CTAP: A Web-Based Tool Supporting Automatic Complexity Analysis

Xiaobin Chen and **Detmar Meurers**
LEAD Graduate School and Research Network
Department of Linguistics
Eberhard Karls Universität Tübingen, Germany
{xiaobin.chen,detmar.meurers}@uni-tuebingen.de

Abstract

Informed by research on readability and language acquisition, computational linguists have developed sophisticated tools for the analysis of linguistic complexity. While some tools are starting to become accessible on the web, there still is a disconnect between the features that can in principle be identified based on state-of-the-art computational linguistic analysis, and the analyses a teacher, textbook writer, or second language acquisition researcher can readily obtain and visualize for their own collection of texts.

This short paper presents a web-based tool development that aims to meet this challenge. The Common Text Analysis Platform (CTAP) is designed to support fully configurable linguistic feature extraction for a wide range of complexity analyses. It features a user-friendly interface, modularized and reusable analysis component integration, and flexible corpus and feature management. Building on the Unstructured Information Management framework (UIMA), CTAP readily supports integration of state-of-the-art NLP and complexity feature extraction maintaining modularization and reusability. CTAP thereby aims at providing a common platform for complexity analysis, encouraging research collaboration and sharing of feature extraction components to jointly advance the state-of-the-art in complexity analysis in a form that readily supports real-life use by ordinary users.

1 Introduction

Linguistic complexity is a multifaceted construct used in a range of contexts, including the analysis of text readability, modeling the processing difficulty of sentences in human sentence processing, analyzing the writing of second language learners to determine their language proficiency, or for typological comparison of languages and their historical development. To analyze linguistic complexity in any of these contexts, one needs to identify the observable variedness and elaborateness (Rescher, 1998; Ellis, 2003, p. 340) of a text, which can then be interpreted in relation to the nature of the task for which a text is read or written, or the characteristics of the individuals engaged in reading or writing. In this paper, we are concerned with this first step: identifying the elaborateness and variedness of a text, sometimes referred to as absolute complexity (Kusters, 2008).

Measure of absolute complexity for the purpose of selecting reading materials or the analysis of learner language range from more holistic, qualitative perspectives to more analytic, quantitative approaches. While we here focus on the latter, reviews of both can be found in Pearson and Hiebert (2014), Collins-Thompson (2014), Benjamin (2012), Ellis and Barkhuizen (2005) and Wolfe-Quintero (1998).

The present paper describes a system that supports the extraction of quantitative linguistic features for absolute complexity analysis: the Common Text Analysis Platform (CTAP). CTAP is an ongoing project that aims at developing a user-friendly environment for automatic complexity feature extraction and visualization. Its fully modularized framework enables flexible use of NLP technologies for a broad range of analysis needs and collaborative research. In the following sections, we first sketch demands

that a system for complexity analysis and research should satisfy, before providing a brief description of the CTAP modules and how they are integrated to address the demands.

2 Identifying Demands

In order to find out how complexity had been measured in L2 research, Bulté and Housen (2012) reviewed forty empirical studies published between 1995 and 2008 and compiled an inventory of 40 complexity measures used in these studies (pp. 30–31). Although they found that there was "no shortage of complexity measures in SLA studies", most studies used no more than 3 measures to measure complexity. This was largely "due to the lack of adequate computational tools for automatic complexity measurement and the labour-intensiveness of manual computation" (p. 34). The authors were optimistic that some online complexity analyzers would come out in the near future and the situation would change.

As Bulté and Housen predicted, a number of complexity analysis tools were released in the past few years (e.g., Xiaofei Lu's Syntactic and Lexical Complexity Analyzers[1], CohMetrix's Web interface to its 106 complexity features[2], and Kristopher Kyle's Suite of Linguistic Analysis Tools[3], etc.). While they make it possible for researchers to measure absolute linguistic complexity easier and faster, these tools were generally not designed for collaborative research and are limited in terms of usability and platform compatibility, provide no or very limited flexibility in feature management, and do not envisage analysis component reusability. As a result, they are not suitable (and generally were not intended) as basis for collaborative research on complexity, such as joint complexity feature development.

Commercial systems such as ETS's TextEvaluator[4] and Pearson's Reading Maturity Metric[5] also implemented automatic complexity analysis for readability assessment (see Nelson et al. (2012) for a comprehensive review and assessment of such systems.) However, the commercial nature of these systems limits the transparency of the mechanisms they employ and future research cannot be freely developed on this basis. The Text Analysis, Crawling, and Interpretation Tool TACIT (Dehghani et al., 2016) provides an open-source platform for text analysis. While linguistic complexity analyses could be integrated in this framework, it so far is primarily geared towards crawling and text analysis in a social media context, e.g., for sentiment analysis.

These complexity analysis tools overlap in terms of the complexity features offered by different systems. For example, the tools exemplified earlier contain a significant amount of lexical feature overlap across systems. While this can be useful for cross-validating calculated results, it also duplicates analyses options without giving the user the choice of selecting the set of analyses needed to address the specific needs. A more optimal scenario would be based on a common framework where developers of feature extraction tools can collaborate and share analysis components, release analysis tools to be used by researchers who focus on different aspects of the complexity problems (e.g., relative complexity for a specific target audience).

Another issue of existing complexity analysis tools concerns (re)usability. Many of these tools are released as standalone precompiled software packages or program source code. Precompiled packages not only cause cross-platform compatibility problems, but also are difficult to adapt to meet the user's specific needs. The source code option provides maximum flexibility, but are usable only to expert users or programmers. It should be noted that a lot of complexity researchers are linguists, psychologists, or cognitive scientists, but not necessarily computer scientists or programmers. Consequently, developing a complexity analysis system with user-friendly interface and visualization features are on demand.

Last but not least, there is also the challenge of complexity feature proliferation over the past years. Researchers are systematically exploring and identifying new features that contribute to our understanding of linguistic complexity. For example, CohMetrix (McNamara et al., 2014) provides 106 metrics for measuring cohesion and coherence. Housen (2015) identified more than 200 features for measuring L2

[1] http://www.personal.psu.edu/xxl13/download.html

[2] http://cohmetrix.com

[3] http://www.kristopherkyle.com

[4] Formerly SourceRater, cf. https://texteval-pilot.ets.org/TextEvaluator

[5] http://www.pearsonassessments.com/automatedlanguageassessment/products/100000021/reading-maturity-metric-rmm.html#tab-details

complexity. Vajjala (2015) accumulated another 200 features for doing readability assessment. Although features overlap across systems, the number of complexity features used and compared by researchers is large and likely to grow . Not every study needs to use all these features, nor any tool provides a full set. Researchers interested in linguistic complexity arguably would benefit from a system that readily supports them in choosing and applying complexity analyses from a large repository of features, without requiring NLP expertise.

3 System Architecture of CTAP

The CTAP system is designed to address the issues reviewed in the previous section. The goal is a system that supports complexity analysis in an easy-to-use, platform independent, flexible and extendable environment. The system consists of four major user modules—Corpus Manager, Feature Selector, Analysis Generator, and Result Visualizer—as well as a Feature Importer administrative module. Figure 1 shows the system architecture and module relationships.

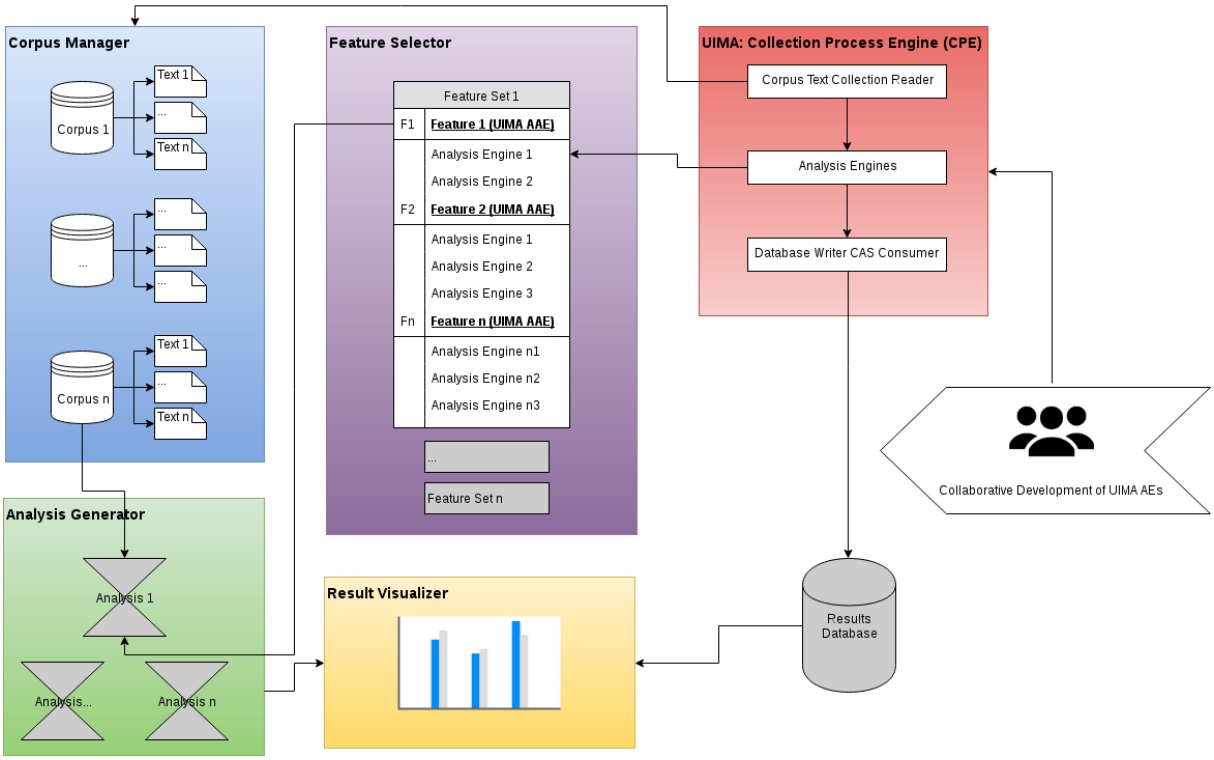

Figure 1: CTAP modules and their relationship

The Corpus Manager helps users manage the language materials that need to be analyzed. They can create corpora to hold texts, folders to group corpora and tags to label specific texts. The text labels will then be used to help filter and select target texts for analysis. They can also be used to group texts for result visualization purposes.

Other complexity analyzers usually limit users to a fixed set of features that the analyzer extracts. The Feature Selector from CTAP enables users to group their selection of the complexity features into feature sets. This flexibility is realized by utilizing the Unstructured Information Management framework (UIMA[6]) provided by the Apache Foundation. By using the UIMA framework, every complexity feature can be implemented as an Aggregate Analysis Engine (AAE) which chains up a series of primitive Analysis Engines (AEs). Each AE may be a general purpose NLP components, such as a sentence segmenter, parser, or POS tagger. It may also be one that calculates some complexity feature values based on analysis results from upstream AEs or components. This setup enables and encourages reusability of

[6] https://uima.apache.org

AEs or analysis components, thus making collaborative development of complexity feature extractors easier and faster.

After collecting/importing the corpora and selecting the complexity features, the users can then generate analyses in CTAP's Analysis Generator. Each analysis extracts a set of features from the designated corpus. Results of the analysis are then persisted into the system database and may be downloaded to the user's local machine for further processing. The user can also choose to explore analysis results with CTAP's Result Visualizer. The UIMA framework supports parallel computing that can easily scale out for handling big data analysis needs.

The Result Visualizer is a simple and intuitive module that plots analysis results for the user to visualize preliminary findings from the analysis. It supports basic plot manipulation and download. Figures 2–5 show screenshots of the user modules.

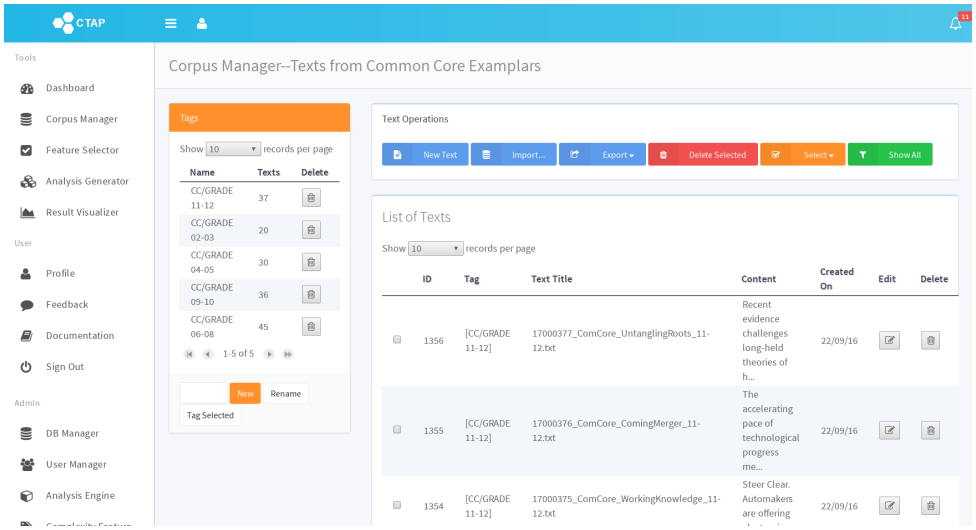

Figure 2: Corpus Manager module screen shot

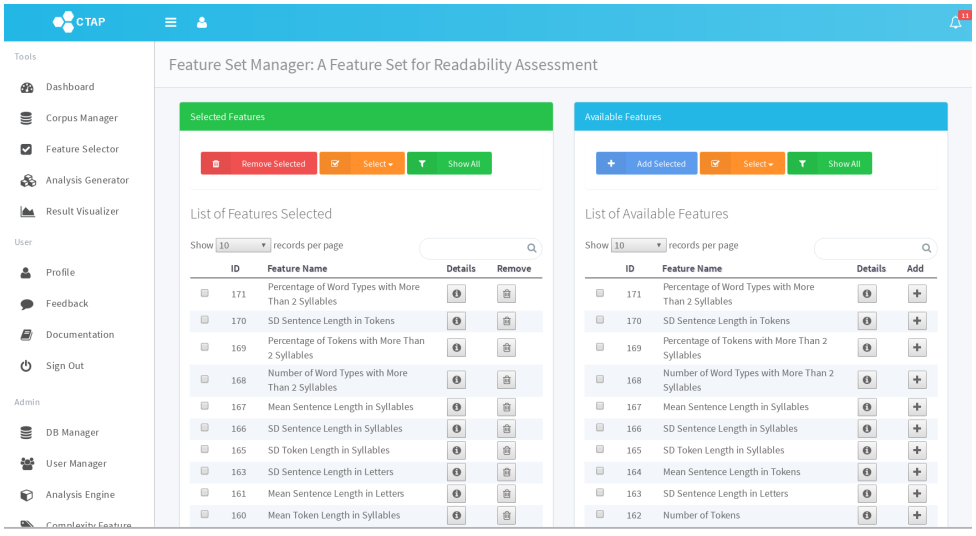

Figure 3: Feature Selector module screen shot

4 Design Features of CTAP

The target users of the CTAP system are complexity feature developers and linguists or psychologists who might not necessarily be computer science experts. As a result, the system features the following design.

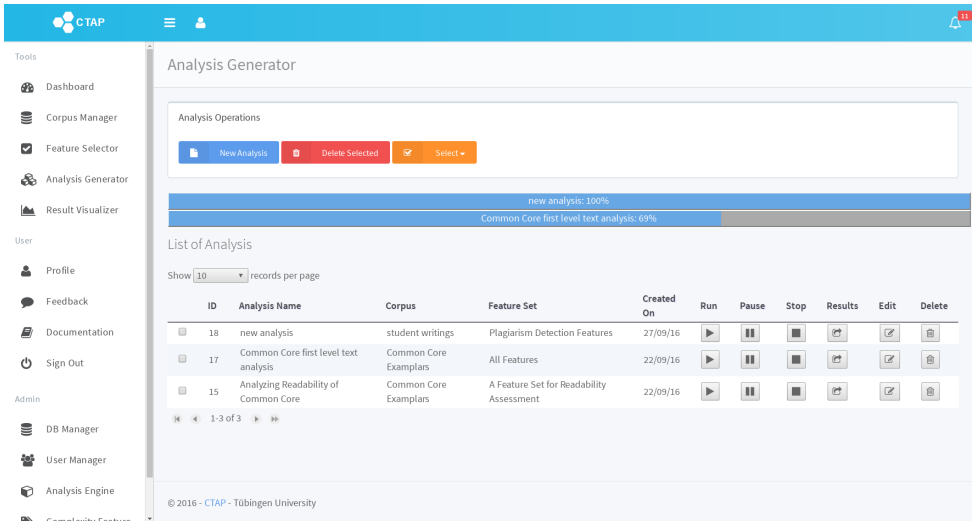

Figure 4: Analysis Generator module screen shot

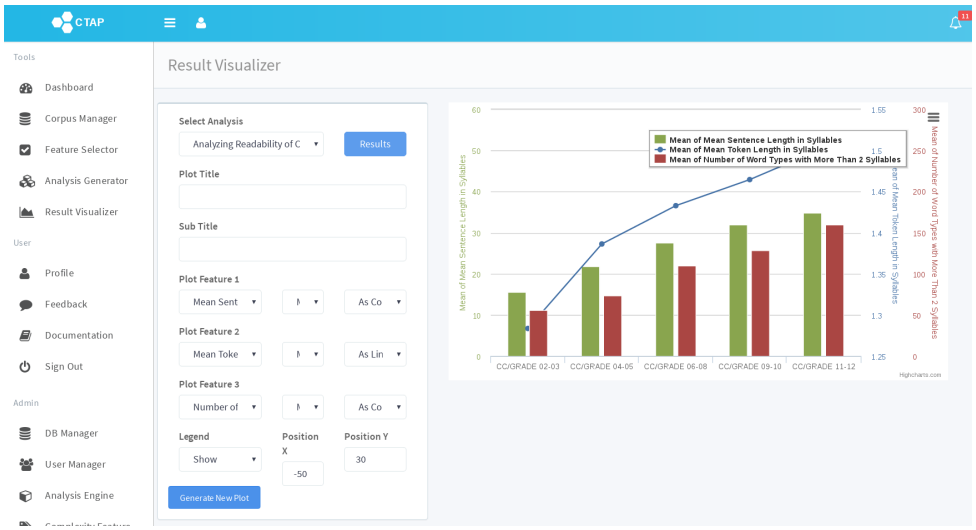

Figure 5: Result Visualizer module screen shot

Consistent, easy-to-use, friendly user interface. The CTAP system is deployed as a Web application, which strikes a balance between usability, flexibility and cross-platform compatibility. The GUI provided on the Web makes it easy to access, user-friendly and platform neutral. The CTAP client frontend was written with Google Web Toolkit[7] (GWT), an open source and free technology that enables productive development of high-performance web applications. This avoids the necessity to compile the software for different operating systems, which has been proved to be a major frustration for small development teams or single developers who do not have enough resources to deal with platform differences.

Modularized, reusable, and collaborative development of analysis components. The CTAP analysis back-end is written under the UIMA framework. Each analysis unit is implemented as a UIMA AE. Since a lot of the AEs are commonly required by different complexity features, modularizing analysis into smaller AEs makes it easier to reuse and share components. The AEs included into CTAP are open sourced and we encourage contribution from feature developers. A community effort will enhance complexity research to a greater extent.

[7] http://www.gwtproject.org

Flexible corpus and feature management. This feature is a luxury in light of the existing complexity analysis tools. However, this feature is of special value to users with lower information and communication technology competence. Users choose from the feature repository the system provides a set of features that meet their needs, the CTAP system then generates a UIMA AAE to extract the chosen feature values. It frees users from tediously editing analyzer source code, which is also often error-prone.

5 Summary and Outlook

The CTAP project is under active development at the moment. A demo version of the system has been finished (`http://www.ctapweb.com`), establishing the feasibility of the design, architecture, and the features described in this paper. Additional functionality, such as allowing users to add their own feature extractors and providing modules supporting machine learning to combine the collected evidence will be added in the near future. We are currently working on populating the system with complexity feature extractors implemented as UIMA AEs by either migrating existing analyzer code as well as reimplementing features reported on in other complexity studies. To validate and exemplify the approach, we plan to replicate the state-of-the-art linguistic complexity analyses for English (Vajjala and Meurers, 2014) and German (Hancke et al., 2012) using CTAP, making the components on which the analyses are based readily available.

In making the tool freely available under a standard Creative Commons by-nc-sa licence, we would also like to call for contribution from other researchers. Interested parties are encouraged to join and contribute to the project at `https://github.com/ctapweb`. Only by making use of joint effort and expertise can we envisage a production level system that can support joint progress in the complexity research community, while at the same time making the analyses readily available to ordinary users seeking to analyze their language material—be it to study language development or to develop books better suited to the target audience.

Acknowledgments

We would like to thank the anonymous reviewers for their comments. This research was funded by the LEAD Graduate School & Research Network [GSC1028], a project of the Excellence Initiative of the German federal and state governments, and received support through grants ANR-11-LABX-0036 (BLRI) and ANR-11-IDEX-0001-02 (A*MIDEX).

References

Rebekah George Benjamin. 2012. Reconstructing readability: recent developments and recommendations in the analysis of text difficulty. *Educational Psychology Review*, 24(1):63–88.

Bram Bulté and Alex Housen. 2012. Defining and operationalising l2 complexity. In Alex Housen, Folkert Kuiken, and Ineke Vedder, editors, *Dimensions of L2 Performance and Proficiency: Complexity, Accuracy and Fluency in SLA*, pages 21–46. John Benjamins, Amsterdam.

Kevyn Collins-Thompson. 2014. Computational assessment of text readability: A survey of past, present, and future research. *International Journal of Applied Linguistics*, 165(2):97–135.

Morteza Dehghani, Kate M. Johnson, Justin Garten, Reihane Boghrati, Joe Hoover, Vijayan Balasubramanian, Anurag Singh, Yuvarani Shankar, Linda Pulickal, Aswin Rajkumar, and Niki Jitendra Parmar. 2016. Tacit: An open-source text analysis, crawling, and interpretation tool. *Behavior Research Methods*, pages 1–10.

R. Ellis and G. P. Barkhuizen. 2005. *Analysing learner language*. Oxford University Press, Oxford.

Rod Ellis. 2003. *Task-based Language Learning and Teaching*. Oxford University Press, Oxford, UK.

Julia Hancke, Detmar Meurers, and Sowmya Vajjala. 2012. Readability classification for German using lexical, syntactic, and morphological features. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 1063–1080, Mumbay, India.

Alex Housen. 2015. L2 complexity—a difficult(y) matter. Oral presentation given at the Measuring Linguistic Complexity: A Multidisciplinary Perspective workshop, Université catholique de Louvain, Louvain-la-Neuve.

Wouter Kusters. 2008. Complexity in linguistic theory, language learning and language change. *Language complexity: Typology, contact, change*, pages 3–22.

Danielle S. McNamara, Arthur C. Graesser, Philip M. McCarthy, and Zhiqiang Cai. 2014. *Automated Evaluation of Text and Discourse with Coh-Metrix*. Cambridge University Press, New York, NY.

J. Nelson, C. Perfetti, D. Liben, and M. Liben. 2012. Measures of text difficulty: Testing their predictive value for grade levels and student performance. Technical report, The Gates Foundation.

P. David Pearson and Elfrieda H. Hiebert. 2014. The state of the field: Qualitative analyses of text complexity. *The Elementary School Journal*, 115(2):161–183.

Nicolas Rescher. 1998. *Complexity: A philosophical overview*. Transaction Publishers, London.

Sowmya Vajjala and Detmar Meurers. 2014. Readability assessment for text simplification: From analyzing documents to identifying sentential simplifications. *International Journal of Applied Linguistics, Special Issue on Current Research in Readability and Text Simplification*, 165(2):142–222.

Sowmya Vajjala. 2015. *Analyzing Text Complexity and Text Simplification: Connecting Linguistics, Processing and Educational Applications*. Ph.D. thesis, University of Tübingen.

Kate Wolfe-Quintero, Shunji Inagaki, and Hae-Young Kim. 1998. *Second Language Development in Writing: Measures of Fluency, Accuracy & Complexity*. Second Language Teaching & Curriculum Center, University of Hawaii at Manoa, Honolulu.