Parametrizable exercise generation from authentic texts: Effectively targeting the language means on the curriculum

Tanja Heck
Universität Tübingen / Germany
tanja.heck@
uni-tuebingen.de

Detmar Meurers
Universität Tübingen / Germany
detmar.meurers@
uni-tuebingen.de

Abstract

We present a parametrizable approach to exercise generation from authentic texts that addresses the need for digital materials designed to practice the language means on the curriculum in a real-life school setting. The tool builds on a language-aware search engine that helps identify attractive texts rich in the language means to be practiced. Making use of state-of-the-art NLP, the relevant learning targets are identified and transformed into exercise items embedded in the original context.

While the language-aware search engine ensures that these contexts match the learner's interests based on the search term used, and the linguistic parametrization of the system then reranks the results to prioritize texts that richly represent the learning targets, for the exercise generation to proceed on this basis, an interactive configuration panel allows users to adjust exercise complexity through a range of parameters specifying both properties of the source sentences and of the exercises.

An evaluation of exercises generated from web documents for a representative sample of language means selected from the English curriculum of 7th grade in German secondary school showed that the combination of language-aware search and exercise generation successfully facilitates the process of generating exercises from authentic texts that support practice of the pedagogical targets.

1 Introduction

With digital learning contexts becoming increasingly common in Foreign Language Teaching and Learning, automatic exercise generation arguably will become a crucial tool for making individualized practice materials available that are adapted to the learner's individual needs and competencies (Liu et al., 2005). An ideal system for this purpose will generate exercises of parametrizable complexity for a given input text.

Form-focused exercises lend themselves especially well to automatic generation as their answer space is limited enough to support automatic evaluation (Sysoyev, 1999; Zanetti et al., 2021; Schwartz et al., 2004). Approaches in this domain can be subdivided into two categories: systems that generate simple exercise sentences using a rule-based approach, and tools that extract sentences which contain the targeted constructions from existing texts (Perez-Beltrachini et al., 2012). Working with authentic texts has been argued to have positive effects on learner motivation (Peacock, 1997), especially if as much context as possible is preserved (Romney, 2016). Since motivation is highest when the topic and contents of the text is of interest to the learner, allowing them to provide their own texts as input to exercise generation is advantageous (Zhuomin, 2010). Yet, authentic texts often do not include sufficient examples for the language means to be practiced (Chinkina et al., 2016). It is therefore important to assist learners in finding suitable documents that are of interest and richly represent the language means on the syllabus that are to be practiced, which has been referred to as input enrichment (Chinkina and Meurers, 2016).

More recently, an important need for automatically generated exercises is arising in the context of adaptive language tutoring systems (Pandarova et al., 2019). Adaptivity comprises elements both at the micro level and at the macro level (Rus et al., 2014). With respect to micro-adaptivity, scaffolding feedback is used to guide the learner towards the correct answer. Macro-adaptivity refers to the system capability to provide sequences of exercises at the right level for a given learner. Such systems thus need to either manipulate exercise difficulty in real-time (Beinborn, 2016) or to maintain large pools of exercises of varying complexity levels (Pandarova et al., 2019). Real-time manipulation is most feasible for aspects of the exercise, such as the number of distractors or hints, but not for linguistic

Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022), pages 154 - 166

features of the seed sentence or the choice of the target item. This approach to macro-adaptivity in the language learning context has mainly been limited to C-tests (Beinborn, 2016; Lee et al., 2019).

Most educational institutions use some Learning Management System (LMS, Zabolotniaia et al., 2020). To be able to integrate generated exercises into regular classes, they should be compatible with the LMS system used. Exercises would thus be most beneficial to instructors when provided as globally usable web components or in a format that complies with standards such as xAPI[1] and cmi5[2]. In addition, it would be important to provide interfaces that make it possible to edit exercises that were generated to be able to correct or modify them to suit the instructor's needs and preferences.

Summing up the requirements mentioned above and in the wider literature, an exercise generation tool should provide input enrichment mechanisms for user-selected texts, be parametrizable and editable by the educators themselves, integrate feedback into the exercises, provide the exercises in a portable format, and support use of the exercises within the original context. To our knowledge, no tool has been developed so far which complies with all these features. Indeed, no fully automated exercise generation system for grammar exercises we know of even offers some of the features mentioned, such as an integrated input enrichment approach.

In this paper, we thus present an exercise generation extension of the language-aware search engine FLAIR[3] to address this gap. We start with section 2 introducing the research context on automatic generation of grammar exercises. Section 3 describes the implementation of the exercise generation extension of FLAIR and outlines its functionality and use. Section 4 evaluates the tool before section 5 summarizes and concludes with an outlook.

2 Related work

Our approach integrates automatic exercise generation into an educational document retrieval system. Therefore, we will first elaborate on previous work on educational information retrieval systems before discussing existing tools for form-based grammar exercise generation with respect to the outlined criteria we impose on such a system.

Similar to tutoring systems, educational **document rankings systems** often leverage information from a learner model to identify texts that match a learner's individual proficiency level. Two examples for such an approach are *REAP* and *TextFinder* (Bennöhr, 2005). The learner model is maintained within the system based on the learner's interaction with the tool. While *REAP* can work with texts from anywhere on the web, *TextFinder* operates on its own database of online news articles.

A less automated approach relies more on user interaction. Examples include the standalone tool *READ-X* (Miltsakaki and Troutt, 2008) and the web extension *LAWSE* (Ott and Meurers, 2011). Both tools calculate readability scores for web documents. While LAWSE merely displays them for the analyzed documents, READ-X matches the scores against a readability level which users need to specify, and filters the documents accordingly.

For narrowly defined use cases, tools may filter documents without either a learner model or user input. *SourceFinder* constitutes an example for such a system. It applies a binary filter to its corpus of online journals and identifies texts suitable in academic contexts (Sheehan et al., 2007).

Systems with the highest degree of flexibility allow users to filter documents according to contained grammatical constructions. This approach is for example realised in the authoring assistance tool *Sakumon* (Hoshino and Nakagawa, 2008) and the language-aware search engines FLAIR[4] (Chinkina et al., 2016) and KANSAS[5] (Dittrich et al., 2019). Sakumon maintains information on the article's reading level as well as on contained grammatical constructions in its database. FLAIR and KANSAS, the latter being based on FLAIR and specializing on low literacy in German, analyze web texts on demand. In addition to the filtering functionality, these two systems also allow users to rank all retrieved documents according to the occurrence of linguistic constructions. This kind of re-ranking of search results allows to identify documents containing certain linguistic constructions, such as those targeted by grammar exercises. Such a tool therefore lends itself well as basis into which we can integrate an exercise generation component.

Table 1 provides an overview of existing work on **automatic generation of grammar exercises** highlighting that while many of these systems incorporate some of the characteristics we consider rele-

[1] http://github.com/adlnet/xAPI-Spec
[2] http://aicc.github.io/CMI-5_Spec_Current
[3] The authors kindly made the source code available to us.
[4] http://flair.schule
[5] http://kansas-suche.de

	DR	CT	CS	AC	PO
Mgbeg					
GramEx			(•)		
KillerFiller			(•)		•
Task Generator		•	(•)	(•)	
MIRTO		•	•	(•)	(•)
GEG		•		(•)	
FAST			(•)		
ArikIturri		•			•
WebExperimenter			(•)	(•)	
Sakumon	•	(•)	•	(•)	
VIEW		•	(•)	•	•
ClozeFox		•	(•)	•	•
LEA		•	•	(•)	•
Lärka	(•)		(•)		
COLLIE		•		(•)	
Language Muse		•	(•)		

Table 1: Exercise generation system functionalities
Document ranking (DR), custom text input (CT), configurable settings (CS), authentic context (AC) and portable output (PO) marked by • if offered, by (•) if partially offered.

vant to automatically generated, form-based grammar exercises, none of them combines all features. Especially the selection of suitable documents is hardly targeted at all. Only *Sakumon*, which is also listed among the document ranking systems, offers full-fledged document filtering. Since it was developed as an assistant system, exercise generation has, however, not yet been fully automated.

Support for preserving the authentic context varies considerably from one system to the other. Rule-based systems do not rely on authentic texts at all. Examples include the *Mgbeg* exercise generator (Almeida et al., 2017) and *GramEx* (Perez-Beltrachini et al., 2012). Among the tools using authentic texts, a couple use only decontextualized, single sentences. This encompasses for example *Lärka*[6] (Volodina et al., 2014), *ArikIturri* (Aldabe et al., 2006) and *FAST* (Chen et al., 2006). A range of systems integrate the exercises into the base text, yet visual context such as markup elements and images are removed. This is, for instance, the case in the Tutor Assistant's *Task Generator* (Toole and Heift, 2001), *MIRTO* (Antoniadis et al., 2004), the *Grammar Exercise Generator (GEG)* (Melero and Font, 2001), the *Language Exercise App (LEA)* (Perez and Cuadros, 2017) and *COLLIE*[7] (Bodnar and Lyster, 2021). Visual context is only preserved in those exercise generation tools implemented as web plugins. Prominent examples include *VIEW*[8] (Meurers et al., 2010; Reynolds et al., 2014) and *ClozeFox*[9] (Colpaert and Sevinc, 2010).

Most of the exercise generation tools provide the generated exercises within the system and do not offer any export functionalities. Noticeable exceptions include *KillerFiller*, *ArikIturri* and the *LEA*. The web plugins *VIEW* and *ClozeFox* are also portable by nature.

The degree to which users can influence the exercises to be generated is generally rather low. Although in most systems, users can upload their own texts, they often have only rudimentary influence on the properties of the generated exercises. The most highly configurable applications include *MIRTO*, *Sakumon*, *Language Muse* (Madnani et al., 2016) and the *LEA*. Since *Sakumon* is an assistant system, instructors can and must select the target items and distractors manually from among the tool's suggestions. The *LEA* also allows to specify target constructions, bracket contents and distractors. *MIRTO* in addition lets users specify interactive supportive elements such as links to lookup pages. *Language Muse* generates a range of different activities for each text from which the user can choose the one which best suits their needs. These can be edited to allow further customization.

Our approach aims to combine the strengths of these systems into a single application.

3 System description

We integrate the exercise generation functionality into the language-aware search engine FLAIR. While the exercise generation is fully integrated into this application, we also considered the interface supporting integration of the generated exercises in the LMS serving as deployment platforms.

3.1 Implementation

FLAIR serves as base system to search the web for documents on user-specified topics. Just like ordinary web search engines, it supports restricting the search space to specific sites using operators. As illustrated in Figure 1, the system provides additional functionalities to filter and re-rank those documents based on the linguistic criteria selected by the user (left part).

[6] https://spraakbanken.gu.se/larkalabb/
[7] https://www.collietool.ca/
[8] http://purl.org/view
[9] https://wiki.mozilla.org/Education/Projects/JetpackForLearning/Profiles/ClozeFox

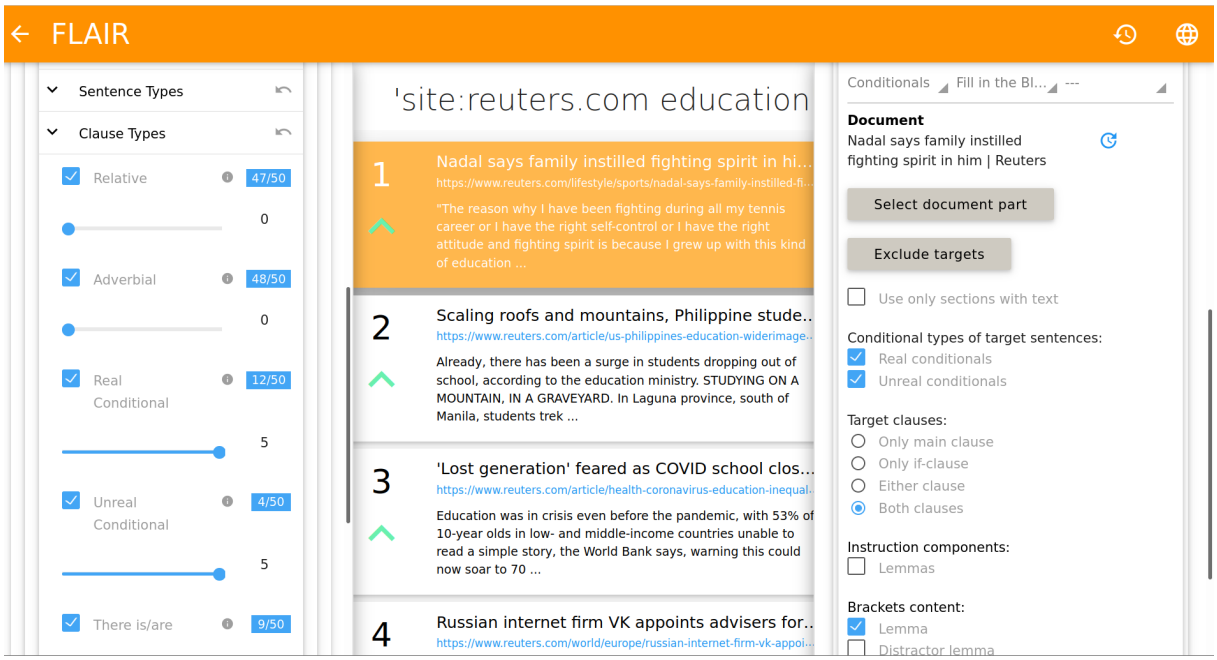


Figure 1: Exercise configuration in FLAIR

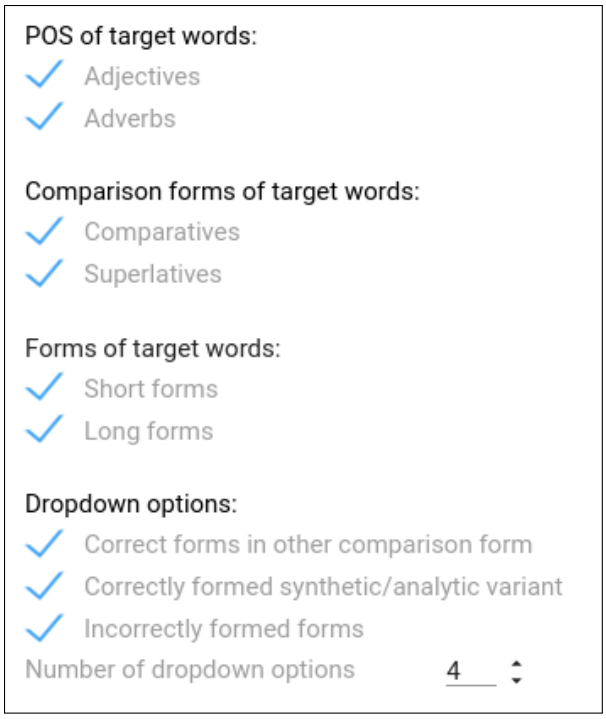
	FiB	SC	DD	MtW	M	JS
Simple present	●	●		●	●	●
Past tenses	●	●	●	●	●	●
Conditionals	●	●	●			●
Relatives	●	●	●	●		●
Comparatives	●	●	●	●	●	●
Passive	●		●			●

Table 2: Exercise types per topic

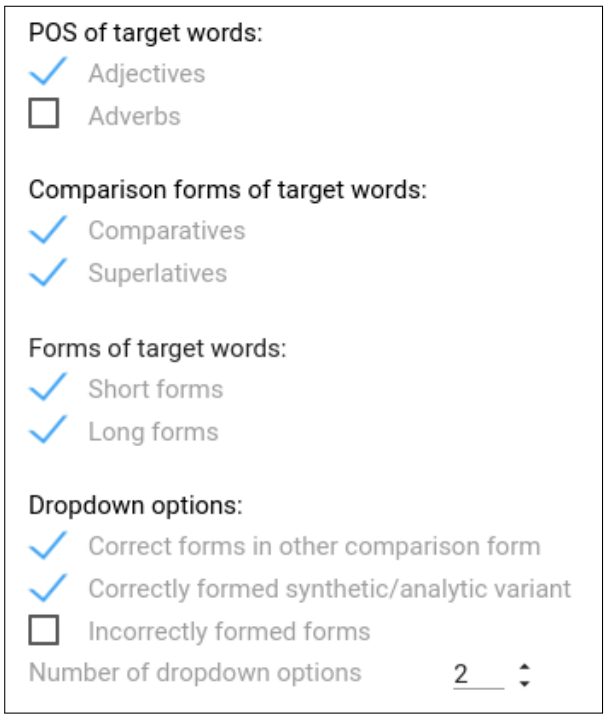
Fill-in-the-Blanks (FiB), Single Choice (SC), Drag and Drop (DD), Mark-the-Words (MtW), Memory (M), and Jumbled Sentences (JS) marked by ● if offered.

Exercise generation comes into play after the documents have been retrieved and ranked. It offers the configuration panel displayed on the right of Figure 1. Supported language means for the exercises are based on the pedagogical goals of German 7th grade high schools (Ministerium für Kultus, 2016) and include Comparatives, Present and Past tenses, Passive, Conditionals and Relative pronouns. The available exercise types depend on the language means. Table 2 shows that while Fill-in-the-Blanks exercises are supported for all language means, other exercise types we generate, such as Single Choice, Mark the Words, Drag and Drop, Memory, and Jumbled Sentences exercises, are not universally applicable. Users are shown only those exercise settings that are applicable to the selected text. These settings, on the one hand, comprise a characterization of the exercises such as the exercise type or the number and features of distractors and, on the other hand, features to restrict the choice of seed sentences. For exercises on Passive, the parametrizable characteristics of seed sentences encompass the tense (past, present and future), the aspect (simple, perfect and progressive) and the voice (active and passive). Exercises targeting Tenses support parameters for the targeted tenses and the aspect, as well as for negated and interrogative contexts. For Simple present, additional parameters allow to exclude regular or irregular forms. Seed sentences for Comparatives can be selected to contain synthetic or analytic comparative or superlative forms of adjectives or adverbs, or both. Parameters for Conditionals include the conditional type. For exercises on Relative pronouns, sentences can be restricted to those containing specific relative pronouns. The parameters for seed sentences lead to a more fine-grained subdivision of each language means into target constructions. Some of the parameters serve to manipulate exercise complexity by including or excluding additional language means, such as questions or negation, from the exercises. Other parameters which are specific to the language means, such as active and passive voice, allow to put the focus of the exercise either on the acquisition of a specific form or on the distinction between multiple forms.

Figure 2 illustrates that even for the same lan-



(a) Example 1: Parameters and generated exercise for default settings



(b) Example 2: Parameters and generated exercise for custom configuration

Figure 2: Comparison of generated exercises for different parametrizations

guage means and exercise type, in this case *Comparatives* and *Single Choice* respectively, the resulting exercises differ although they are based on the same document.[10] In Figure 2a where the default configuration is used, all comparative and superlative adjectives and adverbs are transformed into targets and the distractors also contain ill-formed forms. In Figure 2b, adverbs are excluded and the distractors contain only well-formed forms.

In order to generate an exercise from a document and an exercise specification, the algorithm automatically separates all of the markup elements in the web page from the plain text. It relies on the linguistic annotations used by the base system for document ranking and post-processes them in order to generate an abstract exercise definition. For most of the possible exercise configurations, the base system's distinction between linguistic constructions is not fine-grained enough to identify target items that comply with all activated settings options to

[10]The exercises were generated from Reuters article https://www.reuters.com/world/europe/poverty-education-levels-draw-battle-lines-french-election-2022-04-12/ and uploaded into a Moodle instance.

select seed sentences. Target items thus correspond to elements which are assigned multiple annotations by FLAIR. As the scopes of the annotations relevant to the different parameters of an exercise target are not always congruent, the resulting scope of the exercise target is determined individually for each combination of settings parameters through a set of manually defined rules.

Apart from the target construction, the abstract exercise definition contains elements such as task descriptions, distractors and pre-compiled feedback. This feedback is obtained from the already established feedback generation algorithm of the FeedBook (Rudzewitz et al., 2018). The required exercise components differ slightly from one exercise type to another. While for Mark the Words and Drag and Drop exercises, no additional content is generated, Fill-in-the-Blanks exercises need hints which are displayed in parentheses, such as the lemma of the exercise item, and Single Choice exercises require distractors. For Memory tasks, the targeted language means determines what content has to be generated for the second card. Jumbled Sentences do not need supplementary content, yet they need further processing in order to determine the parts into which sentences containing a target construction are split. The particular characteristics of additional elements and processing depend on the settings defined by the user who may, for instance, select any or multiple of lemma, distractor lemma, tense or other options depending on the language means for hints in parentheses. Distractors are equally configurable, allowing users to include well-formed but context inappropriate options such as incorrect tenses or POS, as well as ill-formed options. While the generation of well-formed elements applies NLP technology including lemmatization and Natural Language Generation (NLG), ill-formed elements are generated based on manually defined transformation rules. Markup elements are also added to the exercise definition so that the authentic context of the document can be reconstructed.

This abstract representation is then used to generate the exercises in a portable format. Since most of the relevant standards are rather complex (Griffiths, 2020), we use the H5P[11] format which integrates multiple standards and offers a library of predefined, open source exercise types (López et al., 2021; Magro, 2021).

[11] https://h5p.org/

3.2 Usage

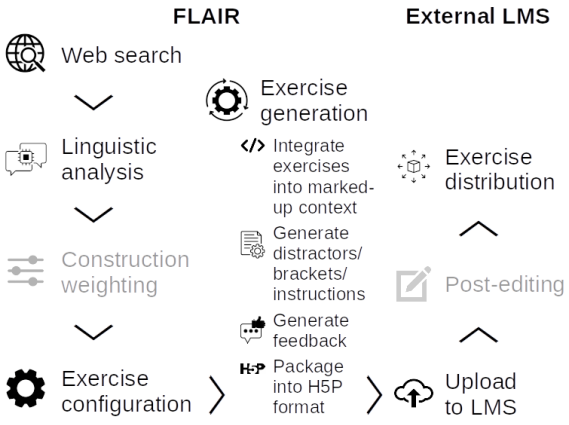


Figure 3: Exercise generation workflow

As outlined in Figure 3, the end user, typically a language instructor, will interact with FLAIR on the one hand in order to generate an exercise, and with the LMS on the other hand in order to provide the exercise to learners. The prototypical workflow starts in FLAIR where the instructor performs a web search which returns linguistically analyzed documents. Weighting linguistic constructions to re-rank the results is optional. After choosing a document from the results, the instructor configures one or multiple exercises. When exercise generation is triggered, a H5P exercise is generated, including the original mark-up, exercise components and pre-compiled feedback. The instructor then uploads the file to the LMS where he or she may edit the generated exercise and make it available to students as illustrated in Figure 4.[12] While working on the exercise, students will receive instant, dynamic feedback based on the pre-compiled feedback until they complete the task.

4 Evaluation

The quality of the exercise generation extension depends on its ability to identify documents which contain linguistic constructions that can successfully be transformed into exercise items. We conducted a three-step, pilot evaluation in order to determine the tool's performance in this respect. Due to time restrictions, the gold standard annotations and evaluations results were produced only by one of the authors.

[12] The exercise was generated from Reuters article https://www.reuters.com/lifestyle/sports/nadal-says-family-instilled-fighting-spirit-him-2022-03-13 and uploaded into a Moodle instance.

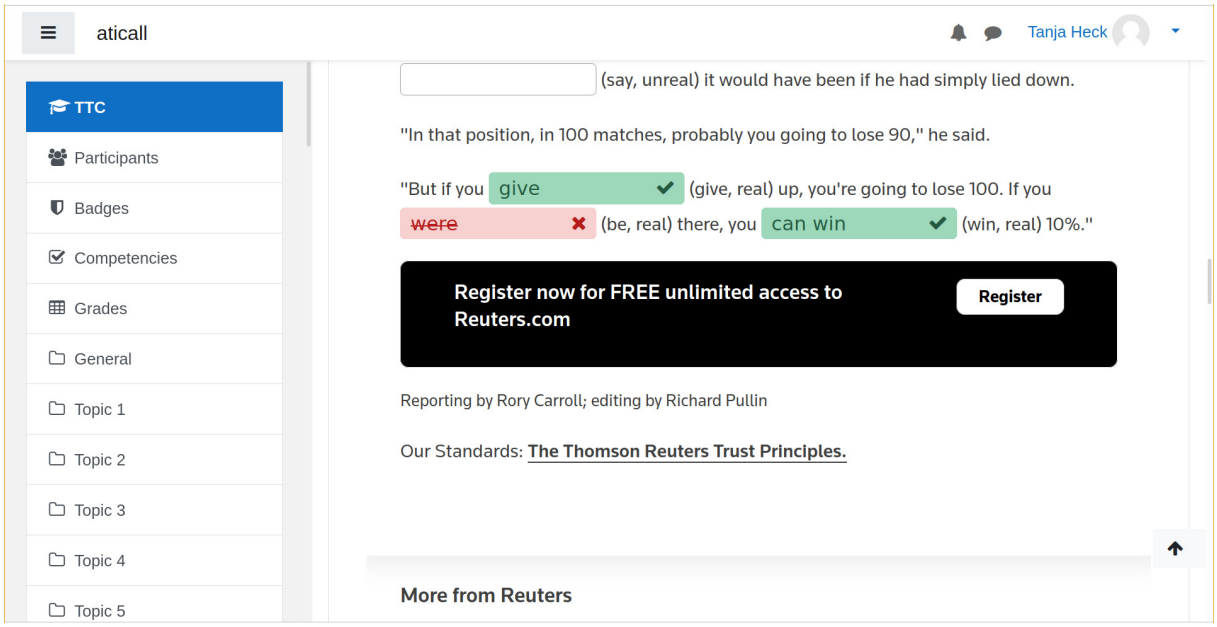


Figure 4: Excerpt of a generated H5P exercise in the LMS

4.1 Methodology

4.1.1 Suitable document selection

Suitable documents need to contain the targeted grammatical constructions relevant to the selected language means. The tool's performance in identifying them depends mostly on the reliability of the base system. We evaluated all supported language means, i.e., *Simple present*, *Past tenses*, *Comparatives*, *Conditionals*, *Relative pronouns* and *Passive sentences*. For each language means, we determined a binary score of whether the highest-ranked search result for the search term *education* contained constructions which could be transformed into exercise items. We used an additional flag to indicate whether this was possible with the default settings or only with the help of FLAIR's document ranking. When document ranking was applied, maximum weights were set for all constructions associated with the currently assessed language means.

Relevant construction identification Relevant constructions targeted by the language means can only be used for exercise generation if they are correctly annotated. Since the exercise generation extension uses a more fine-grained distinction between linguistic constructions than the base system, the performance of construction identification depends both on the base system's ability to correctly identify rather coarse-grained linguistic constructions and on the exercise generation tool's ability to correctly identify exercise targets from multiple, overlapping constructions. To this purpose, we sampled up to 10 occurrences for each type of exercise target from 100 arbitrarily selected web pages. Identical occurrences of target constructions were not considered and only web pages which contained at least one construction were taken into account. We report the precision for the identified constructions since the quality of most of our exercise types depends on the correctness of the used constructions, whereas recall is less important as long as sufficient exercise opportunities are found.

Target generation The target generation ratio is defined as the ratio of the number of actual exercise items in the generated exercises to the number of potential target constructions before post-processing. Although the identified constructions form the basis for exercise generation, some of them may be rejected during post-processing so that they cannot be transformed into exercise items. A perfect ratio of 1 indicates that all potential target constructions could be turned into an exercise target. Rejecting all constructions decreases the ratio to 0. In this evaluation which exclusively targets the performance of the exercise generation tool, we built on the search results obtained in the first evaluation step. The generation ratio was calculated for all supported language means-type combinations.

4.2 Results

4.2.1 Suitable document selection

The degree of difficulty in identifying **suitable documents** varied from one language means to another. For ***comparatives*** and ***simple present***, documents containing the targeted constructions were plentiful so that exercises could be generated on FLAIR's default settings. In order to find documents containing ***conditional***, ***passive*** or ***relative clause*** constructions, however, FLAIR's construction weighting needed to be applied. Setting high weights for conditional clauses, passive voice or relative pronouns respectively yielded high-ranked documents containing potential exercise items. ***Past tense*** exercises were also possible on documents identified with the standard settings, yet with little variety in the targeted tenses. When setting high weights for past tenses, the highest-ranked document contained past progressive constructions in addition to the previously included simple past and present perfect findings. Increasing the number of search results to 50 allowed to also target past perfect and present perfect progressive when setting construction weights. Only for past perfect progressive, none of the documents returned for the given search term contained any occurrences.

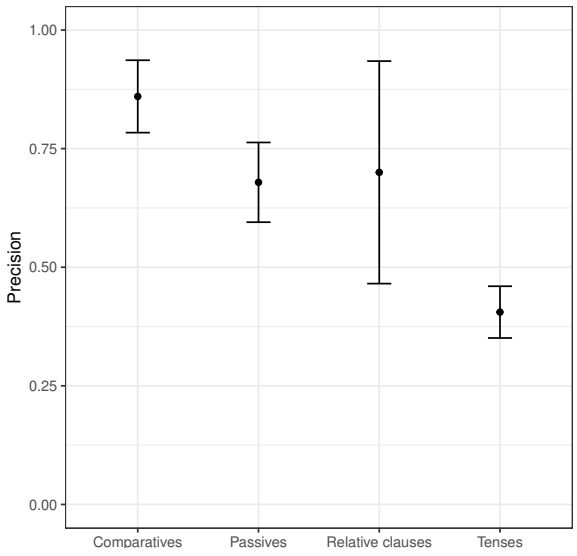


Figure 5: Precision of construction identification[13]

Relevant construction identification As illustrated by the plot in Figure 5 showing precision and standard error of pedagogical construction identification, the precision differs considerably between the language means as well as for the different instances of the language means. The full list of linguistic constructions subtypes relevant for each of these pedagogical language means is included in the Appendix. ***Comparative*** constructions all obtained fairly high precision values. Errors are mostly attributed to incorrectly assigned POS tags and thus already introduced in FLAIR's initial annotation. With respect to ***conditionals***, the performance for the two types differed considerably. While real conditionals were detected at high precision, most findings of unreal conditionals are in fact real conditionals. Performance with ***active and passive*** constructions was slightly lower on average. Tenses in simple aspect were rarely mislabelled for both active and passive voice. Constructions with progressive aspect, on the other hand, were often mislabelled, especially when combined with perfect aspect. Incorrect labels concern either aspect or voice. Precision values are only slightly better for active than for passive constructions. The performance for ***tenses*** was generally rather poor. Interrogative and negation annotations were not always correct, especially when the sentence constituted a question where the clause containing the construction was not in interrogative form. Past tenses in addition produced issues similar to those encountered with passive constructions that are not related to the active-passive distinction. The most prominent cause for incorrect labelling which was responsible for the overall poor performance in this category, with only 103 out of 232 occurrences labelled correctly, consists in the distinction between regular and irregular verbs. This generally resulted from the presence of an irregular auxiliary verb in the construction scope which incorrectly triggered the *irregular* label. Since simple present constructions do not distinguish between regular and irregular forms, performance for those was slightly better with 48 out of 80 occurrences labelled correctly. ***Relative pronouns*** performed very well for the most common pronouns *who*, *which* and *that* with 28 out of 30 occurrences labelled correctly. Only occurrences categorized as relative pronouns other than these three pronouns were incorrect, so the average precision is comparable to that of *Passives*.

Target generation Figure 6 depicts the target generation ratios for all language means. It shows that the results for the **target generation ratio** are generally reasonably high, although they vary from one language means to another. For ***past tense*** and ***relative pronoun*** exercises of all types, all predicted target constructions could be turned into

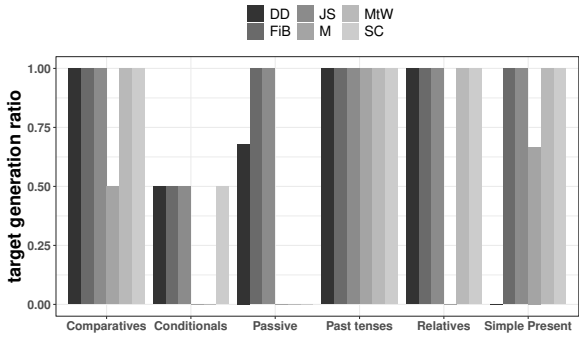


Figure 6: Target generation ratios

For each language means, the ratios are given for the following exercise types (left to right): Drag and Drop (DD), Fill-in-the-Blanks (FiB), Jumbled Sentences (JS), Memory (M), Mark-the-Words (MtW), and Single Choice (SC).

exercise targets. ***Comparative*** and ***simple present*** also exhibited good performance; only for Memory tasks did their resulting item number fall short of the prediction as some construction values were identical. Ratios for ***passive*** attained also maximum values except for Drag and Drop exercises where the NLP analysis in some cases failed to detect the relevant sentence parts targeted by this exercise. For ***conditional clause*** exercises, half of the constructions could be transformed into exercise targets. The other half deviated too much from standard tense and aspect constellations of conditional clauses so that they could not be analyzed by the NLP pipeline.

5 Conclusion

We presented a tool for automatic generation of English form-based grammar exercises from authentic web texts. It uses a language-aware search engine to address the challenge of identifying documents rich in the pedagogically targeted language means. While the integration of feedback aims at micro-adaptivity of the exercises, the tool also supports macro-adaptivity by allowing generation of parallel exercises at different levels of complexity. High parametrization of the exercise generation gives instructors control over the characterisitics of the generated exercises.

An evaluation of the current implementation yielded promising results. The tool robustly generates functional exercises that comply with the user configurations. While the evaluation considered the performance aspects in isolation, in the future we plan to perform an end-to-end evaluation in an authentic education context.

Limitations of our tool arise from building on an existing system for input enrichment before performing more detailed linguistic analyses to support exercise generation. As a result, some of the language material provided by the input enrichment system is rejected during the exercise generation phase. We are thus considering to enrich the initial linguistic analysis performed in the input enrichment component to the more fine-grained level that will make it possible to use it for both the document ranking component and the exercise generation.

Future work also will be important to determine the effect of the parameter settings on the exercise complexity as experienced by the learner and to determine which parameter constellations are appropriate for generating developmentally proximal exercises for a given target population. This will open the path for adaptive sequencing algorithms to offer exercises optimally adapted to the learner's current proficiency level and cognitive capabilities.

References

Itziar Aldabe, Maddalen Lopez de Lacalle, Montse Maritxalar, Edurne Martínez, and Larraitz Uria. 2006. Arikiturri: An automatic question generator based on corpora and nlp techniques. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems (ITS'06), Jhongli (Taiwan)*, pages 584–594. Springer-Verlag.

J. João Almeida, Eliana Grande, and Georgi Smirnov. 2017. Exercise generation on language specification. In *Recent Advances in Information Systems and Technologies*, pages 277–286, Cham. Springer International Publishing.

Georges Antoniadis, Sandra Echinard, Olivier Kraif, Thomas Lebarbé, Mathieu Loiseau, and Claude Ponton. 2004. Nlp-based scripting for call activities. In *Proceedings of the Workshop on eLearning for Computational Linguistics and Computational Linguistics for eLearning*, pages 18–25.

Lisa Beinborn. 2016. *Predicting and manipulating the difficulty of text completion exercises for language learning*. Ph.D. thesis, Department of Computer Science, Technische Universität Darmstadt.

Jasmine Bennöhr. 2005. *A web-based personalised textfinder for language learners*. Ph.D. thesis, Master's thesis, School of Informatics, University of Edinburgh.

Stephen Bodnar and Roy Lyster. 2021. Choose Your Own Content: a technology-based approach for automatically creating tailored grammar practice exercises from the web. 9th International Conference on Second Language Pedagogies.

Chia-Yin Chen, Hsien-Chin Liou, and Jason S. Chang. 2006. FAST – an automatic generation system for grammar tests. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 1–4, Sydney, Australia. Association for Computational Linguistics.

Maria Chinkina, Madeeswaran Kannan, and Detmar Meurers. 2016. Online information retrieval for language learning. In *Proceedings of ACL-2016 System Demonstrations*, pages 7–12.

Maria Chinkina and Detmar Meurers. 2016. Linguistically-aware information retrieval: Providing input enrichment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 188–198, San Diego, CA. ACL.

Jozef Colpaert and Emre Sevinc. 2010. ClozeFox: Gap Exercise Generator with Scalable Intelligence for Mozilla Firefox. `https://github.com/emres/clozefox`. [Online; accessed 25-March-2022].

Sabrina Dittrich, Zarah Weiss, Hannes Schröter, and Detmar Meurers. 2019. Integrating large-scale web data and curated corpus data in a search engine supporting german literacy education. In *Proceedings of the 8th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2019), September 30, Turku Finland*, 164, pages 41–56. Linköping University Electronic Press.

Colleen Griffiths. 2020. All about the lms. *Learning Management Systems*.

Ayako Hoshino and Hiroshi Nakagawa. 2008. A cloze test authoring system and its automation. In *Advances in Web Based Learning – ICWL 2007*, pages 252–263, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ji-Ung Lee, Erik Schwan, and Christian M Meyer. 2019. Manipulating the difficulty of c-tests. *arXiv preprint arXiv:1906.06905*.

Chao-Lin Liu, Chun-Hung Wang, Zhao-Ming Gao, and Shang-Ming Huang. 2005. Applications of lexical information for algorithmically composing multiple-choice cloze items. In *Proceedings of the Second Workshop on Building Educational Applications Using NLP*, pages 1–8, Ann Arbor, Michigan. Association for Computational Linguistics.

Sergio-Ramón Rossetti López, Ma Teresa García Ramírez, and Isaac-Shamir Rojas Rodríguez. 2021. Evaluation of the implementation of a learning object developed with h5p technology. *Vivat Academia*, 154:1–24.

Nitin Madnani, Jill Burstein, John Sabatini, Kietha Biggers, and Slava Andreyev. 2016. Language muse: Automated linguistic activity generation for english language learners. *Grantee Submission*.

Juliana Magro. 2021. H5p. *Journal of the Medical Library Association: JMLA*, 109(2):351.

Maite Melero and Ariadna Font. 2001. Construction of a Spanish Generation Module in the framework of a general-purpose, Multilingual Natural Language Processing System. *VII International Symposium on Social Communication*.

Detmar Meurers, Ramon Ziai, Luiz Amaral, Adriane Boyd, Aleksandar Dimitrov, Vanessa Metcalf, and Niels Ott. 2010. Enhancing authentic web pages for language learners. In *Proceedings of the 5th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 10–18, Los Angeles. ACL.

Eleni Miltsakaki and Audrey Troutt. 2008. Real time web text classification and analysis of reading difficulty. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications (BEA-3) at ACL'08*, pages 89–97, Columbus, Ohio. Association for Computational Linguistics.

Jugend und Sport Ministerium für Kultus. 2016. Bildungsplan des Gymnasiums: Englisch als erste Fremdsprache [academic school track curriculum: English as first foreign language].

Niels Ott and Detmar Meurers. 2011. Information retrieval for education: Making search engines language aware. *Themes in Science and Technology Education*, 3(1-2):9–30.

Irina Pandarova, Torben Schmidt, Johannes Hartig, Ahcène Boubekki, Roger Dale Jones, and Ulf Brefeld. 2019. Predicting the difficulty of exercise items for dynamic difficulty adaptation in adaptive language tutoring. *International Journal of Artificial Intelligence in Education*, 29(3):342–367.

Matthew Peacock. 1997. The effect of authentic materials on the motivation of EFL learners. *ELT Journal*, 51(2):144–156.

Naiara Perez and Montse Cuadros. 2017. Multilingual CALL framework for automatic language exercise generation from free text. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 49–52, Valencia, Spain. Association for Computational Linguistics.

Laura Perez-Beltrachini, Claire Gardent, and German Kruszewski. 2012. Generating grammar exercises. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 147–156. Association for Computational Linguistics.

Robert Reynolds, Eduard Schaf, and Detmar Meurers. 2014. A view of Russian: Visual input enhancement and adaptive feedback. In *Proceedings of the third workshop on NLP for computer-assisted language learning*, NEALT Proceedings Series 22 / Linköping Electronic Conference Proceedings 107, pages 98–112, Uppsala. ACL.

Cameron Romney. 2016. Considerations for using images in teacher made materials. *The*, pages 280–289.

Björn Rudzewitz, Ramon Ziai, Kordula De Kuthy, Verena Möller, Florian Nuxoll, and Detmar Meurers. 2018. Generating feedback for English foreign language exercises. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 127–136, New Orleans, Louisiana. Association for Computational Linguistics.

Vasile Rus, Dan Stefanescu, William Baggett, Nobal Niraula, Don Franceschetti, and Arthur C Graesser. 2014. Macro-adaptation in conversational intelligent tutoring matters. In *International Conference on Intelligent Tutoring Systems*, pages 242–247. Springer.

Lee Schwartz, Takako Aikawa, and Michel Pahud. 2004. Dynamic language learning tools. *Proceedings of InSTIL/ICALL Symposium 2004*.

Kahtleen M. Sheehan, Irene W. Kostin, and Yoko Futagi. 2007. Sourcefinder: A construct-driven approach for locating appropriately targeted reading comprehension source texts. In *Proceedings of the 2007 Workshop of the International Speech Communication Association, Special Interest Group on Speech and Language Technology in Education*.

Pavel V Sysoyev. 1999. Integrative l2 grammar teaching: Exploration, explanation and expression. *The internet TESL journal*, 5(6):1–13.

Janine Toole and Trude Heift. 2001. Generating learning content for an intelligent language tutoring system. In *Proceedings of NLP-CALL Workshop at the 10th Int. Conf. on Artificial Intelligence in Education (AI-ED). San Antonio, Texas*, pages 1–8.

Elena Volodina, Ildikó Pilán, Lars Borin, and Therese Lindström Tiedemann. 2014. A flexible language learning platform based on language resources and web services. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3973–3978, Reykjavik, Iceland. European Language Resources Association (ELRA).

Mariia Zabolotniaia, Zhichao Cheng, Evgenij Dorozhkin, and Anton Lyzhin. 2020. Use of the lms moodle for an effective implementation of an innovative policy in higher educational institutions. *International Journal of Emerging Technologies in Learning (iJET)*, 15(13):172–189.

Arianna Zanetti, Elena Volodina, and Johannes Graën. 2021. Automatic generation of exercises for second language learning from parallel corpus data. *International Journal of TESOL Studies*, 3(2):55–71.

Sun Zhuomin. 2010. Language teaching materials and learner motivation. *Journal of Language Teaching and Research*, 1.

Appendix A: Detailed target identification results

Up to ten occurrences were randomly sampled for all target constructions. From these samples, the numbers of correctly and incorrectly labelled instances were determined and precision was calculated.

Grammatical construction	# correct	# incorrect	Precision
Conditional: real	9	1	0.9
Conditional: unreal	2	8	0.2
Passive: present simple	9	1	0.9
Passive: past simple	10	0	1.0
Passive: future simple	0	2	0.0
Passive: present perf.	10	0	1.0
Passive: past perf.	10	0	1.0
Passive: future perf.	1	7	0.125
Passive: present prog.	0	0	
Passive: past prog.	0	0	
Passive: future prog.	0	2	0.0
Passive: present perf. prog.	0	4	0.0
Passive: past perf. prog.	1	2	0.3333
Passive: future perf. prog.	0	0	
Active: present simple	6	4	0.6
Active: past simple	10	0	1.0
Active: future simple	10	0	1.0
Active: present perf.	10	0	1.0
Active: past perf.	10	0	1.0
Active: future perf.	6	4	0.6
Active: present prog.	10	0	1.0
Active: past prog.	10	0	1.0
Active: future prog.	10	0	1.0
Active: present perf. prog.	7	3	0.7
Active: past perf. prog.	5	5	0.5
Active: future perf. prog.	1	1	0.5
Past simple: stmt., affirm., reg.	9	1	0.9
Past simple: stmt., affirm., irreg.	10	0	1.0
Past simple: stmt., neg., reg.	1	9	0.1
Past simple: stmt., neg., irreg.	6	4	0.6
Past simple: quest., affirm., reg.	4	6	0.4
Past simple: quest., affirm., irreg.	4	6	0.4
Past simple: quest., neg., reg.	1	0	1.0
Past simple: quest., neg., irreg.	4	6	0.4
Present perf.: stmt., affirm., reg.	9	1	0.9
Present perf.: stmt., affirm., irreg.	10	0	1.0
Present perf.: stmt., neg., reg.	9	1	0.9
Present perf.: stmt., neg., irreg.	6	4	0.6
Present perf.: quest., affirm., reg.	4	6	0.4
Present perf.: quest., affirm., irreg.	3	7	0.3
Present perf.: quest., neg., reg.	3	1	0.75
Present perf.: quest., neg., irreg.	4	3	0.5714
Past perf.: stmt., affirm., reg.	0	0	
Past perf.: stmt., affirm., irreg.	4	6	0.4
Past perf.: stmt., neg., reg.	0	0	

Grammatical construction	# correct	# incorrect	Precision
Past perf.: stmt., neg., irreg.	5	5	0.5
Past perf.: quest., affirm., reg.	0	0	
Past perf.: quest., affirm., irreg.	0	9	0.0
Past perf.: quest., neg., reg.	0	0	
Past perf.: quest., neg., irreg.	0	1	0.0
Past prog.: stmt., affirm., reg.	0	3	0.0
Past prog.: stmt., affirm., irreg.	3	7	0.3
Past prog.: stmt., neg., reg.	0	0	
Past prog.: stmt., neg., irreg.	1	9	0.1
Past prog.: quest., affirm., reg.	0	0	
Past prog.: quest., affirm., irreg.	0	10	0.0
Past prog.: quest., neg., reg.	0	0	
Past prog.: quest., neg., irreg.	0	1	0.0
Present perf. prog.: stmt., affirm., reg.	0	7	0.0
Present perf. prog.: stmt., affirm., irreg.	3	7	0.3
Present perf. prog.: stmt., neg., reg.	0	1	0.0
Present perf. prog.: stmt., neg., irreg.	0	1	0.0
Present perf. prog.: quest., affirm., reg.	0	1	0.0
Present perf. prog.: quest., affirm., irreg.	0	4	0.0
Present perf. prog.: quest., neg., reg.	0	0	
Present perf. prog.: quest., neg., irreg.	0	0	
Past perf. prog.: stmt., neg., reg.	0	0	
Past perf. prog.: stmt., neg., irreg.	0	2	0.0
Past perf. prog.: quest., affirm., reg.	0	0	
Past perf. prog.: quest., affirm., irreg.	0	0	
Past perf. prog.: quest., neg., reg.	0	0	
Past perf. prog.: quest., neg., irreg.	0	0	
Present simple: stmt., affirm., 3rd pers.	7	3	0.7
Present simple: stmt., affirm., not 3rd pers.	8	2	0.8
Present simple: stmt., neg., 3rd pers.	9	1	0.9
Present simple: stmt., neg., not 3rd pers.	8	2	0.8
Present simple: quest., affirm., 3rd pers.	3	7	0.3
Present simple: quest., affirm., not 3rd pers.	4	6	0.4
Present simple: quest., neg., 3rd pers.	5	5	0.5
Present simple: quest., neg., not 3rd pers.	4	6	0.4
Relative pronouns: who	10	0	1.0
Relative pronouns: which	9	1	0.9
Relative pronouns: that	9	1	0.9
Relative pronouns: other relative pronoun	0	10	0.0
Adjective: comparative, synthetic	10	0	1.0
Adjective: superlative, synthetic	8	2	0.8
Adjective: comparative, analytic	9	1	0.9
Adjective: superlative, analytic	10	0	1.0
Adverb: comparative, synthetic	9	1	0.9
Adverb: superlative, synthetic	5	0	1.0
Adverb: comparative, analytic	10	0	1.0
Adverb: superlative, analytic	9	1	0.9