TICCLops: Text-Induced Corpus Clean-up as online processing system

Martin Reynaert
TiCC - Tilburg University and CLST - Radboud Universiteit Nijmegen
The Netherlands
reynaert@uvt.nl

Abstract

We present the 'online processing system' version of Text-Induced Corpus Clean-up, a web service and application open for use to researchers. The system has over the past years been developed to provide mainly OCR error post-correction, but can just as fruitfully be employed to automatically correct texts for spelling errors, or to transcribe texts in an older spelling into the modern variant of the language. It has recently been re-implemented as a distributable and scalable software system in C++, designed to be easily adaptable for use with a broad range of languages and diachronical language varieties. Its new code base is now fit for production work and to be released as open source.

1 Introduction

The spelling and OCR-error correction system Text-Induced Corpus Clean-up[1] (TICCL) we gradually developed in prior projects is now TICCLops (TICCL online processing system). TICCLops is a fully operational web application and RESTful web service. This is thanks to the Computational Linguistics Application Mediator (CLAM), more fully described in our companion paper (van Gompel and Reynaert, 2014). CLAM comes with its own extensive documentation and is maintained at GitHub[2]. CLAM and TICCLops are the result of a project in the initial phase of the CLARIN-NL programme, a Dutch initiative in the wider European CLARIN framework. The project TICCLops ran in 2010. The new version of TICClops described here is the result of project @PhilosTEI[3], which now runs in the final phase of the CLARIN-NL programme.

TICCL is a later reincarnation of TISC (Text-Induced Spelling Correction) which was developed as part of the author's PhD work (Reynaert, 2005). TICCL extends some of the ideas explored in TISC to the often graver and sometimes different types of errors present in digital texts as produced by machines, in contrast to texts produced by humans. Human errors are commonly ascribed to mechanical failure as in typing, dubbed 'typos', or to cognitive causes (not knowing, failure to recall, ...) dubbed 'cognitive errors'. In contrast, machines' text digitization errors may have a plethora of causes. Their combined main effect in text digitised by means of e.g. Optical Character Reading (OCR) is nevertheless inaccurate text – even if the system in the final analysis faithfully managed to reproduce the original human typesetter's lapse. For this reason, we think OCR post-correction is an essential step in the process of overall quality enhancement of digital libraries. Also, with more and more diachronical texts becoming available digitally, more and more researchers are interested in further linguistically enriching them. To this end many wish to automatically obtain a modernised version of the texts in order not to have to adapt their synchronic linguistic annotation tools to the diachronic texts. This, too, is a use TICCL may fruitfully be put to.

[1] TICCL is now set to be released under GPL v.3. Please see: http://ticclops.uvt.nl

[2] https://github.com/proycon/clam/ Its documentation can also be found at: http:/ilk.uvt.nl/clam

[3] http://axiom.vu.nl/PhilosTEI.html

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations, pages 52–56, Dublin, Ireland, August 23-29 2014.

2 TICCL

In this section we briefly describe the heavy-duty version of TICCL geared at offline production work. We have recently described the separate C++ modules that make up the current TICCL (Reynaert, 2014). TICCL is now meant to be easily adaptable to a new language or language variety. This is achieved by first deriving the alphabet for the language from an appropriate source. To this end we currently recommend the available Aspell dictionaries in their expanded form, if the user has no other lexicon at his disposal. Such an expanded list is converted by the TICCL helper module TICCL-lexstat into a character frequency list. Characters are ranked descendingly according to their observed frequency in the dictionary. One may then decide that some characters do not really belong to the language, i.e. are infrequently present and due to loan words having been incorporated in the lexicon. A good example from the German Aspell dictionary is the character 'ñ', derived from the 7 expanded word forms for the Spanish loan word 'Señor'. By means of a frequency threshold infrequent characters for one's language can be virtually neutralized: they will still be handled by TICCL, but incorporating them as-is would simply enlarge the search space unnecessarily. This is because TICCL will perform an exhaustive lookup of alle possible character combinations up to the Levenshtein or edit limit one specifies. Each additional character retained unnecessarily in the language's alphabet would result in many thousands additional lookups having to be performed. Based on the set of characters retained for the language, TICCL-lexstat next produces the list of all possible character confusions given the chosen Levenshtein distance or LD.

A core element of TICCL, TICCL-indexer, is a purely numerical step and is in large part what allows TICCL to scale to very large corpora sizes. Based on 'anagram hashing' (Reynaert, 2010) it affords fast efficient identification of all typographical near-neighbours for the words types in a corpus. For each anagram value in the character confusion list TICCL-indexer collects the values in the corpus hash that when summed with the particular character confusion list value give a hit on another value in the anagram hash. It thus builds an index to all word pairs displaying all the possible confusions given the LD limit. This means we perform an exhaustive lookup for all possible character confusions regardless of where they occur in the word forms represented by the corpus hash. The output of TICCL-indexer can be reconstructed as a full Spellnet in the terms of (Choudhury et al., 2007), i.e. a full graph of all the word forms in the lexicon and the corpus linked by an edit distance of at most the LD imposed. TICCL-indexer represents the character confusion approach to spelling correction as introduced in (Reynaert, 2010). We describe an equivalent to the focus word approach discussed in the same paper in the next paragraph.

Most books have a vocabulary which is far smaller than that present in an extended lexicon for the particular language. Building the full Spellnet over both resources would therefore be overkill in terms of processing performed, time required and superfluous variant pairs output. We have therefore produced a new implementation which for smaller corpora is far quicker in so far that it searches for variants only for those word types actually present in the text to be corrected. This module is called TICCL-indexerNT. It can be invoked to use several processor threads, i.e. to work in a distributed fashion.

The decision whether to use TICCL-indexer rather than TICCL-indexerNT might be taken on an informed basis by comparing the number of character confusion lookups to be performed given the LD one wants to work to with the number of word types actually present in the corpus to be corrected. The character confusion approach, for an alphabet of 38 characters, amounts to 275,651 character confusion values and therefore lookups given LD 2. Very few single books have this amount of word types. For single book size correction jobs the use of TICCL-indexerNT is therefore indicated. This module is then what is invoked when TICCLops runs.

3 TICCLops

The aim of the TICCLops system is to make TICCL available to researchers whose text correction needs are modest, whose personal computer facilities may perhaps be insufficient to run the full system and whose computer skills are based more on Graphical User Interfaces rather than command-line environments.

TICCL is geared to automatically correct large corpora. Recent evaluations on about 10.000 Dutch

books (Reynaert, 2014) have shown that the extra lexical information the system derives from the corpus to be corrected effectively helps it to better perform its task. TICCLops, the TICCL 'online processing system' is however not meant for production work on massive digitized text collections. It is oriented more towards helping users to improve the lexical quality of e.g. a single book for which they have obtained an OCRed version.

The system has been designed to be user-adaptable. This is first and foremost expressed in the facilities it offers to work with different languages. In its simplest form, adapting TICCLops to a new language would involve uploading a lexicon for the language and have the system derive a language specific alphabet from the lexicon. The user has the option of imposing a character frequency threshold. The lexicon might well be e.g. a list of the expanded word forms available for almost 100 languages in open source for the spelling checker Aspell[4].

TICCL is highly modular. It may be halted after each main processing step if one so desires, to be restarted from the point it left off at the user's convenience. Only the very first and last corpus processing steps require FoLiA XML (van Gompel and Reynaert, 2013). The major first processing step is converting the text extracted from FoLiA XML into a corpus frequency list, which simply lists all the word types and their observed frequencies in the corpus. This means that the user has the option of not actually submitting his texts to the online service. This one may well not be able or be inclined to do for e.g. copyright or privacy reasons. In this case it is possible to simply upload a frequency list one has obtained by one's own means. Alternatively, the user may have his texts converted to FoLiA XML. The system has conversion facilities from plain text, Alto and Page XML and hOCR HTML, i.e. the major OCR output formats. If the corpus is in FoLiA XML, the user has the option of performing language classification on it, at the paragraph level. This is performed by FoLiA-langcat, typically before the corpus frequency list is built by the module FoLiA-stats. Given mixed language texts, one then has the option of correcting only the paragraphs in a particular language and to not deal with the others.

The web application actually shields the user from the further processing steps. One may as well log off and just wait to be notified e.g. by email that the process has run. In practice, however, this should not require more than a few minutes given a single book input.

3.1 Running TICCLops

After logging in to the system, see Figure 1, the first thing a user is required to do is to specify a 'project name', i.e. in fact create a repository for his input files and for the system's output files. Next, the users are presented with the main interface page. Here, one is able to upload the input files for one's (OCR post-)correction or transcription project. These can be e.g. OCRed book pages in FoLiA format or the frequency list derived by the users from the corpus they want to have corrected. The users may further upload their own lexicon or opt to use one of the default lexicons available.

Next, the user may specify some system parameters. For all these the most reliable defaults have already been specified. We briefly describe them here. A drop-down list allows the user to specify the language of the input corpus, by default this is Dutch, but most European languages are available. In so far as it is not advisable to let TICCL work on very short words, a limit in lower character length needs next to be specified. For this the default is 6 characters, with lower settings the system may become less precise. The LD to be imposed on the retrieved Correction Candidates or CCs is by default 2. The default number of CCs that will be output at the end of the variant identification and retrieval phase of the correction process is 5 CCs. Optionally this may be set to one, e.g. when the user wants to go for fully-automatic spelling correction (where only the best-first ranked CC is in fact relevant). Alternatively, this may be set to maximally 33 CCs – with a sensible range of in-between values – in case the user intends to present these to human annotators in e.g. a crowd sourcing setting aimed at corpus correction. Please see Figure 2.

Only if the input was FoLiA XML has the user the option of letting TICCL edit his corpus on the basis of the list of ranked CCs produced and output. If this option is activated, TICCL will create copies of the OCR layer paragraph elements and in these replace identified variants with their best-first ranked

[4] ftp://ftp.gnu.org/gnu/aspell/dict/0index.html

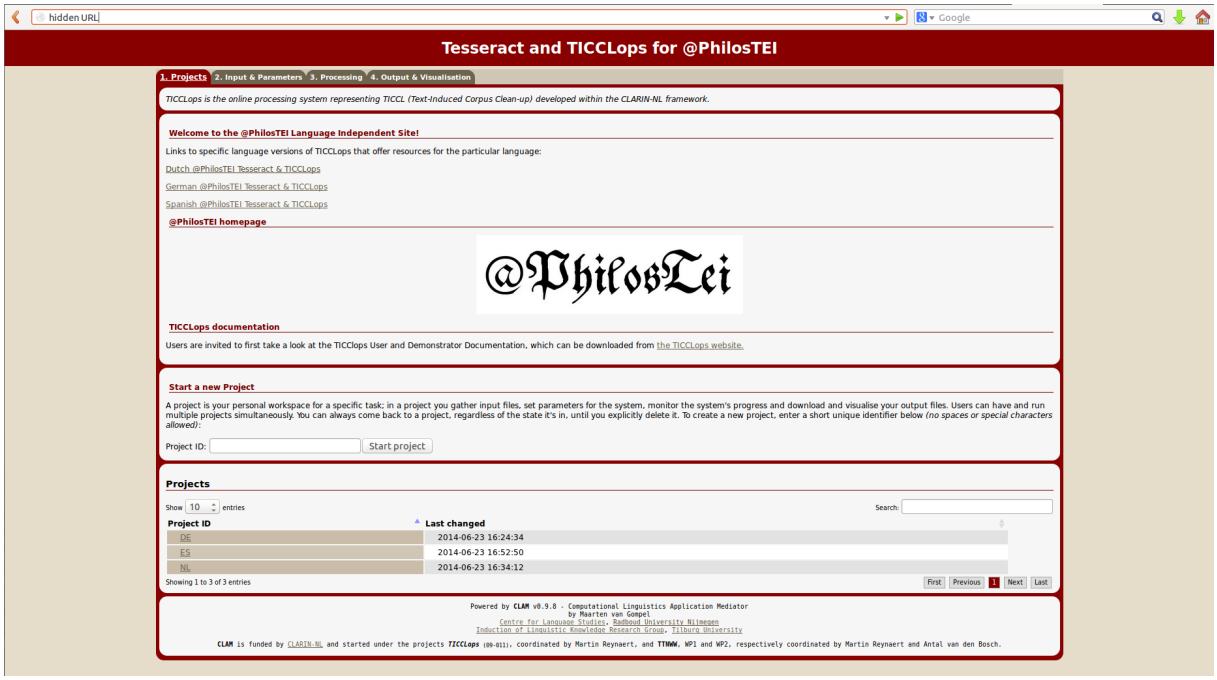

Figure 1: The @PhilosTEI start-up screen offering the user access to the specific language demonstrators, the @PhilosTEI website, the user manual, as well as the option to start a new project or visit one of the previous projects run by the system.

CC. The edited paragraph elements are identified by the attribute 'TICCL'. It will further create string elements for the user-specified number of CCs and insert them. These string elements are identified as correction 'suggestion'.

3.2 Related work

To the best of our knowledge there are no other systems geared at fully automatically performing correction of isolated non-words (whether typos or OCR errors). The open-source spelling checker Aspell can be set to automatically replace supposed errors in the text by its best-first ranked CC. However this is likely to lower the accuracy of the text rather then raise it as is shown by the evaluation results presented in (Reynaert, 2005) for both English and Dutch.

A newly open-sourced system produced by German colleagues is PoCoTo (Post Correction Tool) (Vobl et al., 2014). It is geared at interactive correction of e.g. a digitised book and shares with TICCL that it takes into account information derived from the whole book in the correction process. The correction process adapts to the text to be corrected on the basis of both text and error profiles derived from the text. It seems oriented primarily to the quite possibly very large numbers of systematic errors occurring in the text. TICCL uses as input only the electronic OCRed text version, also aiming for non-systematic errors, which may well account for a large proportion of the actual number of errors in an OCRed text. PoCoTo has an edge in that it also derives useful information from the text images that were used for the actual OCRing. An evaluative comparison of the respective strengths and potential weaknesses of both systems should be enlightening.

4 Conclusion

We have presented an overview of the CLAM-based web service and application TICCLops. This is an online instantiation of Text-Induced Corpus Clean-up not geared at large scale production work on massive digital text collections, but rather at the more modest needs of perhaps individual researchers who want to enhance the quality or modernise the spelling of single manuscripts or digitised books.

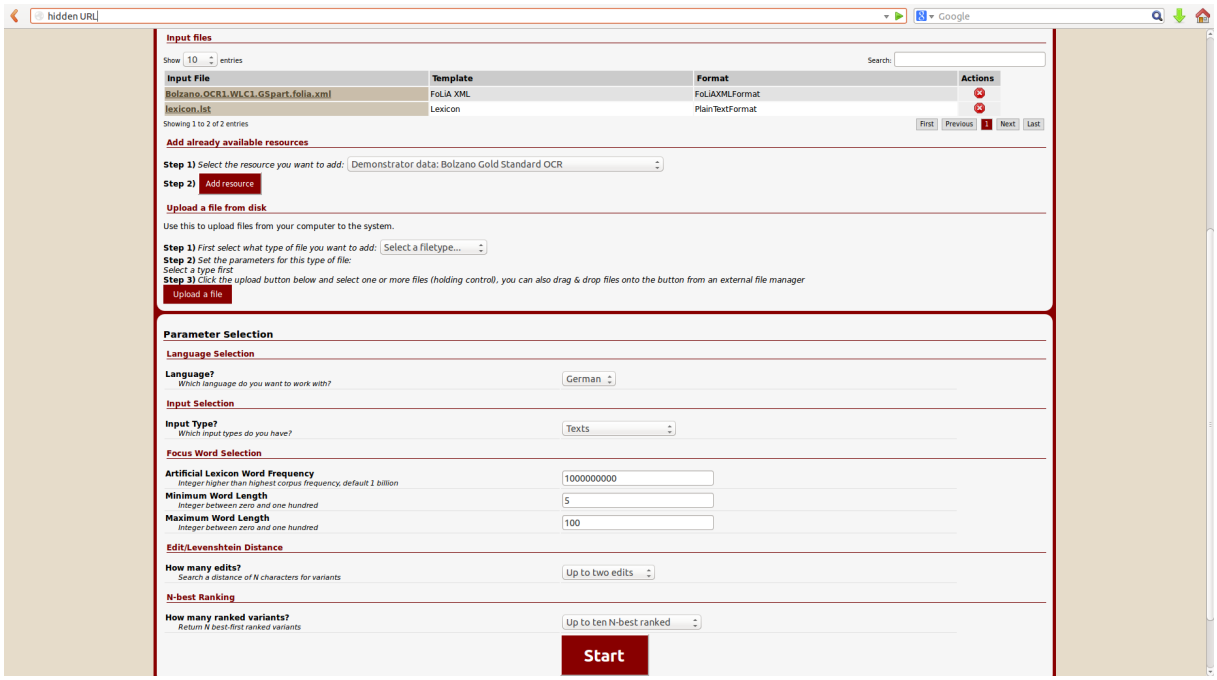

Figure 2: Having started a new project, the user has loaded the necessary input files, set some system parameters or accepted their most sensible defaults, and is now ready to run by a simple mouse click on the START button..

Acknowledgements

The author, Martin Reynaert, and TiCC senior scientific programmer Ko van der Sloot gratefully acknowledge support from CLARIN-NL in projects @PhilosTEI (CLARIN-NL-12-006) and OpenSoNaR (CLARIN-NL-12-013). The author further acknowledges support from NWO in project Nederlab.

References

Monojit Choudhury, Markose Thomas, Animesh Mukherjee, Anupam Basu, and Niloy Ganguly. 2007. How difficult is it to develop a perfect spell-checker? A cross-linguistic analysis through complex network approach. *Proceedings of the second workshop on TextGraphs: Graph-based algorithms for natural language processing*, pages 81–88.

Martin Reynaert. 2005. *Text-induced spelling correction*. Ph.D. thesis, Tilburg University.

Martin Reynaert. 2010. Character confusion versus focus word-based correction of spelling and OCR variants in corpora. *International Journal on Document Analysis and Recognition*, 14:173–187. 10.1007/s10032-010-0133-5.

Martin Reynaert. 2014. Synergy of Nederlab and @PhilosTEI: diachronic and multilingual Text-Induced Corpus Clean-up. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. ELRA.

Maarten van Gompel and Martin Reynaert. 2013. FoLiA: A practical XML Format for Linguistic Annotation - a descriptive and comparative study. *Computational Linguistics in the Netherlands Journal*, 3.

Maarten van Gompel and Martin Reynaert. 2014. CLAM: Quickly deploy NLP command-line tools on the web. In *this volume*. COLING 2014.

Thorsten Vobl, Annette Gotscharek, Ulrich Reffle, Christoph Ringlstetter, and Klaus Schulz. 2014. PoCoTo - An Open Source System for Efficient Interactive Postcorrection of OCRed Historical Texts. In *Proceedings of Datech 2014*. ACM.