Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 4817–4826
Marseille, 20-25 June 2022

The Tembusu Treebank: An English Learner Treebank

Luis Morgado da Costa ♠, **Francis Bond** ♠, **Roger V P Winder** ♡
♠Asian Studies, Palacký University Olomouc, ♡LCC, Nanyang Technological University
lmorgado.dacosta@gmail.com

Abstract

This paper reports on the creation and development of the Tembusu Learner Treebank — an open treebank created from the NTU Corpus of Learner English, unique for incorporating *mal-rules* in the annotation of ungrammatical sentences. It describes the motivation and development of the treebank, as well as its exploitation to build a new parse-ranking model for the English Resource Grammar, designed to help improve the parse selection of ungrammatical sentences and diagnose these sentences through *mal-rules*. The corpus contains 25,000 sentences, of which 4,900 are treebanked. The paper concludes with an evaluation experiment that shows the usefulness of this new treebank in the tasks of grammatical error detection and diagnosis.

Keywords: treebank, learner corpus, error detection, error diagnosis, parsing

1. Introduction

Treebanks are valuable language resources, where information pertaining to syntactic and/or semantic structure is provided alongside a corpus. Treebanks have many usages, both in empirical linguistic research and in NLP. They have proved to be invaluable resources for tasks such as parsing, machine translation, information retrieval, and others.

In general, most NLP tasks have something to gain from richer annotation schemas, and this is why treebanks are generally seen as worthwhile endeavors, despite requiring a large investment of resources to create.

Current attempts to address the profoundly linguistic problems of grammatical error detection and diagnosis rely on fairly shallow annotations. Mainstream shared-tasks generally bypass any notion of grammatical structure when providing labeled training data. We believe this information is important for these tasks, and aim to fill this gap. The Tembusu Treebank is new kind of data resource, designed to aid the tasks of grammatical error detection and diagnosis with deep structural annotations of syntax and semantics, along with grammatical error annotations in the form of *mal-rules*.

2. Related Work

2.1. The NTU Corpus of Learner English

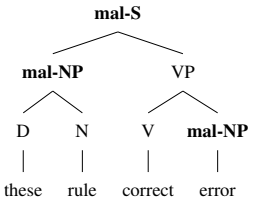
The Tembusu Treebank is based on the NTU Corpus of Learner English (Winder et al., 2017, **NTUCLE**). The NTUCLE is an open corpus of learner English, made up of assignments submitted by first year undergraduate engineering students from a major university in Singapore (NTU). It was hand-tagged by six professional English lecturers, and uses a new annotation schema largely based on pre-existing tagsets such as the ones used by the NUS Corpus of Learner English (Dahlmeier et al., 2013) and the Cambridge Learner Corpus (Nicholls, 2003). NTUCLE's first release contained 180 tagged documents, containing 9,571 sentences out of which 2,751 were considered problematic.

2.2. The English Resource Grammar

The Tembusu Treebank is built from parses generated by the English Resource Grammar (Flickinger, 2000; Copestake and Flickinger, 2000, **ERG**). The ERG is an open-source, broad-coverage computational grammar for English. It has a very large lexicon and wide coverage of syntactic phenomena capable of producing high-precision syntactic and semantic representations for English. It follows the theoretical framework of Head-Driven Phrase Structure Grammar (Pollard and Sag, 1994; Sag et al., 1999, **HPSG**) and produces Minimal Recursion Semantics (Copestake et al., 2005, **MRS**). Despite a high standard of linguistic accuracy, the ERG has an impressive coverage over unseen English text across a variety of genres — between 81.2% and 96.8% (Flickinger, 2011).

Of special interest for the Tembusu Treebank is the fact that the ERG has incorporated substantial work on the design of *mal-rules* (Bender et al., 2004; Flickinger and Yu, 2013; Suppes et al., 2014; Flickinger et al., 2016) — which allow the grammar to identify, diagnose and potentially correct a variety of ungrammatical and stylistically deprecated sentences. *Mal-rules* were first proposed by Schneider and McCoy (1998), and extend descriptive grammars in order to allow specific ungrammatical phenomena.

(1) * These rule correct error.



Taking (1) as an example, there are multiple rules in the English grammar that should prevent (1) from forming a proper sentence: i) the single noun *error* should

not be able to form a bare noun phrase (NP); ii) the NP *these rule* should not be able to form due to agreement constraints; iii) and, finally, if we assume a singular subject, there are also agreement issues between the subject and the main verb of the sentence. In (1), the nodes in the syntactic tree that would need to correspond to *mal-rules* are marked with the suffix '**mal-**'.

Early work with *mal-rules* in the ERG targeted elementary and middle-school English language education in the USA (Suppes et al., 2014), and showed very positive results in its ability to provide corrective feedback and to improve students' language use.

Systems based on the ERG participated in the 2013 CoNLL Shared Task on Grammatical Error Correction (Flickinger and Yu, 2013) and the 2016 Shared Task on Automated Evaluation of Scientific Writing (Flickinger et al., 2016). Despite not being the top-ranked systems, the ERG-based systems were able to consistently identify multiple problems in the 'gold' annotations provided as development and evaluation data in these tasks (e.g., missing, incorrect or unnecessary error annotations in the gold data, and missing plausible corrections in the test sets). More recently, the ERG has also been successfully adapted into a system designed to provide corrective feedback to undergraduate engineering students — the iTELL Automated Writing Support System (Morgado da Costa et al., 2020).

From an analysis of the results of the above-mentioned systems, it became clear that the ERG's ability to diagnose grammatical problems is potentially very good, but inherently dependent on the parse ranking models used by the grammar. However, even though the ERG has been used for grammatical error detection and diagnosis for many years, there have been no attempts to train a new model with learner data. The Tembusu Treebank is the first step to close this gap.

2.3. Similar Treebanks

Our treebank is similar in kind to other available treebanks. It is most similar to the Redwoods treebank (Oepen et al., 2002) and the DeepBank (Flickinger et al., 2012) – two other English treebanks built with the help of the ERG. In the same group we have the Japanese Hinoki Treebank (Bond et al., 2004; Bond et al., 2008) built from JACY (Siegel and Bender, 2002; Siegel et al., 2016) (for Japanese), and JATI (Moeljadi, 2017), a treebank for the Indonesian language built with INDRA (Moeljadi et al., 2015).

All these treebanks share a similar structure, and include very rich syntactic and semantic outputs. The treebanks include a list of possible parses from each grammar (representing the ambiguity generated by the source grammar) and, when possible, the single most appropriate parse selected by a human annotator. The treebanks include fairly general structures, such as simple labeled syntactic trees, as well as formalism-specific outputs such as MRS semantics, and a full derivation tree that stores enough information to replicate the full syntactic analysis done by the computational grammar. These treebanks also share infrastructure that supports their maintenance and evolution, as their respective grammars evolve over time. This includes a shared set of tools that can be used to annotate and update the treebanks, as well as to train stochastic parse-ranking models that can be used by their source grammars (Oepen et al., 2004).

The main differences between the Tembusu Treebank and other pre-existing treebanks is the fact that it is built from learner data and the fact that it uses *mal-rules*. This new treebank was then used to produce a new parse-ranking model for the ERG, with the goal of improving its error detection capabilities. The evaluation of this model is discussed in Section 4.

2.4. Similar Projects

The Tembusu treebank shares some similarity with SALLE — Syntactically Annotating the Language of Learner English (Ragheb and Dickinson, 2012; Ragheb and Dickinson, 2014) and the Universal Dependencies for Learner English (Berzak et al., 2016), which overtly follows in SALLE's footsteps.

These projects use syntactic dependency-style analysis to hand-annotate learner data, which can be useful to inform tasks such as grammatical error detection and/or correction. However, their main concern is to increase the robustness of statistical parsers, and their ability to reasonably deal with non-canonical language. Although related, these goals are not fully-aligned with those of our project.

These two projects focus on establishing a reasonable layer of dependency annotations when presented with sentences that would not be able to be annotated using standard guidelines (designed for canonical language). Neither project attempts to explicitly diagnose or annotate the source or kind of errors present in the data. And even though the second project (Berzak et al., 2016) provides a corrected version of each ungrammatical sentence (tagged using the standard universal dependency guidelines), neither project explicitly elaborates on how these annotations can be used to improve the tasks of error detection, error correction, or in the provision of corrective feedback. In general, these projects are essentially working towards certain classes of errors (or non-canonical language) being ignored by parsers by attempting to '*reduc[e] the impact of grammatical errors in automatic annotation*' (Berzak et al., 2016).

Some of the main differences between the Tembusu Treebank and the projects mentioned above are:

- our treebank uses a grammar to annotate trees, while other projects use direct human labeling. This has advantages and disadvantages. Using a grammar assumes that a theoretical model of a phenomenon/construction has been previously developed, which helps provide deeper morphosyntactic and semantic information for annotated trees. The downside is the impossibility to provide annotations

for sentences with phenomena or errors not covered by a grammar (i.e. not all sentences are annotated);
- our treebank includes full derivation trees that store enough information to replicate the complete syntactic analysis provided by the ERG — including *mal-rules*; this means that our annotations can be used to describe which constraints are being violated by an ungrammatical sentence. The above-mentioned projects label ungrammatical data in a way similar to that of grammatical data — and hence cannot easily describe *where* or *why* a sentence is ungrammatical;
- because it is produced from the ERG, our treebank is uniquely suited to train this grammar without compromising its flexibility or precision. The inherent upside of having detailed annotations is the fact that these annotations can be used by simpler systems (e.g., converting fine-grained linguistic labels into coarser ones). However, the reverse is not true. As such, while the data provided by the Tembusu Treebank could, in theory, be converted into Universal Dependencies (with some amount of work to produce adequate mappings), the reverse is not possible.

3. The Treebank

The Tembusu Treebank is built from an enhanced version of the NTUCLE (Winder et al., 2017). Data collection continued until 2021, under the same conditions and for similar student populations. Today, it contains slightly over 800 documents (≈25,000 sentences).

The Tembusu Treebank gives a new life to this learner corpus. It releases all previously unavailable data under a Creative Commons Attribution 4.0 International license, and defines a new future for this data — a learner treebank. The treebank has been in continuous development since 2020, and this paper reports on its first release (with ≈20% of the data tagged). The treebank will be made available on Github,[1] and will include the full dataset (including untagged documents).

3.1. ACE Tools

The treebanking process relied heavily on the ACE Tools:[2] a suite of open-source applications based on the Answer Constraint Engine (**ACE**). ACE is a highly efficient HPSG unification engine that supports both parsing and generation for grammars written in Type Description Language (Krieger and Schäfer, 1994). In addition to the main parsing engine, the ACE Tools also include the Full Forest Treebanker (Packard, 2015, **FFTB**) and ready-to-use binaries to train parse-ranking models from full-forest treebanks.

The Tembusu Treebank uses a slightly enhanced version of the FFTB tool. These enhancements included small changes to be able to securely serve the FFTB as a web-service (so the results of remote annotation could be centralized in a server), and improvements to the user-interface of the FFTB (i.e., providing in-tool access to grammar documentation, and making mal-rules visually distinct from other rules).

3.2. The Treebanking Process

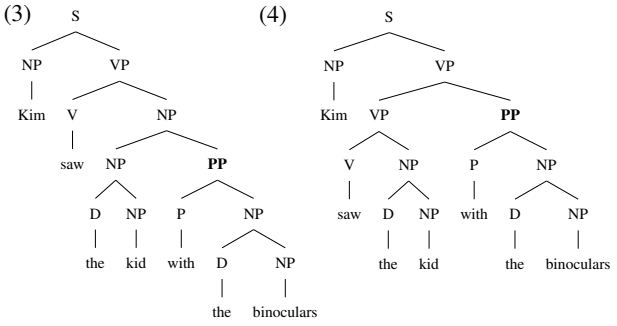
This first release of the Tembusu Treebank was tagged with the help of five student assistants (four undergraduates and one graduate student), all majoring in Linguistics and Multilingual Studies. All students had successfully completed a Syntactic Theory course, where they were introduced to the HPSG framework. However, even though students may have a detailed understanding of the theoretical inner-workings and assumptions of HPSG, treebanking sentences with a real grammar is a very different experience. As such, all five students went through an intensive training exercise.

While developing a grammar, grammarians often need to decide on the best among many possible analyses for each linguistic phenomena. As such, real grammars (in this case, the ERG) have their own assumptions, which are not always intuitive and need to be learned. For example, the destination (e.g., 'to Beijing') in a sentence such as 'She went to Beijing.' is treated as an adjunct in the ERG — while other English grammars might treat it as a complement. Much of the variety that can be found in implemented grammars mirrors current linguistic discussions. It is not always clear what is the best analysis for certain linguistic phenomena. Our treebankers needed to learn many of the decisions taken while developing the ERG.

Apart from this layer of idiosyncrasy, the main task of treebanking is dealing with ambiguity. Treebanking a sentence involves resolving any inherent ambiguity that it may have. Sentence (2) shows a classic example of PP attachment ambiguity.

(2) Kim saw the kid with the binoculars.

Consider (3) and (4) as two possible analyses of (2), which show the syntactic ambiguity. In the analysis (3), the kid was carrying or using the binoculars, while the reading captured by (4) describes a situation where the binoculars were used as an instrument in the act of seeing (i.e., it was Kim holding/using the binoculars).



Whenever there is context, the decision of how to resolve ambiguity should be done with the available context in mind. Sometimes, common-sense knowledge also plays a part in this disambiguation process. For

[1] https://github.com/lmorgadodacosta/the-tembusu-treebank

[2] http://sweaglesw.org/linguistics/acetools/

example, (4) makes more sense, especially given the common sense knowledge of what binoculars are used for. However, more often than not, the context is not enough to resolve all available ambiguity.
In sentence (5), for example, it is arguably more difficult to decide if the purchase happened on Tuesday or if the concert will happen on Tuesday. In such cases, treebanking projects need to create guidelines for the treebankers on how to deal with these problems.[3]

(5) Yasu bought tickets for the concert on Tuesday.

Finally, as previously mentioned, one of the main differentiators of the Tembusu Treebank is the fact that it uses *mal-rules*. In this case, a new set of guidelines was also developed to help treebankers understand how to use *mal-rules*, including situations where they should not be used even if they were available. In the presence of multiple ways of correcting a sentence, treebankers were instructed to select the most natural correction (similar to what is done for disambiguation — using context and common sense knowledge). Whenever *mal-rules* were available but none of the possible corrections was diagnosing/reconstructing a plausible interpretation given the context, treebankers were instructed to not treebank these sentences (i.e., told to reject them). In the future, the addition of more *mal-rules* to the ERG may allow these sentences to be tagged.
Treebankers had to go through a training exercise where they learned how to work with the treebanking tools, how to access the grammar documentation, and how to inspect previous treebanks created from the ERG (which could be used as a guide for the treebanking process). During their training, treebankers also had sessions with the main developer and maintainer of the ERG to learn and discuss the main assumptions and idiosyncrasies of this grammar. The goal of this training exercise was to annotate a set of 500 sentences, which was thoroughly tagged and discussed through multiple adjudication sessions by all the treebankers.
Adjudication sessions can only happen when more than one person tags the same set of data, and are used to discuss and harmonize annotations. These sessions are especially important in annotation tasks dealing with high complexity or ambiguity — both true in the case of treebanking. Adjudication exercises can be very important to bring treebankers to the same *wavelength* — i.e., being aware of the thought processes of the other treebankers. By requesting treebankers to discuss discrepancies and to decide on a single analysis, the annotation process becomes more streamlined, and some types of discrepancies tend to dissipate over time as treebankers are asked to adjudicate discrepancies multiple times during the treebanking process.
During training, each treebanker had the opportunity to adjudicate different subsets of sentences with multiple other treebankers. These sessions were supervised by at least one experienced linguist/treebanker, ensuring that all discussions were productive. However, despite being highly desirable, adjudicating a full treebank in this manner would be too expensive and time consuming. For this treebank, adjudication sessions happened heavily during the training process, and then more sparsely during the the main annotation exercise. This was done to save resources, while maintaining the ability to measure inter-treebanker agreement, and to ensure treebankers kept following the guidelines throughout the process.
Table 1 shows a summary of the entire treebanking process. A total of 4,900 sentences were tagged. The first dataset (ID 0) was the largest, with 500 sentences, and was used during the training sessions. All other sets had 200 sentences each.[4] The remaining 22 sets were tagged by either one or two treebankers. In total, 1700 sentences (≈35%) were tagged by two or more treebankers. Whenever a set was treebanked by two people, it was also adjudicated before moving on with further sets. It was important for the treebanking process to guarantee that adjudication sessions happened at various stages of the process, to ensure that quality remained stable throughout the entire process.
From the 4,900 treebanked sentences, 890 contain at least one *mal-rule*. In total they contain 1,253 *mal-rule* instances, distributed over 133 types – which is still only a fraction of the more than 270 *mal-rule* types currently available in the ERG.[5] As expected, the distribution of errors is strongly skewed towards types that were common among the student population that produced this dataset (Winder et al., 2017). These include: the misuse or absence of necessary articles, problems concerning verb and quantifier agreement, tense asymmetry, run-on sentences, among many others. An unfortunate consequence of this skewed distribution is that a fairly large portion of *mal-rule* types (n=58) have only one instance in the treebank – showing that there is still room to improve this dataset.

Measuring the Quality of the Treebank

Evaluating the quality of the treebank process is not a simple endeavor, especially since there is no gold standard to measure against. One could argue that a human annotated treebank is, in itself, a gold standard for future attempts to automatically select the best parse/tree for a given sentence. However, as discussed above, the task of treebanking includes common-sense reasoning, as well as true ambiguities that prevent this task from being treated as having a single correct answer. In most cases, the task is picking the *best possible analysis* from a pool of plausible analyses. As such, it is difficult to discuss quality in a very explicit way.

[3] For example, when context alone was not enough to clear up ambiguity, our treebankers were told to prefer higher attachment of PPs — i.e., prefer trees like (4), instead of (3).

[4] Although sets do not reflect document boundaries, individual documents can still be retrieved from the dataset

[5] A list of all *mal-rule* types can be found in the source code of the ERG (http://svn.delph-in.net/erg/trunk)

Datasets		Treebankers				
ID	**Size**	**A**	**B**	**C**	**D**	**E**
0	500	★	★	★	★	★
1	200	★	★			
2	200			★	★	
3	200	☆				
4	200		☆			
5	200					☆
6	200				☆	
7	200	☆				
8	200		☆			
9	200					☆
10	200				☆	
11	200	★				★
12	200		★		★	
13	200					☆
14	200		☆			
15	200					☆
16	200				☆	
17	200					☆
18	200		☆			
19	200					☆
20	200				☆	
21	200				★	★
22	200		★			★
Total	4900	1300	1900	700	1900	2300

Table 1: Treebank Summary (filled stars indicate sets tagged by two or more treebankers)

The common practice to measure the quality of a treebank relies on portions of the treebank that have been tagged by two or more people. The same metrics used by computational parsers are applied to double annotated subsets of the corpus, producing a measure of how much two treebankers' choices overlap, without necessarily defining which one is the correct tree. To measure this $\approx 35\%$ of this treebank was tagged by two people, with overlapping sets prepared as part of the treebanking process.

The metric used to measure the overlap between two treebankers is derived from the PARSEVAL metric, first proposed in Black et al. (1991). PARSEVAL is useful for constituency-based parsers, and is able to calculate how much the constituents defined by two different parse trees overlap.

The implementation used for this paper follows Collins (1997) to define Labeled Precision. In this definition, a constituent is only deemed equivalent if: a) it spans over the same set of words in the sentence; and b) has the same label. Unlabeled Precision is a similar metric where a constituent only needs to span over the same set of words in the sentence to be considered equivalent (i.e., the label may or may not match) — which is useful if there are many similar labels available.

When used to evaluate parsers, one tree is defined as canonical ('gold' or 'target'), and the tree produced by the parser is evaluated against that tree. In its canonical form, PARSEVAL precision is calculated using the formula shown in (6). And it essentially measures the percentage of constituents in the generated tree that exist in the gold standard. PARSEVAL Recall (also labeled or unlabeled) is a related metric and can be calculated using the formula in (7). Recall can be seen as a measure of completeness or, in other words, the percentage of the gold tree that is matched by constituents in the generated tree.

However, when no tree is canonical — as is the case for treebank adjudication — this algorithm needs to be slightly modified, so it is not biased to any particular tree. Agreement is, essentially, a measure between precision and recall — the formulas can be seen in (8) and (9), for Labeled Agreement (**LA**) and Unlabeled Agreement (**UL**), respectively.

These formulas ensure that if two trees are considered equivalent, then the denominator (Number of the sum of constituents in both trees) is exactly the same as twice the numerator (Number of labeled/unlabeled constituents equivalent between both trees) — yielding a score of 1. If there are no constituents considered equivalent in both trees, then it produces a score of 0. A partial overlap of the two trees will yield a score between 0 and 1, proportional to the amount of overlap, but not biased toward either of the two trees.

$$(6)\quad \text{P} = \frac{\text{No. of generated constituents that also exist in the GOLD tree}}{\text{Number of constituents in the generated tree}}$$

$$(7)\quad \text{R} = \frac{\text{No. of generated constituents that also exist in the GOLD tree}}{\text{Number of constituents in the GOLD tree}}$$

$$(8)\quad \text{LA} = \frac{2 \times \text{Number of labeled constituents equivalent in both trees}}{\text{Number of the sum of constituents in both trees}}$$

$$(9)\quad \text{UA} = \frac{2 \times \text{Number of unlabeled constituents equivalent in both trees}}{\text{Number of the sum of constituents in both trees}}$$

Scores for these two metrics were computed for each sentence before being adjudicated and then averaged across all sentences in a given set. The results can be seen in Table 2. The unlabeled precision is, naturally, slightly higher than the labeled precision (by roughly 5%). It is possible to observe a slight tendency to increase the overlap in later sets, which shows that the treebankers are getting more accurate with experience. An average agreement score of 73.1% for labeled agreement and 78% for unlabeled agreement is in line with what was expected. These numbers are comparable to those provided by Tanaka et al. (2005) — reporting 83.5% for a similar metric of labeled agreement across annotators, when building the Hinoki Treebank with a fairly large HPSG grammar of Japanese. In comparison, the ERG is a much larger grammar, capable of generating a lot more ambiguity. The NTUCLE also contains longer sentences than Hinoki. This explains why the precision is lower than those presented by Tanaka et al. (2005).

In total, 76.3% of the 4,900 annotated sentences in the Tembusu treebank had a suitable parse (i.e., 23.7% of all sentences were rejected). These numbers can be explained by two factors: i) grammatical sentences for which the ERG cannot produce an adequate analysis were rejected (see Section 2.2); ii) ungrammatical sentences for which there were no available mal-rules were also rejected. These numbers are expected to improve as the ERG's syntactical and lexical coverage increases

overtime, and as the mal-rule repertoire expands to currently unavailable classes of errors.

ID	Size	Overlap					LA	UA
0	500	**A**	**B**	**C**	**D**	**E**	0.681	0.747
1	200	**A**	**B**				0.738	0.778
2	200			**C**	**D**		0.690	0.730
11	200	**A**				**E**	0.773	0.812
12	200		**B**		**D**		0.773	0.820
21	200				**D**	**E**	0.775	0.816
22	200		**B**			**E**	0.761	0.807
	1,700						0.731	0.780

Table 2: Agreement of Overlapped Sets

3.3. New Parse Ranking Model

As introduced above, one of the main reasons to create the Tembusu Treebank was to gather and annotate data to train a new *mal-rule* enhanced parse-ranking model for the ERG. This new model should, in principle, perform better at tasks such as parse-selection and error diagnosis when using *mal-rules* within the ERG.

One of the current issues of using the ERG with *mal-rules* enabled but with a model not trained using mal-rules is the fact that the model does not know the relative likelihood of *mal-rules* when compared with the other available rules in the grammar. *Mal-rules* are essentially initiated with neutral weights, when many other rules show up with negative weights inside the model. This effectively makes the grammar select parses with *mal-rules* even when other plausible grammatical parses are still available. Training a parse ranking model on a treebank containing *mal-rules* allows the model to store the right relative weights of all rules. With enough data, theoretically, this means that the grammar should be able to prefer parses without *mal-rules* whenever a plausible parse is available.

With this in mind, the Tembusu Treebank was used to train a new maximum entropy parse ranking model for the ERG. This model was trained using available binaries in the ACE Tools (see Section 3.1), following the standard parameters used to train other ERG models (e.g., grandparenting=3, see Toutanova et al. (2005)). In the next section we discuss a small experiment designed to evaluate this new model.

4. Evaluation

We set up an experiment to determine if the new parse ranking model had measurable impact on the performance of the error detection and diagnosis capabilities of the *mal-rule* enhanced ERG (**edERG**). It measured only the ERG's ability to better detect and diagnose the classes of errors it was already designed to capture (i.e., errors for which *mal-rules* already existed), although its results could be extrapolated for future *mal-rules*, if suitable treebanked data becomes available.

We selected a test set of 1,000 sequential sentences, collected from the unannotated portion of the NTUCLE (approx. 30 student assignments). We then prepared and compared five configurations of the ERG, including systems with and without *mal-rules* enabled, and systems using the new and old parse ranking models.

We also compared a two-step approach for error detection, as proposed in Morgado da Costa et al. (2020). In this two-step approach, i) the standard release of the ERG is used to provide a filter to likely ungrammatical sentences (i.e., sentences rejected by the standard release of the ERG are considered potentially problematic); ii) potentially problematic sentences are further processed by the *mal-rule* enhanced version of the ERG (edERG) to perform error diagnosis.

This two-step approach helped deal with the high rate of misdiagnoses generated by the edERG without this filtering step. This happened because there was no available model trained using *mal-rules* — which was one of the main motivations to create this new treebank.

The systems compared in this experiment are:

- **ERG (orig.)**: the broad-coverage standard release of the ERG grammar (without *mal-rules*), using its original parse ranking model;
- **edERG (orig.)**: the *mal-rule* enhanced version of the ERG using its original parse ranking model;
- **edERG (new)**: the *mal-rule* enhanced version of the ERG using the new, *mal-rule* enhanced parse ranking model described in Section 3.3;
- **2-step (orig.)**: the two-step approach proposed in Morgado da Costa et al. (2020). Both the standard release of the ERG (1st step) and the *mal-rule* enhanced version of the grammar (2nd step) use ERG's original parse ranking model;
- **2-step (new)**: the same two-step approach, with the main difference that the *mal-rule* enhanced version of the grammar uses the new, *mal-rule* enhanced parse ranking model;

The five systems were created using the 'trunk' branch of ERG's SVN repository[6] — the same version used to create the Tembusu Treebank. Each sentence was parsed by all five systems, using ACE with the same parameters set to create the treebank.[7]

Table 3 shows an initial summary of the results obtained by analyzing the top/best parse for each system. Sentences are classified in three categories: 'w/o errors' (i.e., the top parse did not include *mal-rules*), 'w/ errors' (i.e., the top parse includes at least one *mal-rule*), and 'no parse' (i.e., the system was unable to produce a parse for that sentence).

	w/o errors	w/ errors	no parse
ERG (orig.)	0.920	0.001	0.079
edERG (orig.)	0.589	**0.315**	**0.096**
edERG (new)	0.703	0.201	**0.096**
2-step (orig.)	**0.921**	0.037	0.042
2-step (new)	**0.921**	0.037	0.042

Table 3: Results of top/best parses (n=1,000)

[6] http://svn.delph-in.net/erg/trunk (Revision 29199)

[7] Parse-chart=15Gb; Unpacking=16Gb; Timeout=300s

The results show that the system edERG with the original model chooses a parse with at least one *mal-rule* for 31.5% of the sentences in the test set (i.e., classifies them as problematic in some way). The system edERG with the newly developed model is less greedy, diagnosing only 20.1% of these sentences as problematic. This decrease is welcomed, as it was known that using ERG's original model provided many spurious diagnoses. However, both numbers are still quite large when compared to the output of the two-step systems, which identify only 3.7% of the test set as problematic. Also noteworthy is the fact that the standard release of the ERG (introduced earlier as not containing *mal-rules*) identifies 0.1% of sentences as problematic. In fact, the standard release of the ERG contains a very select number of *mal-rules* designed to accommodate, for example, common misspellings. In this particular case, this 0.1% refers to a single sentence that was correctly identified as problematic due to a mix-up between the possessive pronoun *its* and the contraction *it's*.

Next, we wanted to confirm if the reduced number of sentences identified as problematic was being achieved by an actual decrease in misdiagnoses. We performed a second-stage evaluation that looked at the subset of 349 sentences (from the original 1000) that were flagged as problematic by at least one of the systems. Table 4 shows the summary of this analysis. Each sentence was classified into one of four categories: sentences correctly classified as problematic (i.e., that have at least one error); sentences incorrectly classified as problematic (i.e., that were classified as problematic by the system but not by human annotation); problematic sentences ignored by the system (i.e., a system classifies a sentence as grammatical but the human analysis identified at least one error in it); and sentences correctly ignored by the system (i.e., both the system and the human analysis classify a sentence as grammatical).

The results shown in Table 4 confirm that the system edERG using the new parse ranking model performs much better at correctly ignoring sentences without problems. While it is possible to observe a slight decrease in its ability to correctly identify problematic sentences, it more than halves (from 49% to 21.5%) the number of sentences misclassified as ungrammatical.

	Correctly Problem.	Incorrectly Problem.	Ignored Problem.	Correctly Ignored
ERG (orig.)	0.003	0.000	**0.430**	**0.567**
edERG (orig.)	**0.413**	**0.490**	0.020	0.077
edERG (new)	0.361	0.215	0.072	0.352
2-step (orig.)	0.095	0.011	0.338	0.556
2-step (new)	0.095	0.011	0.338	0.556

Table 4: Human grammaticality judgments (n=349)

The results presented in Table 4 also show that both two-step systems have a much lower incidence of misclassified sentences (of just around 1.1%) — which was why it was the approach used in Morgado da Costa et al. (2020). However, sustaining such a low rate of misdiagnoses comes at a cost — many problematic sentences are ignored by these two-step systems. This happens because the standard release of the ERG is able to parse most sentences without providing any warning (Table 3 shows it was able to parse 92% of the test set). As such, it becomes clear that using the standard release of the ERG might not be the most desirable filter, since many problematic sentences are actually being ignored without a warning. This does not necessarily mean that the standard ERG is able to parse strictly ungrammatical sentences, but only that there might be viable parses with extremely implausible interpretations.

	Precision	Recall	F1
ERG (orig.)	**1.000**	0.007	0.013
edERG (orig.)	0.457	**0.954**	0.618
edERG (new)	0.627	0.834	**0.716**
2-step (orig.)	0.892	0.219	0.351
2-step (new)	0.892	0.219	0.351

Table 5: Error Detection Summary

Table 5 shows very similar results to those presented in Table 4, but from a system-relative perspective of precision, recall and F1 measures. The interpretations are very similar, but through a different lens. It should not be a surprise that the system edERG using the original model offers the highest recall, since it was the system who classified most sentences as problematic. This, however, leads to a much lower precision score (i.e., too many false positives). On the other hand, the same system using the new model shows a strong boost in precision (around 17%) but a slightly worse recall measure. Overall, the F1 measure shows that the new model performs better, when considering a balance between the losses in recall and the improvements in precision. Nevertheless, in real educational contexts, the precision in error detection and diagnosis should be rated much higher than recall — as telling a student that a sentence is incorrect when it is not can have a much worse impact than not being able to recognize a sentence as problematic.

Evaluating Error Diagnosis

Finally, the new model was also evaluated with regard to its ability to properly diagnose the errors in a sentence. To achieve this, a further subset of 151 sentences was selected. This was the subset of sentences that had been confirmed to be problematic from the earlier subset of 349 sentences discussed above. Each system was then evaluated by its ability to select a parse that would diagnose a plausible error in that sentence.

Sentences were classified as one of three categories: correct diagnosis (i.e., every *mal-rule* provided by the system's top/best parse pointed to a plausible correction for that sentence), incorrect diagnosis (i.e., the top/best parse provided by the system included at least one *mal-rule* that did not point to a plausible correction for the sentence in question), and missed diagnosis (i.e., the system provided either a parse without *mal-rules* or did not provide a parse at all). An effort was made to consider multiple possible corrections for each

particular sentence. The main criterion for this evaluation was that all *mal-rules* included in a parse had to lead to a plausible correction. Sentences with more than one *mal-rule* that presented a mix of plausible and implausible *mal-rules* were classified as 'incorrect diagnosis'. However, if a sentence that had more than one problem but included only a single plausible *mal-rule* (i.e., a partial fix), that sentence was classified as 'correct diagnosis'.

The reason for this is the fact that *mal-rules* are essentially used to generate corrective feedback for problematic sentences. As such, it was not deemed acceptable to receive a mix of adequate and inadequate diagnoses for the same sentence, since an inadequate diagnosis could lead students to make further mistakes. However, it was deemed acceptable to receive plausible feedback to only a subset of errors present in a sentence, as it could still help the student improve the sentence.

	Precision	Recall	F1
ERG (orig.)	**1.000**	0.007	0.013
edERG (orig.)	0.556	**0.920**	0.693
edERG (new)	0.770	0.795	**0.782**
2-step (orig.)	0.667	0.157	0.254
2-step (new)	0.848	0.192	0.313

Table 6: Error Diagnosis Summary

Table 6 shows the summary of this evaluation. The results show that using the new *mal-rule* enhanced model trained with our new treebank generated a strong increase in precision of error diagnosis (between 18% and 22%). And that, despite a slight decrease in recall, the F1 measure shows an overall improvement in the full resolution of the task (between 6% and 9%).

Even though the design of the two-step systems prevented the new model from contributing much towards the task of error detection, this new model does show improvements in the quality of error diagnosis in both the single and the two-step system designs.

5. Conclusions and Future Work

In sum, our evaluation experiments show the importance of training a model that includes annotations for error detection and diagnosis using *mal-rules*. Creating and exploiting resources like the Tembusu Treebank allow us to increase the precision of the feedback which should be the main focus of these tasks when applied to educational contexts.

The parse-ranking model trained with our new treebank increased ERG's error detection precision by around 17%, and the precision of its error diagnosis by around 22% (when measured without any filtering techniques).

There is, however, still room to improve. One area of concern is that even with the improvements our new model provides, systems still produce a number of false positives that may not be tolerable in a pedagogical setting — at least not without filtering, as employed by the two-step systems.

One way forward would be to continue improving the treebank. We suggest three areas for improvement:

i) The size of the treebank. Currently, the Tembusu Treebank contains only 4,900 annotated sentences. The ERG normally uses a treebank trained on the Redwoods treebank (Oepen et al., 2002) — with over 85,000 sentences. While the size of our efforts was sufficient to show improvements, we expect that a larger treebank will produce better results.

ii) The sparsity of ungrammatical sentences. Grammatical sentences are two to three times more frequent than ungrammatical sentences in our corpus (see Table 3). And the Zipfian distribution of language would support the idea that many classes of errors are very likely only sparsely annotated and even missing in our treebank. The number of ungrammatical structures effectively caps the learning potential of the model. As such, another good direction for future work would be to focus specifically on acquiring and enriching our treebank with ungrammatical sentences.

iii) The *mal-rule* repertoire. The ERG is not able to detect and diagnose all grammatical errors. This also hurts the system's performance, especially in its ability to correctly diagnose errors. In the absence of a plausible correction, the ERG often suggests less plausible corrections — this is imposed by the nature of the parse-ranking models, and how they deal with ambiguity. As such, continuing work on the repertoire of *mal-rules* within the ERG is also a worthy direction of future work.

Finally, and concerning future work that does not necessarily depend on further improving the treebank, we also believe that this new treebank can be very useful for the training of new statistical parsers. Even though our experiments have focused on using this new resource to train a parse-ranking model for the English Resource Grammar, this treebank could be used to train parsers capable of producing *mal-rule* enhanced trees without the need of a formal grammar — which would be ideal for situations where extra robustness would be required. A hybrid solution, in the form of a PCFG model on top of the ERG would also be an interesting area of future study (Zhang and Krieger, 2011).[8]

6. Acknowledgements

This research project received support from Nanyang Technological University through a Research Scholarship and an EdeX Teaching and Learning Grant administered by the Teaching, Learning and Pedagogy Division, and from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement H2020-MSCA-IF-2020 CHILL – No.101028782. We would like to thank Dan Flickinger for making himself available during the training of our treebankers, and to the five student assistants who helped tag the treebank.

[8]The ACE Tools have binaries available for this purpose

7. Bibliographical References

Bender, E. M., Flickinger, D., Oepen, S., Walsh, A., and Baldwin, T. (2004). Arboretum: Using a precision grammar for grammar checking in CALL. In *InSTIL/ICALL Symposium 2004*.

Berzak, Y., Kenney, J., Spadine, C., Wang, J. X., Lam, L., Mori, K. S., Garza, S., and Katz, B. (2016). Universal Dependencies for learner English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 737–746, Berlin, Germany, August. Association for Computational Linguistics.

Black, E., Abney, S., Flickinger, D., Gdaniec, C., Grishman, R., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J. L., et al. (1991). A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*.

Bond, F., Fujita, S., Hashimoto, C., Kasahara, K., Nariyama, S., Nichols, E., Ohtani, A., Tanaka, T., and Amano, S. (2004). The Hinoki treebank. a treebank for text understanding. In *International Conference on Natural Language Processing*, pages 158–167. Springer.

Bond, F., Fujita, S., and Tanaka, T. (2008). The Hinoki syntactic and semantic treebank of Japanese. *Language Resources and Evaluation*, 42(2):243–251.

Collins, M. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of ACL '98/EACL '98*, page 16–23, USA. Association for Computational Linguistics.

Copestake, A. and Flickinger, D. (2000). An open source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second Linguistic Resources and Evaluation Conference*, pages 591–600, Athens, Greece.

Copestake, A., Flickinger, D., Pollard, C., and Sag, I. A. (2005). Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332.

Dahlmeier, D., Ng, H. T., and Wu, S. M. (2013). Building a large annotated corpus of learner English: The NUS corpus of learner English. In *BEA@ NAACL-HLT*, pages 22–31.

Flickinger, D. and Yu, J. (2013). Toward more precision in correction of grammatical errors. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 68–73.

Flickinger, D., Zhang, Y., and Kordoni, V. (2012). Deepbank. a dynamically annotated treebank of the wall street journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories*, pages 85–96.

Flickinger, D., Goodman, M., and Packard, W. (2016). UW-Stanford System description for AESW 2016 shared task on grammatical error detection. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 105–111.

Flickinger, D. (2000). On building a more effcient grammar by exploiting types. *Natural Language Engineering (Special Issue on Efficient Processing with HPSG)*, 6(1):15–28.

Flickinger, D. (2011). Accuracy vs. robustness in grammar engineering. *Language from a cognitive perspective: Grammar, usage, and processing*, 201:31–50.

Krieger, H.-U. and Schäfer, U. (1994). TDL – a type description language for constraint-based grammars. *arXiv preprint cmp-lg/9406018*.

Moeljadi, D., Bond, F., and Song, S. (2015). Building an HPSG-based Indonesian resource grammar (indra). *ACL-IJCNLP 2015*, page 9.

Moeljadi, D. (2017). Building jati: A treebank for indonesian. In *Proceedings of The 4th Atma Jaya Conference on Corpus Studies (ConCorps 4)*, pages 1–9.

Morgado da Costa, L., Winder, R. V. P., Li, S. Y., Liang, B. C. L. T., Mackinnon, J., and Bond, F. (2020). Automated writing support using deep linguistic parsers. In *Proceedings of the 12th Conference on Language Resources and Evaluation*, Marseille, France, may. European Language Resources Association (ELRA).

Nicholls, D. (2003). The Cambridge learner corpus: Error coding and analysis for lexicography and elt. In *Proceedings of the Corpus Linguistics 2003 conference*, volume 16, pages 572–581.

Oepen, S., Toutanova, K., Shieber, S. M., Manning, C. D., Flickinger, D., and Brants, T. (2002). The LinGO Redwoods treebank: Motivation and preliminary applications. In *COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes*.

Oepen, S., Flickinger, D., and Bond, F. (2004). Towards holistic grammar engineering and testing — grafting treebank maintenance into the grammar revision cycle. In *Beyond Shallow Analyses — Formalisms and Statistical Modelling for Deep Analysis (Workshop at IJCNLP-2004)*, Hainan Island.

Packard, W. (2015). Full Forest Treebanking. Master's thesis, University of Washington, USA.

Pollard, C. and Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar*. The University of Chicago Press.

Ragheb, M. and Dickinson, M. (2012). Defining syntax for learner language annotation. In *Proceedings of COLING 2012: Posters*, pages 965–974, Mumbai, India, December.

Ragheb, M. and Dickinson, M. (2014). Developing a corpus of syntactically-annotated learner language for English. In *Proceedings of the 13th Interna-*

tional Workshop on Treebanks and Linguistic Theories (TLT13), pages 292–300, Tübingen, Germany.

Sag, I. A., Wasow, T., Bender, E. M., and Sag, I. A. (1999). *Syntactic theory: a formal introduction*, volume 2. CSLI Stanford.

Schneider, D. and McCoy, K. F. (1998). Recognizing syntactic errors in the writing of second language learners. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 2*, pages 1198–1204, Stroudsburg, PA, USA. Association for Computational Linguistics.

Siegel, M. and Bender, E. M. (2002). Efficient deep processing of Japanese. In *Proceedings of the 3rd workshop on Asian language resources and international standardization-Volume 12*, pages 1–8. Association for Computational Linguistics.

Siegel, M., Bender, E. M., and Bond, F. (2016). *Jacy: An implemented grammar of Japanese*. CSLI Publications.

Suppes, P., Liang, T., Macken, E. E., and Flickinger, D. P. (2014). Positive technological and negative pre-test-score effects in a four-year assessment of low socioeconomic status k-8 student learning in computer-based math and language arts courses. *Computers & Education*, 71:23–32.

Tanaka, T., Bond, F., Oepen, S., and Fujita, S. (2005). High precision treebanking — blazing useful trees using POS information. In *Proceedings of the 43rd annual meeting of the Association for Computational Linguistics*, pages 330–337.

Toutanova, K., Manning, C. D., Flickinger, D., and Oepen, S. (2005). Stochastic HPSG parse disambiguation using the Redwoods corpus. *Research on Language and Computation*, 3(1):83–105.

Winder, R. V. P., MacKinnon, J., Li, S. Y., Lin, B., Heah, C., Morgado da Costa, L., Kuribayashi, T., and Bond, F. (2017). NTUCLE: Developing a corpus of learner English to provide writing support for engineering students. In *Proceedings of the 4th Workshop on NLP Techniques for Educational Applications (NLPTEA 2017)*, Taipei, Taiwan.

Zhang, Y. and Krieger, H.-U. (2011). Large-scale corpus-driven PCFG approximation of an HPSG. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 198–208, Dublin, Ireland, October. Association for Computational Linguistics.