Predicting CEFRL levels in learner English on the basis of metrics and full texts

Taylor Arnold¹, Nicolas Ballier², Thomas Gaillat³, and Paula Lissón⁴

¹University of Richmond, USA
²Université Paris Diderot, France
³The Insight Centre for Data Analytics NUI Galway, Ireland
⁴Université Paris Diderot, France

Abstract

This paper analyses the contribution of language metrics and, potentially, of linguistic structures, to classify French learners of English according to levels of the Common European Framework of Reference for Languages (CEFRL). The purpose is to build a model for the prediction of learner levels as a function of language complexity features. We used the EFCAMDAT corpus [GAK13], a database of one million written assignments by learners. After applying language complexity metrics on the texts, we built a representation matching the language metrics of the texts to their assigned CEFRL levels. Lexical and syntactic metrics were computed with LCA and LSA [Lu14] and koRpus [mm17]. Several supervised learning models were built by using Gradient Boosted Trees and Keras Neural Network methods and by contrasting pairs of CEFRL levels. Results show that it is possible to implement pairwise distinctions, especially for levels ranging from A1 to B1 (A1→A2: 0.916 AUC and A2→B1: 0.904 AUC). Model explanation reveals significant linguistic features for the predictiveness in the corpus. Word tokens and word types appear to play a significant role in determining levels. This shows that levels are highly dependent on specific semantic profiles.

Keywords: Learner corpora, criterial features, lexical complexity, syntactic complexity, automatic language scoring, NLP, supervised learning.

1 Introduction

This paper focuses on the detection of language levels in Second Language Acquisition. Foreign language education centers provide courses that are tailored according to the different levels of their learners and this leads to two requirements. In the process of learning, it is paramount to provide regular evaluations to both learners and teachers so as to help them focus on specific areas to train upon. There is also a growing demand to group learners homogeneously in order to set adequate teaching objectives and methods. These two requirements rely on language assessment tests whose design and organization are labor intensive and thus costly. Currently, language centers rely on instructors to design and manually correct tests. Alternatively, they use specifically designed short-context and rule-based on-line exercises in which a discrete number of specific language errors is used as a paradigm for level assignment. This creates a bias to use errors as the sole criterion for assessment. Recent research has shown that metrics used in the domain of text mining and NLP can help characterize complexity and thus levels. Consequently, there is a need to use error independent tools to compute levels.

The objective of our experiment is to show that a supervised learning approach is possible. By creating a vector representation of texts matched with language complexity metrics, we build a predictive model for language levels. We use the EFCAMDAT⁴ corpus [GAK13] in which learner texts are classified according to the six levels of the Common European Framework.

¹The training and test data were selected and manipulated independently of direct involvement from the EF and Cambridge research teams. This corpus is publicly available at https://corpus.mml.cam.ac.uk/efcamdat2/public_html/

*taylor.arnold@acm.org
2 Related Work

There is a large body of research in automatic language scoring starting with [Pag68]. Over the last four decades, there has been various methods to provide language level analysis. Methods evolved from rule-based approaches focused on pattern matching to Machine Learning (ML) approaches including unsupervised and supervised learning methods.

Rule-based systems have included analyses relying on error detection [LC03, Mit02] while other systems identify features assumed to be to be indicative of proficiency [HXZW11]. With the advent of ML techniques, probabilistic models have appeared. Some approaches use unsupervised learning [Ton13] but, since the task of score assignment relies on human annotated corpora, most ML strategies rely on supervised learning with SVM [YB12] or linear regression and decision tree [CZ11]. Our proposal falls into this category but it uses a Neural Network including several LSTM and dense layers. The input layer includes a multi-dimensional feature representation of written essays and the output corresponds to language levels.

This area of research is closely linked to the research on criterial features that define levels. All the aforementioned studies include a large number of features based on morpho-syntactic patterns, word counts, text and readability metrics [Vaj18]. Authors tested their significance in terms of correlation or classification performance. Another perspective is to specifically focus on weighing feature significance, e.g. by applying strategies based on entropy [FH15], errors [Ton13] or lexical metrics [BG16]. In our experiment, we use lexical and syntactic complexity features [Lu10, Lu12]. Our approach supports two perspectives, i.e. classification according to features and modeling for feature significance analysis with a gradient boost tree model.

Few studies make use of the CERFL as a standard for level description. In English, many studies use other language level scales such as [FH15] with the NICT-JLE corpus levels, [CSMJ11] with TOEFL levels and [YBM11] for the Cambridge Learner Corpus (CLC). We built our data set with texts from the EF-Cambridge Open Language Database (EFCamDat) as a Gold Standard and conducted experiments on English texts classified according to the CERFL.

3 Method

3.1 Data extraction

All the available raw texts from French learners were downloaded from the EFCAMDAT database in separate XML files. Each XML file corresponded to the EFCAMDAT levels associated with each one of the CERFL levels. In total, 41,626 texts (approx. 3,298,343 tokens), corresponding to 128 units and 7,695 French learners were downloaded.

3.2 Lexical Diversity Metrics

Most of the lexical diversity metrics are based on the relationship between the numbers of types and tokens within a given text. That is, for example, the case of the TTR: (types/tokens), and some of its mathematical transformations, such as the MSTTR (types/tokens, with fragments of n tokens), Herdans C (logTypes/logTokens), Guirauds RTTR (types/√tokens), Carrols CTTR (types/ 2√tokens), Dugasts Uber Index (logTokens/logTypes - log tokens), Summers Index (S) (log(logTypes) / log(logTokens)), Maas a: a2 = (logTypes/logTokens)/logTokens, Maas log (logTypes/logTypes/logTokens)/√1 - logTypes/logTokens, and Yules K (10^4(Σ(fX * X^2) - tokens)/tokens^2, where X is a vector with the frequencies of each type, and fX is the frequency for each X).

However, most of these metrics are said to be unreliable because they are highly dependent on text length [TB98]. As a consequence, a second generation of metrics, with more complex mathematical transformations, has been developed: MTLD: (types/factors, where factors are segments that have reached the stabilization point of TTR; MATTR (mean of moving TTR, generated through a 'window' technique of variable sizes that computes TTR of samples of the text); MTLD-MA, which combines both factors and the window technique, or the HDD-D metric, which computes, for each type, the probability of finding any of its tokens in a random sample of 42 words taken from the text.
3.3 Complexity Metrics

Complexity metrics were generated using Xiaofei Lu’s software [Lu12], the L2 Syntactic Complexity Analyzer (L2SCA). There are nine syntactic measures: the number of words (W), the number of sentences (S), of verbal phrases (VP), of clauses (C), of T-units (T), dependent clauses (DC), complex T-units (CT), coordinated phrases (CP) and complex noun phrases (CN). The central unit for these metrics is the T-unit, defined as one main clause plus any subordinate clause or non-clausal structure that is attached to or embedded in it (Hunt 1970:4).

Fourteen indices of syntactic complexity: the mean length of the sentence (MLS), the mean length of the T-unit (MLT), the mean length of the clause (MLC), the number of clauses per sentences (C/S), the number of verbal phrases per T-units (VP/T), the number of clauses per T-units (C/T), the number of dependent clauses per clauses (DC/C), the number of dependent clauses per T-units (DC/T).

3.4 Readability metrics

Readability metrics have traditionally been used to assess the difficulty of texts, i.e. how comprehensible or "readable" a text is for a particular audience. In that sense, for example, some of the metrics were initially designed to determine whether texts were suitable for particular school or college years, such as the Dale and Chall formula or the Bormuth’s formulae.

Most of the metrics take into account the average number of words per sentence as well as the average word length, such as the Automatic Readability Index (ARI = words per sentence + 9 · word length), or the LIX (number of words per sentence + percentage of words with more of 6 characters) and RIX (number of long words / number of sentences) formulae; whereas other metrics also include syllable count, such as the well-known Flesch-Kincaid (0.39 · average sentence length + 11.8 · average number of syllables per word - 15.59) formula, the Fog (0.4 · (average sentence length + number of words with more than two syllables), the FORCAST (20 · number of one-syllable words / 10), and the Linsear Write (number of one-syllable words + · number of sentences) indexes. A couple of metrics also take into account the ‘complexity’ of the vocabulary deployed in the text by including parameters related to the use of ‘difficult’ or ‘hard’ words as defined by word lists previously created on the basis of native use, such as the Dale and Chall formula (0.1579 · percentage of difficult words + (0.496 · average sentence length) + 3.6365), the Spache grade (0.141 · average sentence length + (0.086 · percentage of unfamiliar words) + 0.839), or Bormuth’s four formulae.

4 Learner classification models

4.1 Prediction task set-up

The language learner ability levels in the available training data are highly skewed towards beginners, from 17,605 samples of learners at the A1 level to only 76 at the C2 level. One way to account for this in building a predictive classifier is to only compare adjacent classes. Rather than a single multinomial model, therefore, we use 5 pairwise models. While motivated by computational necessity, this approach makes sense from a linguistic perspective as well. The most interesting distinguishing marks between learners should come from adjacent classes. Also, in most practical applications the utility of a predictive model comes from distinguishing between subtle differences in proficiency.

A second challenge in establishing a reliable classification algorithm comes from differences in the tasks given as prompts in the EFCAMDAT dataset. Each of the essays is a response to a given question prompt from a chosen topic such as Attending a robotics conference or Covering a news story. Each of the 128 topics is provided as a prompt to only one specific learner level. If a modeling dataset was built by randomly assigning documents to training and testing sets, it is likely that models could be predicting the topics rather than language proficiency (particularly if specific word counts are used). For example, we may find that use of the word ‘robot’ is a good predictor of a learner achieving C2 proficiency because only those learners were asked to write about a robotics conference. In order to avoid these spurious predictors, we split the data into training and testing sets such that all of the essays from specific topics are grouped together. Similar strategies are commonly used in authorship prediction tasks to separate the effects of topical and stylistic predictors.

4.2 Readability metrics

The first set of models use only the computed readability metrics. Gradient boosted trees are used for this task because they are generally very good general-purpose algorithms and, unlike linear models, are particularly robust to highly correlated inputs. Trees are also ideal for metric inputs defined on a scale, such as readability, that is primarily designed as an ordinal measurement. Experimentation of the training set led
---|---|---|---|---|---|---|---
Metrics | GBT | train | 0.897 | 0.888 | 0.903 | 0.854 | 0.948
Metrics | GBT | test | 0.895 | 0.897 | 0.778 | 0.821 | 0.587
Term Freq. | Elastic Net | train | 0.972 | 0.977 | 0.895 | 0.998 | 0.949
Term Freq. | Elastic Net | test | 0.865 | 0.847 | 0.686 | 0.629 | 0.550
Word Seq. | LSTM | train | 0.985 | 0.985 | 0.942 | 0.777 | 0.634
Word Seq. | LSTM | test | 0.863 | 0.824 | 0.548 | 0.525 | 0.535
Metrics+ | GBT | train | 0.931 | 0.899 | 0.927 | 0.860 | 0.853
Metrics+ | GBT | test | **0.916** | **0.904** | 0.753 | 0.746 | 0.558

Table 1: AUC Metrics for binary classification models across four methods.

...to the choice of a max tree depth of 4, learning rate 0.01 and a total of 100 trees. The AUC metrics for training and testing sets are shown in Table 1. We see that the classification task generally becomes harder for distinguishing between more advanced learners. Partially this is the result of differences being more subtle and difficult to pick up from a small snippet of text. Also, the amount of training data decreases when working with the higher proficiency learners. The models for B1–>B2 and C1–>C2 seem to be overfit to the training data, but the other three models produce models with AUC values that generalize favorably to the test set.

The top five features from each gradient boosted tree are shown in Table 2 based on the sum of model gain taken from all nodes that use a given variable as a split point. The two strongest variables across the first four models are wordtokens (number of unique terms used) and wordtypes (number of word inflections used).

### 4.3 Term frequencies matrix

In the second set of models, we use the actual word frequencies to predict the language level of a learner. To do this, we tokenised the raw input text for each learner and constructed a term frequency matrix $X$ of word counts per document. Words that were used in less than 2% of the corpus were filtered out. A logistic elastic net regression was used to learn a predictive model from these word counts. The elastic net model is a generalized linear regression model with an extra penalty term on the negative log-likelihood. The elastic net works well with datasets with a large number of columns, such as a term frequency matrix, because the $\ell_1$-penalty causes many of the values of $\beta$ to be equal to zero (known as a parsimonious model). The degree of penalization was set using cross-validation.

In Table 1 we see that the word-based model improves the training-set AUC scores significantly but leads to less predictive models on the test set. This is largely due to overfitting caused by learning the topics discussed by the learner categories rather than stylistic features that would generalize across tasks. However, the words along do perform relatively well for distinguishing the A1–>A2 and A2–>B2 tasks. The dominant features in these models are the use of prepositions and verb forms that may correlate with an increase in proficiency.

### 4.4 Incorporating word order

One distinguishing feature of advanced learners is their ability to construct complex sentences and correctly use advanced features such as zero-relative clauses. It would therefore seem that a model that takes into account word-order would perform better for learning prediction compared to methods based on term-frequency matrices. In order to incorporate word-order into a model, we applied an LSTM model — a particular type of recurrent neural network popular in text analysis — to the four classification tasks. Given the relatively small data size, and following experimentation on the training set, we selected an LSTM model that uses an embedding layer with 128-dimensions, 32 recurrent units, and a high degree of dropout (80%). The network was trained using the stochastic gradient descent, early stopping, and the learning rate schedule based on the ADAM algorithm. Results in Table 1 show that the LSTM model performs similarly to the elastic net for the first two tasks but not quite as well for classifying the advanced learners. So, while in theory word-order should improve the model, the inconsistent topics and small amount of data stops this from appearing empirically in the classification model.

### 4.5 Custom metrics

Our final set of models constructs three new sets of metrics to add the default readability metrics and re-applies the whole set of metrics using the same gradi-
Table 2: Most relevant metrics for pairwise level distinctions

<table>
<thead>
<tr>
<th></th>
<th>A1→A2</th>
<th>A2→B1</th>
<th>B1→B2</th>
<th>B2→C1</th>
<th>C1→C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>wordtokens</td>
<td>W</td>
<td>wordtypes</td>
<td>wordtokens</td>
<td>wordtokens</td>
<td>ndwerz</td>
</tr>
<tr>
<td>W</td>
<td>ssv1</td>
<td>W</td>
<td>DC,C</td>
<td>vs1</td>
<td>DC.C</td>
</tr>
<tr>
<td>wordtypes</td>
<td>MLC</td>
<td>vs2</td>
<td>swordtypes</td>
<td>swordtypes</td>
<td>slextypes</td>
</tr>
<tr>
<td>MLS</td>
<td>CN</td>
<td>lextokens</td>
<td>ttr</td>
<td>W</td>
<td>MLC</td>
</tr>
<tr>
<td>DC.T</td>
<td>CN,T</td>
<td></td>
<td></td>
<td>CN,T</td>
<td></td>
</tr>
</tbody>
</table>

5 Discussion and Conclusions

Taking into account the topic of the essay to be written, our experiment confirms the observations that task-based corpora entail strong overfitting [AMMM17]. Another issue that cropped up with this type of data is that we have not quite overcome the initial skewed distribution of the data. For any expert system that we would like to build to automatically classify learner texts into learner levels, we have to face the fact that in most learner corpora, there will be more beginner and intermediate data than advanced (not to say "expert") data. Our experiment also suggests that more metrics, and not only readability formulae, may help improve classification rate, which somehow confirms the results presented in ([VM12]), where a classification accuracy of 93.3% was achieved with 46 metrics. Following on this lead it is tempting to resort to even more complex or detailed systems producing metrics, such as the whole range of metrics produced by the Common Text Analysis Platform [CM16].

5.1 Analyzing POS-tags

We have shown that some POS-tag patterns could improve the classification rate. We have only used the Universal Part Of Speech (UPOS) tagset, which simplifies the analysis. More subtle tagsets (such as the Penn Treebank tagset) would possibly yield more fine-grained results at the expense of precision (these tagsets include more tags and are therefore linguistically more relevant but more error-prone. We have not included punctuation in the analysis, but it is likely to play a role in the way more advanced learners conceive information packaging and structure sentence initial segments. As an aside, we ran frequency inventories of UPOS-grams for each level, to investigate whether we could see any specific patterns. Initial levels exhibited the sequence noun / punctuation / pronoun / verb. The typical interpretation of this sequence is a sentence ending with a noun (end weight principle) and sentences beginning with a pronoun as a subject. In contrast, more advanced levels also favored more elaborated sequences with adverbs and postpositions. It should be noted that tagsets do not distinguish between commas and full stops (both labeled "punct") so that the analysis mostly holds for full stops, as learners do not always abide by expected punctuation guidelines (commas, and semi-colons are underused by learners). One strategy could consist in re-annotating the data with a specific tag for 'comma' and for 'full stop'. This would allow us to retrieve more specific information structure strategies across levels of proficiency. For instance, we would then be able to investigate which grammatical categories are used to express focus after the 'full stop'. As to the potential investigation of the use of 'comma', clustering the 'comma' with its corresponding dependency structure may help classifying C1/C2 users by means of the required 'comma' for appositive clauses, for instance. Because learner punctuation might be erratic (especially for online-based data collection), 4-grams of POS-tags
involving punctuation have proved to be more robust than 3-grams.

5.2 L1-based features

The experiment has used data from French learners, but the metrics used are supposed to be language-independent. We could try to refine the classification by using features based on potential errors made by a given population of learners, such as their native language (L1). In this respect, further research could resort to specific problems for French learners, such as the expression of definiteness. For example, we would expect French speakers to overuse the and a little whereas much, few, and fewer would be underused.

5.3 Lexically-based features

It should be pointed out that some metrics rely on tokenized data, whereas others are computed on the basis of the raw texts. This is of paramount importance when it comes to learners because their (sometimes alternative and variable) spelling may artificially inflate the number of tokens. An expert system for learner levels should take this into account, especially for beginner levels. Discourse-based metrics could be used, such as the number of repetitions, more frequent in the A1 group. Last, we have mostly considered frequency as a potential cue for the use of the lexicon by learners, but more subtle techniques could be used, such as word embeddings.

References


