# CoCoGen - Complexity Contour Generator

*Automatic Assessment of Linguistic Complexity Using a Sliding-Window Technique*

Ströbel, M.; Kerz, E.; Wiechmann, D.; Neumann, S.

**Publication date**
2016

**Document Version**
Final published version

**Published in**
CL4LC 2016 : Computational Linguistics for Linguistic Complexity

**License**
CC BY

Link to publication

**Citation for published version (APA):**
Ströbel, M., Kerz, E., Wiechmann, D., & Neumann, S. (2016). CoCoGen - Complexity Contour Generator: Automatic Assessment of Linguistic Complexity Using a Sliding-Window Technique. In D. Brunato, F. Dell'Orletta, G. Venturi, T. François, & P. Blache (Eds.), *CL4LC 2016 : Computational Linguistics for Linguistic Complexity: proceedings of he workshop : December 11, 2016, Osaka, Japan* (pp. 23-31). The COLING 2016 Organizing Committee. https://www.aclweb.org/anthology/W16-4103/

# CoCoGen - Complexity Contour Generator: Automatic Assessment of Linguistic Complexity Using a Sliding-Window Technique

**Marcus Ströbel**
Department of English Linguistics
RWTH Aachen University
stroebel@anglistik.rwth-aachen.de

**Elma Kerz**
Department of English Linguistics
RWTH Aachen University
kerz@anglistik.rwth-aachen.de

**Daniel Wiechmann**
Institute for Language Logic and Computation
University of Amsterdam
d.wiechmann@uva.nl

**Stella Neumann**
Department of English Linguistics
RWTH Aachen University
neumann@anglistik.rwth-aachen.de

## Abstract

We present a novel approach to the automatic assessment of text complexity based on a sliding-window technique that tracks the distribution of complexity within a text. Such distribution is captured by what we term *complexity contours* derived from a series of measurements for a given linguistic complexity measure. This approach is implemented in an automatic computational tool, *CoCoGen – Complexity Contour Generator*, which in its current version supports 32 indices of linguistic complexity. The goal of the paper is twofold: (1) to introduce the design of our computational tool based on a sliding-window technique and (2) to showcase this approach in the area of second language (L2) learning, i.e. more specifically, in the area of L2 writing.

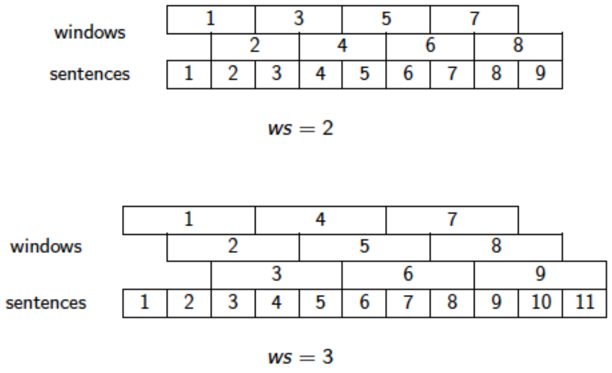
## 1 Introduction

Linguistic complexity has attracted a lot of attention in many research areas, including text readability, first and second language learning, discourse processing and translation studies. Advances in natural language processing have paved the way for the development of computational tools designed to automatically assess the linguistic complexity of spoken and written language samples. There are a variety of computational tools available which measure a large number of indices of linguistic complexity. Such tools afford speed, flexibility and reliability and permit the direct comparison of numerous indices of linguistic complexity. *Coh-Metrix* is a well-known computational tool that measures cohesion and linguistic complexity at various levels of language, discourse and conceptual analysis (McNamara, Graesser, McCarthy & Cai, 2014). Considerable gains have been made from the use of *Coh-Metrix*. In particular, an important contribution has been made to the identification of reliable and valid measures or proxies of linguistic complexity and their relation to text readability (Crossley, Greenfield, & McNamara, 2008), writing quality (Crossley & McNamara, 2012) and speaking proficiency (Crossley, Clevinger, & Kim, 2014). *Coh-Metrix* measures are also shown to serve as proxy for more complex features of language processing and comprehension (cf. McNamara et al. 2014). More recently, a number of tools have been developed that feature a large number of classic and recently proposed indices of syntactic complexity (*Syntactic Complexity Analyzer*, Lu, 2010; *TAASSC*, Kyle,

2016) and lexical sophistication (*Lexical Complexity Analyzer*, Lu 2012; *TAALES*, Kyle & Crossley, 2015). These tools provide comprehensive assessments of text complexity at a global level. They provide as output for each measure a single score that represents the complexity of a text, i.e. a summary statistics. We present a novel approach to the assessment of linguistic complexity that enables tracking the progression of complexity within a text. In contrast to a global assessment of text complexity based on summary statistics, the approach presented here provides a series of measurements for a given complexity dimension and in this way allows for a local assessment of within-text complexity. The goal of the paper is twofold: (1) to introduce the design of our computational tool which implements such an approach by using a sliding-window technique and (2) to showcase this approach in the area of second language (L2) learning.

## 2 Automatic Assessment of Linguistic Complexity Using a Sliding-Window Technique

The present paper introduces a computational tool – *Complexity Contour Generator* (*CoCoGen*) – designed to automatically track the changes in linguistic complexity within a target text. *CoCoGen* uses a sliding-window technique to generate a series of measurements for a given complexity dimension allowing for a local assessment of complexity within a text. A sliding window can be conceived of as a window with a certain size that is moved across a text. The window size (*ws*) is defined by the number of sentences it contains. The window is moved across a text sentence-by-sentence, computing one value per window for a given complexity measure. For a text comprising *n* sentences, there are $w = n - ws + 1$ windows. Given the constraint that there has to be at least one window, a text has to comprise at least as many sentences at the *ws* is wide ($n \geq ws$). Figure 1 illustrates how sliding windows of two exemplary *ws* (2 and 3) are mapped to sentences within a text.



*ws* = 2

*ws* = 3

*Figure 1: Mapping of windows for ws = 2 and ws = 3 to sentences*

The series of measurements obtained by the sliding-window technique represents the distribution of linguistic complexity within a text for a given measure and is referred here to as a *complexity contour*. The shape of a complexity contour is affected by the *ws*, a user-definable parameter. Setting the *ws* parameter to *n* will yield a single value representing the average global complexity of the text. To track the progression of complexity within a text there has to be a sufficient number of windows. As a rule of thumb, there should be at least ten times as many sentences as the window is wide to have at least ten completely distinct (i.e. non-overlapping) windows. Figure 2 illustrates the smoothed curve produced by *CoCoGen*'s sliding window approach for a sequence of 50 random numbers between 0 and 10 presents complexity contours for *ws* of 5 and 10 compared to raw data.

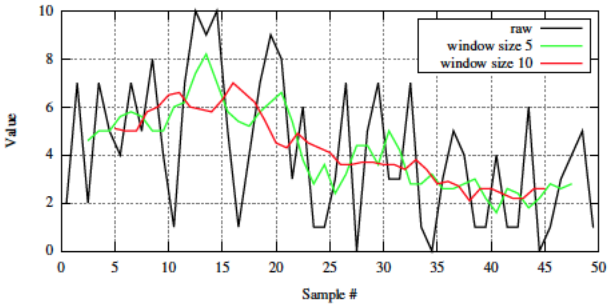


*Figure 2: Sliding windows for window sizes 5 and 10 compared to raw data*

In what follows, we address how a value for a window is obtained and how the comparison of texts of different sizes is afforded by a text-time scaling technique. There are different methods of obtaining a value for a window: One method is similar to using simple moving averages with a length equal to the *ws* over a set of measurements for all sentences in a text. This way the value is computed once per sentence and can be cached and reused in other windows that include that particular sentence. In addition, the values obtained for individual sentences can be cached for recalculating the window values for different *ws*. Another method is to apply the measure function directly to the contents of a window rather than a single sentence. As the number of windows can never exceed the number of sentences, compared to the first method, fewer calls to a measure function are needed, however, the number of the calls is greater. Furthermore, with this method it is not possible to reuse the values for different *ws*. Another disadvantage is that the *ws* may directly influence the resulting values for simple counting measures like word counts. The method implemented in CoCoGen is a compromise between the two methods discussed above: the measure function is called for each sentence, but it returns a fraction rather than a fixed value. The denominators and numerators of those fractions are then added to form the denominator and numerator of the resulting value. For complexity measures based on ratios, the result is the same as when the measure function is directly applied to the contents of a window. For counting measures, a fixed denominator of 1 is used, resulting in the arithmetic mean of the results for the sentences in the window. The idea behind this method is to obtain values for windows that do not depend on the *ws* chosen, allowing comparison of results for different window sizes.

$$(1) \qquad window_n = \frac{num_n + num_{n+1} + \ldots + num_{n+ws}}{den_n + den_{n+1} + \ldots + den_{n+ws}}$$

A scaling technique is implemented in the tool to allow comparing complexity contours across texts. As texts tend to vary in length given in number of sentences, the number of available windows will differ across texts. The scaling algorithm fits the number of windows $w_T$ for a text $T$ into a user-defined number of windows $w_{scaled}$. It is recommended to adjust the number of scaled windows to be at most as high as the largest number of windows in a text. In case the number of scaled windows is exceeded, the scaling algorithm will still work by linearly interpolating the missing information. However, the interpolated information will not contain actual data and thus won't be of much use. For that reason, the program issues a warning message if the number of scaled windows is higher than the window count for one of the input text files.

In its current version, *CoCoGen* supports 32 measures of linguistic complexity mainly derived from language learning research (Table 1 provides an overview, cf. Ströbel 2014 for details). Importantly, *CoCoGen* was designed with extensibility in mind, so that additional complexity measures can easily be added. It uses an abstract MEASURE class for the implementation of complexity measures.

Prior to the computation of complexity measures, *CoCoGen* pushes raw English text input through an annotation pipeline. While several open-source natural language analysis toolkits are available, CoCoGen uses several annotators from one of the most used toolkits, *Stanford CoreNLP* (Manning et al. 2014): tokenizer, sentence splitter, POS tagger, lemmatizer, named entity recognizer and syntactic parser.

*Table 1: Overview of complexity measures currently implemented in CoCoGen*

| Measure | Label | Formula |
|---|---|---|
| Kolmogorov Deflate | KOLMOGOROV | Ehret & Szmrecsanyi 2011 |
| Lexical Density | LEX.DEN | $N_{lex}/N$ |
| Number of different words / sample | LEX.DIV.NDW | $N_{w\ diff}$ |
| Number of diff. words / sample (cor.) | LEX.DIV.CNDW | $N_{w\ diff}/N_w$ |
| Type-Token Ratio | LEX.DIV.TTR | $T/N$ |
| Corrected Type-Token Ratio | LEX.DIV.CTTR | $T/\sqrt{2N}$ |
| Root Type-Token Ratio | LEX.DIV.RTTR | $T/\sqrt{N}$ |
| Sequences Academic Formula List | LEX.SOPH.AFL | $Seq_{N\ AWL}$ |
| Lexical Sophistication (ANC) | LEX.SOPH.ANC | $N_{slex\_ANC}/N_{lex}$ |
| Lexical Sophistication (BNC) | LEX.SOPH.BNC | $N_{slex\_BNC}/N_{lex}$ |
| Words on New Academic Word List | LEX.NAWL | $W_{N\ AWL}$ |
| Words not on General Service List | LEX.NGSL | $W_{N\ GSL}$ |
| Morphological Kolmogorov Deflate | MORPH.KOLMOGOROV | Ehret & Szmrecsanyi 2011 |
| Mean Length of Words (characters) | SYN.MLWC | $N_{char}/N_w$ |
| Mean Length of Words (syllables) | SYN.MLWS | $N_{syl}/N_w$ |
| Noun Phrase Postmodification (words) | SYN.NPPOSTMODW | $N_{NP\ Pre}$ |
| Noun Phrase Premodification (words) | SYN.NPPREMODW | $N_{NP\ Post}$ |
| Clauses per Sentence | SYN.CS | $N_C/N_S$ |
| Clauses per T-Unit | SYN.CT | $N_C/N_T$ |
| Complex Nominals per Clause | SYN.CNC | $N_{CN}/C$ |
| Complex Nominals per T-Unit | SYN.CNS | $N_{CN}/N_T$ |
| Complex T-Units per T-Unit | SYN.CTT | $N_{CT}/N_T$ |
| Coordinate Phrases per Clause | SYN.CPC | $N_{CP}/N_C$ |
| Coordinate Phrases per T-Unit | SYN.CPT | $N_{CP}/N_T$ |
| Dependent Clauses per Clause | SYN.DCC | $N_{DC}/N_C$ |
| Dependent Clauses per T-Unit | SYN.DCT | $N_{DC}/N_T$ |
| Syntactic Kolmogorov Deflate | SYN.KOLMOGOROV | Ehret & Szmrecsanyi 2011 |
| Mean Length Clause | SYN.MLC | $N_W/N_C$ |
| Mean Length Sentence | SYN.MLS | $N_W/N_S$ |
| Mean Length T-Unit | SYN.MLT | $N_W/N_T$ |
| T-Units per Sentence | SYN.TS | $N_T/N_S$ |
| Verb Phrases per T-Unit | SYN.VPT | $N_{VP}/N_T$ |

## 3 Application Domain: Second Language Learning

Linguistic complexity has received considerable attention in the assessment of second language (L2) performance and proficiency (cf., e.g., Ortega, 2003, 2012; Larsen-Freeman, 2006; Housen et al., 2012). It is assumed that with an increasing level of proficiency L2 writing becomes more complex and sophisticated, i.e. consisting of more advanced structures and vocabulary (Wolfe-Quinterno, Inagaki & Kim, 1998). For this reason, measures of linguistic complexity have been seen as basic descriptors of L2 performance and as indicators of L2 proficiency. While there is still much controversy as to how linguistic complexity should be defined, operationalized and measured (cf., Larsen-Freeman, 2009; Housen et al., 2012; Connor-Linton & Polio, 2014), there is a general consensus that it is a multidimensional construct affected by a number of dimensions at various levels of linguistic description (e.g. Bulté & Housen, 2014).

As mentioned in the introduction, L2 learning research has benefited tremendously from the development of computational tools designed to automatically assess linguistic complexity of texts based on a wide range of indices. More specifically, the development of such tools has made an important contribution to the identification of reliable and valid measures of linguistic complexity and their relation to L2 written and spoken performance and proficiency. A number of studies have demonstrated that automatically computed indices of linguistic complexity can successfully predict human judgments of L2 text quality (e.g. McNamara, Crossley & McCarthy, 2009) and L2 speaking proficiency (e.g. Kyle & Crossley, 2015) and can be used to discriminate between L1 and L2 texts (e.g. Crossley & McNamara 2009). More recently, a number of computational tools have been developed featuring a wide range of classic indices, fine-grained indices as well as indices informed by recent insights from language learning and processing research. One such tool is *TAALES* (Kyle & Crossley, 2015) that supports 104 classic and new indices of lexical sophistication, covering indices of frequency, range, academic language and psycholinguistic word information. Another tool is *TAASSC* (Kyle 2016) that covers 372 classical and fine-grained indices of syntactic complexity. *CRAT* (Crossley, Kyle, Davenport & McNamara, 2016) is another recently developed tool that includes over 700 indices related to lexical sophistication, cohesion and source text/summary text overlap. The coverage of such a large number of indices allows for an extensive and comprehensive assessment of text complexity and enables the identification of the most predictive and reliable indices of L2 performance and proficiency.

The sliding-window approach implemented in *CoCoGen* adds a new perspective on the assessment of text complexity. We showcase how this approach can be put to use in the area of L2 learning. The focus here is on the advanced stages of L2 English learning which in recent years have received growing attention, primarily grounded in what has come to be known as learner corpus research (cf. Granger, Gilquin & Meunier, 2015). This line of research has provided valuable insights into how and to what extent advanced L2 learners' performance deviate from target-like behavior. The vast majority of previous studies conducted in this line of research have made L1-L2 comparisons using L2 data from corpora such as the *ICLE* (cf. Granger, Dagneaux, Meunier & Paquot, 2009) and L1 data from comparable corpora such as the *LOCNESS* (cf. https://www.uclouvain.be/en-cecl-locness.html). These corpora include writing of a general argumentative, creative or literary nature and consist of relatively short texts (e.g. average text length: *ICLE* = 617 words). These texts do not represent academic writing in a narrow sense (cf. Callies & Zaytseva, 2013), a register characterized by its compressed style (cf., Biber & Gray, 2010) as well as its own phraseology/formulaic language (e.g., Ellis et al., 2008). The mastery of this register constitutes a learning target in both L1 and L2 learning (cf., Biber et al., 2011, 2013; Hyland & Tse, 2007). Correspondingly, advanced L2 learners' performance is best evaluated against an expert writer baseline (cf., Bolton, Nelson & Hung, 2002; Römer, 2009; Kerz & Wiechmann, 2015 for discussions).

The L2 learner data used in our paper come from a corpus of 110 academic research papers on a linguistic topic written by 2nd and 3rd year students enrolled in the bachelor programmes of the English Department at the RWTH Aachen University (N ~ 486,000 words, average text length = 4,500 words). All students are L1 speakers of German and meet the criteria for advanced learner status of English based on their institutional status (undergraduates with 7-9 years of formal instruction of English before entering university) (cf. Callies, 2009:116f.). The expert corpus consists of 110 research articles on linguistic topics published in peer-reviewed journals (N ~867,000 words, average text length = 7,880 words).

We are interested in whether and to what extent the progression of complexity within L2 texts deviates from the expert-writer target. We address this question for each of the 32 complexity measures currently implemented in *CoCoGen* (cf. Table 1). We used a supervised machine learning classifier to distinguish L2 texts from expert texts based on the measurements computed by the tool. The guiding idea is that in cases where the classifier cannot distinguish between learner and expert texts, L2 performance is target-like. Conversely, in cases where the classifier can distinguish between learner and expert texts, L2 performance deviates from target-like behavior. We also want to know whether there is any advantage of using complexity contours in the classification task, rather than using summary statistics. If this is the case, we would expect classification accuracy to be higher for a classifier fed with complexity contour information compared to one fed with summary statistics information.

The complexity of 220 texts in our corpora was automatically assessed using *CoCoGen*. Figure 2 below provides a visual representation of complexity contours of a single text – a randomly selected text from our expert corpus – for two selected complexity measures: SYN.CNS and LEX.SOPH.BNC. The two plots show the progression of complexity over 100 scaled windows, indicating that for both measures complexity is not uniformly distributed.

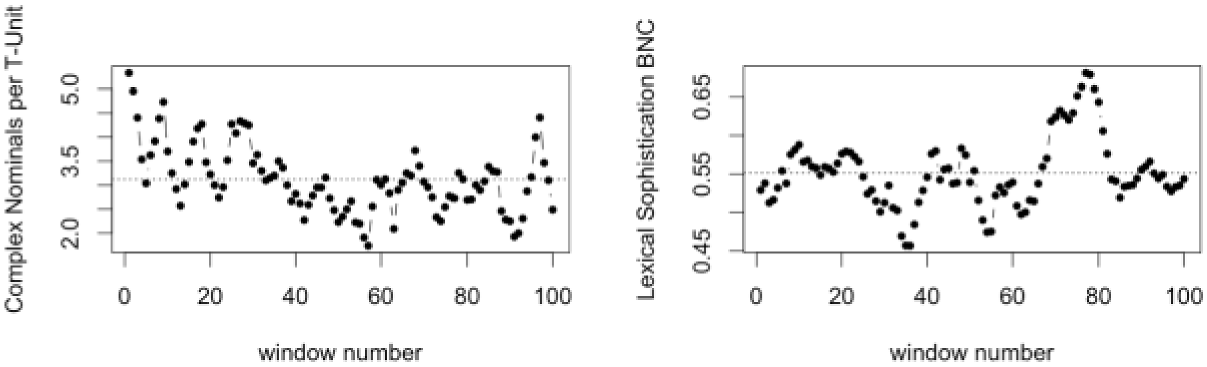


*Figure 2: Visual representation of complexity contours for two selected measures in a single expert text*

Figure 3 below illustrates the distribution of complexity for a single measure – SYN.MLS – in the two corpora. The thick solid lines describe the distributions of mean complexity using text-time scaling for both learners (blue) and experts (red). The shaded areas represent the corresponding interquartile ranges for both groups.

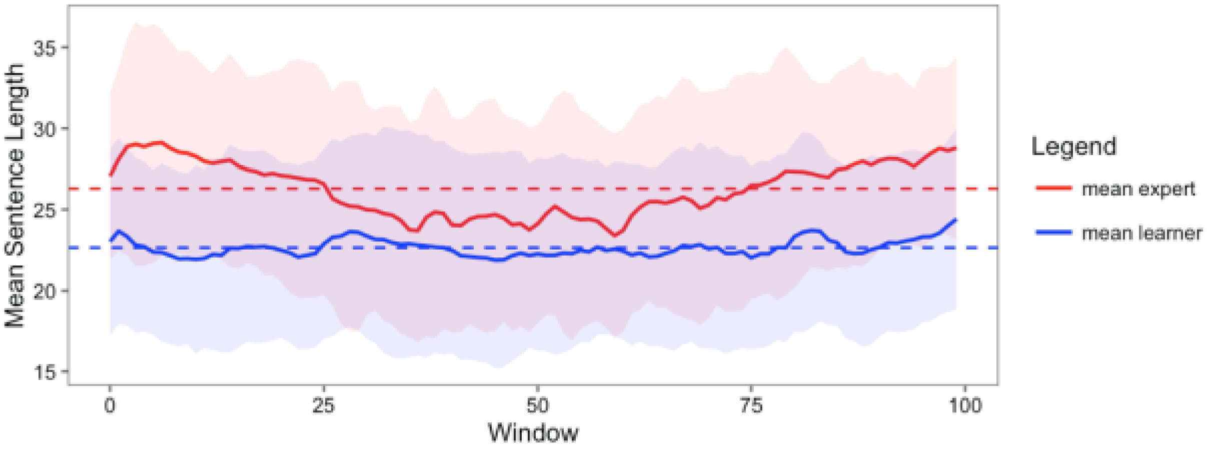


*Figure 3: Distribution of complexity for a single measure in the learner and the expert corpus*

For our classification task, we used a simple and transparent supervised machine learning technique: For each complexity measure, we identified the *empirical threshold complexity value*, i.e. the value that discriminates most strongly between L2 learner and expert texts in our data. This value served as the decision boundary for discriminating between the two groups. For the summary statistics-based approach, the description of text complexity of our corpus yielded 220 point estimates of text complexity – one score for each of the 110 learner and 110 expert texts. Each midpoint between any two values of the rank-ordered vector of complexity scores was used to divide the data into two groups. The optimal empirical threshold complexity value was found by maximizing the rand index (RI, Rand, 1971) and was validated using 10-fold cross-validation. For the sliding window approach, the empirical threshold values were determined for each window separately and the classification was determined by majority vote.[1]

Figure 5 visualizes the type of information available to the contour-based and summary statistics-based classification. The plot presents all measurements obtained for the LEX.DIV.CTTR complexity measure. Red dots represent expert texts, whereas blue dots represent L2 learner texts. The black vertical line separates the data used for the contour-based classification (left) from the data used for the summary statistics-based classification (right). The horizontal lines mark the empirical threshold complexity,

[1] Classification can also be informed by the accumulated deviation of the observed values from the threshold. We opted against this option here in the interest of transparency. However, the inclusion of this information – as well as information concerning weights of vector positions (feature weighting) – can only improve the performance of the contour-based approach advocated here.

which were used by the classifier to predict the class of a text (L2 learner/expert). In case of the summary statistics-based approach, this is a single value. In the contour-based approach, a threshold value was determined for each of the ten scaled windows. It is important to note that the thresholds found for each of the windows follow a nonlinear curve, which cannot be adequately captured by a single value.

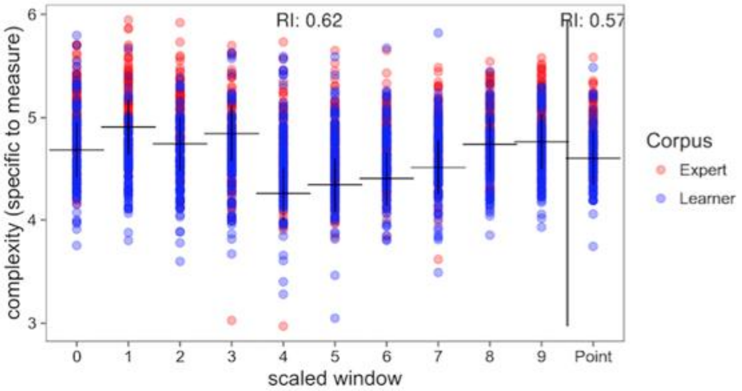


*Figure 4: Measurements of complexity and empirical threshold values for* LEX.DIV.CTTR *for ten non-overlapping windows*

Table 2 presents the Accuracy (RI), Recall, Precision, and F-measure (harmonic mean) for all measures for the summary-statistics-based and contour-based classification.

*Table 2: Results of summary statistics-based (SSB) and contour-based (CB) classification*

| | **Accuracy** | | **Recall** | | **Precision** | | **F** | |
|---|---|---|---|---|---|---|---|---|
| **Measure** | SSB | CB | SSB | CB | SSB | CB | SSB | CB |
| KOLMOGOROV | 0.6 | 0.62 | 0.59 | 0.62 | 0.59 | 0.65 | 0.59 | 0.63 |
| LEX.DEN | 0.64 | 0.66 | 0.63 | 0.65 | 0.64 | 0.67 | 0.64 | 0.66 |
| LEX.DIV.CNDW | 0.54 | 0.58 | 0.54 | 0.58 | 0.55 | 0.62 | 0.55 | 0.6 |
| LEX.DIV.CTTR | 0.58 | 0.62 | 0.58 | 0.61 | 0.62 | 0.7 | 0.6 | 0.65 |
| LEX.DIV.NDW | 0.6 | 0.64 | 0.59 | 0.63 | 0.6 | 0.66 | 0.59 | 0.64 |
| LEX.DIV.RTTR | 0.58 | 0.62 | 0.57 | 0.61 | 0.59 | 0.68 | 0.58 | 0.64 |
| LEX.DIV.TTR | 0.54 | 0.58 | 0.54 | 0.58 | 0.55 | 0.62 | 0.54 | 0.6 |
| LEX.SOPH.AFL | 0.51 | 0.57 | 0.51 | 0.56 | 0.58 | 0.68 | 0.54 | 0.61 |
| LEX.SOPH.ANC | 0.57 | 0.62 | 0.56 | 0.61 | 0.58 | 0.65 | 0.57 | 0.63 |
| LEX.SOPH.BNC | 0.6 | 0.65 | 0.6 | 0.64 | 0.6 | 0.67 | 0.6 | 0.65 |
| LEX.NAWL | 0.54 | 0.59 | 0.54 | 0.58 | 0.54 | 0.69 | 0.54 | 0.63 |
| LEX.NGSL | 0.6 | 0.62 | 0.59 | 0.61 | 0.62 | 0.64 | 0.61 | 0.63 |
| MORPH.KOLMOGOROV | 0.57 | 0.62 | 0.57 | 0.62 | 0.57 | 0.65 | 0.57 | 0.63 |
| SYN.MLWC | 0.54 | 0.6 | 0.53 | 0.59 | 0.67 | 0.66 | 0.59 | 0.62 |
| SYN.MLWS | 0.55 | 0.6 | 0.54 | 0.59 | 0.64 | 0.67 | 0.58 | 0.63 |
| SYN.NPPOSTMODW | 0.56 | 0.6 | 0.56 | 0.59 | 0.59 | 0.65 | 0.57 | 0.62 |
| SYN.NPPREMODW | 0.57 | 0.6 | 0.56 | 0.6 | 0.62 | 0.65 | 0.59 | 0.62 |
| SYN.CS | 0.5 | 0.58 | 0.5 | 0.57 | 0.53 | 0.61 | 0.52 | 0.59 |
| SYN.CT | 0.5 | 0.57 | 0.5 | 0.57 | 0.56 | 0.65 | 0.53 | 0.6 |
| SYN.CNC | 0.62 | 0.63 | 0.62 | 0.63 | 0.62 | 0.65 | 0.62 | 0.64 |
| SYN.CNS | 0.6 | 0.63 | 0.59 | 0.62 | 0.62 | 0.67 | 0.6 | 0.64 |
| SYN.CTT | 0.51 | 0.57 | 0.51 | 0.57 | 0.63 | 0.67 | 0.56 | 0.61 |
| SYN.CPC | 0.52 | 0.57 | 0.52 | 0.57 | 0.56 | 0.67 | 0.54 | 0.61 |
| SYN.CPT | 0.52 | 0.57 | 0.52 | 0.57 | 0.54 | 0.61 | 0.53 | 0.59 |
| SYN.DCC | 0.51 | 0.57 | 0.5 | 0.57 | 0.58 | 0.63 | 0.54 | 0.6 |
| SYN.DCT | 0.5 | 0.57 | 0.5 | 0.57 | 0.53 | 0.63 | 0.52 | 0.6 |
| SYN.KOLMOGOROV | 0.59 | 0.63 | 0.58 | 0.62 | 0.59 | 0.65 | 0.59 | 0.64 |
| SYN.MLC | 0.63 | 0.63 | 0.63 | 0.63 | 0.64 | 0.64 | 0.63 | 0.64 |
| SYN.MLS | 0.59 | 0.63 | 0.59 | 0.63 | 0.59 | 0.65 | 0.59 | 0.64 |
| SYN.MLT | 0.58 | 0.62 | 0.57 | 0.61 | 0.64 | 0.67 | 0.61 | 0.64 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Syn.TS | 0.53 | 0.57 | 0.52 | 0.57 | 0.63 | 0.66 | 0.57 | 0.61 |
| Syn.VPT | 0.5 | 0.57 | 0.5 | 0.57 | 0.52 | 0.62 | 0.51 | 0.59 |

We found that the contour-based classifier outperformed the summary statistics-based classifier for all measures of complexity. The contour-based classifier also identified a larger number of complexity measures that discriminate between L2 learner and expert texts: In the global, summary statistics-based classification more than a third of the measures received a predictive accuracy (RI score) < 0.55, which is not significantly different from chance ($p_{\text{binomial test}} > 0.05$). In response to our first research question, these findings indicate that for all measures investigated in this study L2 learners' performance deviates from that of the expert-target (classification accuracy ≥ 0.57, $p_{\text{binomial test}} = 0.025$). The top three measures are all measures of lexical sophistication. In response to our second research question, we found that using complexity contours information in the classification task provides a more accurate picture of differences between learner and expert texts. While these results look promising, further work is needed to include a larger set of complexity measures proposed in the relevant literature and to investigate how the contour-based approach can contribute to the identification of the most reliable and valid complexity measures that serve as proxies of L2 performance and proficiency.

Most importantly, however, the contour-based approach opens up new interesting research questions. One possible research question concerns the identification of "gold standards" for the within-text distribution of complexity for different text type (register/genre) and to what extent compliance to such standards is related to perceived text quality. A related question is whether human ratings of text quality are affected by the "global" complexity of a text captured in terms of summary statistics, or by the "local" complexity of specific passages, captured in terms of complexity contours: For example, do human raters judge a text quality primarily based on an early partition (anchoring effects), do they judge it based on properties of a late partition (recency effects)? Another question is whether there is evidence for "local compensatory effects", i.e. whether a high level of complexity at one level of linguistic analysis (e.g. syntax) is compensated for by a low level of complexity at another level (e.g. lexicon).

## 4 Conclusion

We introduced *CoCoGen* (*Complexity Contour Generator*), a tool designed to automatically track the progression of linguistic complexity within a text. *CoCoGen* uses a sliding-window technique to generate a series of measurements (complexity contours) for a given complexity dimension, providing a novel approach to the automatic assessment of text complexity. For the purposes of the present study, we decided to showcase this approach in the area of L2 learning. In future work we intend to apply this approach to other research areas, in particular, readability research and discourse processing.

## Reference

Biber, Douglas and Bethany Gray. 2010. Challenging stereotypes about academic writing: Complexity, elaboration,explicitness. *Journal of English for Academic Purposes,* 9(1):2–20.

Biber, Douglas, Bethany Gray, and Kornwepa Poonpon. 2011. Should we use characteristics of conversation to measure grammatical complexity in L2 writing development? *TESOL Quarterly,* 45(1):5–35.

Biber, Douglas and Kornwepa Poonpon, 2013. Pay attention to the phrasal structures: Going beyond t-units-a response to WeiWei yang. *TESOL Quarterly,* 47(1):192–201.

Bolton, Kingsley, Gerald Nelson, and Joseph Hung. 2002. A corpus-based study of connectors in student writing: Research from the International Corpus of English in Hong Kong (ICE-HK). *International Journal of Corpus Linguistics,* 7(2): 165-182.

Bulté, Bram, and Alex Housen. 2014. Conceptualizing and measuring short-term changes in L2 writing complexity. *Journal of Second Language Writing,* 26: 42–65.

Callies, Marcus. 2009. *Information Highlighting in Advanced Learner English*. John Benjamins Publishing.

Callies, Marcus and Ekatarina Zaytseva. 2013. The Corpus of Academic Learner English (CALE): A new resource for the assessment of writing proficiency in the academic register. *Dutch Journal of Applied Linguistics,* 2(1): 126-132.

Connor-Linton, Jeff and Charlene Polio. 2014. Comparing perspectives on L2 writing: Multiple analyses of a common corpus. *Journal of Second Language Writing*, 26:1-9.

Crossley, Scott A., Zhiqiang Cai, and Danielle S. McNamara. 2012. Syntagmatic, Paradigmatic, and Automatic N-gram Approaches to Assessing Essay Quality. In G. M. Youngblood and P. M. McCarthy (Eds.), *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*, Palo Alto, California, pp. 214–219. The AAAI Press.

Crossley, Scott. A. and Danielle S. McNamara. 2009. Computational Assessment of Lexical Differences in L1and L2 writing. *Journal of Second Language Writing*, 18(2):119–135.

Crossley, Scott. A. and Danielle S. McNamara. 2014. Does writing development equal writing quality? A computational investigation of syntactic complexity in L2 learners. *Journal of Second Language Writing,* 26:66–79.

Ehret, Katharina and Benedikt Szmrecsanyi. 2011. *An information-theoretic approach to assess linguistic complexity. Complexity and isolation*. Berlin: de Gruyter.

Ellis, Nick. C., R.I.T.A Simpson-Vlach, and Carson Maynard. 2008. Formulaic language in native and second language speakers: Psycholinguistics, corpus linguistics, and TESOL. *TESOL Quarterly*, 42(3): 375–396.

Granger, Sylviane, Estelle Dagneaux, Fanny Meunier, and Magali Paquot. 2009. *The international corpus of learner English*. Presses universitaires de Louvain.

Granger, Sylviane, Gaëtanelle Gilquin, and Fanny Meunier (Eds.). 2015. *The Cambridge Handbook of Learner Corpus Research*. Cambridge University Press.

Hyland, Ken and Polly Tse. 2007. Is there an "academic vocabulary"? *TESOL Quarterly,* 41(2):235–253.

Kerz, Elma and Daniel Wiechmann. 2015. Second language construction learning: investigating domain specific adaptation in advanced L2 production. *Language and Cognition*, 1–33.

Kyle, Kristopher. 2016. *Measuring syntactic development in L2 writing: Fine grained indices of syntactic complexity and usage-based indices of syntactic sophistication* (Doctoral Dissertation).

Kyle, Kristopher and Scott A. Crossley. 2014. Automatically assessing lexical sophistication: Indices, tools, findings, and application. *TESOL Quarterly,* 49(4):757–786.

Larsen-Freeman, Diane. 2006. The emergence of complexity, fluency, and accuracy in the oral and written production of five Chinese learners of English. *Applied Linguistics,* 27(4):590–619.

Larsen-Freeman, Diane. 2009. Adjusting expectations: The study of complexity, accuracy, and fluency in second language acquisition. *Applied Linguistics*. 30(4):579–589.

Lu, Xiaofei. 2010. Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15(4):474–496.

Lu, Xiaofei. 2012. The relationship of lexical richness to the quality of ESL learners' oral narratives. *The Modern Language Journal,* 96(2):190–208.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schutze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

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

Ortega, Lourdes. 2003. Syntactic complexity measures and their relationship to L2 proficiency: A research synthesis of college-level L2 writing. *Applied Linguistics,* 24(4):492–518.

Römer, Ute. 2009. English in academia: Does nativeness matter? *International Journal of English Studies,* 20(2):89–100.

Ströbel, Marcus. 2014. *Tracking complexity of l2 academic texts: A sliding-window approach*. Master's thesis. RWTH Aachen University.

Team, R Core. 2013. *R: A Language and Environment for Statistical Computing*.R Foundation for Statistical Computing. Vienna, Austria.

Wolfe-Quintero, Kate, Shunji Inagaki, and Hae-Young Kim. 1998. Second Language Development in Writing: Measures of Fluency, Accuracy, & Complexity. No. 17. University of Hawaii Press, 1998.