

Level-Up: Learning to Improve Proficiency Level of Essays

Wen-Bin Han, Jhih-Jie Chen, Ching-Yu Yang, Jason S. Chang

Department of Computer Science

National Tsing Hua University

{vincent.han, jjc, chingyu, jason}@nlpplab.cc

Abstract

We introduce a method for generating suggestions on a given sentence for improving the proficiency level. In our approach, the sentence is transformed into a sequence of grammatical elements aimed at providing suggestions of more advanced grammar elements based on originals. The method involves parsing the sentence, identifying grammatical elements, and ranking related elements to recommend a higher level of grammatical element. We present a prototype coaching system, *Level-Up*, that applies the method to English learners' essays in order to assist them in writing and reading. Evaluation on a set of essays shows that our method does assist user in writing.

1 Introduction

Many essays (e.g., “*Amazingly, the child is so fashionable and creative that he makes the ugly house modern.*”) are submitted to tutoring services by English learners on the Web every day, and an increasing number of services on the Web specifically target learners' essays. For example, *LanguageToolPlus* (languagetoolplus.com) uses rule-based model with n-grams extracted from the numerous data to inspect essays for grammatical errors, while *Grammarly* (grammarly.com), *Ichecker* (lchecker.com) and *Ginger* (gingersoftware.com) use proprietary neural network approaches to proofread texts, check grammar, review style, and enrich vocabulary.

Tutoring services such as *Write&Improve* (writeandimprove.com) and *WhiteSmoke* (whitesmoke.com) typically correct and grade essays as a whole. However, very few systems provide focused suggestions on how to raise the level of proficiency. Learners could raise their level of grammatical proficiency, if a system can

identify grammatical elements and suggest the learner to use related elements with a higher level.

Consider the essay “*Amazingly, the child is so fashionable and creative that he makes the ugly house modern.*”. The useful suggestion for this sentence is definitely not just a level for the whole essay, which is pointless for learners, but the levels of grammatical elements with explanation and level-up grammatical elements. A helpful suggestion for an essay should not only contain the pairs of level and explanation such as: “*B1 - make the ugly house modern - Can use adjectives as object complement after ‘make’.*” but also suggest improvement: “*B2 - Can use a limited range of degree adjectives (‘real’, ‘absolute’, ‘complete’) before a noun to express intensity.*”. These grammatical elements can be retrieved from English Grammar Profile (EGP) with more than a thousand grammatical elements with levels stipulated by Common European Framework of Reference (CEFR). Intuitively, by categorizing grammatical elements, we can provide more informative instruction for learners to improve their essays.

We present a new system, *Level-Up*, that parses essays into trees expected to recommend related advanced grammatical elements. An example *Level-Up* recommending for the essay “*Amazingly, the child is so fashionable and creative that he made the ugly house modern.*” is shown in Figure 1. *Level-Up* has identified several grammatical elements (e.g., {*make NOUN ADJ*}) for the given essay. *Level-Up* detects these grammatical elements by matching patterns against the parse trees. We describe the *Level-Up* model in more detail in Section 3.

At run-time, *Level-Up* starts with a given essay submitted by the learner (e.g., “*Amazingly, the child is so fashionable and creative that he made the ugly house modern.*”), which is first converted into a set of grammatical elements. Then,

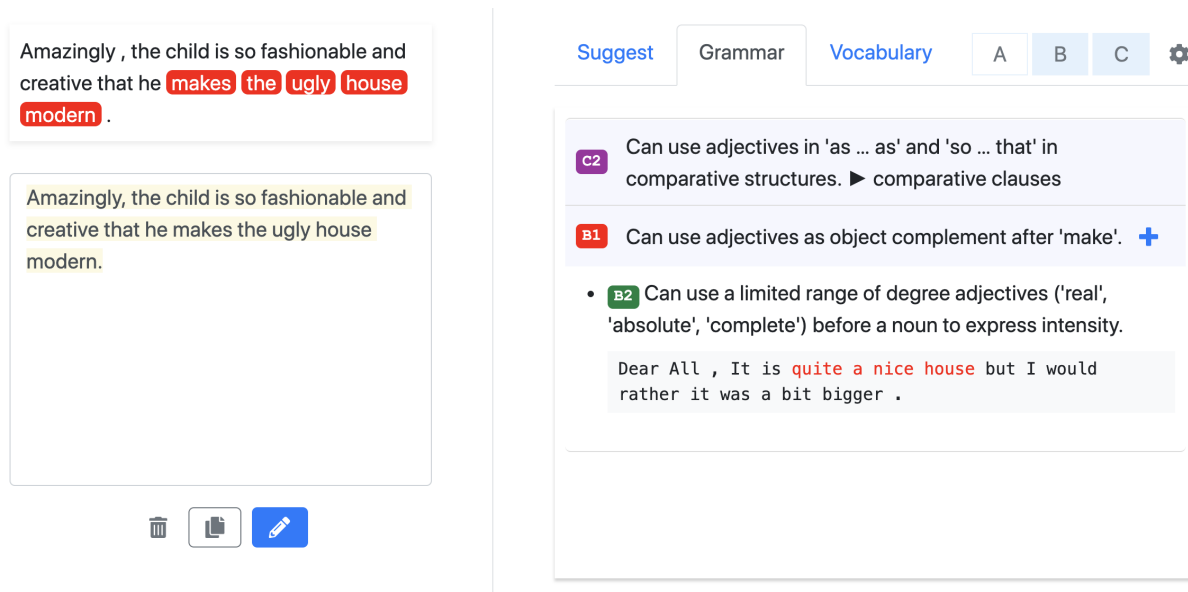


Figure 1: The screenshot of *Level-Up*

Level-Up ranks categorized elements and retrieves the related elements with a higher level as suggestions. In our prototype, *Level-Up* returns detected grammatical elements and recommendations of higher level elements to English learners directly (see Figure 1); alternatively, the elements and levels returned by *Level-Up* can be used as input to an essay scoring system.

The rest of the paper is organized as follows. We review the related work in the next section. Then we present our method for detecting grammatical elements in learners' essays expected to suggest more advanced elements. In our evaluation, *Level-Up* can provide useful collocations with levels for learners during writing.

2 Related work

English Language Teaching (ELT) has been an area of active research in Applied Linguistics and Computational Linguistics. Recently, the state-of-the-art research in ELT has been represented in the 13th Workshop on Innovative Use of NLP for Building Educational Applications (Tetreault et al., 2018) in the Association for Computational Linguistics (ACL) community. The workshop involves developing applications based on NLP approaches for teachers and learners of English as a Second Language (ESL) in educational settings. For example, Bryant and Briscoe (2018) build a competitive system only requiring minimal annotated data by using a simple Language Model ap-

proach. In our work, we address an aspect of English Language Teaching which is not the focus of correcting errors. Instead, we concentrate on how to analyze grammatical elements and suggest more advanced elements for learners to level up their essays.

More specifically, we focus on grammatical analysis for assisting learners in writing English, namely, suggesting grammatical elements at higher proficiency level based on identified grammatical elements in learner's writing. Grammaticality improvement for learners has been the focus of ELT research with much works concentrating on Grammar Error Correction (GEC). In general, GEC systems are aimed at correcting errors in learners' essays without considering the levels of grammatical elements used in the essays. In contrast, we will analyze the levels of grammatical elements in a given essay and provide more grammatically and lexically advanced elements to inform learners of how to refine and level up their essays.

The most commonly used criteria for measuring the proficiency levels is the Common European Framework of Reference for Languages (CEFR) (Council of Europe, 2001) with six proficiency levels: the basic level (A1 and A2), independent level (B1 and B2) and proficient level (C1 and C2). As an aid to defining levels for learning, teaching and assessment, CEFR describes what language learners can do ("can-do" statement) at different learning stages. (e.g., level A1 - Can use

Regex	Level	Statement
<i>JJR and JJR</i>	B1	Can use 'and' to join a limited range of comparative adjectives.
<i>too JJ TO VB</i>	B1	Can use 'too' before adjectives followed by 'to'-infinitive.
<i>very JJ</i>	A1	Can use 'very' to modify common gradable adjectives.

Table 1: Example regular expressions

commas and “and” to join more than two adjectives, after “be”.) Moreover, Cambridge University Press organizes a wealth of information related to CEFR, including English Grammar Profile (EGP) and English Vocabulary Profile (EVP). EGP grades learners’ ability in terms of grammatical form and CEFR levels, while EVP defines words and phrases of different CEFR levels.

Previous works targeting CEFR level detection of learners’ essays include Hancke and Meurers (2013) for German and Vajjala (2014) for Estonian based on annotated learner data. To cope with high cost of collecting learners’ data, Pilán et al. (2016) investigated the benefits of using texts from language learning coursebooks to train classifiers for predicting proficiency levels of learners’ texts. Vajjala (2017) and Tack et al. (2017) present methods for identifying the linguistic variables that are indicative of writing quality to evaluate a learner’s proficiency. Their researches and Bartning et al. (2019) all use CEFR to assess proficiency levels. We also utilize CEFR criteria in our research to evaluate essays, but focusing more on grammatical elements laid out in Cambridge EGP.

In a study more closely related to our work, *Write&Improve*¹ (Andersen et al., 2013; Yanakoudakis et al., 2018) supports self-assessment and learning by correcting common errors and returning an overall score for an essay. Furthermore, it also indicates potentially worst sentences. In contrast, we focus on providing specific information on raising the proficiency level as the learner writes.

Researches have pointed out that supplying suggestions while writing is more helpful than suggesting after the fact (Hearst, 2015). *Grammarly* tries to correct grammatical errors and provides the explanation while the user is writing. *WriteAhead*² (Yen et al., 2015) provides real-time writing suggestions on what to write next in the form of grammar patterns and example sentences. Similarly, *ColloCaid*³ (Lew et al., 2018) checks if the

¹www.writeandimprove.com

²www.writeahead.nlpweb.org

³www.collocaid.uk

collocation is used correctly and provides frequent collocates so that writers can choose words that go well together.

In contrast to the previous research in English Language Teaching and Grammatical Error Correction, we present a tutoring system, *Level-Up*, that provides writing assistance, focusing on analyzing grammatical elements and suggesting higher level elements during writing.

3 The *Level-Up* System

To improve learners’ essays, grammatical error correction (GEC) is not sufficient. Unfortunately, very few Language tools go beyond GEC and provide suggestions on proficiency level improvement for learners. In this section, we address such a problem. *Level-Up* displays a set of suggestions based on leveled grammatical elements for improving an unfinished sentence or complete sentences in essays. We transform criteria (e.g., EGP) into a pattern-matching program to identify grammatical elements and example n-grams from the corpus. We describe the process of our solution to this problem in the subsections that follow.

3.1 Extracting Grammatical Elements

Due to the lack of annotated data on grammatical elements, we attempt to extract grammatical elements using rules representing grammatical elements. The method involves using regular expressions, a lexical dictionary, and a parser, since regular expression is straightforward for matching patterns. Table 1 shows the examples of regular expression corresponding to EGP elements.

After converting these elements into regular expressions, we first parse the given corpus, and then retrieve grammatical elements and example n-grams for recommendation later at run-time. Since regular expressions have some limitations on flexibility, we solve this problem by using a dependency parser. Therefore, we take advantage of dependency tree to generate all phrasal elements from the parse tree layer by layer, and then match all the rules with these element candidates. We

- (1) Parse the sentence into POS tags and keywords.
S: Actually, the child is very nice and friendly.
“ADV, DET NOUN be ADV ADJ and ADJ.”
- (2) Generate all element candidates layer by layer.
S: Actually, the child is very nice and friendly.
“is”, “Actually, child is nice”, “Actually, the child is very nice and friendly.”
“be”, “ADV, NOUN be ADJ”, “ADV, DET NOUN be ADV ADJ and ADJ.”
- (3) Match candidates against all regular expressions.
 - (a) “ADV”
 - (b) “be ADJ”
 - (c) “ADJ and ADJ”
- (4) Extract these matches with the corresponding n-grams.
 - (a) Actually, “ADV”
 - (b) is nice, “be ADJ”
 - (c) nice and friendly, “ADJ and ADJ”

Figure 2: Outline of the process used to identify elements in an example sentence

- (1) Obtain n-grams belonging to the given element.
- (2) Remove the n-grams not containing the last word in the unfinished sentence.
- (3) Calculate Language Model of the unfinished sentence with each n-gram.
- (4) Retain top 10 highest n-grams of Language Model.
- (5) Calculate the n-grams of the average word level.
- (6) Select the highest n-gram and its corresponding grammatical elements.

Figure 3: Outline of the process used to select top 1 n-gram to exemplify a given suggested grammatical element.

then record every matching phrase and sentence for generating suggestions at run-time. Figure 2 shows the process of identifying grammatical elements in an example sentence.

3.2 Automated Writing Suggestion for Leveling up

Once the grammatical elements and n-grams are automatically extracted and counted from the given corpus, they are stored as suggestion candidates. *Level-Up* constantly returns suggestions based on the last word the user types in the writing area. With the last word as a query, *Level-Up* retrieves and displays n-grams, ranked by Language Model and the level of words. Furthermore, each n-gram exemplifies different grammatical elements and is accompanied with three example

- (1) Obtain grammatical elements in the same subcategory.
- (2) Retain elements with higher level than the identified element.
- (3) Select the first one element for recommendation

Figure 4: Outline of the process of recommending level-up elements for identified grammatical elements.

sentences. The process of selecting n-grams is shown in Figure 3.

3.3 Analyzing Elements and Ranking Suggestions

Level-Up also analyzes essays after users finish writing. The process of analyzing is the same as described in Subsection 3.1. However, we do not display all the matches to the user. Instead, if the grammatical element is completely overlapped by the other element, we only retain the one with higher level.

Our system not only identifies grammatical elements but also suggests level-up elements. EGP contains the broad categories of the grammatical elements, including adjectives and adverbs. Furthermore, every category includes several subcategories (e.g., adjectives - comparatives and superlatives). We group those elements by the categories and then select the most related level-up element in the same group. The process of selecting level-up elements is shown in Figure 4.

4 Experiments and Results

In this section, we describe the details of our experiments and the results. First, we introduce how we preprocess the corpus and extracted grammatical elements. Then, we explain the analysis of vocabularies in *Level-Up*. Finally, we describe the program architecture and the toolkits used, and show the evaluation results of our system.

4.1 English Grammar Profile

CEFR lists total 1,222 grammatical elements in EGP. In our prototype, we experimented with two categories, adjectives and adverbs, as a pilot study to prove that our approach is effective. Some specific types of words, such as degree adverbs, are enumerated from Sinclair (2005). For preprocessing, we used British National Corpus (BNC) (Corpus, 2001), containing over four million sentences, to collect example n-grams for grammatical elements. To be more specific, we parsed all the sentences and generated grammatical element candidates by using SpaCy parser (Honnibal and Montani, 2017). Then, we matched all the candidates against all the rules. Every detected match of grammar pattern is stored with its n-gram and sentence.

4.2 English Vocabulary Profile

In addition to EGP, we also utilize EVP in our system. *Level-Up* not only analyzes the levels of vocabularies defined by EVP in learner’s essays but also provides similar vocabularies at higher level. For example, *Level-Up* can suggest “strive” (C2 level) for the verb “try” (A2 level). However, analyzing vocabulary in our study is not described in Section 3 due to our focus of EGP. First of all, we obtained the full six-level vocabularies from EVP⁴, which covers levels A1-C2 of CEFR. However, disambiguating the meanings of polysemy is still an open problem. Therefore, we use the lowest level of a word directly. In other words, after tokenizing sentences with SpaCy, we match these tokens against the lookup table, EVP, to obtain the lower level.

For word suggestions, we use the pre-trained 300 dimension Word2Vec (Mikolov et al., 2013) trained on Google News to generate top 100 similar words as candidates using Gensim (Řehůřek and Sojka, 2010). Then, we filter out those candidates that are not in EVP, at a lower level, or

⁴vocabulary.englishprofile.org/staticfiles/about.html

Suggestion	Count	Percent	Precision
1st suggestions	22	0.53	0.44
2nd suggestions	13	0.32	0.26
3rd suggestions	6	0.15	0.12
Top 3 suggestions	41	-	0.82
Not in Top 3	9	-	0.18

Table 2: Human evaluation of *Level-Up*

different POS tags. Finally, we choose the top 10 candidates to show to learners.

4.3 Technical Architecture

Level-Up was implemented in Python with the Flask Web framework. We stored the suggestions in JSON format and read the content into memory for fast access. *Level-Up* server obtains client input from a popular browser (Safari, Chrome, or Firefox) dynamically with AJAX techniques.

4.4 Evaluating *Level-Up*

To evaluate the performance of *Level-Up*, we randomly sampled sentences from learner’s corpus. For simplicity, we tested if learners can acquire suitable n-grams with advanced grammatical structures from using *Level-Up* and evaluate the performance. We randomly selected 50 sentences with adjectives or adverbs from EFCAM-DAT) as test data and segmented each sentence from start to adjective or adverb word for recommended n-grams. After typing it in *Level-Up*, we considered n-grams from first one to third one, and counted the position of good suggestions. We assumed that learners can fit the n-grams to the input with less tolerance of edit. Finally, we manually determined the appropriateness of suggestions based on the precision of the Top-3 suggestions. Table 2 shows the performance of *Level-Up*.

5 Future Work and Conclusion

Many avenues exist for future research and improvement of our system, *Level-Up*. For example, other categories in EGP could be handled. The method of ranking n-grams could be improved by considering the relevancy of n-gram to learner’s sentence more precisely. NLP and Machine Learning techniques could be applied to identify and rank grammatical elements. Additionally, an interesting direction to explore is recommending well-spoken words and phrases to level up learner’s essays lexically. For example,

we could suggest “the better part of a week” to level up “almost a week”. Similarly, we could suggest “in the making” for “happening.” Yet another direction of research is evaluating an essay as a whole based on the detected grammatical elements. In other words, teachers can assess students’ essays more efficiently using *Level-Up*.

In summary, we have proposed a method for analyzing grammatical elements and suggesting level-up elements while a user is writing. The approach involves extracting, retrieving, and ranking grammatical elements and examples. We have implemented and evaluated the proposed approach as applied to a large corpus with promising results.

References

- Øistein E. Andersen, Helen Yannakoudakis, Fiona Barker, and Tim Parish. 2013. Developing and testing a self-assessment and tutoring system. In *BEA@NAACL-HLT*.
- Inge Bartning, M Martin, and Ineke Vedder. 2019. Communicative proficiency and linguistic development: intersections between sla and language testing research.
- Christopher Bryant and Ted Briscoe. 2018. Language model based grammatical error correction without annotated training data. In *BEA@NAACL-HLT*.
- British National Corpus. 2001.
- Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Common European Framework of Reference for Languages: Learning, Teaching, Assessment. Cambridge University Press.
- Julia Hancke and Detmar Meurers. 2013. Exploring cefr classification for german based on rich linguistic modeling. pages 54–56.
- Marti A. Hearst. 2015. Can natural language processing become natural language coaching? In *ACL*.
- Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.
- Robert Lew, Ana Frankenberg-Garcia, Geraint Paul Rees, Jonathan C Roberts, and Nirwan Sharma. 2018. Collocaid: A real-time tool to help academic writers with english collocations. In *The XVIII EU-RALEX International Congress*, page 165.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Distributed representations of words and phrases and their compositionality](#). In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS’13*, pages 3111–3119, USA. Curran Associates Inc.
- Ildikó Pilán, Elena Volodina, and Torsten Zesch. 2016. Predicting proficiency levels in learner writings by transferring a linguistic complexity model from expert-written coursebooks. In *COLING*.
- Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. <http://is.muni.cz/publication/884893/en>.
- J. Sinclair. 2005. *Collins Cobuild English grammar*. Collins Cobuild. HarperCollins.
- Anaïs Tack, Thomas Franois, Sophie Roekhaut, and Cédric Faron. 2017. Human and automated cefr-based grading of short answers. In *BEA@EMNLP*.
- Joel Tetreault, Jill Burstein, Ekaterina Kochmar, Claudia Leacock, and Helen Yannakoudakis. 2018. [Proceedings of the thirteenth workshop on innovative use of nlp for building educational applications](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.
- Sowmya Vajjala. 2014. Automatic cefr level prediction for estonian learner text.
- Sowmya Vajjala. 2017. Automated assessment of non-native learner essays: Investigating the role of linguistic features. *International Journal of Artificial Intelligence in Education*, 28:79–105.
- Helen Yannakoudakis, Øistein E. Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. Developing an automated writing placement system for esl learners.
- Tzu-Hsi Yen, Jian-Cheng Wu, Jim Chang, Joanne Boisson, and Jason Chang. 2015. Writeahead: Mining grammar patterns in corpora for assisted writing. *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 139–144.