

Master Thesis

Automated Verb Order Error Detection for Learners of Dutch as a Second Language

Noah-Manuel Michael

*a thesis submitted in partial fulfilment of the
requirements for the degree of*

MA Linguistics

(Text Mining)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities



Supervised by: Dr. Lisa Beinborn, Dr. Camille Welie
2nd reader: Dr. Luís Guilherme de Passos Morgado da Costa

Submitted: August 15, 2023

Abstract

Correct verb placement in Dutch can be difficult to acquire for second-language learners because it depends on a variety of factors such as the (non-)finiteness of the verb or the clause type it appears in. This leads to verb order being a common source of errors. Within the natural language processing community, automated grammatical error correction is a task that has received continuous attention. Yet, word order errors are severely underrepresented in all benchmark datasets. This leaves the potential performance of natural language processing models at detecting word and verb order errors essentially unexplored. In the case of transformer-based models, which have become the current de facto standard for solving a variety of natural language processing tasks, their ability to reliably solve tasks that require syntactic understanding remains an open research question. In this thesis, I test how different natural language processing model architectures can be used for the detection of word order errors. By comparing the performance of a classifier that has access to syntactic information to the performance of a classifier that does not, I can show that syntactic information plays a vital role in the detection of word order errors. Transformer-based classifiers unanimously exhibit almost perfect performance scores in the detection of generic word and generic verb order errors, trained and tested on synthetic datasets. However, depending on the model architecture and the method of pseudo data generation, their general capability in detecting erroneous word order translates differently to the detection of learner-informed verb order errors, i.e., verb order errors that learners are likely to make. The best model is a GPT-2-based classifier trained on a pseudo dataset consisting of both correct Dutch sentences and random permutations of the same sentences. Despite being trained on randomly permuted data, the model appears to have learned to identify learner-informed verb order errors while its performance at detecting generic verb order errors is significantly lower.


Keywords: automated grammatical error correction – detection of word and verb order errors – Dutch – transformer models – pseudo data generation

Declaration of Authorship

I, Noah-Manuel Michael, declare that this thesis, titled *Automated Verb Order Error Detection for Learners of Dutch as a Second Language* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Master of Arts degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: August 15, 2023

Signed: 

Acknowledgements

An diesem Punkt möchte ich mich vor allem einmal herzlich bei meinen Eltern bedanken, deren permanente Unterstützung es mir erlaubt hat, meine bisherige Laufbahn nach meinen Wünschen zu gestalten.

Ik wil mijn partner bedanken die bijna de hele master al aan mijn zijde staat. Hij heeft me begeleid in een intensieve tijd waar ik veel over mezelf ben komen te leren.

I want to thank all the people that embarked on this exciting and intensive journey with me in September last year, not entirely sure what to expect but always cheering each other on whenever the program turned out to get really busy.

Ich möchte mich auch bei meinen Freunden zu Hause bedanken, die mir trotz Perioden längerer Abwesenheit nie fremd geworden sind.


And I want to thank all the friends I have made along the way who have made the journey of studying truly an adventure. I am excited to see what is to come next.

I want to thank the CLTL department and its staff for the high-quality education we were allowed to experience. I feel that the program allowed me to learn a lot and I feel well-prepared for what is ahead of me.

I would like to thank Dr. Camille Welie, my co-supervisor, for his input about word order in Dutch and the idea of the project.

And finally, I would like to thank my supervisor, Dr. Lisa Beinborn, for her continuous support throughout the process of writing this thesis. Her valuable input has enabled me to choose a focus for this work that I am really interested in. She played a vital role in the success of this project.

A special thanks to Jeremias Graf and Irma Tuinenga for thoroughly proofreading this thesis in its final form and their suggestions for stylistic improvements.

In Gedenken an die kleine Amy ... 

Contents

Abstract	i
Declaration of Authorship	ii
Acknowledgements	iii
List of Figures	vi
List of Tables	vii
1 Introduction	1
2 Related Work	5
2.1 Word Order Transfer Errors	5
2.2 Automated Grammatical Error Correction	6
2.3 Pseudo Data for Grammatical Error Correction	7
2.4 Part-of-Speech Taggers	8
2.5 Syntactic Parsers	8
2.6 Transformer Models	9
2.6.1 Model Architecture	10
2.6.2 Word Order Information in Transformer-Based Models	11
3 Detection of Generic Word and Verb Order Errors	15
3.1 Generation of Pseudo Data: Datasets RAND and VERBS	15
3.1.1 Shuffle Methods	16
3.1.2 Seed Corpora	20
3.2 Standard Evaluation Metrics	23
3.3 Models	24
3.3.1 Part-of-Speech Tagger	24
3.3.2 Syntactic Parser	24
3.3.3 Transformer Models	25
3.4 Experimental Configurations	26
3.4.1 PARSE LOOKUP	26
3.4.2 POS CLASSIFIER	29
3.4.3 PARSE CLASSIFIER	30
3.4.4 TRANSFORMER CLASSIFIER	30
3.5 Results	31
3.6 Discussion	32
3.6.1 PARSE LOOKUP	32

3.6.2	POS CLASSIFIER	34
3.6.3	PARSE CLASSIFIER	35
3.6.4	TRANSFORMER CLASSIFIER	36
4	Detection of Learner-Informed Verb Order Errors	39
4.1	Generation of Learner-Informed Pseudo Data: Dataset INFO	39
4.1.1	Dutch Verb Order	40
4.1.2	Generating Target Hypotheses	43
4.1.3	Analyzing Verb Order Errors	44
4.1.4	LEUVEN corpus	46
4.1.5	Identified Error Tendencies	48
4.1.6	Curation of Evaluation Dataset INFO	51
4.2	Evaluation Metric: $F_{0.5}$ Score	54
4.3	Results	55
4.4	Discussion	56
4.4.1	PARSE LOOKUP	57
4.4.2	POS AND PARSE CLASSIFIER	57
4.4.3	TRANSFORMER CLASSIFIER	57
4.4.4	Opportunities for Future Research	59
5	Outlook: Generative Artificial Intelligence Models as Virtual Teachers	61
6	Conclusion	63
A	Abbreviations and Symbols	65
B	Illustration Verbs Shuffles	66
C	Categories for Phrasal Analysis	67
D	Results Overview	68
E	Data Statements	72
E.1	KU Leuven - Instituut voor Levende Talen - Leerdercorpus	72
E.2	Synthetic Dataset	73
F	Annotation Prompt	74
G	Generative AI Statement	75
H	Licenses	76
H.1	LASSY License	76
H.2	WAI-NOT License	80

List of Figures

3.1	Distribution of sentence length in the LEUVEN corpus	22
3.2	Tree A	26
3.3	Tree B	26
3.4	Tree A	27
3.5	Tree C	27
3.6	Tree A SIMPLE	27
3.7	Tree C SIMPLE	27
3.8	SPACY-TUP tuple format	29
3.9	DOP-TUP-ORIG tuple format	30
3.10	DOP-TUP-SIMPLE tuple format	30
4.1	Distribution of first languages in selected learner sentences	47
4.2	Distribution of CEFR levels in selected learner sentences	47
4.3	Distribution of errors per verb / complement type	48
4.4	Distribution of finite verb error types	48
4.5	Distribution of incorrect positions of finite verb in main and subordinate clauses	49
4.6	Recreation of a clause-final before NFV error in main clause	53
4.7	Recreation of a clause-internal after SUBJECT error in subordinate clause	53
4.8	F_1 scores per model and test dataset	55
5.1	Distribution of accepted and challenged initial target hypotheses	61

List of Tables

2.1	Share of word order errors in shared tasks on grammatical error correction	6
3.1	Seed corpora for pseudo datasets	20
3.2	Train/Test data split	22
3.3	Effects of tree simplification	28
3.4	PARSE LOOKUP models	28
3.5	POS CLASSIFIER model	29
3.6	PARSE CLASSIFIER models	30
3.7	TRANSFORMER CLASSIFIER models	31
3.8	Average F_1 score of all models on Test RAND and Test VERBS	31
3.9	Confusion matrix DOP-TREE-ORIG on Test RAND	33
3.10	Confusion matrix DOP-TREE-ORIG on Test VERBS	33
3.11	Confusion matrix DOP-TREE-SIMPLE on Test RAND	33
3.12	Confusion matrix DOP-TREE-SIMPLE on Test VERBS	33
3.13	Confusion matrix SPACY-TUP Train RAND on Test VERBS	34
3.14	Confusion matrix SPACY-TUP Train VERBS on Test VERBS	34
3.15	Confusion matrix DOP-TUP-ORIG Train RAND on Test RAND	35
3.16	Confusion matrix DOP-TUP-SIMPLE Train RAND on Test RAND	35
3.17	Best confusion matrix ROBBERT Train RAND on Test RAND	36
3.18	Best confusion matrix ROBBERT Train VERBS on Test VERBS	36
3.19	Illustration of mislabeled VERBS permutations	38
4.1	Default positions per clause and verb/complement type	43
4.2	Clause type error distribution in INFO	51
4.3	Distribution of error position types in main and subordinate clauses in INFO	52
4.4	Average $F_{0.5}$ score of all models on Test INFO and Recall score of all models on Test LEARN	56
4.5	Best predictions of GPT-2 RAND per incorrect position in INFO	58
C.1	Symbols used for analysis of clause structure	67
D.1	Combined Results of all Experiments	68

Chapter 1

Introduction

Ik wil geen boeken kopen.
I want no books buy.

If English followed the syntax of Dutch, a simple sentence such as “I don’t want to buy books.” would already look considerably different from its current form. A key factor in this is the different positions verbs can occupy in a Dutch sentence. Verbs in Dutch can typically either appear in the second or in the final position in the clause, depending on conditions such as the type of the clause they occur in or whether they are (non-)finite. In the example above, both verb positions are filled. Due to the multitude of factors that can influence the correct placement of verbs in Dutch, verb placement is often challenging for second-language learners.

This can result in transfer errors, where learners of Dutch as a second language misplace verbs in the Dutch sentences they produce based on how they would place verbs in their first language. Similarly to how the pseudo-English sentence in the example above without context would be rather difficult to understand for an English-speaking person, Dutch sentences with verb order errors can lead to challenging situations in communication with Dutch speakers. As verb order is an error-prone area within Dutch grammar (Jordens, 1988; Verhagen, 2011), this offers the opportunity for examining how natural language processing models can be leveraged to help learners of Dutch as a second language detect the verb order errors they make in writing.

In the context of this task, I want to explore the performance of four classification approaches based on different natural language processing model architectures: a lookup approach based on the output of a syntactic parser, a classifier with access to the output of a part-of-speech tagger, classifiers with access to the output of a syntactic parser, and classifiers based on the output of transformer models. By comparing the performance of classification approaches that have access to syntactic information versus those that do not, I can determine whether syntactic information is a vital component in the detection of word order errors. This leads me to the first research question this thesis will focus on:

To what degree is syntactic information helpful for the detection of word order errors?

This question is particularly interesting for the models based on the transformer architecture because their (in)ability to reliably solve tasks that require syntactic understanding remains an open research question. Transformers model natural language by

representing it in the form of embeddings. Embeddings are high-dimensional vector representations of tokens, where tokens can be equivalent to words or smaller units (subwords). These embeddings are trained on large amounts of data and manage to capture contextual information about the tokens they represent. By mathematically manipulating these embeddings depending on the desired output, transformer models can be trained to perform a variety of different natural language processing tasks. The uncertainty as to whether these models are able to effectively represent and access syntactic information leads me to the second research question of this thesis:

Are transformer-based classifiers able to reliably detect word and verb order errors?

One of the key challenges in investigating these questions, however, is the lack of data annotated for word order errors. Within the natural language processing community, the task of automated grammatical error correction is not unheard of. Yet, benchmark datasets for the task often only contain a small fraction of word order errors, of which verb order errors are a subset, if any at all. This leaves the targeted investigation of how to efficiently and reliably detect word order errors essentially unexplored. Being able to inform a learner about the presence of word order errors in their writing can, however, be beneficial as it can raise awareness for these types of errors amongst the learners. This would be especially useful for learners of Dutch, where the acquisition of correct verb placement is particularly difficult.

Another aggravating factor is the general scarcity of genuine learner data. Thus, creating an entirely new dataset for the detection of word and verb order errors is not trivial either. To the best of my knowledge, at this point in time, genuine Dutch learner data professionally annotated for word order errors does not exist. Therefore, I opt to base the training of the classification approaches I am going to explore on synthetic data, or pseudo data.¹ I propose two different methods for the generation of pseudo data based on correct Dutch sentences sourced from a variety of corpora: randomly permuting the positions of all tokens within a sentence (equal to generic word order errors) and randomly permuting the positions of only the verb tokens within a sentence (equal to generic verb order errors). This creates an opportunity for me to explore whether it is possible to train classifiers for the detection of generic word and verb order errors on pseudo data exclusively, which leads me to the third research question that this thesis is concerned with:

Can pseudo data successfully be leveraged to train classifiers for the detection of generic word and verb order errors?

Once the general capability of the different classification approaches in detecting erroneous word order is established, I additionally test the performance of the proposed models on a final synthetic evaluation dataset that approximates genuine learner data by including only learner-informed verb order errors, i.e., common error types I extract by means of structural analysis of genuine learner data. The genuine learner sentences analyzed serve as an additional means of evaluation. This facilitates the fourth research question this thesis aims to investigate:

Do the models' performance scores at the detection of generic word and verb order errors translate to the detection of learner-informed verb order errors?

¹Following Kiyono et al. (2019), I will henceforth use the terms synthetic data and pseudo data interchangeably. Genuine data describes authentic learner data.

Finally, the two different methods of pseudo data generation allow for investigating how training on generic word order errors impacts model performance compared to training on generic verb order errors. As this thesis focuses on the detection of verb order errors for learners of Dutch, all of the errors introduced to the learner-informed evaluation dataset are verb order errors. Intuitively, classifiers trained on generic verb order errors should exhibit a higher performance on learner-informed verb order errors than classifiers trained on generic word order errors, where all tokens can change their positions randomly. This leads to the fifth and final research question this thesis concerns itself with:

How does the method of pseudo data generation influence the performance
of the models?

Outline. In order to address these questions, in the following chapter, I want to lay the theoretical foundations by briefly introducing the concept of word order transfer errors, the task of automated grammatical error correction, and how pseudo data has successfully been used in the training of grammatical error correction models. Additionally, I introduce the basic architectures of the natural language processing model families I explore in the context of the experiments I conduct, and I briefly summarize the current state of knowledge concerning transformer models and their (in)ability to represent, process, and access word order information for solving natural language processing tasks (Section 2. Related Work).²

After introducing the necessary background information, I dedicate the first part of this thesis to the detection of generic word and verb order errors. I formally define the two proposed methods of pseudo data generation and introduce the seed corpora that serve as a source for the correct Dutch sentences to be permuted. Subsequently, I briefly introduce the evaluation metric according to which I judge the models' general capability in detecting erroneous word order before introducing the specific natural language processing models and the classification approaches I build upon their basic architecture. Once the experimental configurations have been established, I present the results and reflect on the performance of each of the classification approaches (Section 3. Detection of Generic Word and Verb Order Errors).

The second part of this thesis focuses on the detection of learner-informed verb order errors. I begin by introducing the most common positions verbs can occupy in Dutch sentences and link these positions to error types. I introduce the concept of target hypotheses, i.e., hypotheses about what the learner intended to express with a given sentence as they are important for the analysis of learner data. I then proceed to explain how I perform a structural analysis of a selection of genuine learner sentences taken from the LEUVEN corpus of genuine learner data. I present the error tendencies that result from the analysis and go on to explain how I create a learner-informed evaluation dataset that tries to approximate genuine learner data as closely as possible. Before presenting and discussing the results the established classification approaches are able to achieve on the learner-informed evaluation dataset and the selected learner sentences, I introduce an additional evaluation metric that attributes more importance to the precision of the classifier. This is important for in a real-life application scenario, high precision is crucial in order to not discourage learners. Lastly, I briefly summarize the limitations of this work (Section 4. Detection of Learner-Informed Verb Order

²I use the terms *word order information* and *syntactic information* interchangeably.

Errors).

With generative artificial intelligence models recently gaining more and more public recognition, before concluding this thesis, I briefly illustrate their potential as readily available end-to-end solutions for automated grammatical error correction (Section 5. Outlook: Generative Artificial Intelligence Models as Virtual Teachers).

Finally, I conclude this thesis by summarizing and contextualizing its main findings (Section 6. Conclusion).

Chapter 2

Related Work

I begin this chapter by introducing the concept of word order transfer errors, which are errors that result from transferring syntactic patterns of one of the languages a learner of a certain language already speaks into the language they are currently learning, i.e., their target language. An ungrammatical sentence such as the opening sentence of this thesis can often be their result. I then introduce the natural language processing task of automated grammatical error correction, which is typically evaluated on benchmark datasets that treat grammatical error correction as a holistic task, focusing on the correction of all or at least a variety of potential errors a learner could make. This has resulted in word order errors being severely underrepresented in those datasets. Verb order in particular has not yet been targeted explicitly, although isolating it and providing feedback about it to the learner is valuable since verb order in Dutch is a common source of errors. My thesis focuses on the detection of these word and verb order errors as the first step in the correction and feedback generation pipeline. I also briefly explain how synthetic data, or pseudo data, has been used successfully in grammatical error correction as I create pseudo datasets to train and evaluate classifiers in the detection of word and verb order errors. Additionally, I introduce the basic architectures of the natural language processing models that I explore in my experiments: part-of-speech taggers, syntactic parsers, and transformer models. Finally, I provide a brief overview of why it is unclear whether transformer-based models have access to word order information and make use of it for solving natural language processing tasks that require such information to be solved.

2.1 Word Order Transfer Errors

Transfer errors in general are a very common phenomenon that can begin with learners of second languages having an accent when speaking in their target language. A first language-based accent is a neat example of a transfer error because the source of the error is immediately apparent – the accent is typically associated with the learner’s first language; parts of the phonology of the first language are carried over into the second language speech.

With other types of transfer errors, the source of the error may not be as immediately apparent. One such case is word order errors, which can result from transferring syntactic structures from one language into another.³ However, just like accents, de-

³Note that word order errors can also occur as a result of other phenomena unrelated to transfer.

pending on their severity, word order errors can negatively impact the communication experience, and being able to form syntactically correct sentences is a desirable skill to master when trying to acquire a second language. Therefore, reliably detecting word order errors, and in the case of Dutch, verb order errors in particular, is a natural language processing task worth exploring.

There are a number of publications concerned with word order acquisition in general and in Dutch specifically. Jordens (1988) discusses the acquisition of word order in Dutch and German as first languages versus as second languages. He groups both languages together as their verb order is rather similar: Both languages exhibit both SOV and V2 syntax patterns,⁴ illustrating the fact that verb order in Dutch seems to be difficult to master. Schepens (2015) finds that the linguistic distance of a learner’s first language can have an impact on the learnability of Dutch. Finally, Erdocia and Laka (2018, p. 8) find that the first language can influence the processing of word order in the second language: “We observed that when L1 and L2 differ, their cues compete, resulting in a negative transfer from L1.”⁵ Knowing that word and verb order can be sources of grammatical errors, I will now introduce what has already been done in the domain of automated grammatical error correction.

2.2 Automated Grammatical Error Correction

Automated grammatical error correction is a natural language processing task that has been researched for more than two decades now. Automated word order error detection and automated word order error correction, on the other hand, are subtasks that until today have received much less attention. According to Grundkiewicz et al. (2020), who provide an informative tutorial on automated grammatical error correction, interest in the task became widespread with the advent of a number of shared tasks in recent years, as illustrated in Table 2.1:

Table 2.1: Share of word order errors in shared tasks on grammatical error correction

Shared Task	Reference	% WOE
HOO	Dale and Kilgarriff (2011)	< 7.5%
CoNLL	Ng et al. (2013)	0.0%
CoNLL	Ng et al. (2014)	2.4%
BEA	Bryant et al. (2019)	1.6%

WOE – *Word order errors*

Yet, as the table also illustrates, word order errors only made up a subset of a category of errors labeled *Other* in the HOO shared task,⁶ which itself only made up about 7.5% of all the errors in the data used in the shared task. Moreover, the CoNLL-2013 shared task does not concern itself with word order errors at all,⁷ and in the CoNLL-2014

⁴SOV: Subject-Object-Verb. V2: Verb-second. V2 languages require verbs to occupy the second position in a clause or sentence under certain conditions.

⁵L1: First language. L2: Second language.

⁶HOO: Helping Our Own.

⁷CoNLL: Conference on Computational Natural Language Learning.

shared task, their share within the whole dataset amounts to a mere 2.4%. This is aggravated by the fact that by definition, word order errors must be a *span correction problem* as opposed to the vast majority of single token correction problems within the shared tasks. The term span correction problem refers to errors that stretch over more than a single token. In a phrase such as **de man die ken goed ik* ‘the man that I know well’, there is no single token that could be identified as erroneous – the error stretches over the span **ken goed ik*, which should be *ik goed ken* ‘I know well’. None of the tokens in the incorrect version are in their correct relative position to the other tokens. By contrast, in a phrase such as **de man die ik goed kent*, the erroneous token is *kent* ‘knows’ (3SG),⁸ which should be *ken* ‘know’ (1SG).

With its negligible impact on the overall scoring of the participating systems and its difficulty for being a span correction problem, this situation has resulted in an almost complete disregard of word order errors by the participating teams. Out of the 13 participating systems in the 2014 shared task, only four managed to achieve any recall at all on the underrepresented word order errors. Furthermore, the above-mentioned shared tasks were organized before the popularization of deep neural networks, with transformer models being a subtype. Due to their promising performances, they have become the current de facto standard for solving a variety of complex natural language processing tasks. Thus, the shared tasks’ solutions most likely have lost their state-of-the-art status by now. Even with the most recent shared task on grammatical error correction, the BEA-2019 shared task,⁹ when transformer architectures had already been available, word order errors only made up 1.6% of all errors in the dataset. This essentially leaves the subtasks of automated word order error detection and automated word order error correction as well as the currently popular language models’ potential performances on them unexplored. This thesis will focus on the detection of word order errors.

2.3 Pseudo Data for Grammatical Error Correction

Studies have shown that incorporating pseudo data into the training process of machine learning models can significantly increase performance at the task of grammatical error correction (Kiyono et al., 2019; Xu et al., 2019). This is important as there is no genuine Dutch data available that is annotated for word and verb order errors. While Xu et al. (2019) do incorporate *transposition errors* into their dataset, which make certain tokens change their absolute position with one of their adjacent tokens, these errors can hardly represent all of the different word order errors learners of a language are likely to make. Kiyono et al. (2019) find that both the source corpus for the synthetic data and the method of generating the data can influence model performance. Both teams test their solutions primarily on the benchmark datasets introduced earlier (e.g., CoNLL-2014, BEA-2019). However, even though Xu et al. (2019) include a certain type of word order error during the training process, evaluating on these datasets suggests that word order errors are underrepresented in the evaluation.

Both teams treat grammatical error correction as a holistic task, i.e., they do not explicitly focus on isolating certain error types. In this thesis, I want to take a different approach by isolating single capabilities a grammatical error correction model should

⁸When providing morphological information, I follow the categories and notation proposed in the Leipzig Glossing Rules (Comrie et al., 2008).

⁹BEA: Building Educational Applications.

possess (detecting erroneous word order and detecting erroneous verb order), which is a parallel that my synthetic datasets share with challenge datasets (Ribeiro et al., 2020). Creating pseudo datasets that differ only in the type of errors present within them allows for isolating the behavior of the tested models under different conditions. I then evaluate how well different classifiers based on different natural language processing model architectures perform on these two tasks, which in the case of the transformer models allows me to draw insights about whether transformer models can solve tasks that require an understanding of word and verb order information to be solved.

When describing the properties of my datasets, I will mostly adhere to the terminology used by Kiyono et al. (2019). After exploring the models’ performance at the detection of generic word and generic verb order errors, I also generate a learner-informed synthetic dataset that imitates error tendencies I extract from genuine learner data. I will now introduce the basic model architectures I use in my experiments, i.e., part-of-speech taggers, syntactic parsers, and transformer models, and I will briefly explain why there is uncertainty concerning the transformer models’ ability to solve tasks that require an understanding of word order information.

2.4 Part-of-Speech Taggers

Automated part-of-speech tagging is a natural language processing task that traditionally forms the basis of many other tasks that require linguistic analysis of the input, one of which is syntactic parsing. Its objective is to assign each token of an input sequence a corresponding part-of-speech tag (Jurafsky and Martin, 2021, pp. 163–164):

Tagging is a disambiguation task; words are ambiguous—have more than one possible part-of-speech—and the goal is to find the correct tag for the situation. For example, *book* can be a verb (*book that flight*) or a noun (*hand me that book*). [...] The goal of POS-tagging is to resolve these ambiguities, choosing the proper tag for the context.¹⁰

The resulting output can then serve as a basis for other linguistic analyses. Over the years, part-of-speech tagging has developed from being performed manually to being performed automatically by means of rule-based systems, statistical systems, and finally neural network-based approaches (Chiche and Yitagesu, 2022). As part-of-speech tagging is traditionally performed as an early step in the syntactic parsing pipeline (Jurafsky and Martin, 2021), in my experiments I illustrate the difference in performance of classification approaches based on the output of a mere part-of-speech tagger and a syntactic parser. The following section introduces the general architecture of syntactic parsers.

2.5 Syntactic Parsers

In automated grammatical error correction, syntactic parsers have been used to solve a variety of tasks, among which the correction of spelling and agreement errors (Wagner et al., 2007), determiner selection (Turner and Charniak, 2007; Rozovskaya and Roth, 2010), and the correction of verb forms (Lee and Seneff, 2008). However, to the best of my knowledge, they have not yet been thoroughly explored in the context of word and

¹⁰POS: Part-of-speech.

verb order error detection. Syntactic parsers are traditional natural language processing architectures that are used to represent the syntax of an input sequence, usually in the form of hierarchical tree structures. There are two main types of parsing for syntactic analysis: constituency and dependency parsing (Zhang, 2020).

Constituency parsers organize the input they receive into different, labeled *constituents*, i.e., groupings of words that are grammatically related in some way, often by agreement, and usually fulfill a single function in a sentence. The noun phrase *het kleine meisje* ‘the small girl’ in the sentence *Het kleine meisje ging naar huis*. “The small girl went home.”, for example, serves as the subject of the sentence. Early constituent parsers often make use of *context-free grammars* (Jurafsky and Martin, 2021, p. 262):

A context-free grammar consists of a set of rules or productions, each of which expresses the ways that symbols of the language can be grouped and ordered together, and a lexicon of words and symbols.

Later models, according to Zhang (2020), often incorporate statistical information in addition to a grammar’s production rules, and neural network-based constituency parsers have redefined previous performance limits. The DISCO-DOP parser I will introduce in the context of my experimental setup in Section 3.3.2 is a constituency parser that, among others, uses a statistically enriched version of context-free grammar: probabilistic context-free grammar, which allows for the calculation of the likelihood of a parse of a sentence.

Dependency parsers, on the other hand, make use of dependency grammars (Zhang, 2020, pp. 1–2): Here, “[...] words are directly connected by dependency links, with labels indicating their syntactic or semantic relations.” However, groupings of words (i.e., constituents or phrases) are typically not labeled, which makes it harder to summarize and abstract these elements. As I am focusing on verb-related word order, simplifying verb-unrelated constituents can potentially help machine learning systems focus on the relative positions of verb tokens in relation to verb-unrelated constituents. This means that dependency parsers are less suitable for the classification approaches I explore in my experiments. However, I make use of the SPACY dependency parser for various preprocessing tasks (Section 3.1.2) and when creating the learner-informed evaluation dataset as will be explained in Section 4.1.6.

2.6 Transformer Models

Transformer models, as opposed to syntactic parsers, are large language models that capture more than just syntactic information. They represent language, i.e., input tokens, in the form of embeddings, or high-dimensional vector representations. By mathematically modifying these vector representations based on learned parameters during their pre-training and fine-tuning, they are able to model natural language. Due to their high dimensionality, the behavior of transformer models is difficult to analyze, and it is not in fact clear whether these models are able to reliably solve tasks that need an understanding of word order information. Therefore, in the following sections, I will briefly introduce the general transformer architecture as well as a number of studies that investigate the representation and processing of word order information in transformer models.

2.6.1 Model Architecture

Transformer models are variants of deep artificial neural networks.¹¹ Neural networks are a machine learning architecture that is inspired by biological neural networks in that vectorized inputs are mathematically modified when passed through layers of interconnected “neurons”. A neural network can be described as deep when in its architecture, multiple of these neural layers are stacked upon one another (Goodfellow et al., 2016). Transformer models are special in that they typically make use of encoder and decoder architectures, as well as the so-called *attention mechanism*.¹² Vaswani et al. (2017, p. 3), who are the first to introduce the transformer architecture, describe attention as follows:¹³

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

Essentially, the decoder can obtain information about which parts of the input to focus on when generating the output by computing a query. The query is used to compute attention weights over the encoder’s outputs, which consist of a set of key-value pairs. Each key in this set is compared with the query to calculate a compatibility score, which measures the relevance of the respective part of the input to the current step of the output generation. The model then pays most attention to the parts of the input that exhibit the highest compatibility scores when generating the output for the current step of the output generation, i.e., it assigns higher weights to the values of the input components that are determined to be more relevant.

In this thesis, I will be exploring the performance of transformer models based on the following architectures:

1. Bidirectional Encoder Representations from Transformers (BERT)
2. Robustly Optimized BERT Pretraining Approach (RoBERTa)
3. Generative Pre-trained Transformer 2 (GPT-2)

BERT is a transformer model that only makes use of the encoder architecture. During its pre-training, where it is tasked to both fill masked tokens (masked language modeling) and predict whether, in a pair of sentences, the second sentence is likely to follow the first sentence (next sentence prediction), it is trained on very large amounts of data (Devlin et al., 2019). By constantly adjusting its parameters during the pre-training process to generate more reliable predictions, the model is able to effectively model natural language.¹⁴ RoBERTa uses the same basic architecture as BERT but optimizes the

¹¹I will henceforth refer to artificial neural networks as neural networks.

¹²An encoder maps an input sequence to a vector of fixed dimensionality (Sutskever et al., 2014). The input vector is then modified as it is passed through the hidden layers of the neural network. Finally, a decoder maps the output to the desired output sequence (Sutskever et al., 2014).

¹³Vaswani et al. (2017) introduce the transformer architecture. Attention mechanisms, however, had already been commonly in use. The innovation in their approach lies in transformer models making use of multi-head attention exclusively, i.e., the attention mechanism in the transformer architecture is not combined with other architectures for generating the desired output sequence (Vaswani et al., 2017).

¹⁴The original BERT model was trained on English data.

pre-training process by, for example, training on additional data, increasing the amount of training time, fine-tuning hyperparameters, and, most notably, dropping the next sentence prediction pre-training task (Liu et al., 2019). RoBERTa is hence pre-trained on the masked language modeling task only. Finally, GPT-2 differs from BERT in that it only uses the decoder architecture, it is not bidirectional but processes sequences from left to right, and its pre-training objective is next-word prediction (Radford et al., 2019), which makes it especially well-suited for generative natural language processing tasks.

In my experiments, I compare the performance of all of the models on the discriminative task that is word order error detection. The models I use in my experiments have been specifically adapted for Dutch, as I will explain in Section 3.3.3. Due to the depth of transformer models, it is difficult to analyze and explain what types of information the models learn to represent during their pre-training. Therefore, it is an open research question whether transformer models are able to reliably solve tasks that require an understanding of word order information. In the following section, I briefly summarize the current state of research on the ways transformer models encode and interpret sequences of words with a focus on positional, i.e., word order information.

2.6.2 Word Order Information in Transformer-Based Models

With the basic architecture and mechanisms of transformer-based models explained, I now want to draw attention to how transformer models process word order. Most of the studies that try to investigate this question do so by testing the models' behavior and the impact ablation studies have on it.

O'Connor and Andreas (2021) investigate the impact augmenting word order has on *usable information* in long-range contexts,¹⁵ i.e., in parts of the input sequence that are relatively far away from the target. They find that shuffling sentences within these long-range contexts while preserving the internal word order of the sentences only has a very moderate effect on the usable information. Likewise, shuffling trigrams within sentences while preserving the order of the sentences, and shuffling word order within trigrams while preserving the order of the trigrams show a similarly low impact on the usable information. Randomly shuffling the word order within the whole long-range context, however, causes a more significant drop in performance. Thus, they conclude that usable information in long-range contexts is mostly contained in content words and local ordering statistics. Nonetheless, even transformations of the type *man bites dog* → *dog bites man* that have significant implications for an utterance's semantic content hardly influence the model's performance if they occur in long-range contexts. Therefore, while word order does have an impact on model performance in contexts close to the target, they find this impact to be significantly smaller in long-range contexts, as long as local ordering is generally preserved: “[P]rediction accuracy depends on information about local co-occurrence, but not fine-grained word order or global position” (O'Connor and Andreas, 2021, p. 855).

¹⁵Usable information refers to information that a language model can use in order to make more accurate predictions. O'Connor and Andreas (2021) explain that it is well established that long-range contexts provide additional helpful information to language models. They train a model that has access to long-range contexts, a model that does not, and several models that have access to long-range contexts that have been augmented in some way. By measuring the differences in additional model accuracy (unaugmented vs. augmented contexts) in comparison to the model that does not have access to the long-range contexts, they can measure the impact of the augmentations on the usable information. The model they use is based on the GPT-2 architecture.

Sinha et al. (2021b) take this even further and investigate how word order permutations influence the performance of a variety of models (including transformer models such as BERT and RoBERTa, and non-transformer architectures such as a bidirectional long short-term memory neural network) at the task of *natural language inference*.¹⁶ They find that in almost all of their test cases, there is at least one and usually multiple permutations that still allow the model to predict the correct output. Sometimes, even previously incorrectly predicted samples are predicted correctly after the permutation of the sample’s word order. Accordingly, they conclude that “NLI models (Transformer-based models, RNNs, and ConvNets) are largely insensitive to permutations of word order that corrupt the original syntax” (Sinha et al., 2021b, p. 7337).¹⁷ It is, however, important to note that while they verify their findings with three different English datasets and a Chinese one, their study still only takes into account a single natural language processing task and the tested models’ behaviors in the context of that task alone.

Nonetheless, Sinha et al. (2021a) manage to back up their claim in a later study where they pre-train the RoBERTa masked language modeling architecture on data with permuted word order. They find that this alternation hardly influences the model’s performance on down-stream tasks, and conclude that “MLM’s success is most likely not due to its ability to discover syntactic and semantic mechanisms necessary for a traditional language processing pipeline during pre-training” (Sinha et al., 2021a, p. 2896),¹⁸ but due to the distributional information the model learns.

Until this point, the evidence suggests that word order is not something that is inherently learned by language models, nor do they rely on it for solving the tasks they have been trained for. Abdou et al. (2022), however, challenge this belief. They base their study on Sinha et al. (2021a)’s findings and try to provide a more nuanced insight into the models’ behaviors by conducting additional experiments. In particular, they try to answer the question of why models without position embeddings perform worse than models with position embeddings that are trained on shuffled data when according to Sinha et al. (2021a), word order does not seem to substantially impact model performance. Additionally, they challenge the benchmarks Sinha et al. (2021a) evaluate their models on and investigate whether there are other natural language understanding tasks that require a more substantial understanding of word order than the tasks Sinha et al. (2021a) explore, as they argue that many of the latter have been shown to be solvable by employing “spurious artefacts and heuristics” (Abdou et al., 2022, p. 6911).¹⁹ At first, Abdou et al. (2022) show that language models do in fact contain word order information by training linear classifiers on the models’ final *word representations* for predicting whether two tokens are likely to succeed one another and for predicting the position of a word in a sentence.²⁰ They find that word representations extracted from a model without position encodings produce close to random results, whereas both embeddings extracted from a model trained on unaugmented data and

¹⁶Natural language inference describes the natural language processing task that aims to predict whether in a pair of sentences, a premise and a hypothesis, the premise entails the hypothesis, contradicts it, or is semantically unrelated to it (Bowman et al., 2015; Sinha et al., 2021b).

¹⁷NLI: Natural language inference. RNNs: Recurrent neural networks. ConvNets: Convolutional neural networks.

¹⁸MLM: Masked language modeling.

¹⁹Sinha et al. (2021a) evaluate their models on GLUE (Wang et al., 2018) and on the PAWS dataset (Zhang et al., 2019).

²⁰Word representations equal embeddings.

embeddings extracted from a model trained on shuffled data are able to achieve high accuracy scores on the previously introduced tasks. They argue that this surprising result is related to the point in time when the shuffling takes place: Models trained on data shuffled after *byte pair encoding segmentation* perform significantly worse than models trained on data shuffled before byte pair encoding segmentation, i.e., models trained on data where only full words are shuffled.²¹ All of the models Sinha et al. (2021a) use were trained on data shuffled before byte pair encoding segmentation. Abdou et al. (2022, p. 6910) remark:

When tokens are shuffled *before* BPE segmentation, this leads to word-level shuffling, in which sequences of subwords that form words remain contiguous. Such sequences become a consistent, meaningful signal for language modelling, allowing models to efficiently utilise the inductive bias provided by position embeddings. Thus, even though our pre-trained models have, in theory, not seen consecutive tokens in their pre-training data, they have learned to utilise positional embeddings to pay attention to adjacent tokens.²²

Shuffling after byte pair encoding segmentation, on the other hand, does not seem to sensitize the models to use positional embeddings for attending to adjacent tokens to the same extent. Additionally, Abdou et al. (2022) report that especially in short sentences, up to 12% of the randomly shuffled word bigrams also occur in the original sentences. This “accidental overlap” effect drops to a maximum of about 8% with subword bigrams. Nevertheless, even when shuffling is performed after byte pair encoding segmentation, Abdou et al. (2022) report that at least some information about the original word order is preserved, which they attribute to the correlation between unigram probabilities and sentence length.²³ They verify their findings by testing the models on additional benchmarks such as the WinoGrande dataset (Sakaguchi et al., 2019) and SuperGLUE (Wang et al., 2020), which, according to them, require a more substantial understanding of word order than the benchmarks used by Sinha et al. (2021a). This mostly resulted in significant drops in performance for the models trained on shuffled data and an even more significant drop for the model trained without positional encodings. Thus, they conclude that models trained on augmented pre-training data still retain information about word order. This retained “word order knowledge” exists mostly on the local level, which aligns with O’Connor and Andreas (2021)’s findings. However, Abdou et al. (2022) arrive at the conclusion that models do in fact rely on word order information for solving natural language understanding tasks that require substantial knowledge of word order.

Lasri et al. (2022) take a step back and investigate the importance of position encodings for the pre-training objective, i.e., masked language modeling. They hypothesize that increasing the number of masked tokens during pre-training increases the importance of word order information. This information comes in the form of position encodings, which the model needs to effectively solve the masked language modeling task. By computing the distance of probability distributions for possible completions in

²¹Byte pair encoding segmentation describes the subtokenization algorithms explained in Section 2.6.

²²BPE: Byte pair encoding.

²³Abdou et al. (2022, p. 6910) exemplify this with the phrase *thank you*, which they use to illustrate that “there is an approximately learnable relationship between the distribution of words and sentence boundary symbols.” Formulaic contexts and other genre-dependent commonly short sentences and phrases do not offer a wide range of possibilities for subword shuffling.

both ordered and unordered contexts, they are able to support this hypothesis. They then proceed to test the performance of two BERT models, one of which has access to position encodings while the other does not. They find that the predictions of the model with access to position encodings align with the true probability distribution of the ordered context, whereas the predictions of the model without access align with the probability distribution of the unordered context. Therefore, they conclude that position information is more important when more tokens are masked, and models that have access to position encodings perform better under these circumstances. It is, however, important to note that Lasri et al. (2022) test their hypotheses on artificial language data as it allows for an estimation of the true probability distributions for possible completions. This is more difficult when working with natural languages, whose word order typically is less restricted than the word order of the artificial language investigated in Lasri et al. (2022)'s study. Nonetheless, their findings indicate that word order information in the form of position encodings does play a role in the models' capability of solving the pre-training task of masked language modeling, especially with higher numbers of masked tokens. In this paper, on the other hand, I want to explore how transformer models perform at the downstream task of automated word order error detection, where word order information appears to be the single most important factor necessary for solving these tasks.

Chapter 3

Detection of Generic Word and Verb Order Errors

In the following chapter, I propose a number of initial classification approaches for detecting generic word and verb order errors. I define both tasks as binary sequence classification problems. When presented with a sentence, a classifier should be able to identify whether the word order of the presented sentence is correct or incorrect.²⁴ Training classifiers for these tasks presents a challenge due to the lack of available genuine data annotated for word and verb order errors. The difficulty in obtaining such data stems from the high costs associated with annotation and existing privacy concerns when working with learner data. Consequently, I resort to generating pseudo datasets that include both correct and incorrect sentences. In the first part of this thesis, my focus is on assessing the effectiveness of different classification approaches in detecting generic word and verb order errors within these incorrect sentences. I characterize sentences with generic word order errors as those where any or all words within the sentence may be randomly rearranged. Meanwhile, in sentences with generic verb order errors, only the verbs are subject to potential misplacement. After showcasing the abilities of the classification approaches in detecting erroneous word order, utilizing generic word and verb order errors for demonstration purposes, the second part of this thesis focuses on the detection of verb order errors that learners are likely to make.

3.1 Generation of Pseudo Data: Datasets Rand and Verbs

In order to be able to train classifiers for the tasks of word and verb order error detection, a substantial amount of labeled data is needed. Due to the lack of available genuine data, I opt to curate pseudo datasets for the training and evaluation of the classifiers. Next to correct Dutch sentences, these pseudo datasets contain sentences with word order errors induced according to two different principles of shuffling the tokens of the respective correct sentences.²⁵ These principles are:

1. Random shuffling of all tokens (RAND).
2. Random shuffling of verb tokens (VERBS).

²⁴By correct I refer to sentences that are acceptable according to standard Dutch syntax rules.

²⁵I henceforth use the terms shuffling and permuting interchangeably. A shuffle is equivalent to a permutation.

RAND shuffles are equivalent to generic word order errors. For any given sentence, the set of RAND shuffles contains all possible unique rearrangements of every token in that sentence. Picking one of these permutations randomly results in a random shuffling of the sentence. This simulates any and all potential misplacements of words that could occur within that sentence. VERBS shuffles, on the other hand, are equivalent to generic verb order errors as their set for a given sentence contains all possible unique permutations where only the verb tokens can change their absolute position. The rest of the sentence’s original syntax is preserved, i.e., the verb-unrelated tokens stay in their original relative order to one another. Kiyono et al. (2019) point to three aspects that can impact model performance when incorporating pseudo data into the training of grammatical error correction models:

1. Pseudo Data Generation
2. Seed Corpus
3. Optimization Setting²⁶

Following the first two aspects, I will now describe both the method of pseudo data generation, i.e., the previously introduced shuffle methods, and the seed corpora I use for the curation of the pseudo datasets in more detail.

3.1.1 Shuffle Methods

In the following section, I formally define the RAND and VERBS shuffle methods to illustrate that VERBS shuffles constitute a subset of RAND shuffles. Because of this, it is conceivable that classifiers trained on data shuffled according to the RAND shuffle method could achieve high performance scores when tested on data shuffled according to the VERBS shuffle method, which is one of the hypotheses I will explore in my experiments.

Rand Shuffling

Let s be an original sentence (3.1), represented as a sequence of verb tokens v and verb-unrelated tokens w (3.2):

$$s = (t_1, t_2, \dots, t_n) = (t_i)_{i=1}^n \quad (3.1)$$

$$t_i = \begin{cases} v_i, & \text{if } t_i \text{ is a verb token} \\ w_i, & \text{otherwise} \end{cases} \quad (3.2)$$

Let generic word order errors be the set of all permutations of s , excluding the original sequence s , i.e., the identity permutation (3.3):

$$S_{Rand}(s) = \{s' : s' \text{ is a permutation of } s \text{ and } s' \neq s\} \quad (3.3)$$

²⁶The optimization setting, according to Kiyono et al. (2019), describes the method of combining genuine data and pseudo data during the training process of grammatical error correction models. However, within the scope of this thesis, I am interested in only very specific types of grammatical errors, i.e., word and verb order errors. As there is no genuine data available that is annotated for these types of errors, my datasets consist of pseudo data only.

In $S_{Rand}(s)$, all tokens can appear in all positions randomly (RAND). Its maximum size is calculated by taking the factorial of n minus one to exclude the original, correct sequence s (3.4).²⁷

$$\max(|S_{Rand}(s)|) = n! - 1 \quad (3.4)$$

In a sentence such as *ik koop boeken* ‘I buy books’,²⁸ which has $n = 3$ tokens, this set contains five unique permutations (3.5).

$$\begin{aligned} |S_{Rand}((ik, koop, boeken))| &= 3! - 1 \\ &= 5 \end{aligned} \quad (3.5)$$

This excludes the original, correct permutation $(ik, koop, boeken)$ (3.6).

$$\begin{aligned} S_{Rand}((ik, koop, boeken)) = \\ &\{(ik, boeken, koop), \\ &\quad (koop, ik, boeken), \\ &\quad (koop, boeken, ik), \\ &\quad (boeken, ik, koop), \\ &\quad (boeken, koop, ik)\} \end{aligned} \quad (3.6)$$

Datasets created according to the RAND shuffle method contain both original sentences s and, for each original sentence, one of the permutations contained in $S_{Rand}(s)$. They simulate all word order errors that could occur in a given sentence. The permutation is picked from $S_{Rand}(s)$ randomly. While it is possible that changing the word order randomly in a sentence could result in a permutation that also forms a grammatically correct sequence, this effect is estimated to be low. I illustrate the effect by means of the sentence *ik wil geen boeken kopen* ‘I don’t want to buy books’, which holds five tokens. $S_{Rand}((ik, wil, geen, boeken, kopen))$ holds three permutations other than the original that could naturally occur in the Dutch language, although only the first one could form a proper stand-alone sentence (Example 1).

- (1) $(wil, ik, geen, boeken, kopen)$
 $(ik, geen, boeken, wil, kopen)$
 $(ik, geen, boeken, kopen, wil)$

These permutations constitute the variants of word order as used in polar questions and subordinate clauses, as will be explained in Section 4.1.1. In total, there are 119 possible permutations contained within $S_{Rand}((ik, wil, geen, boeken, kopen))$. The longer the sentences, the more the relative amount of incorrect sentences versus correct sentences in $S_{Rand}(s)$ is expected to grow, and the lower the likelihood of a correct sentence getting randomly picked. The total amount of correct sentences getting mislabeled as incorrect sentences should therefore be negligible. For better readability, I will refer

²⁷The maximum size is equal to the actual size of the set when all tokens within the sequence s are unique. Let the set of unique token types in the sequence s be represented as $T = \{T_i\}_{i=1}^m$, where m is the total number of unique token types. For each T_i in T , let c_{T_i} denote the count (i.e., the number of occurrences) of token type T_i in the sequence s . If the sequence holds duplicate tokens, the actual size can be calculated by dividing the factorial of the length of the sequence by the product of the factorials of the counts c of each distinct token type T_i minus one to exclude the identity permutation: $|S_{Rand}(s)| = n! / (c_{T_1}! * c_{T_2}! * \dots * c_{T_m}!) - 1$. For illustrative purposes, however, I will henceforth only showcase the maximum size of the sets I define.

²⁸I lowercase all tokens and exclude punctuation.

to the method of shuffling, the set of possible permutations for a single sentence, and the resulting dataset simply as RAND whenever a more fine-grained distinction is not necessary.

Verbs Shuffling

In comparison to RAND shuffling, where all tokens can appear in all positions randomly, with VERBS shuffling, only the verb tokens within a sentence can change their positions randomly. The rest of the tokens, i.e., the verb-unrelated tokens, remain in their original relative order to one another. Since all VERBS shuffles are included in the set of RAND shuffles for a given sentence, VERBS shuffles form a subset of RAND shuffles (3.7):

$$S_{Verbs}(s) \subseteq S_{Rand}(s) \quad (3.7)$$

Let W be a sequence of all verb-unrelated tokens, where the order of the verb-unrelated tokens w_i in W is the same as the order of the corresponding t_i in the original sentence s (3.8):

$$W = (w_i)_{i=1}^j \quad (3.8)$$

Let V be a multiset of all verb tokens contained in the original sentence s .²⁹ Being a set, the order of the verb tokens v_i in V does not matter (3.9):

$$V = \{v_i\}_{i=1}^k \quad (3.9)$$

For illustrative purposes, I will first showcase how to calculate the size of $S_{Verbs}(s)$ when the original sentence s contains a single verb token, as is the case in the example sentence *ik koop boeken*,³⁰ which I introduced in the previous paragraph. Calculating the size of the set of verb order error permutations for a given sentence can be defined as a combinations problem. The verb-unrelated tokens of W always remain in their original order. The verb tokens of V are free and can take any position before, after, or in between the W tokens; they fill hypothetical gaps g around the tokens of W (Example 2):

$$(2) \quad _ ik _ boeken _$$

The number of gaps g is calculated as follows (3.10):

$$g = j + 1 \quad (3.10)$$

This enables us to calculate the maximum size of $S_{Verbs}(s)$ by employing the formula for combinations minus one to exclude the identity permutation (3.11).

$$\begin{aligned} \max(|S_{Verbs}(s)|) &= C(g, k) - 1 = \binom{g}{k} - 1 \\ &= \frac{g!}{k!(g-k)!} - 1 \end{aligned} \quad (3.11)$$

²⁹I define V to be a multiset as it allows for duplicates of the same verb tokens.

³⁰Verb tokens are underlined.

In our case, this equates to two possible permutations (3.12):³¹

$$\begin{aligned}
|S_{Verbs}((ik, koop, boeken))| &= C((|ik, boeken| + 1), |\{koop\}|) - 1 \\
&= \binom{3}{1} - 1 \\
&= \frac{3!}{1!(3-1)!} - 1 \\
&= 2
\end{aligned} \tag{3.12}$$

This excludes the original, correct permutation $(ik, koop, boeken)$ (3.13):

$$\begin{aligned}
S_{Verbs}((ik, koop, boeken)) &= \\
&\{(ik, boeken, koop), \\
&(koop, ik, boeken)\}
\end{aligned} \tag{3.13}$$

However, once we look at longer sentences, it becomes the norm rather than the exception that sentences contain more than a single verb token. Therefore, we either need to select more than a single gap when filling in the gaps of W with the tokens of V , or we need to select the same gap(s) multiple times. This calls for the formula for combinations with repetitions. Additionally, as previously explained, the order of the verb tokens does not matter. Thus, we multiply by the factorial of the number of verb tokens k in s to account for all permutations of the set of verb tokens (3.14):³²

$$\begin{aligned}
max(|S_{Verbs}(s)|) &= C(g + k - 1, k) \cdot k! - 1 = \binom{g + k - 1}{k} \cdot k! - 1 \\
&= \frac{(g + k - 1)!}{k!(g - 1)!} \cdot k! - 1
\end{aligned} \tag{3.14}$$

For the previously introduced example sentence $ik \underline{wil} \underline{geen} \underline{boeken} \underline{kopen}$, which holds two verb tokens and three verb-unrelated tokens, this yields a total of 19 possible permutations (3.15):

$$\begin{aligned}
|S_{Verbs}((ik, wil, geen, boeken, kopen))| &= \\
&= \frac{((|ik, geen, boeken| + 1) + |\{wil, kopen\}| - 1)!}{|\{wil, kopen\}|!((|ik, geen, boeken| + 1) - 1)!} \cdot |\{wil, kopen\}|! - 1 \\
&= \frac{(4 + 2 - 1)!}{2!(4 - 1)!} \cdot 2! - 1 \\
&= 19
\end{aligned} \tag{3.15}$$

³¹The formula for combinations here calculates the number of all possible gap selections when we need to select a single gap as there is one verb token in the example sentence. With three available gaps, the number of possible gap selections also equals three. We subtract one to exclude the gap selection that would result in the original, correct sequence s .

³²Here, the maximum size of the set is equal to the actual size when all verb tokens are unique. If there are duplicates within V , the actual size of $S_{Verbs}(s)$ can be calculated in a similar way as previously explained (dividing the maximum size by the product of the factorials of the counts of all unique verb token types before subtracting one for the identity permutation). The tokens of W themselves do not play a role in calculating the actual size of $S_{Verbs}(s)$ as they can be considered a constant and do not change their position during the permutation process. They are only needed to determine the number of gaps to fill.

The permutations contained within $S_{Verbs}((ik, wil, geen, boeken, kopen))$ are illustrated in Appendix B.

Summarizing the previous definitions, I define verb order errors be the set of all unique permutations of the positions of the verb tokens in a sentence s , excluding the original, correct sentence. The relative order between the verb-unrelated tokens of W is preserved (3.16):

$$S_{Verbs}(s) = \{s' : s' \text{ is a permutation of } s \\ \text{with each } v_i \in V \\ \text{inserted into any of the } g \text{ gaps of } W, \\ \text{where gaps may be filled more than once,} \\ \text{and } s' \neq s\} \quad (3.16)$$

As shown in the previous paragraph, the sentence *ik wil geen boeken kopen* has $5! - 1 = 119$ unique RAND shuffles, as opposed to 19 unique VERBS shuffles. Thus, VERBS shuffles are a subset of RAND shuffles in the same way that verb order errors are a subset of word order errors. Datasets curated according to the VERBS shuffle method contain both original sentences s and, for each original sentence, one of the permutations contained in $S_{Verbs}(s)$. They simulate generic verb order errors. Here too, for better readability, I will refer to the method of shuffling, the set of permutations for a single sentence, and the resulting dataset as VERBS.

3.1.2 Seed Corpora

For the generation of the pseudo data, I use the corpora illustrated in Table 3.1 as seed corpora, i.e., as sources for the correct Dutch sentences which I permute according to the shuffle methods explained above.

Table 3.1: Seed corpora for pseudo datasets

Corpus	License	Content
EDIA ³³	CC BY-NC 4.0	news items, fiction, academic journals, educational content, informational content
LASSY ³⁴	see Appendix H.1	newsletters, websites, Wikipedia, press releases, books, brochures, flyers, manuals, legal texts, newspapers, policy docs, proceedings, reports
LEIPZIG ³⁵	CC BY-NC 4.0	newspapers, webcrawls, Wikipedia
WAI-NOT ³⁶	see Appendix H.2	newspaper articles in easy-to-read Dutch

By including the different types of content listed in the table, I try to make the pseudo datasets as representative as possible of the text types learners of a foreign language are likely to be prompted to produce in a classroom setting. The training data contains sentences from all of these corpora.

³³Breuker (2023); <https://www.edia.nl/resources/elg/downloads>, last accessed: 15.08.2023.

³⁴<https://taalmaterialen.ivdnt.org/download/lassy-klein-corpus6/>, last accessed: 15.08.2023.

³⁵Composition: Mixed 2012, Wikipedia 2021, News 2022; https://wortschatz.uni-leipzig.de/de/download/Dutch#nld-nl_web_2019, last accessed: 15.08.2023.

³⁶<https://taalmaterialen.ivdnt.org/download/wai-not-corpus1-0/>, last accessed: 15.08.2023.

For the test data, I want to approximate genuine learner data as closely as possible by only using sentences from the EDIA corpus. The EDIA corpus is a readability corpus comprised of texts annotated for CEFR levels.³⁷ I filter the EDIA data to select sentences from texts that match the CEFR levels present in the LEUVEN corpus of genuine Dutch learner data available to me.³⁸ A detailed introduction to the LEUVEN corpus will be provided in Section 4.1.4. This means that the test data only holds sentences of levels A2 through C1, which reflects the typical learner range. The remaining sentences of the EDIA corpus, i.e., sentences originating from texts annotated as A1 or C2, I add to the training data.

Finally, since I am looking into word and verb order errors, there are a number of additional conditions a sentence must fulfill in order to be included in the pseudo datasets I curate. One of these conditions, naturally, is that the sentence must contain verb tokens. In the following section, I will therefore explain how I preprocess the raw seed corpora before illustrating the final data split.

Preprocessing

The data in the LASSY and LEIPZIG corpora is already pre-segmented into individual sentences. For EDIA and WAI-NOT, I segment the full texts contained within the corpora into individual sentences with the help of the SPACY dependency parser (model: *nl_core_news_lg*).³⁹ Subsequently, for all corpora, I filter out all sentences that fulfill any of the following conditions:

1. Sentence is shorter than 10 characters in length.
2. Sentence does not begin with a capital letter.
3. Sentence does not end with a full stop, question mark, or exclamation mark.
4. Sentence does not contain any verb tokens according to the SPACY part-of-speech tagger.⁴⁰

By doing so, I try to ensure that the pseudo datasets only contain proper sentences as opposed to bullet points, headers, elements of lists, or other syntactically incomplete structures and mis-segmented sequences. That a sentence must contain verb tokens to be able to be permuted according to the VERBS shuffle method is trivial.

Additionally, in order to save computational resources and to approximate genuine learner data even further, I analyze the distribution of sentence length in the LEUVEN corpus of genuine Dutch learner data to establish a maximum sentence length that realistically mirrors the output of learners. As illustrated in Figure 3.1, it seems that

³⁷CEFR: Common European Framework of Reference for Languages.

³⁸All texts in the EDIA corpus have been annotated for their corresponding CEFR level by multiple annotators. When the annotators disagree, I convert the CEFR levels to a point scale (A1 = 1, A1+ = 2, A2 = 3, ..., C2+ = 12) and average by the number of annotators. Finally, I convert the averaged score back to its corresponding CEFR level according to the point scale to obtain the average CEFR level for the given text.

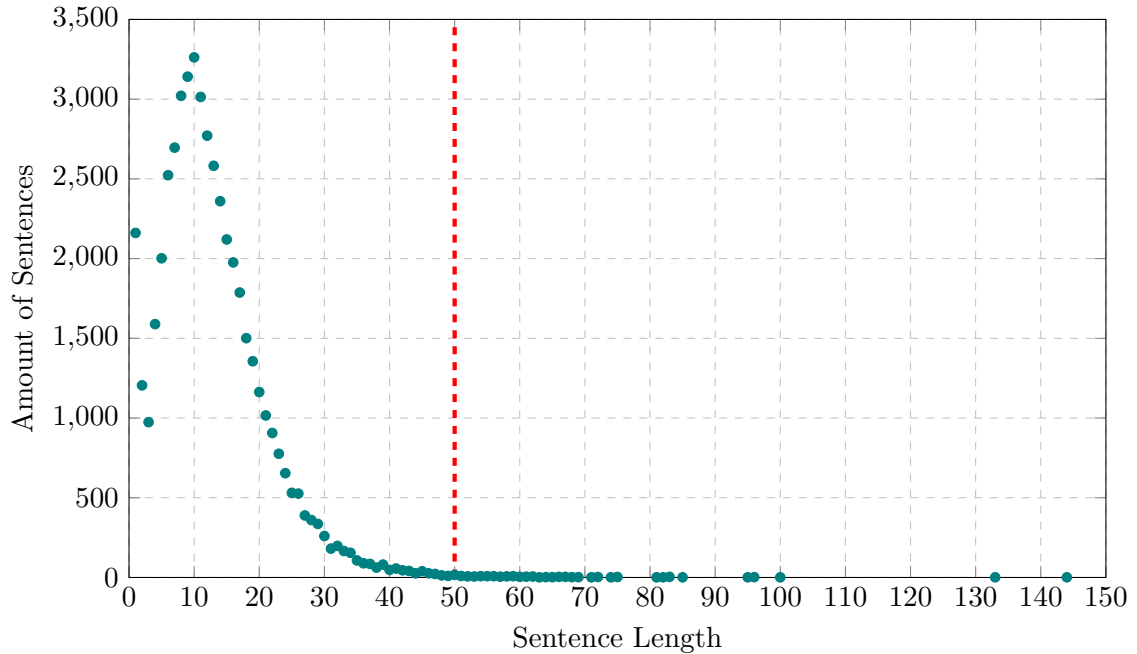
³⁹<https://spacy.io/api/dependencyparser>, last accessed: 15.08.2023.
<https://spacy.io/models/nl>, last accessed: 15.08.2023.

⁴⁰<https://spacy.io/usage/linguistic-features>, last accessed: 15.08.2023.

Uses the same *nl_core_news_lg* model, which is reported to exhibit an accuracy score of 0.96 for part-of-speech tagging. I check for the presence of at least one token in a given sentence that is tagged as a verb or auxiliary.

learners hardly ever produce sentences that are longer than 50 tokens. Therefore, I filter out all sentences that exceed this threshold in length.

Figure 3.1: Distribution of sentence length in the LEUVEN corpus



Finally, I remove all punctuation characters from the sentences to ensure that the machine learning-based classification approaches I explore in my experiments have to rely on linguistic information only as opposed to orthographic cues that could potentially be learned during the training process.⁴¹

Data Split

After preprocessing and filtering the raw seed corpora according to the conditions explained above, I am left with the following data split of correct Dutch sentences (Table 3.2):

Table 3.2: Train/Test data split

Dataset	Size ⁴²	Seed
Train	2,231,602	EDIA ^{A1,C2} , LASSY, LEIPZIG, WAI-NOT
Test	13,586	EDIA ^{A2-C1}

Due to computational restrictions, I treat the train data as a pool of sentences to draw from, i.e., not all classification approaches make use of all the sentences contained within the train data. I specify the precise amount of train sentences each classifier is trained

⁴¹It is, for example, likely that a machine learning-based classifier could learn that a full stop in any place other than the very end of the sentence, as could be rendered when permuting a sentence according to the RAND shuffle method without removing punctuation tokens, is a strong indicator of incorrect word order.

⁴²In number of sentences.

on when introducing the classification approaches in the experimental setup (Section 3.4). The resulting RAND and VERBS datasets contain both the specified numbers of correct sentences and, for each correct sentence, one permutation of it rendered according to the respective shuffle method. The actual size of the datasets is therefore double the indicated amount of sentences.

3.2 Standard Evaluation Metrics

The first part of this thesis focuses on the general capability of the classification approaches I explore in detecting erroneous word order. Therefore, I evaluate the performance of the models according to the standard F_1 metric, which is defined as the harmonic mean of the precision and recall scores and attributes equal importance to both metrics (3.17):

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.17)$$

Precision, for each category, is defined as the number of instances that the classifier correctly predicts as belonging to that respective category (true positives; TP) divided by the sum of the number of true positives plus the number of instances that are incorrectly predicted as belonging to that category (false positives; FP) (3.18):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3.18)$$

In the context of the binary classification tasks that are word and verb order error detection, this means that the precision score indicates the proportion of sentences the classifier predicts as correct that are indeed correct, and the proportion of sentences it classifies as incorrect that are indeed incorrect.

Recall, for each category, is defined as the number of true positives divided by the sum of the number of true positives plus the number of instances that are incorrectly classified as belonging to another category (false negatives; FN) (3.19):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.19)$$

For word and verb order error detection, this means that the recall score indicates the proportion of correct sentences that are correctly classified as such, and the proportion of incorrect sentences that are correctly classified as incorrect.

For a model to exhibit an overall good performance at detecting erroneous word and verb order, both the precision and recall scores should be as high as possible for each category, i.e., the classifier should identify as many sentences as possible as belonging to their respective category (recall) while at the same time, it should not overpredict any of the categories and only classify sentences as belonging to a category if they actually do belong to that category (precision). The F_1 score combines both metrics, attributing equal importance to each of them. Hence, the average F_1 score is a good indicator of the general capabilities of the classification approaches.⁴³

⁴³(Footnote continues on the next page.) For reasons of completeness, in Appendix D, I also provide the accuracy score, which indicates the proportion of total predictions that the classifier correctly made, regardless of category. It is calculated by dividing the sum of the number of true positives plus the number of true negatives (TN), i.e., the total number of correct predictions, by the total number of

3.3 Models

In this thesis, I explore the performance of classifiers based on three different natural language processing model families at the tasks of word and verb order error detection: part-of-speech taggers, syntactic parsers, and transformer models. Specifically, I explore the part-of-speech tagger of the SPACY natural language processing toolkit,⁴⁴ the DISCO-DOP constituency parser,⁴⁵ and transformer models adapted for Dutch based on the BERT, RobBERTa, and GPT-2 architectures introduced earlier.

3.3.1 Part-of-Speech Tagger

In traditional methods, unlike neural network-based parsers that provide end-to-end solutions, part-of-speech tagging is typically required before syntactic parsing. Thus, I aim to compare the performance of classifiers trained on the output of a part-of-speech tagger versus those trained on the more syntactically informative output of a syntactic parser. The part-of-speech tagger I use is the SPACY morphologizer, which is a neural part-of-speech tagger that exhibits an accuracy score of 0.96 for Dutch with the *nl_core_news_lg* model (see also Section 3.1.2).

3.3.2 Syntactic Parser

DISCO-DOP, or discontinuous data-oriented parsing, is a statistical constituency parser implementation that focuses on being able to represent discontinuous constituents. Parse tree structures that allow for discontinuous constituents allow “non-terminal node[s] to dominate a lexical span that consists of non-contiguous chunks.” (Cranenburgh et al., 2016). This can be useful for Dutch as in Dutch, discontinuous constituents are quite common. Consider the example sentences *Ik heb ervan gedroomd*. ‘I have dreamed about it.’ and *Ik heb er niet van gedroomd*. ‘I have not dreamed about it.’. In the first sentence, *ervan* ‘from it’ is a single token, yet in the second sentence, the adverb *niet* ‘not’ splits it up into two tokens, i.e., a discontinuous constituent. The DISCO-DOP parser operates in three stages, making use of three different grammar formalisms, the precise details of which go beyond the scope of this paper and can be found in Cranenburgh et al. (2016).⁴⁶ I use it with a grammar that is trained on both the CGN (van der Wouden et al., 2002) and Lassy (van Noord, 2009) treebanks for Dutch.⁴⁷ For each parsing stage, DISCO-DOP approximates the most probable parse by employing the *relative frequency estimate*. “The relative frequency of a fragment is the number of its occurrences, divided by the total number of occurrences of fragments

instances (3.20):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (3.20)$$

⁴⁴<https://spacy.io/api/morphologizer/>, last accessed 15.08.2023. The implementation is not a pure part-of-speech tagger but additionally offers morphological information.

⁴⁵<https://discodop.readthedocs.io/en/latest/>, last accessed: 15.08.2023.

⁴⁶DISCO-DOP uses a Probabilistic Context-Free Grammar (PCFG) in its first stage. The grammar treats elements of discontinuous constituents as independent from one another but encodes information about discontinuity in the node labels. In the next stage, it uses a Probabilistic Linear Context-Free Rewriting System (PLCFRS). This grammar allows “non-terminals to rewrite tuples of strings instead of just single, contiguous strings” (Cranenburgh et al., 2016, p. 65). Finally, it uses a discontinuous data-oriented parsing grammar (Disco-DOP) that makes use of tree fragments instead of production rules to find the most likely parse for a sentence.

⁴⁷<https://lang.science.uva.nl/grammars/>, last accessed: 15.08.2023.

with the same root node” (Cranenburgh et al., 2016, p. 84). The relative frequencies of all fragments in the input are multiplied and yield the probability of the derivation. I always choose the output of the stage that renders the highest absolute probability when parsing sentences. Finally, I choose the DISCO-DOP parser implementation in particular for its representation of natural language in the form of constituency structures, which allows me to abstract verb order-irrelevant syntactic information as will be explained in Section 3.4.1.

3.3.3 Transformer Models

BERTje. BERTJE is a monolingual transformer model based on the BERT architecture introduced in Section 2.6 and adapted for the Dutch language. It uses the same architecture and parameters but is trained on “a large and diverse [Dutch] dataset of 2.4 billion tokens” (de Vries et al., 2019, p. 1). Additionally, it differs from the original BERT model in that its second pre-training task is sentence order prediction as opposed to next sentence prediction. In next sentence prediction, the original BERT model was tasked to predict whether, in a pair of sentences, the two sentences are consecutive or not. The second sentence could either be an actual consecutive sentence or a random sentence (de Vries et al., 2019). BERTJE, on the other hand, is tasked to predict whether two sentences are consecutive or have been swapped, i.e., the second sentence of the sentence pair is never random but always taken from the same context. According to de Vries et al. (2019), the original next sentence prediction task led the BERT model to learn topic similarity instead of coherence, which they mitigate by adapting the pre-training objective. BERTJE outperforms the multilingual BERT model trained on the full Wikipedias of 104 languages in a variety of fundamental natural language processing tasks in Dutch, among which part-of-speech tagging (de Vries et al., 2019).

RobBERT. ROBBERT, like BERTJE, is a monolingual transformer model adapted for Dutch. It is based on the previously introduced RoBERTa model that optimizes the BERT pre-training process but generally uses the same architecture as BERT (Delobelle et al., 2020). It is trained on a Dutch dataset of 6.6 billion words and outperforms BERTJE in a number of Dutch natural language processing tasks, especially when datasets for fine-tuning for a given task are rather small (Delobelle et al., 2020).

GPT-2 Dutch. Lastly, GPT-2 (recycled for) Dutch is another monolingual transformer model. It is based on the GPT-2 architecture and therefore differs from BERT architecturally. Its pre-training objective is next word prediction and it processes sequences from left to right instead of bidirectionally. Additionally, it differs from both BERTJE and ROBBERT in that the model is not initialized with random parameters when pre-training, but de Vries and Nissim (2021) retrain only the lexical embeddings, i.e., the vector representations of tokens that serve as input for the model with the help of a dataset that is slightly larger in size than the dataset BERTJE was trained on. In their first step, they leave the parameters within the layers of the transformer network untouched. The resulting model is therefore technically identical to the original English model in terms of model parameters. It is only in their following step that de Vries and Nissim (2021) fine-tune the transformer layers based on the newly obtained retrained word embeddings for the target language, which, they report, reduces the recognizability of the model’s output as artificial. They evaluate their model’s

generative capabilities with the help of human annotators and do not provide results for benchmark datasets of common natural language processing tasks. I want to explore whether the difference in architecture in comparison to the previous two models could potentially have an influence on the transformer model’s capability of detecting erroneous word and verb order.

3.4 Experimental Configurations

I will now introduce the different experimental configurations I explore with the help of the previously introduced models. First, I will explore a rule-based approach to classification that is based on the output of the DISCO-DOP parser. It is a naive lookup approach that does not employ machine learning algorithms. I will then introduce the machine learning-based approaches by illustrating the performance of a classifier trained on the output of the SPACY part-of-speech tagger. As part-of-speech tagging in many cases is a prerequisite for the task of syntactic parsing, I will subsequently compare this performance with the performance of a classifier trained on the syntactically more informative output of the DISCO-DOP parser. By doing this, I can confirm that detecting word and verb order errors is a natural language processing task that benefits significantly from having access to syntactic information. Finally, I compare the performances of three different transformer-based models in order to investigate whether transformer-based models are able to reliably solve tasks that require an understanding of word order information and whether different transformer architectures influence the performance scores of these models.

3.4.1 Parse Lookup

The first configuration I explore is a lookup approach based on the output of a syntactic parser, i.e., the DISCO-DOP parser. Because the parser’s processing speed is very slow, this approach makes use of only 80,000 correct sentences picked randomly from the set of available train sentences as explained in Section 3.2. I let the parser parse all

Figure 3.2: Tree A

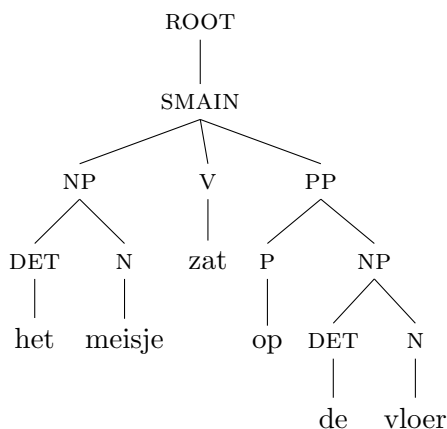
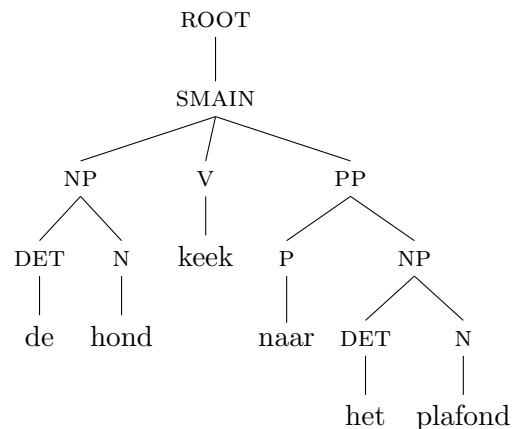


Figure 3.3: Tree B



80,000 correct sentences and save the resulting most probable parses and their tree representations as a POOL of correct parse structures. The assumption here is that a parse structure resulting from parsing a correct sentence is a valid sentence structure

pattern in the target language. The sequences *het meisje zat op de vloer* ‘the girl sat on the floor’ (A) and *de hond keek naar het plafond* ‘the dog looked at the ceiling’ (B) are superficially different but share the same sentence structure. The parser renders similar parses for similarly structured sentences (Figure 3.2 and Figure 3.3).⁴⁸ Consequently, the question arises of whether a POOL of correct parse trees can serve as an effective tool to look up the validity of a parse of a sentence that is not included in that POOL. I name the solution that uses the tree output of the DISCO-DOP parser in its unmodified, original form DOP-TREE-ORIG.

However, once we look at sequences such as *het meisje zat op de vloer* (A) and *het meisje zat op school* ‘the girl was in school’ (C), we face a new challenge: The noun phrase *de vloer* ‘the floor’ loses its determiner when being replaced by *school*. This means that the sentence structure of the sequence is different from the one without the adjective as the tree structure the parser renders holds a representation for every token in the sequence (Figure 3.4 and Figure 3.5):

Figure 3.4: Tree A

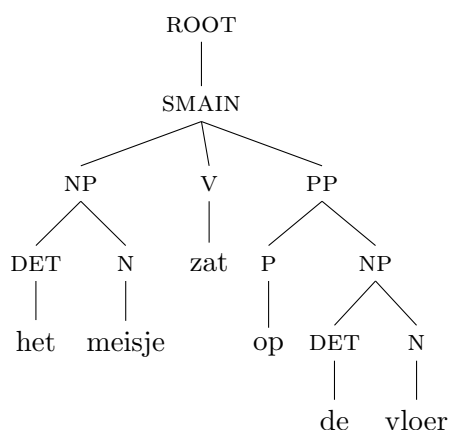
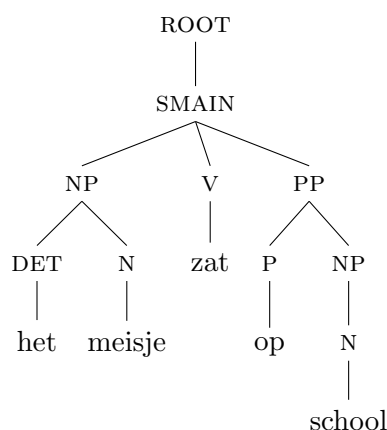


Figure 3.5: Tree C



Yet, the overall sentence structure is still the same and the difference occurs in a constituent that does not contain a verb token. If a constituent does not contain any verb tokens, its internal structure is irrelevant to the relative order of verbs and verb-unrelated constituents. Thus, I collapse constituents that do not contain verb tokens into the highest possible node label before a verb token crosses the path (Figure 3.6 and Figure 3.7):

Figure 3.6: Tree A SIMPLE

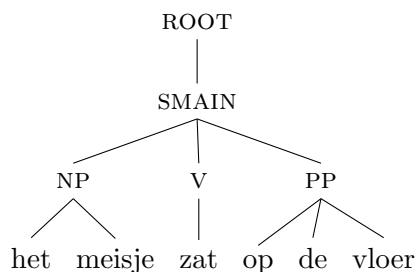
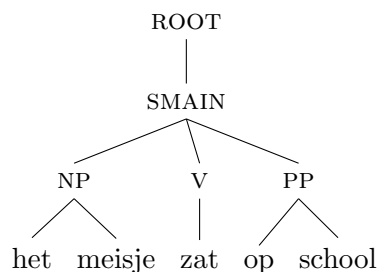


Figure 3.7: Tree C SIMPLE



⁴⁸Node labels adapted by the author for better recognizability.

The DOP-TREE-SIMPLE solution makes use of a POOL consisting of the same 80,000 tree representations as DOP-TREE-ORIG, yet the representations are simplified by recursively iterating through the tree and checking whether a subtree of the current node contains a v (verb) token. If it does, I leave the subtree untouched. If it does not, I collapse the node into the highest possible node label that does not contain a v token, i.e., the current node label. This way, verb-unrelated constituents like the prepositional phrases *op de vloer* and *op school* can be abstracted into being represented by the same node label, ignoring their internal structure. This can potentially make the POOL more representative by allowing for groupings of similarly structured sentences that do not correspond to each other word for word. By applying this technique to all original trees in the POOL, I am able to reduce the number of unique trees in the POOL from 76,186 to 55,197, resulting in a reduction in variety of approximately 27.5%. The reductions in variety in the test datasets vary, as Table 3.3 illustrates:⁴⁹

Table 3.3: Effects of tree simplification

	Sents	Trees Original	Trees Simple	Red.
POOL	80,000	76,186	55,197	27.5%
Test CORRECT	13,586	12,917	9,918	23.2%
Test RAND	13,586	13,505	11,537	14.6%
Test VERBS	13,586	13,453	10,812	19.6%

Unsurprisingly, the reduction is higher in the correct sentences (POOL and Test CORRECT) as natural sentences follow certain syntactic patterns and form constituents. These elements can neatly be abstracted as shown above. Introducing word and verb order errors negatively impacts the potential for abstraction as constituents are broken apart. This effect is strongest in the RAND sentences where none of the original syntax is necessarily preserved. VERBS shuffles, on the other hand, exhibit a reduction in variety closer to the reduction of their corresponding correct sentences, showing that they retain more abstractable information than RAND. Moreover, the POOL of correct sentences exhibits an even greater reduction in unique tree structures. This indicates that the larger the dataset, the more reduction can be expected as more sentences are likely to exhibit similar syntactic patterns.

Table 3.4: PARSE LOOKUP models

Config	Model	Size Pool
PARSE LOOKUP	DOP-TREE-ORIG	80,000
PARSE LOOKUP	DOP-TREE-SIMPLE	80,000

When parsing an unseen sentence, the PARSE LOOKUP solutions search for the resulting parse tree in the POOL of correct parse trees they have access to. If it finds the parse tree in the POOL, the sentence is classified as correct for it has seen a correct

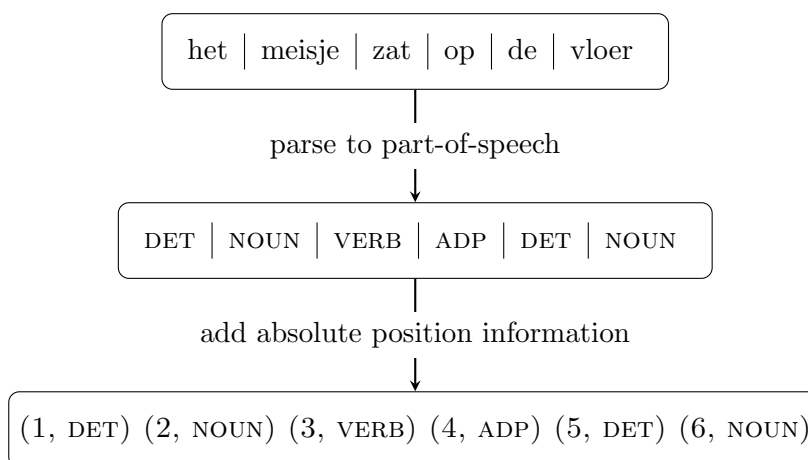
⁴⁹When referring to the RAND and VERBS test datasets, I typically refer to the test sets that contain both the correct sentences and the corresponding shuffled sentences. In order to illustrate the effect of simplifying the parse trees of correct and shuffled sentences independently from one another, here, I report the reduction in the portion of correct sentences in the test datasets separately from the reduction in the portion of shuffled sentences (RAND and VERBS).

sentence with the same sentence structure before. If it does not, the sentence is classified as incorrect. Table 3.4 summarizes the resulting models from the PARSE LOOKUP configuration.

3.4.2 PoS Classifier

Part-of-speech tagging is a common prerequisite for syntactic parsing. To introduce the first machine learning classifier, I take the output of the SPACY part-of-speech tagger and concatenate each part-of-speech label generated with its absolute position in the sequence represented as an integer, resulting in position-part-of-speech tuples. The process is illustrated in Figure 3.8:

Figure 3.8: SPACY-TUP tuple format



The SPACY-TUP solution extracts these tuples for all TRAIN sentences and vectorizes them to make them available as features before training a logistic regression classifier.⁵⁰ The combination of position and part-of-speech information approximates syntactic information in a very simple form. The output of the part-of-speech tagger could be described as a type of morphological information. This allows us to compare the performance of SPACY-TUP, a POS CLASSIFIER that does not have access to explicit syntactic information, to the performance of a classifier that does have access to such information, as will be explained in the following section. SPACY’s fast processing speed allows for training the model on the whole set of train sentences (Table 3.5).

Table 3.5: POS CLASSIFIER model

Config	Model	Size Train
POS CLASSIFIER	SPACY-TUP	2,231,602

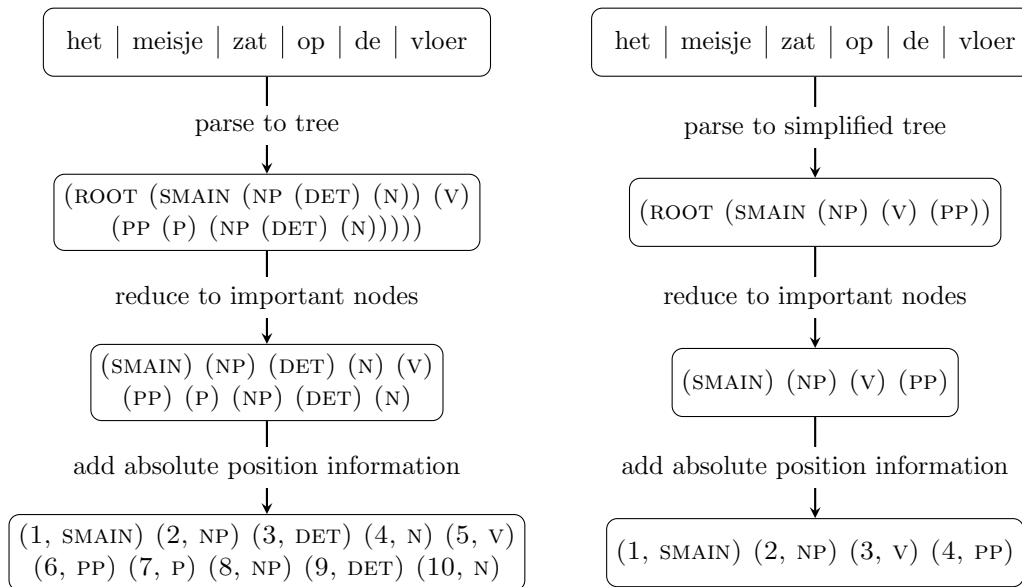
⁵⁰https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html, last accessed: 15.08.2023. The vectorizer generates a vocabulary of all possible tuples that exist in the train data. Each input sentence is converted to a one-hot-vector with an entry of 0 for the absence of a certain tuple and 1 for its presence.

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html, last accessed: 15.08.2023. Uses default parameters except for the maximum number of iterations, which I set to 1000 to ensure convergence.

3.4.3 Parse Classifier

Although the available TRAIN data that could be parsed with the DISCO-DOP parser is small, I want to compare the performance of SPACY-TUP to the performance of classifiers built according to the same principle, but having access to the syntactically richer output of the DISCO-DOP parser. I use the same approach to integrating positional information by creating tuples of position–node information.

Figure 3.9: DOP-TUP-ORIG tuple format Figure 3.10: DOP-TUP-SIMPLE tuple format



As Figures 3.9 and 3.10 illustrate, I test two different tree variants for the PARSE CLASSIFIER configuration, which is in analogy to the PARSE LOOKUP approach. DOP-TUP-ORIG uses tuples extracted from the original tree structures, while DOP-TUP-SIMPLE uses tuples extracted from the simplified tree structures. Table 3.6 summarizes the resulting models from the PARSE CLASSIFIER configuration:

Table 3.6: PARSE CLASSIFIER models

Config	Model	Size Train
PARSE CLASSIFIER	DOP-TUP-ORIG	80,000
PARSE CLASSIFIER	DOP-TUP-SIMPLE	80,000

3.4.4 Transformer Classifier

For all three different transformer models in the TRANSFORMER CLASSIFIER approach, the experimental configuration is identical. I fine-tune the off-the-shelf models by installing a sequence classification head on top of the models.⁵¹ The sequence classification head takes the raw output of the transformer models, which is a vector that has

⁵¹(Footnote continues on the next page.) https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertForSequenceClassification, last accessed: 15.08.2023. https://huggingface.co/docs/transformers/model_doc/roberta#transformers.RobertaForSequenceClassification, last accessed: 15.08.2023.

undergone mathematical modifications according to the transformer models’ architectures and their learned parameters, and predicts whether a given sentence is correct or incorrect based on that output. During the fine-tuning process, it constantly adjusts its parameters to make its predictions more accurate. As previously explained, due to the transformers’ high dimensionality and deep neural architecture, it is uncertain whether transformer models can reliably solve natural language processing tasks that require word order information. Additionally, the experiment lets me explore if and to what extent different transformer architectures influence the transformers models’ performance scores. Due to computational limitations, I limit the amount of correct train sentences for the transformer experiments to one million (Table 3.7):

Table 3.7: TRANSFORMER CLASSIFIER models

Config	Model	Size Train
TRANSFORMER CLASSIFIER	BERTJE	1,000,000
TRANSFORMER CLASSIFIER	ROBBERT	1,000,000
TRANSFORMER CLASSIFIER	GPT-2	1,000,000

3.5 Results

With the exception of the PARSE LOOKUP configuration, which does not require training, I train all models of all configurations on both the RAND and VERBS train datasets with the amount of train sentences specified in the previous sections. As it has been

Table 3.8: Average F_1 score of all models on Test RAND and Test VERBS

Config	Model	Train Test	POOL/RAND		VERBS
			RAND	VERBS	VERBS
PARSE	DOP-TREE-ORIG		0.41	0.41	–
LOOKUP	DOP-TREE-SIMPLE		0.58	0.56	–
POs CLASSIFIER	SPACY-TUP		0.72	0.59	0.68
PARSE	DOP-TUP-ORIG		0.78	0.68	0.73
CLASSIFIER	DOP-TUP-SIMPLE		0.77	0.71	0.73
TRANSFORMER CLASSIFIER	BERTJE		0.99	0.77	0.98
	ROBBERT		1.00	0.74	0.99
	GPT-2		1.00	0.68	0.98

established that VERBS shuffles are a subset of RAND shuffles in the same way that verb order errors are a subset of word order errors, I test the classifiers trained on the RAND train dataset on both the RAND and VERBS test datasets. If classifiers trained on generic word order errors (RAND) could already achieve high performance scores on the more restricted subset of generic verb order errors (VERBS), computational resources

https://huggingface.co/docs/transformers/model_doc/gpt2#transformers.GPT2ForSequenceClassification, last accessed: 15.08.2023.

could potentially be saved by not having to tailor the train data to specific word order error types. Table 3.8 shows the results for all experiments on detecting generic word and verb order errors. For the TRANSFORMER CLASSIFIER configuration, I report the average performance score over three models to account for the random initialization classification heads’ parameters when fine-tuning.

3.6 Discussion

All configurations exhibit higher performance scores on the RAND data than on the VERBS data, indicating that generic word order errors are easier to detect than generic verb order errors. This is unsurprising as sentences permuted according to the VERBS shuffle method generally preserve more of the original syntax of the correct sentences they are derived from. Additionally, training on data permuted according to the VERBS shuffle method typically results in significantly higher performance scores on the VERBS test data than training on data permuted according to the RAND shuffle method. This stresses the importance of task-specific training data, even though VERBS shuffles are a subset of RAND shuffles and RAND-trained classifiers should have seen VERBS shuffles during their training. The naive PARSE LOOKUP approach exhibits the lowest average F_1 score on both the RAND and VERBS test datasets. Yet, simplifying the tree structures results in a significant increase in performance. The combination of absolute positional and part-of-speech information in the POS CLASSIFIER configuration outperforms the PARSE LOOKUP both in the detection of generic word and of generic verb order errors. However, the PARSE CLASSIFIER models, which have access to syntactically richer information than the POS CLASSIFIER model, outperform the latter significantly. This is despite the PARSE CLASSIFIER models having been trained on only a fraction of the data the POS CLASSIFIER has been trained on (3.6%). This confirms that syntactic information is highly beneficial for solving the tasks of word and verb order error detection. Finally, all transformer models are able to achieve almost perfect performance scores when both trained and tested on data permuted according to the same shuffle methods. When RAND-trained models of the TRANSFORMER CLASSIFIER configuration are tested on the VERBS test data, their performance scores differ significantly. As it seems, BERTJE is able to generalize the generic word order errors it has seen during the training process most efficiently when tasked to detect generic verb order errors, outperforming both ROBERT and GPT-2. Nonetheless, the extremely high performance scores all three transformer models exhibit when trained and tested on data permuted according to the same shuffle method clearly indicate that transformer-based models are able to effectively solve the task of detecting word and verb order errors, which, as explained above, benefits greatly from access to syntactic information. The precise mechanisms by which they are able to achieve this, however, will need to be the subject of future studies and could be related to co-occurrence probabilities as suggested by O’Connor and Andreas (2021). I will now discuss each of the configurations and their performances in more detail.

3.6.1 Parse Lookup

A key limitation of the PARSE LOOKUP approach is the fact that natural language is recursive and, in theory, can generate sequences of infinite length. Basing the classification on a POOL of previously seen correct parses cannot account for this generative

power. While limiting the maximum number of tokens in a sentence to 50 and simplifying the tree structures to abstract verb-unrelated constituents can help mitigate this issue and, in the case of the simplification, measurably increases performance, the size of the POOL of correct parses is simply not large enough to be able to effectively represent natural language. This can clearly be seen when looking at the confusion matrices, where both models overpredict the incorrect category (Tables 3.9, 3.10, 3.11, and 3.12):⁵²

Table 3.9: Confusion matrix
DOP-TREE-ORIG on Test RAND

Gold	Predicted	
	incorrect	correct
incorrect	13,438	148
correct	12,476	1,110

Table 3.10: Confusion matrix
DOP-TREE-ORIG on Test VERBS

Gold	Predicted	
	incorrect	correct
incorrect	13,423	163
correct	12,476	1,110

Table 3.11: Confusion matrix
DOP-TREE-SIMPLE on Test RAND

Gold	Predicted	
	incorrect	correct
incorrect	11,009	2,577
correct	8,378	5,208

Table 3.12: Confusion matrix
DOP-TREE-SIMPLE on Test VERBS

Gold	Predicted	
	incorrect	correct
incorrect	10,341	3,245
correct	8,378	5,208

While DOP-TREE-SIMPLE is able to correctly identify almost five times as many correct sentences, the number of incorrect sentences that it misclassifies as correct also drastically increases. While it is possible that due to the method of pseudo data generation, i.e., the shuffle methods, some of these sentences labeled as incorrect are actually correct Dutch sentences as explained in Section 3.1.1, the problem clearly must stem from the combination of shuffled data and the simplification process: Syntactic parsers are typically trained on correct data only. Thus, they are likely to overpredict correct phrases. A common way of nominalizing a verb in Dutch, for example, is by placing a definite article before the infinitive. The verb *eten* can become the noun *het eten* by pre-appending the definite article *het*, resulting in a meaning close to ‘the act of eating’ or, in this particular case, its more common meaning ‘the food’. If in a sentence pair such as in Examples 3 and 4 *eten* was a verb token in the original sentence, which in its shuffled version would appear next to a definite article, it is impossible for the parser to know that the “intended” part-of-speech of the token was of the type verb.

- (3) *Het brood is nog te eten.*
 the bread is still to eat-INF
 ‘The bread is still edible.’

⁵²As both the RAND and VERBS datasets contain the same correct sentences and the PARSER LOOKUP approach is a heuristic classification approach, the predictions for the correct sentences are the same in both datasets, differing only per model.

- (4) *Het eten is nog brood te.*
 the food is still bread to
 ~ ‘The food is still bread to.’

The parser does not have access to information about the original sentence; each parse is independent of the context in earlier parses. Human speakers of Dutch may experience a similar effect. When reading *het eten* at the beginning of the shuffled sentence, a speaker of Dutch may mistake *het eten* for the subject of the sentence, without noticing immediately that the word order of the entire sentence has been permuted randomly. This, in turn, can lead to an oversimplification during the simplification process of the tree structures for all nodes that do not contain verb tokens are collapsed. If the parser is unable to identify verb tokens because their shuffled position causes them to resemble other, verb-unrelated constituents, the simplification process eliminates them entirely when collapsing the node. Therefore, while simplifying tree structures does increase the overall performance, it is a rather unreliable approach that may lead to unintentional exclusions of vital components.

3.6.2 PoS Classifier

The POS CLASSIFIER and all other following machine learning-based approaches, on the other hand, are more robust as they do not rely on simplifying the syntactic information they receive as input in order to be able to represent syntactic patterns. However, the information the POS CLASSIFIER has access to is still very limited. It only has access to morphological information in combination with absolute positional information. While this does allow the classifier to learn that the presence or absence of certain position-part-of-speech tuples can be indicators of correct or incorrect word order, the overall performance is still rather low. When looking at the confusion matrices of the RAND-trained and VERBS-trained classifiers on the VERBS test dataset, we can see that the RAND-trained classifier clearly overpredicts sentences to be correct (Tables 3.13 and 3.14):

Table 3.13: Confusion matrix
 SPACY-TUP Train RAND on Test
 VERBS

	Predicted	
	incorrect	correct
Gold incorrect	6,048	7,538
Gold correct	3,225	10,361

Table 3.14: Confusion matrix
 SPACY-TUP Train VERBS on Test
 VERBS

	Predicted	
	incorrect	correct
Gold incorrect	9,284	4,302
Gold correct	4,299	9,287

This is not very surprising as classifiers trained on the RAND data are likely to expect a completely broken syntax in order to classify a sentence as incorrect. Since VERBS shuffles retain most of the original word order, they are more likely to resemble correct sentences than RAND shuffles. This means that although VERBS shuffles are a subset of RAND shuffles, the classifier seems to not be sensitive enough to verb order errors if trained on generic word order errors only. This tendency can be observed in all following machine learning-based approaches. Additionally, the part-of-speech tagger is subject to the same challenge as explained in the previous section: Since the part-of-speech tagger is trained on correct data only, the permutation of word order can result

in the part-of-speech tagger assigning verbs and verb-unrelated tokens a different part-of-speech tag than it would in the correct sentence, which can lead to an overprediction of sentences to be correct. This is likely one of the reasons that the average F_1 score of POS CLASSIFIER models trained on task-specific data, i.e., classifiers trained on RAND data and tested on RAND data as well classifiers trained on VERBS data and tested on VERBS data is limited to roughly 0.70 – despite being trained on the largest number of sentences out of all approaches.

3.6.3 Parse Classifier

The models of the PARSE CLASSIFIER approach face the same challenges as both the PARSE LOOKUP and the POS CLASSIFIER configurations. The training of the parser on correct data can lead to overprediction of correct sentences by assigning the most likely labels based on the permuted context. Subsequent simplification of parse trees can potentially amplify this effect by omitting elements that would otherwise have been preserved if the parser does not recognize the constituent to contain a verb token.⁵³ Nonetheless, both DOP-TUP-ORIG and DOP-TUP-SIMPLE significantly outperform both the PARSER LOOKUP and the POS CLASSIFIER approaches. This is especially remarkable in the case of the POS CLASSIFIER approach, as the PARSE CLASSIFIER models have been trained on only a fraction of the data the former has been trained on (approximately 3.6%).⁵⁴ This implies that syntactic information is crucial when attempting to solve the tasks of word and verb order error detection. Interestingly, both DOP-TUP-ORIG and DOP-TUP-SIMPLE perform almost equally well when trained on task-specific data, indicating that the evident advantage DOP-TREE-SIMPLE exhibits over DOP-TREE-ORIG does not translate to the machine learning-based approaches. The PARSE CLASSIFIER models are able to learn representations for correct syntactic patterns from the position–node tuples alone and do not require further simplification. In the case of the RAND data, this simplification even seems to be harmful, albeit only slightly. When looking at the confusion matrices for the DOP-TUP-ORIG and DOP-TUP-SIMPLE models trained and tested on RAND data, we can see that the slight drop in performance DOP-TUP-SIMPLE exhibits can be associated with an overprediction of incorrect sentences as correct, which aligns with the previously identified shortcomings of the simplification process (Tables 3.15 and 3.16).

Table 3.15: Confusion matrix
DOP-TUP-ORIG Train RAND on Test
RAND

Gold	Predicted	
	incorrect	correct
incorrect	10,149	3,437
correct	2,477	11,109

Table 3.16: Confusion matrix
DOP-TUP-SIMPLE Train RAND on Test
RAND

Gold	Predicted	
	incorrect	correct
incorrect	9,720	3,866
correct	2,478	11,108

Yet, simplifying the tree structures seems to be helpful when a RAND-trained classifier is tasked to detect verb order errors. This could potentially be due to the fact that

⁵³Rather: What would have been a verb token in the original sentence.

⁵⁴POS CLASSIFIER: 2,231,602 sentences. PARSE CLASSIFIER models: 80,000 sentences.

verb-unrelated constituents are collapsed. Where in the original parses, large subsequences of a given sentence are typically preserved when permuted according to the VERBS shuffle method, in simplified parses, these chunks are likely to be summarized under only a handful of node labels, weakening this effect. This way, the RAND classifier, being less sensitive to verb order errors, can potentially focus on more relevant tuples for identifying them.

3.6.4 Transformer Classifier

Finally, all models of the TRANSFORMER CLASSIFIER approach are able to achieve near-perfect performance scores when trained on task-specific data, clearly indicating that transformer-based models can reliably solve the detection of generic word and verb order errors, which benefits from syntactic information as shown above. The high performance scores on generic verb order errors are especially impressive as here, other word order-unrelated indicators of improbable sequences such as incorrect agreement between verb-unrelated tokens are eliminated for all verb-unrelated tokens remain in their original order. However, training on task-specific data seems necessary in order to sensitize the models for the specific word order error type that needs to be detected, which is a tendency that is reflected in all other machine learning-based approaches as well. Otherwise, the classifiers tend to overpredict the correct category as they expect more disruptions to the syntax when trained on RAND data than present in the VERBS test data which preserves large chunks of the original, correct sentences. The ROBBERT-based models ever so slightly outperform both BERTJE and GPT-2 on both the RAND and VERBS test datasets and the VERBS test dataset, respectively. With performance scores this high, the question arises whether the misclassifications that do happen are in fact shortcomings of the classifier itself or whether the sentences involved are labeled incorrectly due to the method of pseudo data generation. When looking at the best confusion matrices of the ROBBERT model,⁵⁵ we can see that the classifier predicts three gold correct sentences as being incorrect in the RAND test dataset (Tables 3.17 and 3.18):

Table 3.17: Best confusion matrix
ROBBERT Train RAND on Test
RAND

	Predicted	
Gold	incorrect	correct
incorrect	13,553	33
correct	3	13,583

Table 3.18: Best confusion matrix
ROBBERT Train VERBS on Test
VERBS

	Predicted	
Gold	incorrect	correct
incorrect	13,403	183
correct	74	13,512

I illustrate these sentences below. Arguably, none of these sentences, or sequences, can form a grammatically correct stand-alone sentence in Dutch because they are either incomplete or exhibit word order errors.⁵⁶ The sequence *ze autosleutels zoekt* could appear in a subordinate clause. Nonetheless, all three sequences should indeed have been labeled as incorrect in the context of this experiment.

⁵⁵The confusion matrices here show the best predictions per category achieved by any of the three ROBBERT models I trained.

⁵⁶In the case of the incorrect sequences, I do not provide a translation.

- **digitale vaardig zijn*
- **soms ook is er twijfel*
- *ze autosleutels zocht* ‘she was looking for car keys’

When looking at the 33 gold incorrect sentences the classifier predicts to be correct, a total of seven sequences are grammatically possible in Dutch, while only three could be proper stand-alone sentences (sequences that could be proper stand-alone sentences are marked with a +):

- +*je kunt karaktereigenschappen ook erven* ‘you can also inherit character traits’
- +*tandartsen worden door tandartsassistenten bijgestaan* ‘dentists are assisted by dental assistants’
- +*waarom doen ze dat* ‘why do they do that’
- *hoe de beleving was* ‘how the experience was’
- *en er wordt gereorganiseerd* ~ ‘and reorganization is taking place’
- *waar ze zitten* ‘where they sit’
- *wie dat nou doet* ‘who is doing that’

This shows that the amount of grammatically correct sentences the RAND shuffle method generates is indeed negligible. The majority of misclassified instances in this category are indeed incorrect and therefore misclassified. When looking at the misclassifications within the VERBS test dataset, a similar tendency can be observed for the gold correct sentences: Many of the sequences the classifier predicts to be incorrect but are labeled as correct are incomplete or could not typically form stand-alone sentences because of other grammatical reasons. However, the absolute number of these sentences is still small, indicating that the preprocessing criteria were effective. Shifting the focus to the sentences generated according to the VERBS shuffle method, there are 183 sequences the classifier predicts to be correct even though they are labeled as incorrect. Since the set of VERBS permutations for a given sentence is smaller than the set of RAND permutations as explained in Section 3.1.1, the likelihood of a correct permutation being picked randomly is also higher. Table 3.19 showcases a selection of the permutations the VERBS shuffle method rendered that are grammatically possible. The permutations marked as possible stand-alone sentences are thus mislabeled: The first two permutations result in correct sequences because the SPACY part-of-speech tagger mistakes the adjective *bepaalde* ‘certain’ for a verb and rearranges its position in such a way that it happens to appear before nouns, resulting in a grammatically possible sequence (‘the director’s liability also applies to directors of certain associations’; ‘organelles are parts of the cell with a particular function’).⁵⁷ The following two permutations constitute alternative word orders (‘the first fossils of Barosaurus were discovered in 1889’; ‘researchers have shared new pictures of the Mariana Trench’) and the final two permutations would form correct sequences if they appeared in subordinate clauses (‘wonders always happen unexpectedly’; ‘the government must be in

⁵⁷I provide translations for the original sentences. The adjective *bepaalde* is an inflected form of the past participle of the verb *bepalen* ‘to determine’. Its surface form is equivalent to *bepaalde*, which is the finite past form of the verb. This is likely the cause of the assignment of an incorrect part-of-speech.

Table 3.19: Illustration of mislabeled VERBS permutations

Original	Permutation	Type
<i>de bestuurders-aansprakelijkheid geldt ook voor bestuurders van bepaalde verenigingen</i>	⁺ <i>de bestuurders-aansprakelijkheid geldt ook voor bepaalde bestuurders van verenigingen</i>	adjective mistaken for verb and
<i>organellen zijn onderdelen van de cel met een bepaalde functie</i>	⁺ <i>organellen zijn onderdelen van de bepaalde cel met een functie</i>	placed before noun
<i>de eerste fossielen van barosaurus werden ontdekt in 1889</i>	⁺ <i>de eerste fossielen van barosaurus werden in 1889 ontdekt</i>	alternative word order
<i>onderzoekers hebben nieuwe beelden gedeed van de marianentrog</i>	⁺ <i>onderzoekers hebben nieuwe beelden van de marianentrog gedeed</i>	
<i>wonderen gebeuren altijd onverwacht</i>	<i>wonderen altijd onverwacht gebeuren</i>	order correct in
<i>de regering moet in de hoofdstad zijn</i>	<i>de regering in de hoofdstad moet zijn</i>	subordinate clauses

the capital’). While the performance of the TRANSFORMER CLASSIFIER models could potentially be even higher if mislabeled sentences were filtered out, many of the 183 misclassified sentences are in fact incorrect and should have been classified as such. Table 3.19 also conveniently illustrates one of the main challenges learners face when studying Dutch syntax rules: Verbs can appear in different positions depending on the clause type they appear in. Now that the general capabilities of the classification approaches in detecting generic word and verb order errors have been established, I want to look at verb order errors learners of Dutch are in fact likely to make. The following chapter will therefore explore how the established models perform when tested on a dataset that aims to emulate learner error tendencies.

Chapter 4

Detection of Learner-Informed Verb Order Errors

The second part of this thesis focuses on the detection of learner-informed verb order errors. In the first part, I have explored the performances of various classifiers in the detection of generic word and verb order errors. Generic word and verb order errors can, nevertheless, only provide an insight into the general capability of the classification approaches in detecting erroneous word order as they encompass any and all word and verb order errors that could be made given a particular sentence. In reality, however, learners do not typically misplace words randomly but word order errors follow certain patterns, or error tendencies. In Dutch, the correct placement of verbs in a sentence is one of these error-prone areas. In an ideal scenario, a classifier should be trained on data that is annotated for these specific types of errors. Yet, learner corpora are sparse, especially outside of the English domain. To the best of my knowledge, a Dutch dataset annotated for verb order errors does not exist. Therefore, I perform a structural analysis of 200 clauses extracted from 184 unique learner sentences which are obtained from a corpus of genuine learner data graciously made available to me by the Instituut voor Levende Talen at KU Leuven, which I call the LEUVEN corpus. By means of structural analysis, I extract verb order error tendencies which serve as an informational resource for the construction of a final synthetic evaluation dataset INFO. In the following sections, I will describe the LEUVEN corpus, the method of structural analysis, as well as the generation of the learner-informed pseudo data. Finally, I will explore the performance of the already-established classification approaches on both the synthetic evaluation dataset INFO and the 184 genuine learner sentences (LEARN) to assess how well their general capability in detecting erroneous word order translates to real-life application scenarios.

4.1 Generation of Learner-Informed Pseudo Data: Dataset Info

In analogy to the first part of this thesis, I will first describe the method of pseudo data generation. All possible learner-informed verb order errors for a given sentence s intuitively must be a subset of generic verb order errors, as the latter comprise any and all possible misplacements of verbs (4.1):

$$S_{Info}(s) \subseteq S_{Verbs}(s) \subseteq S_{Rand}(s) \quad (4.1)$$

Contrary to the shuffle methods applied in the first part of this thesis, the generation of learner-informed verb order errors is not trivial. It requires an understanding of what kinds of verb order errors learners are likely to make, which in turn requires at least a basic understanding of Dutch syntax. In the following section, I will therefore introduce some of the most prominent syntactic peculiarities of the Dutch language before introducing my method of structural analysis, which I employ to extract verb order error tendencies from the LEUVEN corpus of genuine learner data. Finally, I describe the curation process of the INFO test dataset and present the results the models established in the first part of this thesis are able to achieve on the learner-informed evaluation data.

4.1.1 Dutch Verb Order

Verb order in Dutch can be challenging for second-language learners. As has been mentioned earlier, the placement of verbs in a Dutch sentence depends on a number of factors, among which the clause type, the (non-)finiteness of the verb, and the presence or absence of prepositional and infinitival complements. In principle, there are two main positions in which a verb can occur in Dutch main and subordinate clauses: the verb-second position and the verb-final position (Broekhuis and Corver, 2016).⁵⁸ The verb-second position is exclusively assumed by finite verbs in main clauses. In the context of this thesis, I define finite verbs to be any conjugated verb form that is not the infinitive or the past participle. This entails that in main clauses, non-finite verbs are found in the verb-final position. In subordinate clauses, both the finite verb and any number of non-finite verbs are found in the verb-final position.⁵⁹ Examples (5) through (7) illustrate these general verb order patterns:

- (5) *Ik koop een boek.*
 I buy-1SG.PRES a book
 ‘I buy a book.’

Remark: Main clause. The finite verb occupies the verb-second position.

- (6) *Ik wil een boek kopen.*
 I want-1SG.PRES a book buy-INF
 ‘I want to buy a book.’

Remark: Main clause. The finite verb occupies the verb-second position. The non-finite verb occupies the verb-final position.

⁵⁸The verb-second position is also commonly referred to as the *V2* position. Both verb-second and verb-final could also be understood as clause-second and clause-final as verb-second corresponds to the second element in the clause and verb-final corresponds to the final element in the clause. In order to keep the focus on the verbs and their positioning, however, I will continue to use the terms verb-second and verb-final.

⁵⁹In many cases, the finite verb can appear either before or after the non-finite verb form(s). These two different variations are commonly referred to as the *red order* and the *green order*. The terms result from Pauwels (1953)’s research on dialectal variation in the verb order of the Dutch subordinate clause. She used the two colors to visualize the two different verb orders in illustrations and maps. See Bloem (2021), who finds that this variation appears to be linked to processing complexity, for more details.

- (7) *Hij weet dat | ik een boek zou willen kopen.*
 He knows that | I a book would-1SG.PRES want-INF buy-INF
 ‘He knows that I would want to buy a book.’

Remark: Subordinate clause. Both the finite and the non-finite verbs occupy the verb-final position.⁶⁰

This has resulted in typological resources such as The World Atlas of Language Structures describing Dutch to exhibit both SOV and SVO syntax patterns (Dryer, 2013).⁶¹ Yet, Dutch clauses follow a true SVO pattern only in main clauses that contain a single, finite verb form and start with the subject of the clause, which often is not the case as Example (8) illustrates. The clause-initial position can even be occupied by an entire complement clause serving as a single constituent (Example (9)). In almost all other cases, the default syntax pattern in Dutch can be described as SOV.

- (8) *Morgen koop ik een boek.*
 tomorrow buy-1SG.PRES I a book
 ‘Tomorrow, I am going to buy a book.’

Remark: Main clause. The finite verb occupies the verb-second position. The subject does not occupy the clause-initial position.

- (9) *[Dat jij een boek ging kopen], wist ik al.*
 [that you a book went buy] know-1SG.PST I already
 ‘I already knew that you were going to buy a book.’

Remark: Main clause. The finite verb occupies the verb-second position. The clause-initial position is occupied by a finite complement clause serving as a single constituent.

With the most crucial syntactic patterns covered, I now want to draw the attention to two other noteworthy syntactic phenomena:

- a) In polar questions, the clause-initial position remains phonetically empty, resulting in a surface structure where the finite verb appears to be in verb-first position.
- b) Certain complements can follow verbs that occupy the verb-final position. They occupy the post-verbal position.

For a), consider the open question in Example (10) and the polar question in Example (11):

- (10) *Waarom koop je een boek?*
 why buy-2SG.PRES you a book
 ‘Why do you buy a book?’

Remark: Open question. The finite verb occupies the verb-second position. Analogous to main clause.

⁶⁰Technically, the complementizer *dat* is also part of the subordinate clause. I draw the boundary in the manner shown here to lay the focus on the words that are crucial for illustrating the word order within the subordinate clause. The same applies to all following examples where I have to distinguish between multiple clauses.

⁶¹SOV: Subject-Object-Verb. SVO: Subject-Verb-Object.

- (11) *Koop je een boek?*
 buy-2SG.PRES you a book
 ‘Do you buy a book?’

Remark: Polar question. At the surface level, the finite verb occupies the verb-first position (clause-initial).

In open questions, the finite verb occupies the verb-second position. They function in a manner akin to regular main clauses. In polar questions, according to generative grammar, the finite verb also occupies the verb-second position. According to Broekhuis and Corver (2016), the clause-initial position “remains phonetically empty”, which results in a surface form where the finite verb appears to occupy the clause-initial or verb-first position. As the purpose of this thesis is to lay the foundations for a solution that can provide learners with informative feedback, I opt to utilize the more pragmatic description, designating the correct placement of finite verbs in polar questions as the verb-first position.

For b), consider Examples (12) and (13):

- (12) *Ik wilde voorstellen [(om) zijn boek te lezen].*
 I want-1SG.PST suggest-INF [(COMP) his book to read-INF]
 ‘I wanted to suggest reading his book.’

Remark: Main clause with infinitival complement. The infinitival complement occupies the post-verbal position.

- (13) *Hij weet dat | ik niet nadacht [over het boek].*
 he knows that | I not think-1SG.PST [about the book]
 ‘He knows that I was not thinking about the book.’

Remark: Subordinate clause with prepositional complement. The prepositional complement occupies the post-verbal position.

Contrary to finite subordinate clauses, infinitival complements are non-finite argument clauses required by a substantial collection of Dutch verbs.⁶² They are characterized by the presence of the particle *te* and can often be introduced by the complementizer *om* (Broekhuis and Corver, 2016).⁶³ They typically occur in the post-verbal position. Similarly, prepositional complements can occur in the post-verbal position: “[N]ominal complements normally precede, complement clauses normally follow, and PP-complements can normally either precede or follow the clause-final verb(s)” (Broekhuis and Corver, 2016).⁶⁴ The more complex a prepositional complement is, the more likely it is to be placed in the post-verbal position. Table 4.1 summarizes the default positions of finite verbs, non-finite verbs, and the previously introduced types of complements.⁶⁵ The positions directly translate to the error categories according to which I analyze the LEUVEN learner data. However, in order to identify errors in a sentence, it is necessary

⁶²According to Broekhuis and Corver (2016), infinitives such as *kopen* in Examples (6) and (7) are also referred to as bare infinitivals. I, however, use the term infinitival exclusively for non-finite argument clauses that contain the particle *te*.

⁶³The complementizer *om* is always optional (Broekhuis and Corver, 2015) when invoked by a verb that requires an infinitival complement such as *voorstellen*. Other verbs may require infinitival complements where the use of *om* is prohibited. When *om* is used to indicate a purpose as in *Ik wil naar huis gaan om mijn boek te lezen*. ‘I want to go home to read my book.’, its use is obligatory.

⁶⁴PP: Prepositional phrase.

⁶⁵As mentioned before, prepositional complements can also occur before the verb-final position.

Table 4.1: Default positions per clause and verb/complement type

	Clause Type		Question Type	
	Main	Subordinate	Polar	Open
FV	verb-second	verb-final	verb-first	verb-second
NFV	verb-final	verb-final	verb-final	verb-final
IC		post-verbal		
PC		post-verbal*		

- FV – *Finite verb*
 NFV – *Non-finite verb*
 IC – *Infinitival complement*
 PC – *Prepositional complement*

to initially formulate a hypothesis regarding the specific meaning the learner intended to convey, known as a *target hypothesis*. In the following section, I will therefore introduce the concept of target hypotheses before elaborating on the method of structural analysis in more detail.

4.1.2 Generating Target Hypotheses

When analyzing erroneous learner sentences, it is necessary to establish a target hypothesis, i.e., a hypothesis about what the learner may have intended to express (Lüdeling, 2008). In other words, one must construct a corrected version of the learner sentence against which to evaluate the erroneous sentence produced by the learner. The generation of target hypotheses, however, is not trivial. Consider the sentence in Example (14):

- (14) **Anna heeft maakte zich grote zorgen om Sam.*
 Anna have-3SG.PRES make-3SG.PST herself big sorrows about Sam.
 ‘Anna was very worried about Sam.’ (Russian A2)⁶⁶

The sentence exhibits elements of both the Dutch simple and compound past tenses. Thus, in principle, two target hypotheses are conceivable (Examples (15) and (16)):

- (15) *Anna heeft zich grote zorgen om Sam gemaakt.*
 Anna have-3SG.PRES herself big sorrows about Sam make-PST.PTCP
- (16) *Anna maakte zich grote zorgen om Sam.*
 Anna make-3SG.PST herself big sorrows about Sam

Both hypotheses conform to the grammatical rules of the Dutch language, and a difference in usage could only be argued if context was available and assumed to be grammatically correct. Nonetheless, there is an argument that supports choosing the target

⁶⁶When providing examples from the corpus, I will also provide the learner’s native language and their level of Dutch according to the CEFR.

hypothesis from Example (15) over Example (16): Natural language is both uttered and written in a linear sequence, which means that the learner must first deliberately have made the choice to use the auxiliary *heeft* which suggests an intended usage of the compound tense. It is therefore more likely that the learner failed to produce the correct past participle than it is that the learner did not mean to make use of the compound tense in the first place. Thus, entirely deleting a word the learner chose to use seems more costly than correcting an incorrectly formed verb form. In ambiguous situations like these, I choose the target hypothesis that appears to be the least costly. Within the scope of this thesis, it is sufficient if the target hypothesis is a valid correction of the learner sentence. It does not have to be the single best correction for its purpose is only to identify potential verb order errors. Additionally, I let all target hypotheses, which I generate either manually or with the help of generative AI, be evaluated by two native Dutch speakers (see Section 5). Having defined the concept of target hypotheses, the following section will provide a more detailed description of the method used for analyzing verb order errors.

4.1.3 Analyzing Verb Order Errors

For the analysis of verb order errors, I adopt the default positions of the elements described in Section 4.1.1 and use them to designate error categories. I distinguish between the following error types:⁶⁷

-
- 1: **verb-second** – Finite verb is not found in the verb-second position in main clause or open question.
 - 2: **verb-final** – Non-finite verb is not found in the verb-final position or finite verb in subordinate clause is not found in the verb-final position.
 - 3: **verb-first** – Finite verb is not found in the verb-first position in polar question.
 - 4: **post-verbal** – Post-verbal prepositional complement or infinitival complement is not found in the post-verbal position or exhibits an internal verb order error.
-

I compare learner sentences to their corresponding target hypotheses (TH) and determine the error type. The error types are to be interpreted in the following way: The finite verb form, in a main clause or open question, has to occupy the second position in the clause – if it does not, there is a verb-second error. Non-finite verb forms (infinitives and past participles) typically have to occupy the final position in the clause – if they do not, there is a verb-final error. In subordinate clauses, the finite verb form follows the same principle as the non-finite verb forms and has to occupy the final position in the clause – if it does not, there is a verb-final error. In polar questions, the finite verb has to occupy the first position of the clause – if it does not, there is a verb-first error. Finally, infinitival complements and complex prepositional complements have to appear after the verb-final position – if they do not, there is a post-verbal error. For the sake of simplicity, I also classify internal verb order errors within infinitival complements as post-verbal errors. Examples (17) through (20) illustrate the different error types:

⁶⁷Post-verbal: Internal verb order errors can occur in infinitival complements where the infinitive, being a non-finite verb form, ought to occupy the verb-final position.

- (17) **Bovendien wij de musea kunnen bezoeken.*
 moreover we the museums can-1PL.PRES visit-INF
 ‘On top of that, we can visit the museums.’ (Arabic A2)
TH: *Bovendien kunnen wij de musea bezoeken.*
Error type: main clause – finite verb – verb-second
- (18) **De mensen | die wonen in dorpen | zijn zo conservatief.*
 the people | who live-3PL.PRES in villages | are so conservative
 ‘The people who live in villages are so conservative.’ (Spanish B1+)
TH: *De mensen | die in dorpen wonen | zijn zo conservatief.*
Error type: subordinate clause – finite verb – verb-final
- (19) **De belge bevolking agressiever dan voor worden?*
 the Belgian population more.aggressive than before become-UNK
 ‘Are the Belgian people becoming more aggressive than before?’ (Arabic A2)
TH: *Wordt de belgische bevolking agressiever dan voorheen?*
Error type: polar question – finite verb – verb-first
- (20) ?*Ik wil verduidelijken dat | ik [over de mensen die kleine misdaden hebben gepleegd] spreek.*
 I want clarify that | I [about the people who small offences have committed] talk-1SG.PRES
 ‘I want to clarify that I am talking about the people who have committed minor crimes.’ (Russian B1)
TH: *Ik wil verduidelijken dat | ik spreek [over de mensen die kleine misdaden hebben gepleegd].*
Error type: subordinate clause – prepositional complement – post-verbal

Example (19) also underlines the importance of the target hypothesis, which in this case determines that the verb in question, *worden*, should actually be 3SG.PRES. The verb form the learner makes use of could either be the infinitive or 3PL.PRES, which is impossible to discern and the reason why I mark its morphological information as unknown (UNK). As for the purpose of this study, it is only the position of the verbs within a sentence that is important for extracting learner error tendencies, constructing a valuable target hypothesis bears the advantage of removing all verb order-unrelated errors an original learner sentence may additionally contain.

While the previously introduced error types provide a decent general overview about the types of errors learners are likely to make, they do not possess the representational power that would be needed in order to recreate such errors. In order to account for this, I additionally analyze the structure of the clause containing the verb order error by annotating it for important elements and their order.⁶⁸ The structures for Examples (17) through (20) are illustrated in Examples (21) through (24):

⁶⁸I have adapted the labels of the elements I present here for better readability. A list of all elements for which I analyze the erroneous clauses can be found in Appendix C.

- (21) **(Bovendien) (wij) (de musea) (kunnen) (bezoeken).*
ADVERB – SUBJECT – OBJECT – FINITE VERB – NON-FINITE VERB
- (22) **(die) (wonen) (in dorpen)*
RELATIVE PRONOUN – FINITE VERB – ADVERBIAL
- (23) **(De belge bevolking) (agressieffer) (dan) (voor) (worden)?*
SUBJECT – ADJECTIVE – PARTICLE – ADVERB – FINITE VERB
- (24) *?(ik) (over de mensen die kleine misdaden hebben gepleegd) (spreek)*
SUBJECT – PREPOSITIONAL COMPLEMENT – FINITE VERB

In combination with the respective error type of the clause, I am able to extract patterns of which elements or absolute positions can be associated with the misplacement of verb tokens. I also want to emphasize that there are many ways to analyze phrase structure. My approach is only one way of approximating syntactic patterns, yet it is practical enough to allow for an efficient reconstruction of observed error tendencies using the SPACY natural language processing toolkit as will be explained in Section 4.1.6. In the following section, I will introduce the LEUVEN corpus as the source of the learner data I analyze by means of the just established analytical framework before presenting the error tendencies I am able to extract from it.

4.1.4 Leuven corpus

The original LEUVEN corpus contains 3,121 unique learner essays that are marked for errors with a semi-standardized system as could be used by language teachers in their daily proceedings. The learners authoring the essays speak a total of 82 different first languages, a full list of which can be found in the data statement in Appendix E.1. Their levels of Dutch range from A2 to C1 in the Common European Framework of Reference for Languages (CEFR). In the dataset, the error tag (P) for *positie* ‘position’ is used to indicate word order errors of any kind. For my analysis with the objective of extracting learner-informed verb order error tendencies, I manually extract 500 (P)-sentences from the essays and analyze them for the presence of verb order errors, resulting in a semi-random selection of 184 sentences with 200 main and subordinate clauses containing verb order errors.⁶⁹ The distribution of first languages and CEFR proficiency levels within the 184 selected sentences is illustrated in Figures 4.1 and 4.2, respectively. The languages follow the ISO 639 codes.⁷⁰ I generate target hypotheses for the selected sentences and proceed to analyze them for the nature of the contained verb order errors by the means of the previously introduced analytic framework. The results of this analysis will be presented in the following section.⁷¹

⁶⁹The essays in the corpus do not appear to be ordered based on any criteria such as the learner’s language proficiency level or their first language. Thus, I review the extracted sentences from top to bottom and manually filter out any sentences where the (P) error is not related to verb order or the content of the sentence is too incomprehensible for me to generate a valuable target hypothesis.

⁷⁰<https://www.iso.org/iso-639-language-codes.html>, last accessed: 15.08.2023. Brazilian Portuguese included in *pt*; Farsi included in *prs*.

⁷¹I generate the target hypotheses for half of the selected sentences manually (second-language speaker, CEFR B2–C1) and for the other half with the help of a generative artificial intelligence model. All target hypotheses are evaluated by two first-language speakers. See Section 5 for more details.

Figure 4.1: Distribution of first languages in selected learner sentences

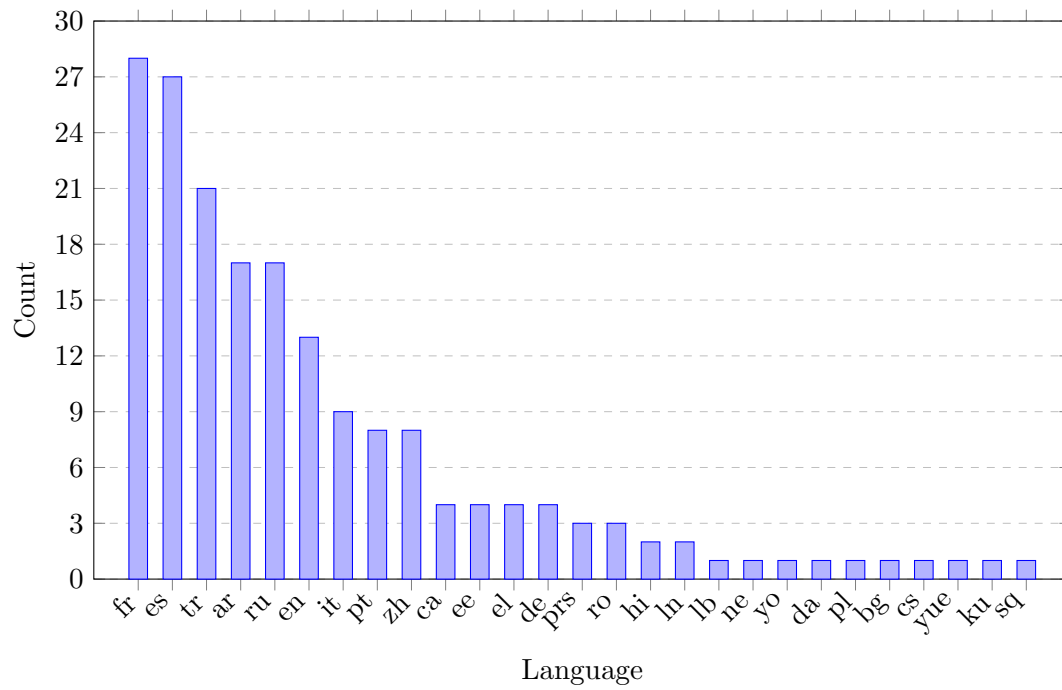
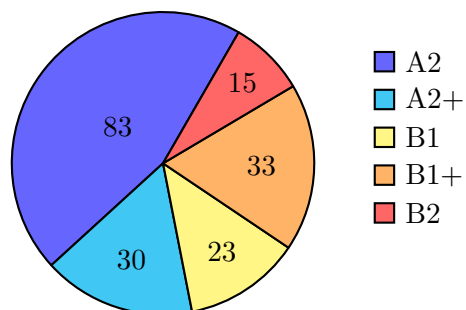


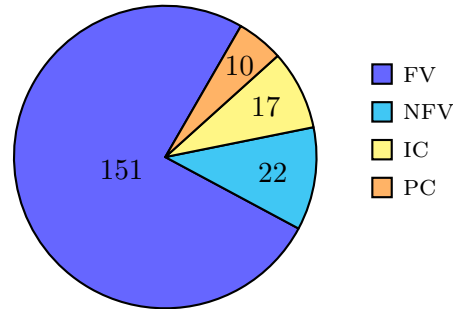
Figure 4.2: Distribution of CEFR levels in selected learner sentences



4.1.5 Identified Error Tendencies

When analyzing the 200 main and subordinate clauses extracted from the learner sentences, we can see that the majority of errors are made in the placement of the finite verb, followed by errors in the placement of non-finite verb forms and lastly by errors resulting from misplacements of infinitival and prepositional complements (Figure 4.3):

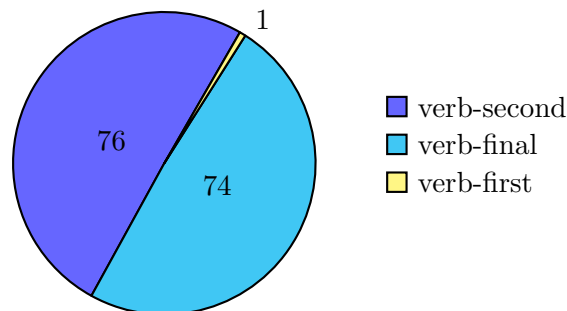
Figure 4.3: Distribution of errors per verb / complement type



Since correctly placing the finite verb in Dutch sentences appears to be the most error-prone challenge for learners by a significant margin (75.5%), this thesis will henceforth focus on verb order errors that result from misplacing the finite verb of a clause. The other error types present interesting directions for future research.

To reiterate, the finite verb in Dutch can assume three positions, depending on which clause type it occurs in: verb-second in main clauses and open questions, verb-final in subordinate clauses, and verb-first in polar questions. Figure 4.4 illustrates the distribution of these error types within the 151 instances of misplaced finite verbs:

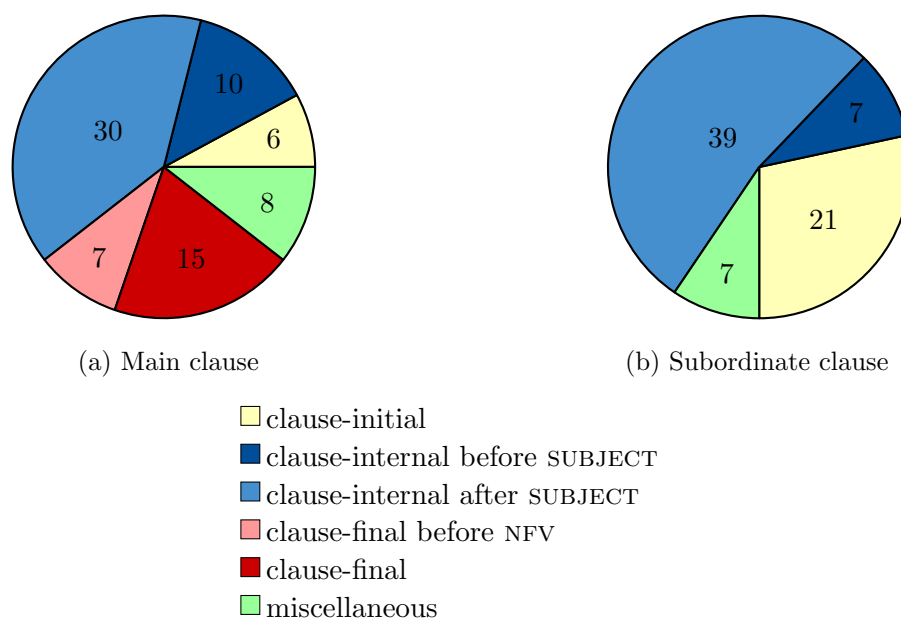
Figure 4.4: Distribution of finite verb error types



As the data indicates, the finite verb is misplaced about equally as often in main clauses (76 instances), where it should occupy the verb-second position, as it is in subordinate clauses (74 instances), where it should occupy the verb-final position. Misplacements of the finite verb in open questions, where it should occupy the verb-first position, occurred only once in the examined sentences.

The structural analysis, the results of which are illustrated in Figure 4.5, reveals that 89.5% of the finite verb errors in main clauses can be associated with either absolute positions (clause-initial and clause-final), the subject of the clause (clause-internal before

Figure 4.5: Distribution of incorrect positions of finite verb in main and subordinate clauses



SUBJECT and clause-internal after SUBJECT), or other, non-finite verb forms (clause-final before NFV).⁷² In subordinate clauses, where the verb-final position is the default position for all verb forms, 90.5% of the finite verb errors can be associated with either the clause-initial absolute position or the subject of the clause (clause-internal before SUBJECT and clause-internal after SUBJECT).⁷³ Misplacements of the finite verb that cannot be associated with any of these categories are classified as miscellaneous. Note that contrary to the error types of Figure 4.4, which denote the desired word order, the incorrect position categories resulting from the structural analysis illustrated in Figure 4.5 denote the positions the finite verb was actually placed in by the learner, which facilitates reconstructing these errors. Examples (25) through (30) illustrate each of the incorrect position categories:

⁷²When the finite verb is found in an absolute position either at the beginning or at the end of the clause, this does not necessarily mean that its positioning cannot be related to other elements. If a finite verb occurs in the clause-initial position, it can still be immediately followed by the subject, for example. I choose to conduct my analysis in terms of absolute positions in the case of the clause-initial and clause-final positions because these absolute positions are more straightforward to replicate in the INFO dataset. At the same time, the results of the analysis do not try to establish a causal relationship between the subject or any other element and the misplacement of the finite verb, although conceivable. It is a mere observation that misplaced finite verbs often occur in the immediate neighboring context of the subject. Further research would be required to establish whether the subject is the cause of the misplacement.

⁷³The categories clause-internal before SUBJECT and clause-internal after SUBJECT denote that the finite verb is found immediately next to the subject but does not occupy any of the absolute positions or the clause-final before NFV position.

- (25) **Ik mag geen alcoholische dranken drinken en | mag ik niet roken.*
 I may no alcoholic beverages drink and | may-1SG.PRES I not
 smoke-INF
 ‘I mustn’t drink and I mustn’t smoke.’ (Kurdish, A2)
TH: *Ik mag geen alcoholische dranken drinken en ik mag niet roken.*
Error type: main clause – finite verb – verb-second
Position: clause-initial
- (26) **De volgende dagen misschien kunnen we een ballonvaart maken.*
 the following days maybe can-1PL.PRES we a balloon_ride make-INF
 ‘Maybe we can go for a balloon ride in the upcoming days.’ (Turkish, A2)
TH: *Misschien kunnen we de volgende dagen een ballonvaart maken.*
Error type: main clause – finite verb – verb-second
Position: clause-internal before SUBJ
- (27) **We leven in een wereld | waarin alles is digitaal geworden.*
 we live in a world | in-which everything be-3SG.PRES digital
 become-PST.PTCP
 ‘We live in a world in which everything has become digital.’ (Lingala, A2)
TH: *We leven in een wereld waarin alles digitaal is geworden.*
Error type: subordinate clause – finite verb – verb-final
Position: clause-internal after SUBJ
- (28) **Bovendien wij de musea kunnen bezoeken.*
 moreover we the museums can-1PL.PRES visit-INF
 ‘On top of that, we can visit the museums.’ (Arabic, A2)
TH: *Bovendien kunnen wij de musea bezoeken.*
Error type: main clause – finite verb – verb-second
Position: clause-final before NFV
- (29) **Twee jaar geleden begon ik met Nederlands leren en | tot nu toe het nog steeds interessant is.*
 two years ago began I with Dutch learn and | until now up it
 still interesting be-3SG.PRES
 ‘I started learning Dutch two years ago, and up until now, it is still interesting.’
 (Arabic, B1+)
TH: *Twee jaar geleden begon ik met Nederlands leren en tot nu toe is het nog steeds interessant.*
Error type: main clause – finite verb – verb-second
Position: clause-final

- (30) **Deze methode soms helpt mij een beetje.*
 this method sometimes help-3SG.PRES me a little
 ‘This method sometimes helps me a little.’ (Turkish, A2)

TH: *Deze methode helpt mij soms een beetje.*

Error type: main clause – finite verb – verb-second

Position: miscellaneous

The following section will explain how I take advantage of these identified error tendencies in order to construct the learner-informed evaluation dataset INFO.

4.1.6 Curation of Evaluation Dataset Info

The error tendencies that result from the structural analysis of the 200 selected learner clauses can serve as a source of information for the curation of the learner-informed evaluation dataset INFO. As a subset of VERBS permutations, INFO permutations only allow for changes in the position of the finite verb that recreate one of the incorrect position categories that result from the analysis. The INFO test dataset is based on the same 13,586 correct sentences as the RAND and VERBS test datasets. The latter two, however, are effectively double in size as they contain each sentence twice, once in its original and once in its permuted form. As the INFO dataset aims to approximate genuine learner data as closely as possible, it only contains each of the 13,586 sentences once, each either retained in its correct form or modified to reflect an incorrect structure based on the identified learner error tendencies. Out of the 13,586 sentences, 2,242 contain at least one subordinate clauses that is identifiable with the SPACY dependency parser,⁷⁴ which is equivalent to 16.5% of all sentences. In order for the dataset to contain both correct and incorrect subordinate clauses, I assume 50% of the subordinate clauses to be erroneous, which translates to 1,121 sentences with incorrect subordinate clauses and equally as many sentences with correct subordinate clauses. In the previous section, it was established that the finite verb is almost equally as likely to be misplaced in a subordinate clause as it is in a main clause (Figure 4.4). Thus, I set the amount of sentences with main clauses to contain errors to 1,121 as well. The resulting distribution of correct and incorrect main and subordinate clauses is illustrated in Table 4.2:

Table 4.2: Clause type error distribution in INFO

Clause	% Total	Value	Amount	%
Main	83.5%	correct	10,223	75.25%
		incorrect	1,121	8.25%
Subordinate	16.5%	correct	1,121	8.25%
		incorrect	1,121	8.25%

In the INFO dataset, the likelihood of encountering an error in a subordinate clause is therefore approximately five times greater than in a main clause, which approximates the proportion of sentences that contain a subordinate clause versus sentences that do not contain a subordinate clause identified above (16.5 : 83.5) to the greatest possible

⁷⁴The criteria for identifying subordinate clauses will be explained below.

extent given the amount of available data.⁷⁵

To partly account for the identified errors resulting from misplacements of the elements this thesis cannot consider in detail (non-finite verb errors, infinitival complement errors, prepositional complement errors), for the miscellaneous finite verb errors, and for entirely unseen error types that may have missed the limited sample of learner sentences analyzed, I introduce an element of randomly verb-permuted sentences R equal to one third of the amount of incorrect clauses.⁷⁶

$$R = 2242/3 \approx 748 \quad (4.2)$$

The random element follows the VERBS shuffle method introduced in the first part of this thesis. This leaves the remaining 747 incorrect sentences per clause type to be divided according to the distribution of the observed learner error tendencies. The distribution of recreated errors in the main and subordinate clauses of the INFO dataset is illustrated in Table 4.3:

Table 4.3: Distribution of error position types in main and subordinate clauses in INFO

Clause	Position Error FV	Amount	%	Analysis
–	R : VERBS	748	–	–
Main	clause-internal after SUBJECT	329	44%	30
	clause-final	164	22%	15
	clause-internal before SUBJECT	112	15%	10
	clause-final before NFV	75	10%	7
	clause-initial	67	9%	6
	Total		747	100%
Subordinate	clause-internal after SUBJECT	433	58%	39
	clause-initial	232	31%	21
	clause-internal before SUBJECT	82	11%	7
	Total		747	100%

I introduce errors into the grammatically correct sentences as follows: For subordinate clauses, I retrieve whether a sentence contains a subordinate clause with the help of the SPACY dependency parser. First, I check whether SPACY detects any subordinative markers (MARK) in a sentence. If there is a token whose dependency relation SPACY identifies to be a subordinative marker, the position of which denotes the beginning of the subordinate clause, I determine the limit of the subordinate clause by looking for any coordinative conjunctions in a position after the subordinative marker. If there are none, the subordinate clause is assumed to be the final clause of the sentence. I then retrieve the subject of the subordinate clause and its finite verb, if present, by

⁷⁵This is a simplification. A sentence typically consists of more than one clause, so the total ratio of main clauses to subordinate clauses in the dataset as a whole is likely to be even greater. However, since I only introduce a single error to each sentence, the $16.5 : 83.5 \approx 1 : 5$ ratio is a good estimate.

⁷⁶I set the size of R to be one third of all erroneous sentences to approximate the amount of identified errors that this thesis cannot consider in more detail: $22_{NFV} + 17_{IC} + 10_{PC} + 15_{FV \text{ miscellaneous}} / 200 = 64/200 \approx 1/3$. R follows the VERBS permutation method to account for unseen verb order error types.

again accessing the dependency relations and SPACY’s representation of morphological information, respectively. If all elements are present, I move the finite verb to the first position behind the subject (clause-internal after SUBJECT), to the first position before the subject (clause-internal before SUBJECT), or to the first position behind the subordinative clausal complement (clause-initial).

For main clauses, I assume their initial token to be the first token of the sentence.⁷⁷ I then check for the limit of the main clause in the same way as in subordinate clauses. I retrieve the first finite verb SPACY detects by accessing the morphological information and move it behind or before the subject (clause-internal after SUBJECT and clause-internal before SUBJECT), before the first non-finite verb form (clause-final before NFV), or into the absolute initial or final position, again following the sample principles as in subordinate clauses (clause-initial and clause-final).

Figure 4.6: Recreation of a clause-final before NFV error in main clause

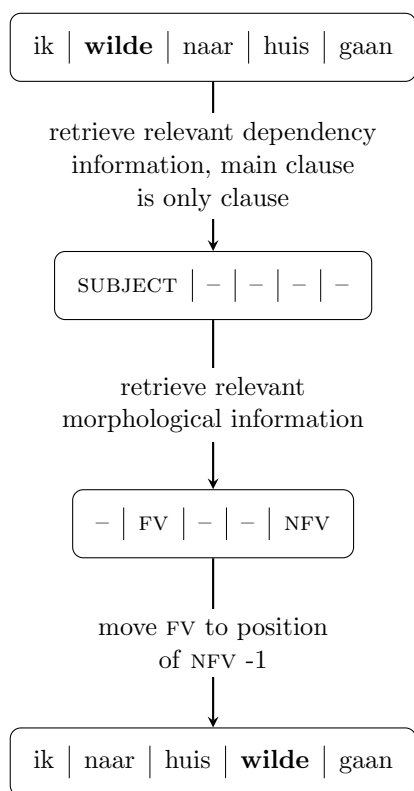
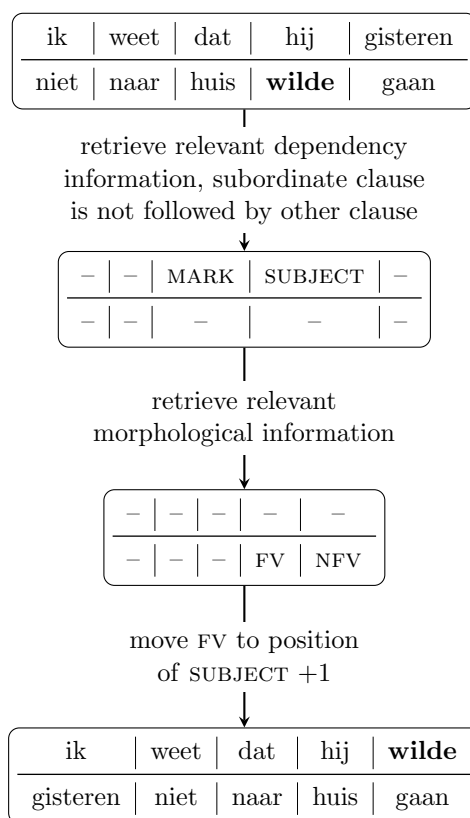


Figure 4.7: Recreation of a clause-internal after SUBJECT error in subordinate clause



Figures 4.6 and 4.7 illustrate how I recreate a clause-final before NFV error in a main clause and a clause-internal after SUBJECT error in a subordinate clause for the sentences *ik wilde naar huis gaan* ‘I wanted to go home’ and *ik weet dat hij gisteren niet naar huis wilde gaan* ‘I know that he did not want to go home yesterday’, respectively.

⁷⁷This effectively means that I introduce errors only to the first main clause of any given sentence.

Now that the types of errors and their distribution in the INFO test dataset have been explained in detail, in the following section I will introduce an additional evaluation metric that is commonly used when working with learner data before presenting the results the previously established classification approaches are able to achieve on both the INFO evaluation dataset and the 200 genuine learner sentences (LEARN) themselves.

4.2 Evaluation Metric: $F_{0.5}$ Score

Where in the first part of this thesis, the average F_1 score presented itself as a suitable evaluation metric because it attributes equal importance to both precision and recall and therefore neatly illustrates the general capabilities of the explored classification approaches in detecting erroneous word order, I now want to introduce an additional evaluation metric that is commonly used in automated grammatical error correction: the $F_{0.5}$ score (4.3).

$$F_{0.5} = \frac{(1 + 0.5^2) \cdot \text{precision} \cdot \text{recall}}{(0.5^2 \cdot \text{precision}) + \text{recall}} \quad (4.3)$$

The $F_{0.5}$ score is especially suitable for grammatical error correction as it deems precision twice as important as recall. Ng et al. (2014) explain:

When a grammar checker is put into actual use, it is important that its proposed corrections are highly accurate in order to gain user acceptance. Neglecting to propose a correction is not as bad as proposing an erroneous correction.

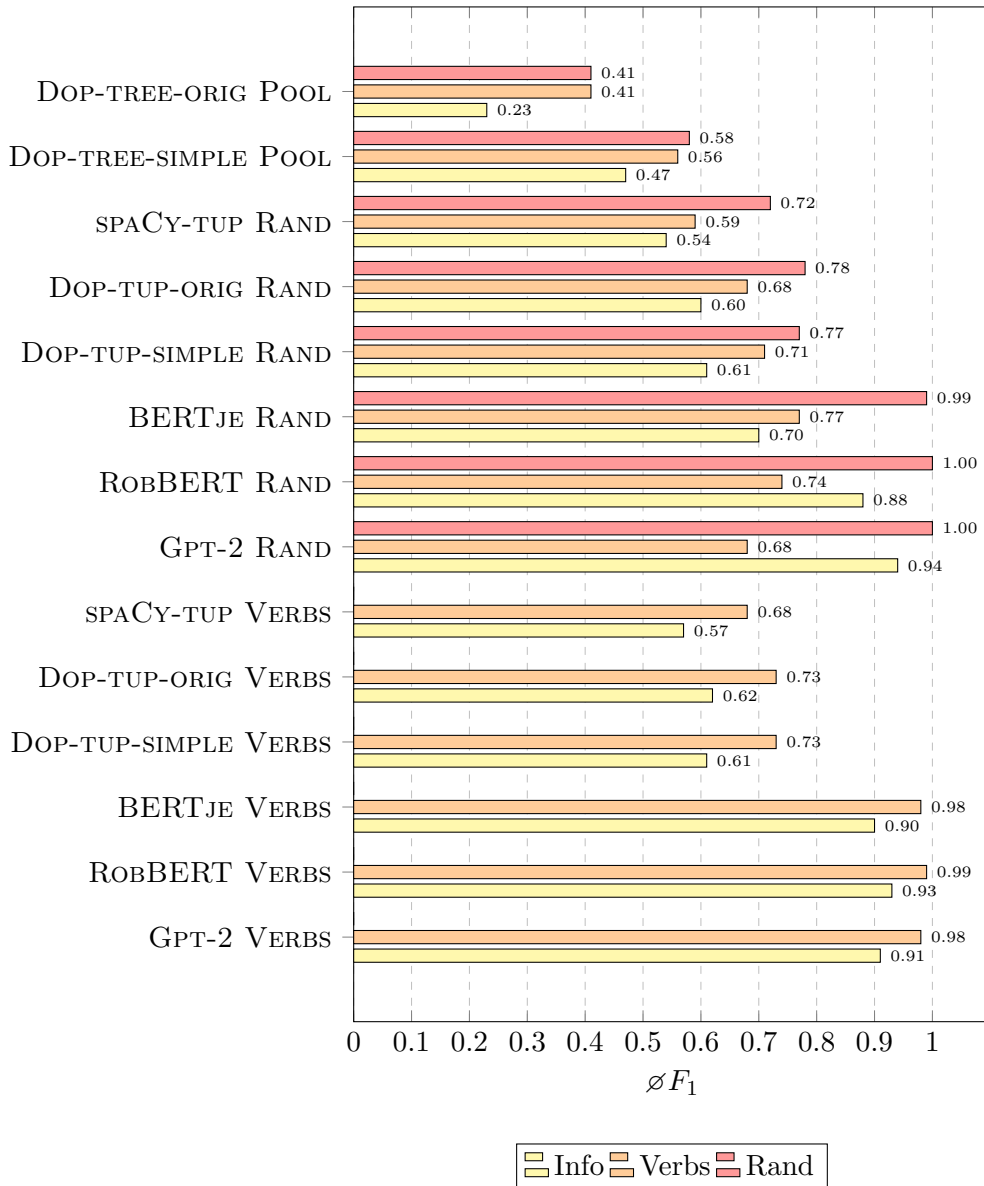
In other words, a classifier that regularly classifies correct sentences as incorrect could discourage learners. Thus, a less sensitive but more accurate classifier would always be preferable if a trade-off between precision and recall was inevitable. This makes the average $F_{0.5}$ score more suitable to represent the performance of the classification approaches I explore if they were to be implemented in a real-life application scenario, which I try to approximate with the INFO test dataset.

Furthermore, for the original learner sentences LEARN, I report the recall score as all of the sentences in the dataset are incorrect. The recall score, therefore, is directly equivalent to the percentage of incorrect sentences the classification approaches identify to be incorrect.

4.3 Results

For comparative purposes, I illustrate the average F_1 scores of all models sorted by their respective train dataset on the RAND, VERBS and INFO test datasets. For the TRANSFORMER CLASSIFIER models, I again report the average performance scores over three models (Figure 4.8):

Figure 4.8: F_1 scores per model and test dataset



Additionally, for each model I report the more practically relevant average $F_{0.5}$ score that emphasizes precision. For the original learner sentences, as all of them are incorrect, I report the recall score of the incorrect category (4.4):

Table 4.4: Average $F_{0.5}$ score of all models on Test INFO and Recall score of all models on Test LEARN

Config	Model	Train Test	POOL/RAND		VERBS	
			INFO	LEARN	INFO	LEARN
PARSE	DOP-TREE-ORIG		0.27	0.99	–	–
LOOKUP	DOP-TREE-SIMPLE		0.51	0.90	–	–
POS CLASSIFIER	SPACY-TUP		0.53	0.27	0.56	0.38
PARSE	DOP-TUP-ORIG		0.60	0.31	0.61	0.36
CLASSIFIER	DOP-TUP-SIMPLE		0.61	0.29	0.60	0.33
TRANSFORMER CLASSIFIER	BERTJE		0.76	0.04	0.92	0.37
	ROBBERT		0.94	0.86	0.95	0.39
	GPT-2		0.96	0.94	0.92	0.29
			$\varnothing F_{0.5}$	Recall	$\varnothing F_{0.5}$	Recall

4.4 Discussion

The average F_1 scores each model is able to achieve typically suffer a slight decrease when tested on the INFO dataset as opposed to the scores the models are able achieve when trained and tested on data curated according to the same permutation method for introducing word order errors (Figure 4.8). This effect is similar to RAND-trained classifiers exhibiting lower scores on the VERBS test set, which has been observed earlier. INFO permutations are a subset of VERBS permutations, which in turn are a subset of RAND permutations. Consequently, RAND-trained classifiers exhibit the highest scores on the RAND test set, followed by the VERBS test set and finally the INFO test set as the error types contained within the test sets become more and more specific while the classifiers become less and less sensitive for these types of errors. Likewise, classifiers trained on the VERBS train data typically exhibit higher performance scores on the INFO test set than classifiers trained on the RAND data. This is due to the fact that the VERBS train data, its permutations being a more restricted superset of the INFO permutations compared to the RAND permutations, is more likely to include the specific error types that the INFO test set contains. A notable exception form the ROBBERT and GPT-2 RAND-trained models, which achieve higher performance scores on the INFO test set than on the VERBS test set. The RAND-trained GPT-2 classifier even outperforms its VERB-trained counterpart and presents itself as the best-performing model on the INFO test set. In the following sections, I will briefly shed light on the performance of the different configurations by means of the more practically relevant performance scores illustrated in Table 4.4.

4.4.1 Parse Lookup

The PARSE LOOKUP approaches, as expected, exhibit the lowest average $F_{0.5}$ scores out of all configurations and models. As has been mentioned before, this likely stems from the insufficient representational power, the overprediction of correct sequences, and the resulting oversimplification of tree structures. Both models, like previously, significantly overpredict the incorrect category. This is also the reason for the extremely high recall scores on the LEARN sentences. In this case, they do not result from the models' excellent performance but from the fact that they classify almost any sentence as incorrect, which is highly undesirable for a real-life application.

4.4.2 PoS and Parse Classifier

The models of the POS and PARSE CLASSIFIER configurations overall exhibit very similar performance scores. The PARSE CLASSIFIER approaches that have access to syntactically richer information still manage to outperform the POS CLASSIFIER approach on the INFO dataset, but the VERBS-trained SPACY-TUP classifier achieves the highest recall on the LEARN sentences. Simplifying the tree structures for the DOP-TUP-SIMPLE implementation of the PARSE CLASSIFIER configuration, just like before, does not prove equally advantageous as it did for the PARSE LOOKUP configuration. The classifiers appear to learn to generalize equally well when presented with the original tree structures as sources for their input. The performance of the PARSE CLASSIFIER approaches, in particular, is impressive as they have been trained on only a fraction of the data the POS CLASSIFIER model has been trained on (3.6%), stressing that syntactic information is vital for solving the task of word order error detection.

4.4.3 Transformer Classifier

In this experiment, the most interesting differences within a configuration can be found in the TRANSFORMER CLASSIFIER models. In the first part of this thesis, BERTJE and ROBERT outperform GPT-2 when tasked to predict generic verb order errors while having seen generic word order errors during training. Task-specific training and testing results in almost perfect performance scores for all models. Here, GPT-2 significantly outperforms the BERT-based models when trained on data permuted according to the RAND method. Training on data permuted to the VERBS method achieves the greatest increase in performance for the BERTJE model, but for GPT-2 it effectively lowers the performance. This is the case both for the INFO data and the LEARN sentences. The differences are drastic: Where the GPT-2-based classifier is able to identify 94% of the LEARN sentences as erroneous, the BERTJE-based model identifies only 4% of them correctly. This indicates that when the transformer models are presented with different types of error generation methods, their learned patterns from the pseudo data they have been trained on translate differently well to the exact subset of errors learners make. Since all classifiers have been trained on the same data, this must be the result of the different architectures of the models. GPT-2 is the model that can identify actual learner errors most accurately and exhibits the highest $F_{0.5}$ score on the learner-informed evaluation data. This tells us that its precision is very good, too, and it is unlikely to overpredict the incorrect category as was the case with the PARSE LOOKUP models that exhibited equally high recall scores on the INFO data. GPT-2 must therefore have a way of representing syntactic information that lets it generalize more

efficiently than the BERT-based models, even though ROBBERT significantly improves on the performance of BERTJE. The recall on the LEARN sentences, when trained on data permuted according to the VERBS permutation method, is also significantly lower with all models, including GPT-2, than that of the RAND-trained GPT-2 model. This suggests that the method of data permutation can affect the capability of all models to generalize. With GPT-2 trained on RAND data being the best model, I want to look at its predictions per incorrect position category (Table 4.5):

Table 4.5: Best predictions of GPT-2 RAND per incorrect position in INFO

Clause	Position	Incorrect	Correct
–	correct	4	11340
	<i>R</i> : VERBS	333	415
Main	clause-initial	67	0
	clause-internal before SUBJECT	112	0
	clause-internal after SUBJECT	329	0
	clause-final before NFV	74	1
	clause-final	162	2
Subordinate	clause-initial	230	2
	clause-internal before SUBJECT	82	0
	clause-internal after SUBJECT	419	14

As it turns out, its performance is almost impeccable on the learner-informed errors introduced, stressing that the RAND-trained GPT-2 seems to have found a way to adjust its parameters in such a way that it is sensitive to errors learners are likely to make, but not as sensitive to randomly introduced verb order errors.

Other than the fact that both sequences are complex sentences, the only two LEARN sentences the best GPT-2 model predicts to be correct although all LEARN sentences are incorrect do not seem to have much in common:⁷⁸

- **Inburgering 1 gaat over de autochtonen van Europese landen die volgens de schrijver en hoogleraar aan de Letterenfaculteit van de Katholieke Universiteit Leuven Joop van der Horst zetten hun hakken in het zand met betrekking tot hun integratie met de rest van de Europese Unie*
- **Als ik moet mijn selectiecriteria opsommen zou ik zeggen dat mijn ideale partner moet grappig zijn en bereid moet zijn om over hun gevoelens te praten*

The first sentence contains a miscellaneous verb order error while the second sentence holds clause-internal before SUBJECT errors in subordinate clauses while also exhibiting a correct word order pattern in another subordinate clause, which could potentially have led to the misclassification. It is even possible that although being trained on randomly permuted data, the classifier learned a similar set of patterns in relation to the subject or absolute positions to identify verb order errors as I have introduced for analytical purposes, for clearly it is able to solve these types of errors while proving to be less sensitive to randomly introduced verb order errors. Yet, how exactly the

⁷⁸For better readability, in this example, I do not lowercase the tokens.

models represent abstract information like this internally needs to be the subject of future research.

From a practical perspective, where in an application scenario, all sentences intuitively follow learner error tendencies for they are the product of learners themselves, this could potentially mean an even better performance on actual learner errors than illustrated on the INFO test set which includes the element R of random VERBS shuffles. While the performance of the RAND-trained GPT-2 classifier appears promising, it is important at this point to acknowledge a number of limitations inherent in the results of this thesis. These limitations could serve as intriguing avenues for subsequent research.

4.4.4 Opportunities for Future Research

In addition to the limitations of the different classification approaches and of the method of pseudo data generation touched upon earlier, which can occasionally lead to the generation of correct sequences, I would like to briefly draw attention to a number of other challenges that could potentially offer interesting starting points for future research:

- **Pseudo data dependency:** Due to the lack of genuine data, both the training and the evaluation of the models described here predominantly rely on pseudo data. Although it is possible to reasonably estimate the models’ real-life performance based on the combination of the INFO and LEARN test sets, their efficacy on genuine data remains to some extent uncertain unless extensively tested on genuine learner data. Moreover, the genuine learner sentences of the LEARN test set are modified to only contain verb order errors, ensuring a consistent input format for the classification approaches. This indicates that the best-performing model can reliably detect verb order errors made by learners. However, its capability to identify these errors in untreated learner data is not guaranteed. A comprehensive annotation study with the objective of obtaining a dataset of genuine learner data annotated for word order errors is thus still desirable.
- **Multiple errors in learner sentences:** Some learner sentences contain multiple errors. In retrospect, it would have been more beneficial to have 200 sentences with a single error each, rather than 184 sentences where some contain multiple errors. This is because the classifiers might find it easier to identify errors in sentences with multiple mistakes. This also relates back to the need for more annotated genuine data.
- **PoS and Parse Classifier input:** The method of feeding positional information into the models of the POS and PARSE CLASSIFIER approaches is rather rudimentary. Apart from absolute position tuples, exploring alternative approaches such as the probability score of parses, tree distance, or incorporating dependency information might yield better results for these classification approaches and could be considered for future research.

- **Explorative nature of the thesis:** Due to the exploratory nature of this thesis and its limited scope, certain aspects such as feature ablation and hyperparameter tuning were not considered. Investigating the optimization of these parameters in future research could provide insights into their impact on model performance.

Finally, with generative artificial intelligence models gaining more and more popular recognition, their potential benefits as readily available end-to-end solutions for a variety of natural language processing tasks cannot be overlooked. In the following section, I will therefore briefly illustrate how a learner could immediately benefit from the use of such a model.

Chapter 5

Outlook: Generative Artificial Intelligence Models as Virtual Teachers

With prompt-based generative artificial intelligence models like OpenAI's Chat-GPT receiving more and more popular attention,⁷⁹ exploring the performance of these models at different tasks can be beneficial as they are very easy to use. On the other hand, however, these models are often proprietary, which makes their behavior difficult to investigate, limits accessibility, and puts them at risk of being subject to undetectable biases.

Figure 5.1: Distribution of accepted and challenged initial target hypotheses



Prompt engineering offers the possibility to investigate the performance of these models. In the applicational context of this thesis, I take on the role of a language learner who wants to use the program to correct sentences I have produced so I can learn from the errors I make. I let it correct half of the learner error sentences of the LEARN test set based on the following prompt:

I will present you with sentences containing errors in Dutch. I want you to correct these sentences while staying as close as possible to the original wording.

⁷⁹<https://openai.com/blog/chatgpt>, last accessed: 15.08.2023.

The other half of the LEARN sentences, I, an advanced learner of Dutch (CEFR B2–C1), correct manually myself. These corrections serve as the target hypotheses I base the analysis of the error type and of the position of the misplacement on. I let all target hypotheses be evaluated by two first-language speakers and compare the number of target hypotheses that are challenged by either one of the first-language speakers, both of the first-language speakers, or none of them.⁸⁰ In case the first-language speakers do not agree with the presented target hypothesis, they are asked to provide a target hypothesis themselves. If the target hypothesis for a given sentence is challenged by both first-language speakers, I compare the suggested target hypotheses by both of the first-language speakers and choose the one that appears to stay closer to the original wording used by the learner, i.e., that tries to avoid as many additions to the sentence as possible. This way, I can ensure that all the target hypotheses I use for the extraction of verb order error tendencies are a possible correction of the given sentence accepted by at least one first-language speaker. This tries to keep the focus of the corrections on the verb order error.⁸¹ At the same time, I can show that the performance of the model at correcting grammatical errors is slightly better than that of an advanced learner, which proves the usefulness of these models for learners, even at higher proficiency levels (Figure 5.1). Therefore, facilitating access to these models and more transparency about their parameters could make them readily available, effective tools in the form of end-to-end solutions. The BEA-2023 shared task whose results were published just a month prior to this thesis investigates the performance of models that are based on a corpus of teacher-student interactions. “The goal of the task was to benchmark the ability of generative language models to act as AI teachers, replying to a student in a teacher–student dialogue” (Tack et al., 2023, p. 1). In this thesis, I have isolated a single capability such an AI teacher should possess, i.e., the detection of verb order errors as the correct placement of verbs seems to be challenging for second language learners.

⁸⁰The instructions provided to the first-language annotators can be found in Appendix F.

⁸¹The correction styles of different annotators can differ and using target hypotheses that divert too much from the original as a basis for the error type and structural analysis can make it harder to analyze verb order errors.

Chapter 6

Conclusion

In this thesis, I have illustrated the performance of several classifiers based on different natural language processing model architectures at the task of detecting word order errors. I have focused on the detection of verb order errors in particular as verb order appears to be challenging for second-language learners of Dutch. Word order error correction, until now, has largely escaped the scope of public interest which is why there is no annotated genuine data available that could be used for the purpose of training classifiers for the task. Therefore, I have created pseudo datasets that introduce generic word and generic verb order errors to correct source sentences.

I have shown that a naive lookup approach based on the tree output of a syntactic constituency parser does not possess the representational power needed to effectively model the allowed syntax patterns of Dutch. In a machine learning setup, however, I have been able to show that classifiers having access to the syntactically richer output of a parser as opposed to the output of a part-of-speech tagger fare significantly better when tasked to detect both generic word and generic verb order errors. This indicates that syntactic information is indeed crucial for solving the task of word order error detection. This holds despite the fact that the parser-based approaches are trained on a fraction of the data that the part-of-speech-based approach is trained on.

Finally, classifiers built upon the architectures of transformer models outperform all previous classification approaches and almost exhibit perfect scores when trained and tested on task-specific data. This indicates that transformers are able to reliably solve natural language processing tasks that require syntactic information. However, the exact mechanisms by which transformers achieve this remain to be explored in subsequent research. By creating a final evaluation set containing only those errors that learners are likely to make based on the structural analysis of genuine learner sentences, I have illustrated how the general capability of the explored classification approaches in detecting erroneous word order translates to more practical application scenarios. While slight declines in performance are observable, I have show that training classifiers on generic errors in the form of pseudo data only is effective and, in the case of the transformer-based models, can generate highly accurate models that would be suitable to be implemented in a real-life scenario with minor adaptations. The best model is a GPT-2-based classifier trained on generic word order errors as opposed to generic verb order errors, which achieves an average $F_{0.5}$ score of 0.96 on the learner-informed evaluation dataset and exhibits a recall score of 0.94 on the genuine learner sentences that served as the basis for the structural analysis of verb order error types. During its fine-tuning on generic word order errors, the model appears to learn to

pay attention to errors that learners commonly make. It exhibits an almost perfect performance on the verb order errors that were introduced according to the learner tendencies that emerged from the structural analysis of genuine learner data, but it struggles to identify generic verb order errors even though these constitute a subset of the errors the model has been trained on.

Lastly, prompt-based conversational agents show promising results in automated grammatical error correction as a whole and could prove to be convenient end-to-end solutions. Yet, providing a learner with feedback about specific areas of their target language's grammar that appear to be particularly challenging such as verb order is also valuable. With this thesis, I have established the foundation for this pipeline, beginning with the detection of verb order errors.

Appendix A

Abbreviations and Symbols

This list solely presents abbreviations not contained in the Leipzig Glossing Rules (Comrie et al., 2008).

ADJ	adjective
ADP	adposition
BEA	Building Educational Applications
BERT	Bidirectional Encoder Representations from Transformers
BPE	Byte Pair Encoding
CEFR	Common European Framework of Reference for Languages
CoNLL	Conference on Natural Language Learning
ConvNet	Convolutional Neural Network
DET	determiner, article
FV	finite verb
GPT-2	Generative Pre-trained Transformer 2
HOO	Helping Our Own
L1	first language
L2	second language
MLM	Masked Language Modeling
NLI	Natural Language Inference
NP	noun phrase
NFV	non-finite verb
P	preposition
PoS	part-of-speech
PP	prepositional phrase
RNN	Recurrent Neural Network
RoBERTa	Robustly optimized BERT approach
SMAIN	full sentence
V	verb
∅	average

Appendix B

Illustration Verbs Shuffles

(ik, wil, geen, boeken, kopen)

_ ik _ geen _ boeken _

(wil, ik, geen, kopen, boeken)

(ik, kopen, geen, wil, boeken)

(ik, wil, kopen, geen, boeken)

(kopen, wil, ik, geen, boeken)

(ik, wil, geen, kopen, boeken)

(ik, geen, wil, boeken, kopen)

(ik, geen, boeken, wil, kopen)

(ik, kopen, geen, boeken, wil)

(kopen, ik, geen, wil, boeken)

(kopen, ik, wil, geen, boeken)

(ik, geen, kopen, wil, boeken)

(ik, geen, kopen, boeken, wil)

(wil, ik, kopen, geen, boeken)

(wil, kopen, ik, geen, boeken)

(ik, geen, wil, kopen, boeken)

(kopen, ik, geen, boeken, wil)

(wil, ik, geen, boeken, kopen)

(ik, geen, boeken, kopen, wil)

(ik, kopen, wil, geen, boeken)

The relative order of the verb-unrelated tokens *ik*, *geen*, and *boeken* is preserved in all permutations.

Appendix C

Categories for Phrasal Analysis

Table C.1: Symbols used for analysis of clause structure

A	adjective, adverb, or adverbial
FC	finite clause
PC	prepositional complement
CE	coordinative element
SE	subordinative element
M	miscellaneous
O	object
PO	prepositional object
om	om (in om + te-infinitivals)
P	preposition
Q	question word
S	subject
te	te (in om + te and te-infinitivals)
FV	finite verb
NFV	non-finite verb

Appendix D

Results Overview

Table D.1: Combined Results of all Experiments

Model	Train	Test	Class	Prec.	Rec.	F_1	$F_{0.5}$	Acc.
Dop-Tree-Original	POOL	RAND	incorrect	0.52	0.99	0.68	0.57	0.54
			correct	0.88	0.08	0.15	0.30	
	POOL	VERBS	incorrect	0.52	0.99	0.68	0.57	0.53
			correct	0.87	0.08	0.15	0.30	
	POOL	INFO	incorrect	0.18	0.99	0.30	0.21	0.24
			correct	0.99	0.09	0.16	0.33	
	POOL	LEARN	incorrect	–	0.99	–	–	0.99
	Dop-Tree-Simple	POOL	RAND	incorrect	0.57	0.81	0.67	0.60
correct				0.67	0.38	0.49	0.58	
POOL		VERBS	incorrect	0.55	0.76	0.64	0.58	0.57
			correct	0.62	0.38	0.47	0.55	
POOL		INFO	incorrect	0.23	0.87	0.36	0.27	0.49
			correct	0.94	0.42	0.58	0.75	
POOL		LEARN	incorrect	–	0.90	–	–	0.90
SpaCy-Tuple		RAND	RAND	incorrect	0.74	0.68	0.71	0.73
	correct			0.70	0.76	0.73	0.71	
	RAND	VERBS	incorrect	0.65	0.45	0.53	0.60	0.60
			correct	0.58	0.76	0.66	0.61	
	RAND	INFO	incorrect	0.22	0.34	0.27	0.24	0.69
			correct	0.85	0.76	0.80	0.83	
	RAND	LEARN	incorrect	–	0.27	–	–	0.27
	VERBS	VERBS	incorrect	0.68	0.68	0.68	0.68	0.68
			correct	0.68	0.68	0.68	0.68	
	VERBS	INFO	incorrect	0.26	0.55	0.35	0.29	0.67
			correct	0.89	0.69	0.78	0.84	
	VERBS	LEARN	incorrect	–	0.38	–	–	0.38
Dop-Tuple-Original	RAND	RAND	incorrect	0.80	0.75	0.77	0.79	0.78
			correct	0.76	0.82	0.79	0.77	
	RAND	VERBS	incorrect	0.75	0.55	0.64	0.70	0.69

Model	Train	Test	Class	Prec.	Rec.	F_1	$F_{0.5}$	Acc.
			correct	0.65	0.82	0.72	0.68	
	RAND	INFO	incorrect	0.31	0.42	0.36	0.33	0.75
			correct	0.88	0.82	0.85	0.86	
	RAND	LEARN	incorrect	–	0.31	–	–	0.31
	VERBS	VERBS	incorrect	0.76	0.67	0.71	0.74	0.73
			correct	0.71	0.79	0.74	0.72	
	VERBS	INFO	incorrect	0.33	0.53	0.41	0.36	0.74
			correct	0.89	0.78	0.84	0.87	
	VERBS	LEARN	incorrect	–	0.36	–	–	0.36
<hr/>								
Dop-Tuple-Simple	RAND	RAND	incorrect	0.80	0.72	0.75	0.78	0.77
			correct	0.74	0.82	0.78	0.76	
	RAND	VERBS	incorrect	0.77	0.60	0.68	0.73	0.71
			correct	0.67	0.82	0.74	0.70	
	RAND	INFO	incorrect	0.33	0.46	0.38	0.35	0.75
			correct	0.88	0.81	0.85	0.87	
	RAND	LEARN	incorrect	–	0.29	–	–	0.29
	VERBS	VERBS	incorrect	0.75	0.68	0.71	0.73	0.73
			correct	0.71	0.77	0.74	0.72	
	VERBS	INFO	incorrect	0.31	0.53	0.39	0.34	0.72
			correct	0.89	0.76	0.82	0.86	
	VERBS	LEARN	incorrect	–	0.33	–	–	0.33
<hr/>								
BERTje I	RAND	RAND	incorrect	0.99	0.99	0.99	0.99	0.99
			correct	0.99	0.99	0.99	0.99	
	RAND	VERBS	incorrect	0.99	0.63	0.77	0.89	0.81
			correct	0.73	0.99	0.84	0.77	
	RAND	INFO	incorrect	0.92	0.31	0.46	0.66	0.88
			correct	0.88	0.99	0.93	0.90	
	RAND	LEARN	incorrect	–	0.04	–	–	0.04
	VERBS	VERBS	incorrect	0.99	0.98	0.98	0.98	0.98
			correct	0.98	0.99	0.98	0.98	
	VERBS	INFO	incorrect	0.91	0.75	0.82	0.87	0.95
			correct	0.95	0.99	0.97	0.96	
	VERBS	LEARN	incorrect	–	0.35	–	–	0.35
<hr/>								
BERTje II	RAND	RAND	incorrect	1.00	0.99	0.99	1.00	0.99
			correct	0.99	1.00	0.99	0.99	
	RAND	VERBS	incorrect	0.99	0.58	0.74	0.87	0.79
			correct	0.71	1.00	0.83	0.75	
	RAND	INFO	incorrect	0.94	0.28	0.43	0.64	0.88
			correct	0.87	1.00	0.93	0.90	
	RAND	LEARN	incorrect	–	0.04	–	–	0.04
	VERBS	VERBS	incorrect	0.99	0.98	0.98	0.98	0.98
			correct	0.98	0.99	0.98	0.98	
	VERBS	INFO	incorrect	0.91	0.78	0.84	0.88	0.95
			correct	0.96	0.98	0.97	0.96	
	VERBS	LEARN	incorrect	–	0.35	–	–	0.35

Model	Train	Test	Class	Prec.	Rec.	F_1	$F_{0.5}$	Acc.
BERTje III	RAND	RAND	incorrect	1.00	0.99	0.99	0.99	0.99
			correct	0.99	1.00	0.99	0.99	
	RAND	VERBS	incorrect	0.99	0.50	0.67	0.83	0.75
			correct	0.67	1.00	0.80	0.72	
	RAND	INFO	incorrect	0.93	0.24	0.38	0.58	0.87
			correct	0.87	1.00	0.93	0.89	
	RAND	LEARN	incorrect	–	0.03	–	–	0.03
			VERBS	VERBS	incorrect	0.99	0.98	0.98
	VERBS	INFO	incorrect	0.98	0.99	0.98	0.98	
			correct	0.91	0.79	0.84	0.88	0.95
VERBS	LEARN	incorrect	0.96	0.98	0.97	0.96		
		correct	–	0.41	–	–	0.41	
RobBERT I	RAND	RAND	incorrect	1.00	1.00	1.00	1.00	1.00
			correct	1.00	1.00	1.00	1.00	
	RAND	VERBS	incorrect	1.00	0.48	0.65	0.82	0.74
			correct	0.66	1.00	0.79	0.71	
	RAND	INFO	incorrect	1.00	0.67	0.80	0.91	0.94
			correct	0.94	1.00	0.97	0.95	
	RAND	LEARN	incorrect	–	0.76	–	–	0.76
			VERBS	VERBS	incorrect	0.99	0.99	0.99
	VERBS	INFO	incorrect	0.99	0.99	0.99	0.99	
			correct	0.96	0.80	0.88	0.93	0.96
VERBS	LEARN	incorrect	0.96	0.99	0.98	0.97		
		correct	–	0.39	–	–	0.39	
RobBERT II	RAND	RAND	incorrect	1.00	1.00	1.00	1.00	1.00
			correct	1.00	1.00	1.00	1.00	
	RAND	VERBS	incorrect	1.00	0.48	0.65	0.82	0.74
			correct	0.66	1.00	0.79	0.70	
	RAND	INFO	incorrect	0.99	0.75	0.86	0.94	0.96
			correct	0.95	1.00	0.98	0.96	
	RAND	LEARN	incorrect	–	0.92	–	–	0.92
			VERBS	VERBS	incorrect	0.99	0.99	0.99
	VERBS	INFO	incorrect	0.99	0.99	0.99	0.99	
			correct	0.96	0.80	0.87	0.92	0.96
VERBS	LEARN	incorrect	0.96	0.99	0.98	0.97		
		correct	–	0.38	–	–	0.38	
RobBERT III	RAND	RAND	incorrect	1.00	1.00	1.00	1.00	1.00
			correct	1.00	1.00	1.00	1.00	
	RAND	VERBS	incorrect	1.00	0.58	0.74	0.87	0.79
			correct	0.71	1.00	0.83	0.75	
	RAND	INFO	incorrect	1.00	0.76	0.86	0.94	0.96
			correct	0.96	1.00	0.98	0.96	
	RAND	LEARN	incorrect	–	0.89	–	–	0.89
			VERBS	VERBS	incorrect	0.99	0.99	0.99
	VERBS	INFO	incorrect	0.99	0.99	0.99	0.99	
			correct	0.99	0.99	0.99	0.99	

Model	Train	Test	Class	Prec.	Rec.	F_1	$F_{0.5}$	Acc.	
	VERBS	INFO	incorrect	0.96	0.80	0.87	0.92	0.96	
			correct	0.96	0.99	0.98	0.97		
	VERBS	LEARN	incorrect	–	0.39	–	–	0.39	
GPT-2 I	RAND	RAND	incorrect	1.00	1.00	1.00	1.00	1.00	
			correct	1.00	1.00	1.00	1.00		
	RAND	VERBS	incorrect	1.00	0.42	0.59	0.78	0.71	
			correct	0.63	1.00	0.78	0.68		
	RAND	INFO	incorrect	1.00	0.81	0.89	0.95	0.97	
			correct	0.96	1.00	0.98	0.97		
	RAND	LEARN	incorrect	–	0.99	–	–	0.99	
			VERBS	VERBS	incorrect	0.99	0.97	0.98	0.98
				correct	0.97	0.99	0.98	0.98	
	VERBS	INFO	incorrect	0.92	0.77	0.84	0.89	0.95	
			correct	0.96	0.99	0.97	0.96		
	VERBS	LEARN	incorrect	–	0.30	–	–	0.30	
GPT-2 II	RAND	RAND	incorrect	1.00	1.00	1.00	1.00	1.00	
			correct	1.00	1.00	1.00	1.00		
	RAND	VERBS	incorrect	1.00	0.43	0.60	0.79	0.71	
			correct	0.64	1.00	0.78	0.69		
	RAND	INFO	incorrect	1.00	0.81	0.89	0.95	0.97	
			correct	0.96	1.00	0.98	0.97		
	RAND	LEARN	incorrect	–	0.97	–	–	0.97	
			VERBS	VERBS	incorrect	0.99	0.97	0.98	0.98
				correct	0.97	0.99	0.98	0.98	
	VERBS	INFO	incorrect	0.92	0.74	0.82	0.87	0.95	
			correct	0.95	0.99	0.97	0.96		
	VERBS	LEARN	incorrect	–	0.28	–	–	0.28	
GPT-2 III	RAND	RAND	incorrect	1.00	1.00	1.00	1.00	1.00	
			correct	1.00	1.00	1.00	1.00		
	RAND	VERBS	incorrect	1.00	0.42	0.59	0.78	0.71	
			correct	0.63	1.00	0.77	0.68		
	RAND	INFO	incorrect	1.00	0.73	0.84	0.93	0.96	
			correct	0.95	1.00	0.97	0.96		
	RAND	LEARN	incorrect	–	0.86	–	–	0.86	
			VERBS	VERBS	incorrect	0.99	0.97	0.98	0.99
				correct	0.98	0.99	0.98	0.98	
	VERBS	INFO	incorrect	0.92	0.75	0.82	0.88	0.95	
			correct	0.95	0.99	0.97	0.96		
	VERBS	LEARN	incorrect	–	0.28	–	–	0.28	

Appendix E

Data Statements

E.1 KU Leuven - Instituut voor Levende Talen - Leerder-corpus

Size: 3121 unique short to mid-length essays.

Curation Rationale: The corpus’s precise curation rationale is unknown. It holds annotations for different kinds of word order errors made by L2 learners of the Dutch language and can therefore serve as a) a source of commonly made errors by learners and b) a guideline for teachers as to how to correct their students’ work.

Language Variety: Standard Dutch as taught in Belgium and as written by learners of the language on the proficiency levels A2 through C1 according to the CEFR.

Speaker Demographic: The learners’ mother tongues represent a range of linguistic backgrounds: Abkhazian, Albanian, Amharic, Antankarana-Malagasy, Randabic, Randmenian, Azerbaijani, Bosnian, Brazilian Portuguese, Bulgarian, Cantonese, Catalan, Chinese, Croatian, Czech, Danish, Dari, Dutch, Edo, English, Estonian, Éwé, Farsi, Filipino, Finnish, French, Georgian, German, Greek, Gujarati, Hebrew, Hindi, Hungarian, Igbo, Indian, Indonesian, Italian, Japanese, Kanarese, Kinyarwanda, Kirundi, Konkani, Korean, Kurdish, Latvian, Lebanese, Lingala, Lithuanian, Luganda, Luxembourgish, Mandarin, Nepali, Oriya, Pashto, Persian (general), Polish, Portuguese, Romanian, Russian, Serbian, Serbo-Croatian, Sindhi, Slovak, Slovenian, Somali, Spanish, Swahili, Swedish, Tabasaran, Tagalog, Taiwanese, Tamil, Telugu, Thai, Tigrinya, Turkish, Twi, Ukrainian, Urdu, Vietnamese, Yoruba, Zulu. The dataset includes information on the learners’ language proficiency, as assessed according to the CEFR. The represented proficiency levels range from A2 to C1. No further information about the speaker demographics is given.

Annotator Demographic: Precise annotator demographics are unknown, but the corpus has been annotated by experienced teachers, most likely native speakers of Dutch. However, the annotations are tentative and are to be understood as corrections of errors as provided to their students by teachers. No annotation guidelines were provided to the teachers.

Speech Situation: The precise speech situation is unknown. However, it is likely that the texts were mostly produced as homework exercises which were then handed in to the teacher for correction.

Text Characteristics: The texts in the corpus were produced as part of the learners' coursework and therefore mostly are short essays as typically written in foreign language classes.

Recording Quality: n.a.

Additional Information: n.a.

Provenance Appendix: n.a.

E.2 Synthetic Dataset

Size: Up to 2,245,188 sentences.

Curation Rationale: The synthetic dataset is intended to represent a variety of text types learners are likely to be prompted to produce in a class room setting.

Language Variety: Standard Dutch.

Speaker Demographic: Unknown.

Annotator Demographic: No annotations. The method of pseudo data generation automatically labels incorrect sentences as incorrect.

Speech Situation: Unknown.

Text Characteristics: The sentences are taken from texts of the following text types: news items, fiction, academic journals, educational content, informational content, newsletters, websites, Wikipedia, press releases, books, brochures, flyers, manuals, legal texts, newspapers, policy docs, proceedings, reports, webcrawls, Wikipedia, newspaper articles in easy-to-read Dutch.

Recording Quality: n.a.

Additional Information: n.a.

Provenance Appendix: For further details refer to
<https://www.edia.nl/resources/elg/downloads>;
<https://taalmaterialen.ivdnt.org/download/lassy-klein-corpus6/>;
https://wortschatz.uni-leipzig.de/de/download/Dutch#nld-nl_web_2019;
<https://taalmaterialen.ivdnt.org/download/wai-not-corpus1-0/>.

Appendix F

Annotation Prompt

The annotators were presented with the following prompt before beginning their annotations process:

You will now see a pair of sentences. The first sentence is a sentence that was produced by a learner of Dutch as a second language. Your task is to evaluate whether the second sentence is a possible correction of the first sentence. For many sentences, there is more than one possible correction. Your task is only to evaluate whether the corrected version presented to you is one of them. For your evaluation, keep in mind that ideally, the corrected version of the sentence should stick as closely as possible to the original wording used by the learner. If you agree with the proposed target hypothesis (the corrected sentence), you will proceed to the next sentence pair. If you do not agree, please provide a corrected version of the learner sentence yourself. Sometimes, the corrected sentence does not cover the whole original sentence produced by the learner. This is usually the case if the learner sentence consists of a combination of phrases or sentences that would typically be separated. In this case, please only evaluate the part that is covered by the corrected sentence presented to you.

Appendix G

Generative AI Statement

This thesis has been written in accordance with the ACL 2023 Policy on AI Writing Assistance (<https://2023.aclweb.org/blog/ACL-2023-policy/>).

If generative artificial intelligence models have been used to help in the generation of code, this is clearly indicated in the respective scripts.

Appendix H

Licenses

H.1 Lassy License

Overeenkomst voor niet-commercieel gebruik

Randtikel 1. Definities

Product: Het Lassy Klein-Corpus, volledige, correcte naam: Lassy Klein-corpus, versie 4.0, Nederlandse Taalunie, 2016, alsmede de totale inhoud van de gedownloade of meegeleverde bestanden, daaronder begrepen maar niet beperkt tot (i) eventueel meegeleverde of van Product deel uitmakende software of computerinformatie en (ii) bijbehorende schriftelijke materialen of bestanden ter uitleg;

Overeenkomst: de onderhavige licentieovereenkomst;

Gebruiker: de natuurlijke persoon die deze Overeenkomst via de Webwinkel heeft geaccepteerd overeenkomstig het in artikel 3 bepaalde;

INT: Het Instituut voor de Nederlandse Taal, gevestigd in Leiden;

Webwinkel: De service beschikbaar op de website van INT waarbij software en data gedownload kan worden.

Randtikel 2. Toepasselijkheid

Deze gebruiksvoorwaarden zijn van toepassing op Product dat door het INT beschikbaar worden gesteld op basis van deze gebruiksovereenkomst en op de daarmee samenhangende rechtsverhouding tussen INT en Gebruiker.

Randtikel 3. Totstandkoming Overeenkomst

De overeenkomst komt tot stand indien Gebruiker bij de bestelprocedure van het product bij de Webwinkel op de knop ‘Akkoord’ heeft geklikt.

Randtikel 4. Gebruiksvoorwaarden

4.1 INT verleent hierbij aan Gebruiker, en Gebruiker accepteert, het niet-exclusieve recht om: a. Product te raadplegen en/of te gebruiken voor eigen onderzoek en ter toelichting bij het eventueel door Gebruiker gegeven onderwijs; b. Product te gebruiken ten behoeve van het ontwikkelen van nieuwe producten (hierna: “Nieuwe Producten”), mits (de inhoud van) Product, een gedeelte van Product of een kwalitatief of kwantitatief substantieel gedeelte daarvan niet herkenbaar in de Nieuwe Producten is opgenomen, verveelvoudigd of overgenomen en mits daarmee niet in strijd wordt gehandeld met Randtikel 4.2 en 4.6 van deze Overeenkomst. Het is Gebruiker niet

toegestaan Product voor commerciële doeleinden te gebruiken.

4.2 De overeenkomstig Randtikel 4.1 sub b van deze Overeenkomst ontwikkelde Nieuwe Producten mogen uitsluitend door Gebruiker worden gebruikt voor eigen oefening, studie of gebruik en niet worden openbaar gemaakt, aan derden verkocht, uitgeleend en/of op andere wijze aan derden ter beschikking worden gesteld, tenzij en voor zover de INT daarvoor uitdrukkelijk schriftelijk toestemming heeft gegeven overeenkomstig Randtikel 8 van deze Overeenkomst.

4.3 Gebruiker is gerechtigd in openbare presentaties, wetenschappelijke publicaties of publicaties voor onderwijsdoeleinden naar Product te verwijzen, melding te maken van het gebruik van Product en/of verslag te doen van werk waarbij van Product gebruik is gemaakt. Gebruiker noemt daarbij expliciet de volledige juiste naam van Product. Gebruiksovereenkomst Lassy Klein-Corpus 2/4 Niet-commercieel

4.4 Voor zover software onderdeel uitmaakt van Product, mag de software uitsluitend worden geïnstalleerd op de computers van de Gebruiker.

4.5 INT verleent aan Gebruiker het recht kopieën van Product te maken die dienen als back-up. Gebruiker is niet gerechtigd om meer kopieën van Product te maken dan strikt noodzakelijk is voor het in Randtikel 4.1 omschreven doel.

4.6 Gebruiker heeft uitsluitend het recht Product te gebruiken voor de doeleinden omschreven in Randtikel 4.1 van deze Overeenkomst. Gebruiker heeft niet het recht (de inhoud van) Product of enig (kwalitatief of kwantitatief substantieel) gedeelte daarvan te reproduceren en/of te vereenvoudigen, tenzij dit noodzakelijk is voor het doeleinde genoemd in Randtikel 4.1 sub b van deze Overeenkomst of wanneer dit is toegestaan op grond van Randtikel 4.4 en/of Randtikel 4.5 van deze Overeenkomst. Gebruiker heeft niet het recht (de inhoud van) Product of enig (kwalitatief of kwantitatief substantieel) gedeelte daarvan openbaar te maken, aan derden te verkopen, uit te lenen en/of op andere wijze aan derden ter beschikking te stellen.

4.7 Gebruiker heeft niet het recht om op eventuele met Product meegeleverde software of software die onderdeel uitmaakt van Product technieken toe te passen waarmee de interne werking kan worden achterhaald, hieronder begrepen maar niet beperkt tot 'reverse engineering'.

4.8 De rechten die middels deze Overeenkomst aan Gebruiker worden verleend, zijn niet overdraagbaar. Gebruiker heeft niet het recht de middels deze Overeenkomst verleende licentie in sublicentie te geven.

Randtikel 5. Vergoeding, betaling en levering

5.1 Voor de aan Gebruiker verleende rechten, zoals bedoeld in Randtikel 4 van deze Overeenkomst, is geen vergoeding verschuldigd.

Randtikel 6. Fouten en onvolkomenheden

6.1 Indien Gebruiker in welk onderdeel dan ook van Product fouten (bugs), onvolkomenheden, inconsequenties e.d. aantreft, dan wordt Gebruiker verzocht die aan INT schriftelijk of elektronisch te melden. INT verplicht zich ertoe gemelde fouten (bugs), onvolkomenheden, inconsequenties e.d. te publiceren. Het publiceren van lijsten van fouten, onvolkomenheden e.d. is voorbehouden aan INT. Het is Gebruiker wel toegestaan om fouten (bugs), onvolkomenheden, inconsequenties e.d. te melden in openbare presentaties, wetenschappelijke publicaties of publicaties voor onderwijsdoeleinden, wanneer verwezen wordt naar Product, melding wordt gemaakt van het gebruik van Product en/of verslag wordt gedaan van werk waarbij van Product gebruik is

gemaakt.

6.2 INT geeft geen enkele garantie en accepteert geen enkele verantwoordelijkheid voor welke beperkingen of fouten in Product dan ook en accepteert geen enkele aansprakelijkheid voor schade, verlies of ongerief dat zou kunnen voortkomen uit het gebruik van Product. 6.3 INT geeft geen enkele garantie dat Product of een gedeelte ervan voor bepaalde specifieke doeleinden kan worden gebruikt.

Randtikel 7. Intellectuele Eigendomsrechten

7.1 In geen geval zal Gebruiker auteursrechten, databankrechten en of andere (intellectuele eigendoms)rechten ten aanzien van Product verwerven.

7.2 Gebruiker erkent dat het gebruik van Product onderworpen is aan de restricties die op grond van Nederlands recht door wetten van intellectuele eigendom en andere vormen van wettelijke bescherming worden opgelegd, waaronder begrepen maar niet beperkt tot auteursrechten, naburige rechten, databankenrechten en rechten op software en dat schendingen van zulke restricties leiden tot wettelijke aansprakelijkheid. Gebruiker onthoudt zich van het schenden van deze restricties. Bijgevolg publiceert Gebruiker geen onderdelen van Product (zoals teksten, geluidsfragmenten of anderszins soortgelijke data en/of tools), anders dan korte voorbeelden in wetenschappelijke publicaties of publicaties voor onderwijsdoeleinden. Gebruiksovereenkomst Lassy Klein-Corpus 3/4 Niet-commercieel

7.3 Voor zover blijkt dat door aanpassingen, bewerkingen en/of aanvullingen van Product door Gebruiker eigen of nieuwe rechten zouden kunnen ontstaan ten aanzien van Product, worden deze hierbij bij voorbaat volledig en onbezwaard door Gebruiker aan INT overgedragen. INT aanvaardt hierbij deze overdracht.

7.4 Voor zover de overdracht zoals bedoeld in het vorige lid middels deze Overeenkomst niet wordt bewerkstelligd of niet mogelijk blijkt, verbindt Gebruiker zich om op eerste verzoek van INT kosteloos alles te doen wat nodig is om de overdracht van alle intellectuele eigendomsrechten ten aanzien van Product te effectueren. Voorts verleent Gebruiker hierbij voor zover nodig tot het moment van volledige overdracht een onbeperkte exclusieve licentie aan INT, welke licentie INT hierbij aanvaardt.

Randtikel 8. Optierecht

8.1 Indien Gebruiker de overeenkomstig Randtikel 4.1 sub b ontwikkelde Nieuwe Producten openbaar wenst te maken en/of aan derden ter beschikking wenst te stellen en/of anderszins wenst te exploiteren, dient hij eerst aan INT een exclusieve licentie aan te bieden inhoudende dat INT gerechtigd is de nieuwe producten middels de Webwinkel ter beschikking te stellen. INT zal binnen twee maanden na ontvangst van het aanbod van Gebruiker beslissen of zij een exclusieve licentie wenst.

8.2 Middels de exclusieve licentie bedoeld in het vorige lid zal Gebruiker aan INT in elk geval de volgende rechten verlenen: a. het recht de Nieuwe Producten openbaar te maken en te verveelvoudigen; b. het recht om sublicenties aan derden te verlenen ten behoeve van het gebruik van de Nieuwe Producten voor onderzoeks- en onderwijsdoeleinden; c. het recht om sublicenties aan derden te verlenen ten behoeve van het ontwikkelen en exploiteren van nieuwe producten door deze derden, onder de voorwaarde dat de Nieuwe Producten of een kwalitatief of kwantitatief substantieel gedeelte daarvan niet herkenbaar in de door de derden te ontwikkelen nieuwe producten zijn opgenomen, verveelvoudigd of overgenomen; d. het recht om de Nieuwe Producten te gebruiken ten behoeve van het (laten) ontwikkelen en exploiteren van nieuwe producten.

Randtikel 9. Duur en einde Overeenkomst

9.1 Deze Overeenkomst vangt aan op het moment dat Gebruiker akkoord is gegaan met deze Voorwaarden en het Product heeft gedownload (zie Randtikel 3).

9.2 INT is gerechtigd deze Overeenkomst tussentijds met onmiddellijke ingang zonder voorafgaande opzegging te beëindigen, indien:

9.3 Licentienemer in strijd handelt met één van de bepalingen van deze Overeenkomst en indien Licentienemer niet binnen 14 (veertien) dagen na aanschrijving dienaangaande door INT is nagekomen;

9.4 Licentienemer zelf het faillissement, surseance van betaling of toepassing van de schuldsaneringsregeling natuurlijke personen aanvraagt, indien Licentienemer failliet is verklaard, indien aan Licentienemer surseance van betaling is verleend of indien ten aanzien van Licentienemer de schuldsaneringsregeling natuurlijke personen van toepassing is verklaard.

9.5 Indien deze Overeenkomst om welke reden dan ook eindigt, is INT niet aansprakelijk voor eventuele schade die Licentienemer lijdt ten gevolge van het beëindigen van de Overeenkomst.

9.6 Vanaf het moment dat de Overeenkomst om welke reden dan ook eindigt, beschikt Licentienemer niet langer over de rechten, die INT middels deze Overeenkomst aan Licentienemer heeft verleend.

9.7 Licentienemer verplicht zich, indien de Overeenkomst om welke reden dan ook eindigt, met ingang van de datum van beëindiging ieder gebruik van (de inhoud van) Product, of een gedeelte daarvan, te staken en gestaakt te houden.

9.8 Licentienemer is verplicht om binnen dertig (30) dagen na de datum van beëindiging van deze Overeenkomst (alle verkregen en gebruikte gegevens en componenten van) Product samen met de door Licentienemer gemaakte back-up(s) te vernietigen en die vernietiging schriftelijk aan INT te bevestigen. Gebruiksovereenkomst Lassy Klein-Corpus 4/4 Niet-commercieel

Randtikel 10. Geschillen en toepasselijk recht

10.1 Op deze Overeenkomst is Nederlands recht van toepassing.

10.2 In geval van geschillen, voortvloeiend uit deze Overeenkomst of uit daarop voortbouwende overeenkomsten, zullen deze worden voorgelegd aan de bevoegde rechter te Den Haag.

10.3 Afwijkende bedingen, wijzigingen van en/of aanvullingen op deze Overeenkomst gelden slechts indien en voor zover deze tussen Licentienemer en INT uitdrukkelijk schriftelijk zijn overeengekomen.

10.4 Indien een bepaling van deze Overeenkomst nietig is of vernietigd wordt, blijven de overige bepalingen volledig van kracht. Licentienemer en INT zullen dan in overleg treden teneinde een nieuwe bepaling ter vervanging van de nietige of vernietigde bepaling overeen te komen, waarbij zo veel mogelijk met het doel en strekking van de nietige of vernietigde bepaling rekening zal worden gehouden.

H.2 Wai-Not License

Overeenkomst voor niet-commercieel gebruik

Randtikel 1. Definities

Product: Het WAI-NOT corpus, volledige, correcte naam: WAI-NOT Corpus, alsmede de totale inhoud van de gedownloade of meegeleverde bestanden, daaronder begrepen maar niet beperkt tot (i) eventueel meegeleverde of van Product deel uitmakende software of computerinformatie en (ii) bijbehorende schriftelijke materialen of bestanden ter uitleg;

Overeenkomst: de onderhavige licentieovereenkomst;

Gebruiker: de natuurlijke persoon die deze Overeenkomst via de Webwinkel heeft geaccepteerd overeenkomstig het in artikel 3 bepaalde.

INT: Het Instituut voor de Nederlandse Taal, gevestigd in Leiden.

Webwinkel: De service beschikbaar op de website van INT waarbij software en data gedownload kan worden.

Randtikel 2. Toepasselijkheid

Deze gebruiksvoorwaarden zijn van toepassing op Product dat door het INT beschikbaar worden gesteld op basis van deze gebruiksovereenkomst en op de daarmee samenhangende rechtsverhouding tussen INT en Gebruiker.

Randtikel 3. Totstandkoming Overeenkomst

De overeenkomst komt tot stand indien Gebruiker bij de bestelprocedure van het product bij de Webwinkel op de knop ‘Akkoord’ heeft geklikt.

Randtikel 4. Gebruiksvoorwaarden

4.1 INT verleent hierbij aan Gebruiker, en Gebruiker accepteert, het niet-exclusieve recht om: a. Product te raadplegen en/of te gebruiken voor eigen onderzoek en ter toelichting bij het eventueel door Gebruiker gegeven onderwijs; b. Product te gebruiken ten behoeve van het ontwikkelen van nieuwe producten (hierna: “Nieuwe Producten”), mits (de inhoud van) Product, een gedeelte van Product of een kwalitatief of kwantitatief substantieel gedeelte daarvan niet herkenbaar in de Nieuwe Producten is opgenomen, verveelvoudigd of overgenomen en mits daarmee niet in strijd wordt gehandeld met Randtikel 4.2 en 4.6 van deze Overeenkomst. Het is Gebruiker niet toegestaan Product voor commerciële doeleinden te gebruiken.

4.2 De overeenkomstig Randtikel 4.1 sub b van deze Overeenkomst ontwikkelde Nieuwe Producten mogen uitsluitend door Gebruiker worden gebruikt voor eigen oefening, studie of gebruik en niet worden openbaar gemaakt, aan derden verkocht, uitgeleend en/of op andere wijze aan derden ter beschikking worden gesteld, tenzij en voor zover de INT daarvoor uitdrukkelijk schriftelijk toestemming heeft gegeven overeenkomstig Randtikel 8 van deze Overeenkomst.

4.3 Gebruiker is gerechtigd in openbare presentaties, wetenschappelijke publicaties of publicaties voor onderwijsdoeleinden naar Product te verwijzen, melding te maken van het gebruik van Product en/of verslag te doen van werk waarbij van Product gebruik is gemaakt. Gebruiker noemt daarbij expliciet de volledige juiste naam van Product. Gebruiksovereenkomst WAI-NOT Corpus 2/4 Niet-commercieel

4.4 Voor zover software onderdeel uitmaakt van Product, mag de software uitsluitend

worden geïnstalleerd op de computers van de Gebruiker.

4.5 INT verleent aan Gebruiker het recht kopieën van Product te maken die dienen als back-up. Gebruiker is niet gerechtigd om meer kopieën van Product te maken dan strikt noodzakelijk is voor het in Randtikel 4.1 omschreven doel.

4.6 Gebruiker heeft uitsluitend het recht Product te gebruiken voor de doeleinden omschreven in Randtikel 4.1 van deze Overeenkomst. Gebruiker heeft niet het recht (de inhoud van) Product of enig (kwalitatief of kwantitatief substantieel) gedeelte daarvan te reproduceren en/of te verveelvoudigen, tenzij dit noodzakelijk is voor het doeleinde genoemd in Randtikel 4.1 sub b van deze Overeenkomst of wanneer dit is toegestaan op grond van Randtikel 4.4 en/of Randtikel 4.5 van deze Overeenkomst. Gebruiker heeft niet het recht (de inhoud van) Product of enig (kwalitatief of kwantitatief substantieel) gedeelte daarvan openbaar te maken, aan derden te verkopen, uit te lenen en/of op andere wijze aan derden ter beschikking te stellen.

4.7 Gebruiker heeft niet het recht om op eventuele met Product meegeleverde software of software die onderdeel uitmaakt van Product technieken toe te passen waarmee de interne werking kan worden achterhaald, hieronder begrepen maar niet beperkt tot 'reverse engineering'.

4.8 De rechten die middels deze Overeenkomst aan Gebruiker worden verleend, zijn niet overdraagbaar. Gebruiker heeft niet het recht de middels deze Overeenkomst verleende licentie in sublicentie te geven.

Randtikel 5. Vergoeding, betaling en levering

5.1 Voor de aan Gebruiker verleende rechten, zoals bedoeld in Randtikel 4 van deze Overeenkomst, is geen vergoeding verschuldigd.

Randtikel 6. Fouten en onvolkomenheden

6.1 Indien Gebruiker in welk onderdeel dan ook van Product fouten (bugs), onvolkomenheden, inconsequenties e.d. aantreft, dan wordt Gebruiker verzocht die aan INT schriftelijk of elektronisch te melden. INT verplicht zich ertoe gemelde fouten (bugs), onvolkomenheden, inconsequenties e.d. te publiceren. Het publiceren van lijsten van fouten, onvolkomenheden e.d. is voorbehouden aan INT. Het is Gebruiker wel toegestaan om fouten (bugs), onvolkomenheden, inconsequenties e.d. te melden in openbare presentaties, wetenschappelijke publicaties of publicaties voor onderwijsdoeleinden, wanneer verwezen wordt naar Product, melding wordt gemaakt van het gebruik van Product en/of verslag wordt gedaan van werk waarbij van Product gebruik is gemaakt.

6.2 INT geeft geen enkele garantie en accepteert geen enkele verantwoordelijkheid voor welke beperkingen of fouten in Product dan ook en accepteert geen enkele aansprakelijkheid voor schade, verlies of ongerief dat zou kunnen voortkomen uit het gebruik van Product.

6.3 INT geeft geen enkele garantie dat Product of een gedeelte ervan voor bepaalde specifieke doeleinden kan worden gebruikt.

Randtikel 7. Intellectuele Eigendomsrechten

7.1 In geen geval zal Gebruiker auteursrechten, databankrechten en of andere (intellectuele eigendoms)rechten ten aanzien van Product verwerven.

7.2 Gebruiker erkent dat het gebruik van Product onderworpen is aan de restricties die op grond van Nederlands recht door wetten van intellectuele eigendom en andere

vormen van wettelijke bescherming worden opgelegd, waaronder begrepen maar niet beperkt tot auteursrechten, naburige rechten, databankenrechten en rechten op software en dat schendingen van zulke restricties leiden tot wettelijke aansprakelijkheid. Gebruiker onthoudt zich van het schenden van deze restricties. Bijgevolg publiceert Gebruiker geen onderdelen van Product (zoals teksten, geluidsfragmenten of andersoortige data en/of tools), anders dan korte voorbeelden in wetenschappelijke publicaties of publicaties voor onderwijsdoeleinden. Gebruiksovereenkomst WAI-NOT Corpus 3/4 Niet-commercieel

7.3 Voor zover blijkt dat door aanpassingen, bewerkingen en/of aanvullingen van Product door Gebruiker eigen of nieuwe rechten zouden kunnen ontstaan ten aanzien van Product, worden deze hierbij bij voorbaat volledig en onbezwaard door Gebruiker aan INT overgedragen. INT aanvaardt hierbij deze overdracht.

7.4 Voor zover de overdracht zoals bedoeld in het vorige lid middels deze Overeenkomst niet wordt bewerkstelligd of niet mogelijk blijkt, verbindt Gebruiker zich om op eerste verzoek van INT kosteloos alles te doen wat nodig is om de overdracht van alle intellectuele eigendomsrechten ten aanzien van Product te effectueren. Voorts verleent Gebruiker hierbij voor zover nodig tot het moment van volledige overdracht een onbeperkte exclusieve licentie aan INT, welke licentie INT hierbij aanvaardt.

Randtikel 8. Optierecht

8.1 Indien Gebruiker de overeenkomstig Randtikel 4.1 sub b ontwikkelde Nieuwe Producten openbaar wenst te maken en/of aan derden ter beschikking wenst te stellen en/of anderszins wenst te exploiteren, dient hij eerst aan INT een exclusieve licentie aan te bieden inhoudende dat INT gerechtigd is de nieuwe producten middels de Webwinkel ter beschikking te stellen. INT zal binnen twee maanden na ontvangst van het aanbod van Gebruiker beslissen of zij een exclusieve licentie wenst.

8.2 Middels de exclusieve licentie bedoeld in het vorige lid zal Gebruiker aan INT in elk geval de volgende rechten verlenen: a. het recht de Nieuwe Producten openbaar te maken en te verveelvoudigen; b. het recht om sublicenties aan derden te verlenen ten behoeve van het gebruik van de Nieuwe Producten voor onderzoeks- en onderwijsdoeleinden; c. het recht om sublicenties aan derden te verlenen ten behoeve van het ontwikkelen en exploiteren van nieuwe producten door deze derden, onder de voorwaarde dat de Nieuwe Producten of een kwalitatief of kwantitatief substantieel gedeelte daarvan niet herkenbaar in de door de derden te ontwikkelen nieuwe producten zijn opgenomen, verveelvoudigd of overgenomen; d. het recht om de Nieuwe Producten te gebruiken ten behoeve van het (laten) ontwikkelen en exploiteren van nieuwe producten.

Randtikel 9. Duur en einde Overeenkomst

9.1 Deze Overeenkomst vangt aan op het moment dat Gebruiker akkoord is gegaan met deze Voorwaarden en het Product heeft gedownload (zie Randtikel 3).

9.2 INT is gerechtigd deze Overeenkomst tussentijds met onmiddellijke ingang zonder voorafgaande opzegging te beëindigen, indien:

9.3 Licentienemer in strijd handelt met één van de bepalingen van deze Overeenkomst en indien Licentienemer niet binnen 14 (veertien) dagen na aanschrijving dienaangaande door INT is nagekomen;

9.4 Licentienemer zelf het faillissement, surseance van betaling of toepassing van de schuldsaneringsregeling natuurlijke personen aanvraagt, indien Licentienemer failliet is verklaard, indien aan Licentienemer surseance van betaling is verleend of indien ten

aanzien van Licentienemer de schuldsaneringsregeling natuurlijke personen van toepassing is verklaard.

9.5 Indien deze Overeenkomst om welke reden dan ook eindigt, is INT niet aansprakelijk voor eventuele schade die Licentienemer lijdt ten gevolge van het beëindigen van de Overeenkomst.

9.6 Vanaf het moment dat de Overeenkomst om welke reden dan ook eindigt, beschikt Licentienemer niet langer over de rechten, die INT middels deze Overeenkomst aan Licentienemer heeft verleend.

9.7 Licentienemer verplicht zich, indien de Overeenkomst om welke reden dan ook eindigt, met ingang van de datum van beëindiging ieder gebruik van (de inhoud van) Product, of een gedeelte daarvan, te staken en gestaakt te houden.

9.8 Licentienemer is verplicht om binnen dertig (30) dagen na de datum van beëindiging van deze Overeenkomst (alle verkregen en gebruikte gegevens en componenten van) Product samen met de door Licentienemer gemaakte back-up(s) te vernietigen en die vernietiging schriftelijk aan INT te bevestigen. Gebruiksovereenkomst WAI-NOT Corpus 4/4 Niet-commercieel

Randtikel 10. Geschillen en toepasselijk recht

10.1 Op deze Overeenkomst is Nederlands recht van toepassing.

10.2 In geval van geschillen, voortvloeiend uit deze Overeenkomst of uit daarop voortbouwende overeenkomsten, zullen deze worden voorgelegd aan de bevoegde rechter te Den Haag.

10.3 Afwijkende bedingen, wijzigingen van en/of aanvullingen op deze Overeenkomst gelden slechts indien en voor zover deze tussen Licentienemer en INT uitdrukkelijk schriftelijk zijn overeengekomen.

10.4 Indien een bepaling van deze Overeenkomst nietig is of vernietigd wordt, blijven de overige bepalingen volledig van kracht. Licentienemer en INT zullen dan in overleg treden teneinde een nieuwe bepaling ter vervanging van de nietige of vernietigde bepaling overeen te komen, waarbij zo veel mogelijk met het doel en strekking van de nietige of vernietigde bepaling rekening zal worden gehouden.

Bibliography

- M. Abdou, V. Ravishankar, A. Kulmizev, and A. Søgaard. Word order does matter and shuffled language models know it. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6907–6919, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.476. URL <https://aclanthology.org/2022.acl-long.476>.
- J. Bloem. *Processing Verb Clusters*. Ph.d. thesis, Universiteit van Amsterdam, 2021. URL <https://www.lotpublications.nl/processing-verb-clusters>.
- S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL <https://aclanthology.org/D15-1075>.
- M. Breuker. *CEFR Labelling and Assessment Services*, pages 277–282. Springer International Publishing, Cham, 2023. ISBN 978-3-031-17258-8. doi: 10.1007/978-3-031-17258-8_16. URL https://doi.org/10.1007/978-3-031-17258-8_16.
- H. Broekhuis and N. Corver. *Syntax of Dutch: Verbs and Verb Phrases. Volume 2*. Amsterdam University Press, Mar. 2015. doi: 10.5117/9789089647313. URL <https://doi.org/10.5117/9789089647313>.
- H. Broekhuis and N. Corver. *Syntax of Dutch: Verbs and Verb Phrases. Volume 3*. Amsterdam University Press, 2016. doi: 10.26530/oopen_614910. URL https://doi.org/10.26530/oopen_614910.
- C. Bryant, M. Felice, Ø. E. Andersen, and T. Briscoe. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4406. URL <https://aclanthology.org/W19-4406>.
- A. Chiche and B. Yitagesu. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9(1), 2022. doi: 10.1186/s40537-022-00561-y. URL <https://doi.org/10.1186/s40537-022-00561-y>.
- B. Comrie, G. G. Corbett, and G. Stone. The leipzig glossing rules: Conventions for interlinear morpheme-by-morpheme glosses, 2008. URL <https://www.eva.mpg.de/lingua/resources/glossing-rules.php>.

- A. V. Cranenburgh, R. Scha, and R. Bod. Data-oriented parsing with discontinuous constituents and function tags. *Journal of Language Modelling*, 4(1), Apr. 2016. doi: 10.15398/jlm.v4i1.100. URL <https://doi.org/10.15398/jlm.v4i1.100>.
- R. Dale and A. Kilgarriff. Helping our own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 242–249, Nancy, France, Sept. 2011. Association for Computational Linguistics. URL <https://aclanthology.org/W11-2838>.
- W. de Vries and M. Nissim. As good as new. how to successfully recycle English GPT-2 to make models for other languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 836–846, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.74. URL <https://aclanthology.org/2021.findings-acl.74>.
- W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim. BERTje: A Dutch BERT Model, 2019. URL <https://arxiv.org/abs/1912.09582>.
- P. Delobelle, T. Winters, and B. Berendt. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.292. URL <https://aclanthology.org/2020.findings-emnlp.292>.
- J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019. URL <https://arxiv.org/abs/1810.04805>.
- M. S. Dryer. Order of subject, object and verb (v2020.3). In M. S. Dryer and M. Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo, 2013. doi: 10.5281/zenodo.7385533. URL <https://doi.org/10.5281/zenodo.7385533>.
- K. Erdocia and I. Laka. Negative Transfer Effects on L2 Word Order Processing. *Frontiers in Psychology*, 9, 2018. doi: 10.3389/fpsyg.2018.00337. URL <https://doi.org/10.3389/fpsyg.2018.00337>.
- I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL <http://www.deeplearningbook.org>.
- R. Grundkiewicz, C. Bryant, and M. Felice. A crash course in automatic grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics: Tutorial Abstracts*, pages 33–38, Barcelona, Spain (Online), Dec. 2020. International Committee for Computational Linguistics. doi: 10.18653/v1/2020.coling-tutorials.6. URL <https://aclanthology.org/2020.coling-tutorials.6>.
- P. Jordens. The acquisition of word order in Dutch and German as L1 and L2. *Second Language Research*, 4(1):41–65, 1988. URL <https://www.jstor.org/stable/43104372>.
- D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*.

- Unpublished, but publicly available., Stanford, CA and Boulder, CO, 3rd edition, 2021. URL <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>.
- S. Kiyono, J. Suzuki, M. Mita, T. Mizumoto, and K. Inui. An empirical study of incorporating pseudo data into grammatical error correction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1236–1242, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1119. URL <https://aclanthology.org/D19-1119>.
- K. Lasri, A. Lenci, and T. Poibeau. Word order matters when you increase masking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1815, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. URL <https://aclanthology.org/2022.emnlp-main.118>.
- J. Lee and S. Seneff. Correcting misuse of verb forms. In *Proceedings of ACL-08: HLT*, pages 174–182, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL <https://aclanthology.org/P08-1021>.
- Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL <https://doi.org/10.48550/arXiv.1907.11692>.
- A. Lüdeling. *Mehrdeutigkeiten und Kategorisierung: Probleme bei der Annotation von Lernerkorpora*, pages 119–140. Max Niemeyer Verlag, Berlin, New York, 2008. ISBN 9783484970342. doi: doi:10.1515/9783484970342.2.119. URL <https://doi.org/10.1515/9783484970342.2.119>.
- H. T. Ng, S. M. Wu, Y. Wu, C. Hadiwinoto, and J. Tetreault. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics. URL <https://aclanthology.org/W13-3601>.
- H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-1701. URL <https://aclanthology.org/W14-1701>.
- J. O’Connor and J. Andreas. What context features can transformer language models use? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 851–864, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.70. URL <https://aclanthology.org/2021.acl-long.70>.
- A. Pauwels. *De plaats van het hulpwerkwoord, verleden deelwoord en infinitief in de Nederlandse bijzin*. M. & L. Symons, Leuven, 1953. URL https://www.dbnl.org/tekst/pauw022plaa01_01/colofon.php.

- A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. URL <https://doi.org/10.1101/2020.12.15.422761>.
- M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL <https://aclanthology.org/2020.acl-main.442>.
- A. Rozovskaya and D. Roth. Training paradigms for correcting errors in grammar and usage. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 154–162, Los Angeles, California, June 2010. Association for Computational Linguistics. URL <https://aclanthology.org/N10-1018>.
- K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019. URL <https://arxiv.org/abs/1907.10641>.
- J. J. Schepens. *Bridging linguistic gaps: The effects of linguistic distance on adult learnability of Dutch as an additional language*. PhD thesis, Radboud University, Nijmegen, The Netherlands, 2015.
- K. Sinha, R. Jia, D. Hupkes, J. Pineau, A. Williams, and D. Kiela. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913, Online and Punta Cana, Dominican Republic, Nov. 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.230. URL <https://aclanthology.org/2021.emnlp-main.230>.
- K. Sinha, P. Parthasarathi, J. Pineau, and A. Williams. UnNatural Language Inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7329–7346, Online, Aug. 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.569. URL <https://aclanthology.org/2021.acl-long.569>.
- I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014. URL <http://arxiv.org/abs/1409.3215>.
- A. Tack, E. Kochmar, Z. Yuan, S. Bibauw, and C. Piech. The BEA 2023 shared task on generating AI teacher responses in educational dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 785–795, Toronto, Canada, July 2023. Association for Computational Linguistics. URL <https://aclanthology.org/2023.bea-1.64>.
- J. Turner and E. Charniak. Language modeling for determiner selection. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 177–180, Rochester, New York, Apr. 2007. Association for Computational Linguistics. URL <https://aclanthology.org/N07-2045>.

- T. van der Wouden, H. Hoekstra, M. Moortgat, B. Renmans, and I. Schuurman. Syntactic analysis in the spoken Dutch corpus (CGN). In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain, May 2002. European Language Resources Association (ELRA). URL <http://www.lrec-conf.org/proceedings/lrec2002/pdf/71.pdf>.
- G. van Noord. Huge parsed corpora in Lassy. In *Proceedings of TLT7*, Groningen, The Netherlands, 2009. LOT. URL https://www.researchgate.net/publication/228385628_Huge_parsed_corpora_in_lassy.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017. URL <https://arxiv.org/abs/1706.03762>.
- J. Verhagen. Verb placement in second language acquisition: Experimental evidence for the different behavior of auxiliary and lexical verbs. *Applied Psycholinguistics*, 32(4):821–858, 2011. doi: 10.1017/S0142716411000087. URL <https://doi.org/10.1017/S0142716411000087>.
- J. Wagner, J. Foster, and J. van Genabith. A comparative evaluation of deep and shallow approaches to the automatic detection of common grammatical errors. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 112–121, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL <https://aclanthology.org/D07-1012>.
- A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL <https://aclanthology.org/W18-5446>.
- A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. SuperGlue: A stickier benchmark for general-purpose language understanding systems, 2020. URL <https://arxiv.org/abs/1905.00537>.
- S. Xu, J. Zhang, J. Chen, and L. Qin. Erroneous data generation for grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–158, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4415. URL <https://aclanthology.org/W19-4415>.
- M. Zhang. A survey of syntactic-semantic parsing based on constituent and dependency structures, 2020. URL <https://doi.org/10.48550/arXiv.2006.11056>.
- Y. Zhang, J. Baldridge, and L. He. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1131. URL <https://aclanthology.org/N19-1131>.