project

Figure 3.1 shows the TICCLops project screen.

The first thing to do when working with TICCL is to make a project. If you have already specified a project before, you can also choose this project from a list. If you want to start a new project, type the name in the field Project ID and click *Start project*. In order to choose an existing project, you can click on its name in the list of Project IDs.

To run the demonstrator, you should create a new project, and choose a name you consider suitable.

3.2 Step 2: Specify input

Figure 3.2 shows the TICCLops screen for input selection.

A choice has to be made what input source TICCL should use. The next step is to specify the corpus files you want TICCL to process. There are several options:

Pre-installed corpora may be available. The TICCLops demonstrator comes with the Martinet book as such a pre-installed corpus. You may

choose to explore the system by way of this. This is done by selecting the first input button *Use uploaded files*. A dropdown box will show the available pre-installed corpora. Choose *OCR* to work with the demonstrator book.

Upload a file from disk

First choose the correct input format of the file that will be uploaded. This is done with the dropdown list. The choices available in this demo are:

- Koninklijke Bibliotheek XML-formaat [utf-8]
- Plain Text Format (not tokenized) [utf-8]

Next, click button *Select and upload a file*, to choose the file from your file system.

Grab a file from the web

When choosing to grab a file that is available on the internet, the correct format must also be specified, using the dropdown list. The available formats are the same as for the previous choice. Next, the URL of the file to be retrieved must be entered, and then click button *Retrieve and add file*.

Add input from browser

It is also possible to create an input file using an editor. In order to do this, click button *Open Live Editor* and an editor will be started, as shown in Figure 3. Now text can be typed or copied into the input field, and saved under a chosen name. Also here the format of the text should be specified using the dropdown list.

Press *Add to input files* to save the document as an input file, or press *Cancel* to leave this screen without saving.

Figure 3.3 shows the TICCLops editor to create a new file.

3.3 Step 3: Parameter selection

Figure 3.4 shows the TICCLops screen for parameter selection.

Next, parameters have to be selected, to run TICCL with the desired options.

Choose a lexicon to use, from the dropdown list. The options available in this demo are:

- Contemporary Dutch lexicon
- Historical/contemporary Dutch lexicon
- No validated Dutch lexicon

TICCL can run very well without a supporting dictionary because the vocabulary present in the corpus plays an integral role in the variant retrieval process. This makes TICCL far less domain sensitive than most other approaches to spelling correction. In fact, experiments have shown that the larger the corpus TICCL is asked to process, the more reliable the observed word frequency statistics are and the better the results.

The selection criteria for the focus word need to be chosen. The word frequency minimum and maximum can have a value between 0 and 10 million, and the minimum and maximum length of the focus word can be set anywhere from 0 to 100.

Selection 1: only words appearing on a certain list are to be selected. Words on the list that do not appear in the corpus are not used as focus words. The list should have extension .lst.

Selection 2: specify the limits in frequency and word length not in relation to the corpus. Minimum and maximum frequency can be specified, and minimum and maximum word length in bytes/iso-8859-1 characters. Because the maximum frequency is not known before running TICCL, it can be useful to set this at a high value. It is not advisable to run TICCL on very short words, i.e. shorter than 5 characters, on badly OCR-ed corpora. When going for shorter words, the LD-limit must be conservative.

Example: 1000000-2-5-50 selects all words whose frequency lies between 2 and one million occurrences (including 2), with a word length between 5 and 50 (including 5 and 50).

Another parameter that needs to be set is the maximum edit distance (Levenshtein distance) between variants. This can be 1, 2 or 3. This parameter defines the scope of the search for lexical variants within the corpus.

Next, you need to specify a name that will be the prefix for all output files.

With N-best ranking the number of output canonical variants of a corpus word is limited. A choice can be made from 1, 2, 3, 5, 10 or 20 N-best ranked. In all cases the variants output are ranked, best candidate first.

Finally there are some options, that can be switched on with the checkboxes. These options are off by default.

- Conversion: Tick this box when your input date is in UTF-8 encoding.
- Text Correction: when this box is checked your input files will be rewritten and the canonical forms for variants within your corpus will be added. The original word form is not overwritten, but rather enriched

with the number of n-best canonical forms requested earlier (if, indeed, so many were found).

- Evaluation: This option only makes sense when indeed a gold standard for the corpus is available, as is the case for the TICCLops demonstrator. By checking this box, the output of TICCLops given its input parameter settings is compared with the expected outcome given the gold standard and scored on various levels. A full report of this can be found in the log file after processing has finished.

For demonstration purposes you might choose the following:

- lexicon: Historical-contemporary Dutch lexicon
- word frequency: 1-250
- word length: 6-100
- maximum edit distance: 2
- output file names prefix: as desired, preferably shorter than 25 characters
- N-best ranking: Up to 3 N-best ranked
- checkboxes Evaluation and Conversion are to be checked
- check box Text Correction if you want the actual input files to be rewritten

Next, click *Start*.

3.4 Step 4: Processing

Figure 3.5 shows the TICCLops screen for monitoring processing progress .

While TICCL is running, the status of the process is shown on the screen, and is regularly updated. It is not necessary to leave the browser open while the process is running. You can close the window and come back later. If you want to reopen the process you were running, start TICCL again and choose the project you started from the list.

It is also possible to stop TICCL running, by clicking *Abort and delete project*. This stops the process, and deletes all the uploaded input files and possibly generated output files, and the project.

3.5 Step 5: Output

Figure 3.6 shows the TICCLops screen for output viewing or downloading.

After TICCL processing is done, the output files are shown. They can be viewed separately by clicking on them, and they can be downloaded bundled together as an archive. Several archive formats are possible: zip, tar.gz and tar.bz2. To download the files in compressed format, click the format of choice. Clicking *Discard output and restart* deletes all the uploaded input files and possibly generated output files, and restarts the project. Clicking *Cancel and delete project* deletes all the uploaded input files and possibly generated output files, and the project itself. If you want to temporarily leave your project, you can close your browser, and open your project again at a later time.

Figure 3.1: TICCL project screen

Figure 3.2: TICCL input specification

Figure 3.3: TICCL input editor

Figure 3.4: TICCL input specification

Figure 3.5: TICCL processing progress

Figure 3.6: TICCL output viewing or downloading

Chapter 4

TICCLops Output

TICCL gives a list as output, consisting of word pairs where the left hand word is the canonical form, and the right-hand word is the normalized variant of the word, as found in the corpus. This variant can differ from the canonical form because of a typesetting error, spelling or typographical error, OCR error, historical spelling difference, or a combination of these.

4.1 Output files

Detailed output (.varout.xml)? Contains the same information as the short output, but in an XML-format. Also the surface forms of the normalized variants are added.

Example:

Log file (.log)? contains information about which corpus files were processed, along with the processing time per file, the number of word forms (types) and the total number of words (tokens) found in the corpus, and the type/token ratio. ?If TICCL was run with an evaluation file, the results of the evaluation are also in the log file. The log file ends with the total time run.

Error log file (error.log)?When something goes wrong during TICCLs processing it will be logged in this file.

Bibliography

[Reynaert(2008)] Reynaert M (2008) All, and only, the errors: more complete and consistent spelling and OCR-error correction evaluation. In: et al NC (ed) Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), European Language Resources Association (ELRA), Marrakech, Morocco

[Reynaert(2010)] Reynaert M (2010) Character confusion versus focus word-based correction of spelling and OCR variants in corpora. International Journal of Document Analysis and Recognition