Features	Model	Partition	A1=>A2	A2=>B1	B1=>B2	B2=>C1	C1=>C2
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 ℓ_1-penalty causes many of the values of $\widehat{\beta}$ 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 reapplies the whole set of metrics using the same gradi-

A1=>A2	**A2=>B1**	**B1=>B2**	**B2=>C1**	**C1=>C2**
wordtokens	wordtypes	wordtokens	wordtokens	ndwerz
W	W	W	adjv	DC.C
svv1	ls2	DC.C	vs1	slextypes
wordtypes	MLC	vs2	swordtypes	MLC
MLS	CN	lextokens	W	lv
DC.T	CN.T	ttr	CN.T	modv

Table 2: Most relevant metrics for pairwise level distinctions

ent boosted trees models shown in Section 4.2. The new metrics incorporate features found when performing automatic part-of-speech and dependency extraction on the corpus, which were extracted using the R package **cleanNLP**. Specifically, we recorded the number of times that (1) each universal part of speech code was used in a text, (2) each universal dependency was used in a text, and (3) words were used from each Zipf-scale categories (a map of each English word into a 7 category set based on its frequency in usage). We choose these metrics based on perceived features missing in the current readability metrics and as an attempt to dig deeper in the the three most influential base metrics: *worktoken* and *wordtypes*. The changes in AUC scores are shown in Table 1. We see that the new features do improve the first models, producing the best AUC across all feature sets, but due to the small sample sizes introduces a modest degree of overfitting. It should be noted that a cascading architecture only yields a 70.34 % acccuracy.

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

[AMMM17] Theodora Alexopoulou, Marije Michel, Akira Murakami, and Detmar Meurers. Task effects on linguistic complexity and accuracy: A large-scale learner corpus analysis employing natural language processing techniques. *Language Learning*, 67(S1):180–208, 2017.

[BG16] Nicolas Ballier and Thomas Gaillat. Classifying French learners of English with written-based lexical and complexity metrics. In *JEP-TALN-RECITAL 2016*, volume 9 of *ELTAL*, pages 1–14, Paris, France, 2016.

[CM16] Xiaobin Chen and Detmar Meurers. Ctap: A web-based tool supporting automatic complexity analysis. In *CL4LC*, pages 113–119, 2016.

[CSMJ11] Scott A. Crossley, Tom Salsbury, Danielle S. McNamara, and Scott Jarvis. Predicting lexical proficiency in language learner texts using computational indices. *Language Testing*, 28(4):561–580, 2011.

[CZ11] Miao Chen and Klaus Zechner. Computing and Evaluating Syntactic Complexity Features for Automated Scoring of Spontaneous Non-native Speech. Association for Computational Linguistics, 2011.

[FH15] Brendan Flanagan and Sachio Hirokawa. The Relationship of English Foreign Language Learner Proficiency and an Entropy Based Measure. *Information Engineering Express*, 1(3):29–38, 2015.

[GAK13] Jeroen Geertzen, Theodora Alexopoulou, and Anna Korhonen. Automatic linguistic annotation of large scale l2 databases: The ef-cambridge open language database (efcamdat). 2013.

[HXZW11] Derrick Higgins, Xiaoming Xi, Klaus Zechner, and David Williamson. A Three-stage Approach to the Automated Scoring of Spontaneous Spoken Responses. *Comput. Speech Lang.*, 25(2):282–306, 2011.

[LC03] Claudia Leacock and Martin Chodorow. C-rater: Automated Scoring of Short-Answer Questions. *Computers and the Humanities*, 37(4):389–405, 2003.

[Lu10] Xiaofei Lu. Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15(4):474–496, 2010.

[Lu12] Xiaofei Lu. The Relationship of Lexical Richness to the Quality of ESL Learners' Oral Narratives. *The Modern Language Journal*, 96(2):190–208, 2012.

[Lu14] Xiaofei Lu. *Computational Methods for Corpus Annotation and Analysis.* Springer, Dordrecht, 2014.

[Mit02] T. Mitchell. Towards robust computerised marking of free-text responses. 2002.

[mm17] m.eik michalke. *koRpus: An R Package for Text Analysis*, 2017. (Version 0.10-2).