Computational Linguistics in the Netherlands Journal 13 (2024) 173-191 Submitted 02/2024; Published 03/2024

Exploring LLMs' Capabilities for Error Detection in Dutch L1 and L2 Writing Products

Joni Kruijsbergen joni.kruijsbergen@ugent.be
Serafina Van Geertruyen serafina.vangeertruyen@ugent.be
Véronique Hoste veronique.hoste@ugent.be
Orphée De Clercq orphee.declercq@ugent.be

LT3, Language and Translation Technology Team, Ghent University, Belgium

Abstract

This research examines the capabilities of Large Language Models for writing error detection, which can be seen as a first step towards automated writing support. Our work focuses on Dutch writing error detection, targeting two envisaged end-users: L1 and L2 adult speakers of Dutch. We relied on proprietary L1 and L2 datasets comprising writing products annotated with a variety of writing errors. Following the recent paradigms in NLP research, we experimented with both a fine-tuning approach combining different mono- (BERTje, RobBERT) and multilingual (mBERT, XLM-RoBERTa) models, as well as a zero-shot approach through prompting a generative auto-regressive language model (GPT-3.5). The results reveal that the fine-tuning approach outperforms zero-shotting to a large extent, both for L1 and L2, even though there is much room left for improvement.

1. Introduction

Some sort of automated writing support relying on Large Language Models (LLMs) is likely to become an integral part of human written language production in the (near) future. While these models can definitely assist with revising or even generating text, they cannot replace the critical thinking, creativity, and effective communication that are inherent to strong human writing skills (Kasneci et al. 2023). Therefore, both students and adults should learn to develop and maintain their writing abilities as a key part of their overall language skills.

Research has indicated that good and timely corrective feedback is beneficial for enhancing writing skills in both L1 and L2 writing (Biber et al. 2011, Kang and Han 2015, Link et al. 2022). But providing this type of feedback manually can be extremely time-consuming (Godwin-Jones 2022), even more so when considering the growing class sizes and shortage of teachers.

Today, writing support systems are prevalent thanks to advances in the field of Natural Language Processing (NLP) and machine learning. Most well-known are the systems for automated essay scoring (AES) - such as e-rater[1] - and automated writing evaluation (AWE) - such as Grammarly[2]. AES systems automatically grade writing products with machine learning by extracting linguistic characteristics from that product. AWE systems use the same techniques to both score a text and to provide the writer with diagnostic feedback. The latter requires the development of error detection or correction modules (Bryant et al. 2023). In the research presented here we want to stress the pedagogical importance of detection over correction, as detection promotes self-correction and language learning by helping learners identify and understand their mistakes (Volodina et al. 2023). A reliable and accurate detection system can effectively identify errors in text, enabling targeted corrections and improving the overall quality of writing (Yuan et al. 2021).

While many systems are readily available for English writing support, research on Dutch is lagging behind. Most of the current systems for Dutch, *Schrijfhulp* in particular (De Wachter et al. 2016),

1. https://www.ets.org/erater.html
2. https://www.grammarly.com/

only apply NLP methods sparingly (Verlinde et al. 2019) and rely mostly on rule-based systems. This causes the system to incorrectly flag errors where there are none, possibly causing confusion in users, which might have a negative impact on motivation. In order to make these systems more intelligent, a machine learning approach could be investigated. However, for Dutch few corpora comprising authentic writing products are publicly available, let alone corpora which have been manually labelled with possible writing errors.

In this respect the paradigm shifts the NLP field experienced in recent years open up new opportunities. While large task-specific training sets with labelled examples were a necessary prerequisite to develop robust NLP systems, pre-trained large language models (LLMs) which are fine-tuned on only a fraction of those labelled examples have revealed an impressive performance on a wide variety of downstream NLP tasks (Min et al. 2023). Moreover, together with the size of these LLMs growing over 100 billion parameters emerged the belief that these models actually possess reasoning abilities and are capable of carrying out specific tasks with no fine-tuning at all, also known as zero-shot learning (Liu et al. 2023).

In this paper we wish to explicitly focus on these two paradigms (fine-tuning and zero-shot learning) for the task of error detection in Dutch writing products, targeting two envisaged audiences, namely L1 and L2 adult speakers of Dutch. For the native Dutch speakers we could rely on an existing dataset (Deveneyns and Tummers 2013). For the learners of Dutch as a second or foreign language we relied on an in-house dataset from the Leuven Language Institute. For fine-tuning, we experimented with current state-of-the-art Dutch LLMs as well as with multilingual models. For zero-shot learning, a generative auto-regressive language model was prompted.

Our results show that fine-tuning outperforms the zero-shot approach to a large extent, both in the L1 and L2 use case. Moreover, we found that evaluating zero-shot output comes with a high post-processing cost, even after extensive prompt engineering.

The remainder of this paper is structured as follows: Section 2 discusses related research in the field of automated writing support. Section 3 then gives a more thorough insight in the two datasets that were used for this research, while Section 4 demonstrates and explains the chosen approaches for the experiments. In Section 5 the results are presented and discussed through means of an in-depth error analysis. Section 6 concludes the paper while also offering prospects for future research.

2. Related Work

Automated writing support systems have been extensively researched for decades, starting in the 1960s (Page 1966). Currently most well-known are the systems for AES and AWE: AES systems automatically grade writing products, whereas AWE systems both assign a score and provide the writer with diagnostic feedback. The latter rely on specialised modules capable of either detecting or correcting errors. Important to note is that this research subfield within NLP is often referred to as grammatical error detection (GED) or correction (GEC), however, the term *grammatical* is used as an umbrella term, encompassing other types of errors, including lexical, orthographic, and syntactical ones (Bryant et al. 2023). By addressing errors and enhancing language proficiency, both tasks aim to improve the overall writing proficiency of learners.

While early research on GED and GEC tasks focused on rule-based systems that identify specific types of errors (Rei and Yannakoudakis 2016), advances were mainly made thanks to employing machine learning approaches and the advent of several shared tasks focusing on GED and GEC, such as CoNLL (Ng et al. 2014), BEA (Bryant et al. 2019), and the recent MultiGED shared task (Volodina et al. 2023).

Recent research in GED seems to follow the overall trend in NLP to fine-tune pre-trained LLMs as they have revealed an impressive performance on a wide variety of downstream NLP tasks (Min et al. 2023). In the MultiGED shared task consisting of a general GED task on L2 sets from five different languages: English, German, Italian, Swedish and Czech (Volodina et al. 2023), the most successful system, by Colla et al. (2023), fine-tuned a pre-trained multilingual language model, XLM-

RoBERTa (Conneau et al. 2020), on each language separately, whereas the runner-up fine-tuned a similar system for all languages at once (Le-Hong et al. 2023). The best approach showed promising $F_{0.5}$ scores of up to 82.32% for German and 82.15% for Italian.

Zero-shot learning is also being explored for GED as ever-growing LLMs, such as OpenAI's GPT models, have proven to be powerful tools for a variety of NLP tasks (Min et al. 2023). Recent research analysing the use of these models (specifically GPT-3.5 and GPT-4) for GED (Coyne et al. 2023) showed some interesting results, with their best performing prompt reaching $F_{0.5}$ scores of 49.66% (GPT-3.5) and 52.79% (GPT-4) on the test set from the BEA-2019 (Bryant et al. 2019) shared task on GEC.

Though work on languages other than English is emerging, most research has been performed on English data, which creates inequality towards other, lower-resourced languages. In addition, multilingual LLMs are known to be biased to English as they are often pre-trained on more English data than any other language (Søgaard 2022, Volodina et al. 2023). Therefore, the focus of the research presented here is on Dutch. Existing tools for Dutch error correction are mainly rule-based. Some examples are the L1 and L2 Schrijfassistent[3] tools from the Leuven Language Institute (ILT) of KU Leuven, LanguageTool[4] and Sapling[5].

The two ILT tools are used by entering a text, processing it and then clicking on specific error types. Erroneous words are highlighted and depending on the error type some general rules regarding the inputted sentences are offered as feedback. Moreover, because the tool is mostly based on database look-ups, it tends to overgenerate. For example, connectives such as "maar" (*EN*: but) are always highlighted and then the user is presented with the feedback: "are you sure you want to express a contradiction here". The user thus needs to be very engaged and motivated to deduce for themselves whether the feedback is actually relevant. On top of that, a good command of the Dutch language is required when reading through the feedback.

With LanguageTool and Sapling, the user also enters a text and after processing, all words containing errors are underlined with a specific colour. The colour depends on the error type, of which there are three: spelling, grammar and punctuation. However, besides indicating these specific error types and their correction, no additional feedback is given. Additionally, these tools tend to overgenerate. The main problem with these rule-based systems is that they are very labour-intensive to develop, as with each exception either comes the adaptation of an existing rule, or a whole new rule.

Research on Dutch is thus lagging behind, which is why we wish to explicitly focus on the two state-of-the-art NLP paradigms for the task of error detection in Dutch writing products.

3. Datasets

We relied on two existing annotated datasets collected for different purposes. The L1 dataset (Section 3.1) was created to map the different writing errors first-year-students in professional bachelor's programmes make, as well as their frequency and spread. The L2 dataset (Section 3.2) is output of an annotation tool used by teachers of Dutch as a second or foreign language to offer feedback on the first draft of writing exercises. As both datasets had been annotated already, and re-annotation fell beyond the scope of the current project, the annotations differ considerably. The sets are part of in-house projects and therefore not publicly available. Table 1 lists the sizes of both datasets, expressed in number of sentences and tokens, as well the respective error rate at the token level. In what follows, both datasets are explained in closer detail.

3. https://schrijfassistent.be/, https://nt2.schrijfassistent.be/
4. https://languagetool.org/spellchecking-dutch
5. https://sapling.ai/lang/dutch

	Sentences	Tokens	Error rate
L1	2534	46531	9.23
L2	5618	102474	13.11

Table 1: Number of sentences and tokens, as well as the error rate for both the L1 and L2 dataset.

3.1 L1

For the L1 data, we relied on a corpus (Deveneyns and Tummers 2013) comprising texts written by first year native Dutch speaking students in a variety of courses of study, all within professional higher education programmes at the former Catholic University College Leuven. Students were prompted to write an argumentative text about social media of around 500 words within a 1 hour time limit. The texts were written using a computer and participants were allowed to use any possible tools to aid in this task. The collected texts were screened for plagiarism using *TurnItIn*, but the specific tools participants may have used have not been documented. It should be noted though that this dataset was collected long before the introduction of generative AI, so these kind of tools were unavailable at the time.

For the experiments presented here, we had access to 90 texts which were manually annotated with the fine-grained error types as listed and exemplified in Table 2. For a translation of the examples, please refer to Appendix A. Important to note is that the dataset was annotated using the codified standard based on established reference works[6], rather than a language user's judgement to identify errors for the sake of objectivity. This very fine-grained error typology also has the side effect that some of the instances labelled as errors do not register as writing errors to most native writers of Dutch.

Error Type	**Explanation**	**Example** (Dutch)	%
Spelling	Incorrect	Dat moet niet *perse*.	6.53
Capitalisation	Incorrect	Kortom *Ik* vindt* het asociaal.	0.67
	Missing	Dit meldt *de morgen*.	
Lexicon	Non-existent	Het *chat-gedeelde* is maar een deel.	27.62
	Incorrect usage	Dit kan wel *is* tegenvallen.	
	Redundant word	Zoals *bijvoorbeeld* bij World of Warcraft.	
	Contamination	Zeker *jongeren* zijn hier zeer kwetsbaar voor.	
	Pleonasm	Je kan al je vrienden toevoegen die je *kent*.	
	Chat language	*Friend requests* worden gedaan*.	
	Loan word	Dit geeft een leuke *touch* aan je profiel.	
Grammar	DT-error	De wereld *bied* meer dan vrienden alleen.	41.32
	Verb congruence	De impact van netwerksites *hebben* gevolgen.	
	Anaphora	Er zijn niet alleen voordelen aan deze *sites*.	
	Incorrect link	Om te beginnen wordt *het* snel verslavend.	
	New referent	Er zijn ook tal van voordelen aan *deze* sites.	
Punctuation	Incorrect	Moet iedereen alles weten over iedereen.	23.85
	Missing	Als je het niet *hebt* hoor je er niet bij.	

Table 2: Error types present in the L1 dataset together with their respective proportions

In total, this dataset comprises 2,534 sentences, with 76.25% (n = 1,939) of the sentences having at least one writing error. Looking at the token level, 9.23% (n = 4,301) of the 46,531 tokens actually

6. Specifically, the reference works used were the *Woordenlijst Nederlandse Taal* (Van Sterkenburg and Beeken 2005) for spelling, *Van Dale Groot Woordenboek der Nederlandse Taal* (Boon and Geeraerts 2008) for lexicon, and the *Algemene Nederlandse Spraakkunst* or *ANS* (Haeseryn 1997) for grammar.

constitute writing errors. Considering the error proportions of the overarching error types, following the annotations, most writing errors are related to grammar (41.32%), followed by lexicon (27.62%), punctuation (23.85%), spelling (6.53%) and capitalisation (0.67%) errors.

3.2 L2

For the L2 data, we have been in contact with the Leuven Language Institute (ILT) of KU Leuven. This institute offers a range of Dutch courses for non-native speakers (NT2) and has developed a designated proprietary tool for giving feedback on writing assignments. The tool is used by all NT2 teachers to annotate erroneous words and potentially offer more specific feedback. While the tool is able to automatically detect spelling mistakes, most of the feedback is still provided manually by the NT2 teachers. This is done by highlighting one or more words and assigning an error type according to a predefined error typology. The full error typology is listed in Table 3. For a translation of the examples, please refer to Appendix A.

The annotation tool is actively being used since February 2019 and over time, the teachers have become more consistent in applying the error typology. However, it should be noted that they use this typology somewhat loosely and, as such, the error annotations are not always consistent. An example are inverted sentences such as *Morgen ik kom niet.* (*EN*: Tomorrow I will not come.), where some teachers indicate *ik* as a position error, some *kom*, some both words as one error and some both words as separate errors.

Error Type	**Explanation**	**Example** (Dutch)	%
Position	Incorrect word position	Morgen *ik kom* niet.	46.49
Spelling	Incorrect spelling	Ik doe het *onmiddelijk*	14.53
Lexicon	Incorrect word usage	Engels is mijn *moedertong.*	17.18
Grammar	Incorrect grammar	Zij *teleurstelde* mij. Dat is een *andere* probleem.	12.07
Punctuation	Incorrect punctuation	Vind je dat interessant.	2.09
Redundant	Redundant word	Dat hangt *er*af van het weer.	7.64

Table 3: Error types present in the L2 dataset together with their respective proportions.

For the experiments presented here we had access to 5,618 sentences which have all been error-annotated following the above-mentioned typology. Important to note is that all sentences in our dataset contain at least one position error. This is because after interviews with NT2 teachers, they indicated that position errors are pervasive mistakes and when thinking about more intelligent writing support, indicating this pervasive error would be the most interesting to include. In total our dataset consists of 13,433 errors, of 102,474 tokens, amounting to an error rate of 13.11%.

Given that our dataset comprises exclusively sentences with at least one position error, it is no surprise that this is also the most frequent error type (46.49%). This is followed by word choice errors (17.18%), spelling errors (14.53%) and grammatical errors (12.07%). Closing the list are redundant words (7.64%) and incorrect punctuation (2.09%).

Even though we rely on both an L1 and L2 dataset to test the current capabilities of LLMs in this paper, it is important to note that the sets differ inherently in various ways. Therefore their performance cannot be compared. The sets have been collected for different purposes, and the writing products come from a different target audience. Additionally, though the error types are comparable, native speakers and learners fundamentally differ in the mistakes they make. For example, the L1 set does not contain any position errors, whereas this is a pervasive error type for learners of Dutch. With regards to the error typology, there are two other differences. Firstly, incorrect capitalisation is a separate error type in the L1 data, whereas for L2, those errors are

classified as spelling errors. Secondly, the error types word choice and redundant word error types in the L2 dataset are included in the lexicon error type group in the L1 dataset.

4. Experiments

Our main objective is to explore LLMs capabilities for token-level error detection in Dutch texts written by both L1 and L2 writers. This boils down to a binary token classification task, where each token is assigned either the label *correct* (c) or *incorrect* (i). To this purpose two approaches have been investigated, representing two recent paradigms in NLP research (Min et al. 2023): fine-tuning both state-of-the-art Dutch monolingual and multilingual LLMs on the datasets at hand (Section 4.1) versus directly prompting a generative LLM (Section 4.2). A multi-class classification task with the various error types was considered, but discarded due to insufficient data.

For the experiments presented in this paper all data had been previously split at the sentence level, and we ensured that all sentences were present in the sets only once. Both datasets were split in a 80:10:10 training, validation, and test set respectively (see Table 4 for the error distribution in both sets). The training and development split were used for the fine-tuning experiments. The resulting model was evaluated on the held-out test set. Given that zero-shot learning does not require any training, the held-out test split was directly used to evaluate that approach.

	train		**dev**		**test**	
	c	i	c	i	c	i
L1	33850	3418	3634	364	4752	513
L2	73496	10560	7036	1318	8509	1555

Table 4: Number of correct (c) and incorrect (i) tokens in the different data splits

4.1 Fine-tuning

For the fine-tuning experiments, we required LLMs which have been pre-trained on Dutch data. To this purpose we relied on four LLMs, more specifically two Dutch or monolingual LLMs – BERTje (de Vries et al. 2019) and RobBERT (Delobelle et al. 2020) – and two multilingual ones – mBERT (Devlin et al. 2019) and XLM-RoBERTa (Conneau et al. 2020).

As for the monolingual models, BERTje (de Vries et al. 2019) and RobBERT (Delobelle et al. 2020) were employed. BERTje was pre-trained on a total of around 2.4 billion words (12GB) of high-quality Dutch texts which include the Dutch Sonar-500 (Oostdijk et al. 2013) and TwNC (Ordelman et al. 2007) corpora, Wikipedia data, historical fiction and a large collection of Dutch online newspaper articles collected over a four-year period. RobBERT, on the other hand, was pre-trained on around 6.6 billion words (39GB) coming from the Dutch section of the Common Crawl corpus (Suárez et al. 2019). For the experiments presented here, we relied on the updated version of RobBERT-v2 (77GB), as it comprises more data overall, but also specifically takes into account evolving language use, e.g. covid-related terms (Delobelle et al. 2022). Considering the difference in pre-training size RobBERT-v2 is thus trained on over six times the amount of Dutch data compared to BERTje.

This brings us to the multilingual models: Multilingual BERT (mBERT, Devlin et al., 2019) and XLM-RoBERTa (Conneau et al. 2020) have both been pre-trained on a multitude of languages (104 and 100, respectively). For mBERT, pre-training was performed on the text passages from the top 104 languages with the largest Wikipedias. While no documentation exist on the exact input used for training mBERT, we do know that the entire Wikipedia dumps at that time were employed.

Considering that the current Dutch Wikipedia has an approximate size of 3.44GB[7], we expect the input for mBERT to have been of a somewhat smaller size. XLM-RoBERTa (Conneau et al. 2020) was pre-trained on the entire Common Crawl corpus, with a total size of 2.5TB, of which 5,025 million Dutch tokens (31.46GB).

Considering the amount of pre-training data, it is clear that mBERT comprises much fewer data than all three other LLMs. This explains why in previous research (de Vries et al. 2019), the monolingual Dutch model BERTje has been found to outperform multilingual BERT on downstream NLP tasks, i.e., mainly because it is trained on a larger and more diverse dataset of Dutch tokens. In the case of XLM-RoBERTa, the Dutch pre-training set is considerably larger than the one used for BERTje (31GB vs 12GB), but much smaller than the RobBERT-v2 one (77GB). Recent research in grammatical error detection revealed that multilingual models, and especially XLM-RoBERTa, often outperform their monolingual counterparts when applied to languages other than English (Volodina et al. 2023).

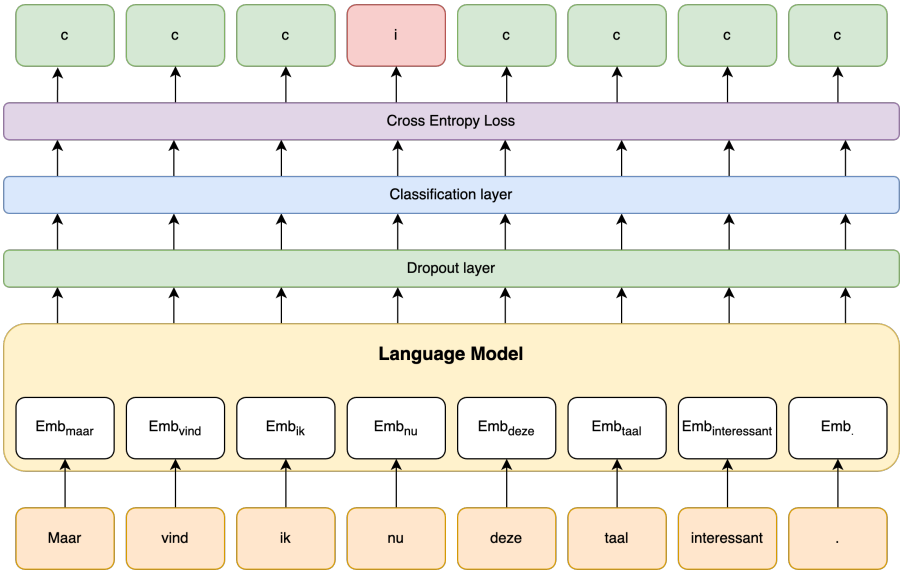

Figure 1: System architecture.

All fine-tuning experiments were performed in Flair using Hugging Face's *Transformerwordembeddings*. A linear classification layer was added on top of the different LLMs, with a dropout layer in between (p=0.1), as well as a locked dropout layer at the end (p=0.5), to prevent overfitting (Allein et al. 2020). Finally, a cross-entropy loss function was utilised to minimise the negative log probability of the correct label, following previous research on error detection (Bell et al. 2019, Kaneko and Komachi 2019, Knill et al. 2019). Figure 1 depicts the architecture.

Following best practices in deep learning (Ulmer et al. 2022) the models were fine-tuned ten times (five for each dataset) in order to be able to report the mean and standard deviation over multiple runs.

4.2 Zero-shot

For the zero-shot experiments, we explored prompting a generative auto-regressive decoder-only language model in order to carry out the error detection task. To this purpose, we relied on OpenAI's

7. As of February 2024, Dutch Wikipedia comprises 2,151,891 articles. With an average of 1,598 bytes per article, the current size can be estimated at 3.44GB

GPT-3.5 Turbo model with the default temperature parameter (n = 1) as this is the LLM underlying OpenAI's free and widely used ChatGPT interface[8]. Given that generative models are very good at generating human-like text (Herbold et al. 2023), we were mainly interested in exploring whether these models can also be used for grammatical error detection, as previous research on using zero-shot methods for grammatical error correction has shown promising results (Coyne et al. 2023). That study, however, was focused on English benchmark datasets for grammatical error correction. As far as we can tell, writing error detection, especially on Dutch text, has been less-studied.

Though prompting generative language models is a brittle task, best prompt engineering practices are emerging. In this respect, we were able to follow the recommendations from Coyne et al. (2023) which conclude that especially a higher level of detail increases the prompt's performance. Regarding the prompt language, we decided to compare both Dutch and English versions of the prompt in order to verify which language works best. Though OpenAI has not released exact details about the proportions of training data for each of the languages in GPT-3.5, it is assumed that it follows the representation of those languages on the World Wide Web, i.e. much more English compared to Dutch. When prompting, it is also important to consider how input and output will be specified, as this highly influences the level of manual post-processing which will be required afterwards.

We experimented with two different types of input. Our first experiments only included the sentences as such in the prompt, however, we quickly noticed that punctuation was not always recognised as a separate token even when this was explicitly stated in the prompt. This is why we decided to include a pre-tokenised version of the sentence so that tokenisation was not left to GPT-3.5. To this purpose, we relied on the sentence's tokenisation as provided in the original datasets. The model was asked to format the output as a table which contains the tokens of the sentence in the first column and the corresponding labels (correct or incorrect) in the second one. Though this format was not always consistently used and required some additional post-processing, it turned out to offer the most reproducible results for evaluation after experimenting with other output types such as the json format.

```
"Pretend you are a Dutch language teacher correcting sentences.[9]
Given are the following Dutch writing errors:
- spelling
- ...

And the following labels:
- "juist"
- "fout"

Create a table with all tokens from the sentence, assigning one
of the aforementioned labels to each token.  If a token contains
one of the given writing errors, it receives a "fout" label, all
other tokens are "juist".  Do not use any numbering or enumeration
marks and only analyse the following sentence:  {text_piece} The
tokens for this sentence are:  {tokenised_text_piece}"
```

Figure 2: Full English base prompt given to GPT-3.5 Turbo

Figure 2 illustrates the final fine-grained prompt used for our experiments. Please note that the final prompts are distinct from one another in two ways: (i) the prompt language (English versus

8. During its Developer Conference on 6 November 2023 OpenAI announced that the ChatGPT service has a 100 million active weekly users.

8. This sentence was not included in the Dutch prompt.

Dutch) and (ii) the dataset (L1 versus L2) and thus also the different possible Dutch writing errors that were given to the model, which correspond to the errors listed in Table 2 and 3.

4.3 Evaluation

Given the high class imbalance in the error detection task at hand (see Table 4) we cannot rely on accuracy as a suitable evaluation metric for this task. An alternative is to use F-score as this measure combines precision (the indicated errors that are actually errors) and recall (how many of the actual errors have been indicated as such). Traditionally, balanced F-score or F_1 is reported resulting in a harmonious mean of precision and recall. For this task, however, we rely on token-based $F_{0.5}$, allocating twice as much weight to precision than recall, following previous research surrounding GED (Volodina et al. 2023), as in Kaneko and Komachi (2019), Rei and Yannakoudakis (2016), Ng et al. (2014). Proposing erroneous corrections can, namely, have a much more negative impact than missing a few errors (Bell et al. 2019), and accurate feedback is thus more important than high coverage in error detection (Kaneko and Komachi 2019, Ng et al. 2014). However, the balance is kept through the use of the $F_{0.5}$-score.

Precision, recall and $F_{0.5}$ are computed using the total numbers of true positives (TP), false positives (FP) and false negatives (FN). Equations 1-3 demonstrate this.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F_{0.5} = 1 + 0.5^2 * \frac{Precision * Recall}{(0.5^2 * Precision) + Recall} \tag{3}$$

An important note when using token-based classification, is that multi-word edits are counted as separate errors, possibly influencing the results. As Volodina et al. (2023) mention, it might be more realistic to consider those as a single error. However, since single word errors are much more common than multi word errors, the impact should be minimal (Volodina et al. 2023).

5. Results & Discussion

In this section we first present the results of the fine-tuning experiments on the held-out test set, after which these are compared to directly prompting GPT-3.5 Turbo. We end this section with a thorough error analysis to shed more light on the (in)capabilities of the LLMs for GED (Section 5.1).

Table 5 and 6 present the results of the fine-tuning experiments for both datasets. The best results are indicated in bold. The results show that, both for L1 and L2, BERTje, RobBERT and XLM-RoBERTa yield similar results, whereas mBERT clearly falls behind, with on average -2.76% $F_{0.5}$ for L1 and -4.18% for L2. As explained in Section 4.1, this is in line with expectations as both BERTje, RoBERTa and XLM-RoBERTa have been pre-trained on more (diverse) Dutch data. Interesting, and in line with the outcome of the MultiGED shared task (Volodina et al. 2023), is that XLM-RoBERTa also outperforms the best monolingual model on the L2 dataset.

	P	**R**	**$F_{0.5}$**	**SD**
BERTje	0.6491	**0.3973**	**0.5761**	± 0.0084
RobBERT	0.6522	0.3680	0.5640	± 0.0134
mBERT	0.6248	0.3583	0.5438	± 0.0131
XLM-RoBERTa	**0.6545**	0.3863	0.5741	± 0.0108

Table 5: L1 results fine-tuning experiments.

	P	R	$F_{0.5}$	SD
BERTje	0.6572	0.5062	0.6202	± 0.0057
RobBERT	0.6842	**0.5124**	0.6412	± 0.0015
mBERT	0.6442	0.4510	0.5930	± 0.0076
XLM-RoBERTa	**0.6880**	0.5097	**0.6429**	± 0.0020

Table 6: L2 results fine-tuning experiments.

When comparing the monolingual LLMs, it is somewhat surprising that BERTje and RobBERT yield similar scores, which is especially the case for the L1 dataset where BERTje even outperforms RobBERT (56.40%) with an $F_{0.5}$ of 57.61%. This is remarkable given their difference in Dutch pre-training data size (12GB versus 77GB, respectively). However, we hypothesise that the better performance of BERTje for L1 stems from the underlying pre-training data and the inherently different errors produced by L1 and L2 writers. The L1 data, as well as the accompanying error annotations, are more formal and strict than the L2 data and annotations, which aligns with the more formal pre-training input from BERTje versus the more colloquial data on which RobBERT is pre-trained. With an overall best performance of 57.61% for L1 and 64.29% for L2, the fine-tuned LLMs show that there still is much room for improvement.

When looking at the output of the zero-shot experiments (Tables 7 and 8), where the best results are again indicated in bold, we observe that performance is significantly worse. For L1, the best $F_{0.5}$ is only slightly over 16% (compared to 57.61% when fine-tuning), and for L2 it barely reaches 30% (compared to 64.29% when fine-tuning). Fine-tuning is thus more suited for error detection, both in terms of error retrieval (recall) and relevance of the errors (precision).

	P	R	$F_{0.5}$
GPT-3.5 - English	**0.2158**	0.0799	**0.1610**
GPT-3.5 - Dutch	0.0957	**0.0819**	0.0926

Table 7: L1 results zero-shot experiments

	P	R	$F_{0.5}$
GPT-3.5 - English	0.3439	**0.1749**	0.2882
GPT-3.5 - Dutch	**0.3662**	0.1742	**0.3001**

Table 8: L2 results zero-shot experiments

In addition to the low zero-shot performance, we would also like to draw attention to the high level of post-processing the output demanded before it could be evaluated. A few examples are: (i) the absence of consistent formatting, (ii) skipping tokens and thus not giving them any label, (iii) inserting new tokens or entire sentences and then labelling them, and (iv) changing or correcting tokens directly instead of merely labelling them.

In order to be able to compare the evaluation of the zero-shot to the fine-tuned models, we decided to label all the skipped, inserted or changed tokens as *unknown* (unk). This unknown label was then each time evaluated as a misclassification, i.e. if the reference label was *correct*, it was counted as an *incorrect* label, and vice versa.

One of the main reasons that the English and Dutch prompts perform so differently for L1 ($F_{0.5}$-scores of 16.10% versus 9.26%) is that there were a lot more instances in the output from the Dutch prompt where tokens or even entire sentences were skipped or mistokenised compared to the output

from the English prompt (290 versus 27). This was not as outspoken in the L2 dataset ($F_{0.5}$-scores of 28.82% versus 30.01%).

5.1 Error Analyses

We decided to take a closer look - both quantitatively and qualitatively - at the false positives (actual correct tokens classified as incorrect) and the false negatives (actual incorrect tokens classified as correct) of both approaches. For the fine-tuning approach, we considered the predictions of the best models, i.e. BERTje for L1 ($F_{0.5}$ of 57.61%) and XLM-RoBERTa for L2 ($F_{0.5}$ of 64.29%). For the zero-shot approach, the output coming from the best prompts was analysed, i.e. the English prompt for L1 ($F_{0.5}$ of 16.10%) and the Dutch one for L2 ($F_{0.5}$ of 30.01%).

5.1.1 Error Analysis on L1

Looking at the confusion matrices presented in Figure 3, especially the difference in the number of true positives (TP) stands out. With 205 TPs for the fine-tuned model and only 41 for the prompt-based method, it is clear why the zero-shot approach has such a low $F_{0.5}$-score.

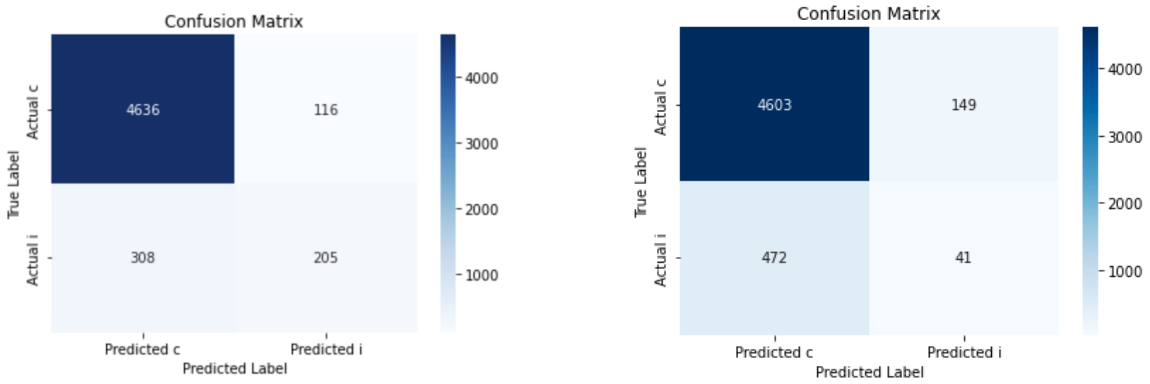

Figure 3: Confusion matrices of the predictions on L1 held-out set using the best fine-tuning (left) and zero-shot (right) approach.

Upon closer inspection of the erroneously classified tokens, it stood out that some of the wrong labels were actually due to issues with the annotations in the reference data. We manually verified all these instances for both models. For the fine-tuned BERTje model, this amounted to 56 (13.21%) out of the 424 misclassifications, bringing the total number of actual false positives back to 94 (22 instances removed) and of the false negatives to 274 (34 instances removed). For the zero-shot approach, 50 instances were considered to not be misclassified by the model, bringing the total number of incorrectly labelled tokens back to 118 actual false positives (31 instances removed) and 453 the actual false negatives (19 instances removed). Some of the most notable issues with the data were:

- Certain error types, specifically those that consist of multiple tokens (e.g. wrongly split verbs, anaphora and the incorrect links), were not always annotated on the same token, because multiple tokens could be classified as erroneous, but only one was labelled as such.

- Some errors present in the data were simply missed by the annotators, but correctly identified as errors by the model.

Because those annotation inconsistencies can have a significant impact on the models' performance, we deemed it interesting to examine the false positives and false negatives for both approaches after excluding the inconsistent annotations.

Regarding the remaining false positives for the fine-tuned model, we could not really identify any clear systematic problems with certain error types. For the zero-shot output the same holds as most false positives could be related to (i) erroneously labeling punctuation marks as a writing error, (ii) misclassifying tokens written all caps or (iii) tokenisation errors present in the reference set.

Type	**Fine-tuning**		**Zero-shot**	
	FN	**%**	**FN**	**%**
Spelling	18	6.57	26	5.74
Capitalisation	1	0.36	2	0.44
Lexicon	99	36.50	116	25.61
Grammar	88	32.12	191	42.16
Punctuation	68	24.82	118	26.05
Total	274	100	453	100

Table 9: Remaining FNs per error type (L1)

If we consider the proportions of the missed error types (false negatives) in Table 9 we observe that mostly Lexicon, Grammar and Punctuation errors are missed. Comparing both approaches, we discern that in the fine-tuned model, the Lexicon errors constitute the largest problem, followed by Grammar; whereas the opposite is true for the zero-shot approach.

5.1.2 Error Analysis on L2

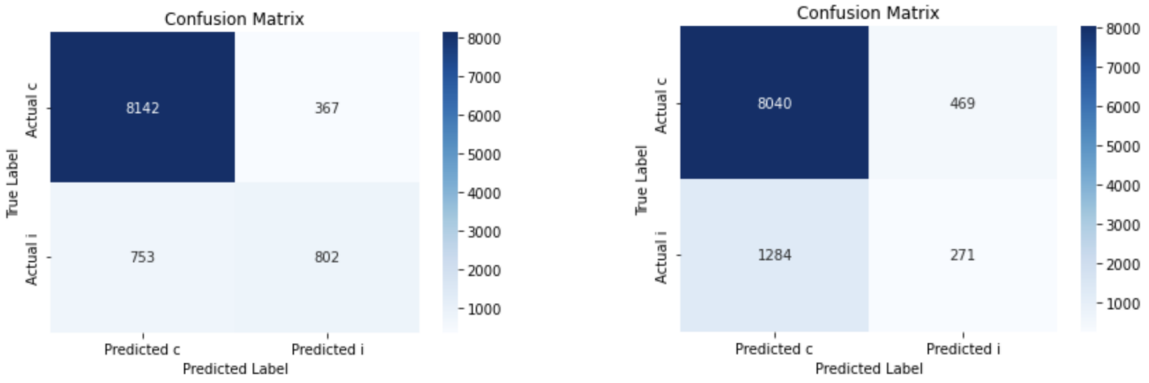

Figure 4: Confusion matrices of the predictions on L2 held-out set using the best fine-tuning (left) and zero-shot (right) approach.

As can be derived from the confusion matrices presented in Figure 4, the fine-tuned model yielded 8,142 true negatives and 802 true positives, whereas the zero-shot model had 8,040 true negatives, and only 271 true positives. This shows that the zero-shot model has more difficulties with identifying the actual errors (TP). Additionally, the fine-tuned model performs better with regards to the false positives (367 versus 469) and the false negatives (753 versus 1,284). The zero-shot system thus misclassifies more tokens (1,753 or 17.42% instances) than the fine-tuned one (1,120 or 11.12% instances), which is supported by the much lower scores for precision (36.62% versus 68.80%), recall (17.42% versus 50.97%) and $F_{0.5}$ (30.01% versus 64.29%) as presented in Table 6 and 8.

Through further examination of the wrongly classified tokens (FP & FN) for both the fine-tuned and zero-shot model, it soon became apparent that also in the L2 dataset there are some annotation inconsistencies.

In total, 427 (38.13%) of the 1,120 incorrectly labelled instances could be considered annotation inconsistencies for the fine-tuned model, bringing the actual false negatives back to 574, and the actual false positives to 248. Of the zero-shot output, 153 (8.73%) tokens were part of such inconsistencies. This brings the actual false positives for the GPT-3.5 output down to 351 and the actual false negatives to 1,249. The most common inconsistencies, which have mainly been caused by a lack of clear annotation guidelines, were that:

- The system indicated one word as erroneous, while the teacher had indicated the other.
- In some sentences, the system indicated two or more words that should be in another spot, whereas the teacher indicated one or more different words, that would amount to the same result, when switched.
- Some errors were simply not annotated as such by the teachers, yet correctly identified by the models.

Since those annotation inconsistencies can considerably affect the results (as explained in Subsection 5.1.1), we excluded them again for the analysis of the false positives and false negatives for both the fine-tuned and zero-shot output.

Regarding the false positives, the same tendencies were present as with the L1 data. For the false negatives, we inspected the error types of the erroneous tokens that the models did not identify as such. The distribution of those errors is presented in Table 10. The proportions show that the fine-tuned model is much better at identifying position errors than the GPT-3.5 model, which makes sense because of the overall high proportion of position errors in the training and development set, since every sentence in all sets contained at least one position error. Since no fine-tuning took place with the GPT-3.5 prompting, it is not surprising that it has more issues identifying the position errors.

Type	**Fine-tuning**		**Zero-shot**	
	FN	**%**	**FN**	**%**
Position	198	34.50	722	57.49
Spelling	84	14.63	109	8.68
Lexicon	126	21.95	188	14.97
Grammar	84	14.63	124	9.87
Punctuation	19	3.31	16	1.27
Redundant	63	10.98	97	7.72
Total	565	100	1256	100

Table 10: Remaining FNs per error type (L2)

6. Conclusion

With this paper, we aimed to explore the capabilities of current state-of-the-art Large Language Models on the task of error detection in Dutch writing products, targeting L1 and L2 adult speakers of Dutch.

Throughout this research, we explored the field of Grammatical Error Detection and its relevance in language education, driven by the increasing demand for effective error feedback. Since current research on automated writing support mainly focuses on English data and models, we aimed to

emphasise the importance of research on languages other than English, namely Dutch. As existing systems for Dutch are mainly rule-based, we wanted to examine the possibilities of both monolingual Dutch (BERTje & RobBERT) and multilingual (mBERT, XLM-RoBERTa & GPT-3.5) LLMs, as these have also shown to exhibit state-of-art performance in GEC for English.

To that end, we compared two recent paradigms in NLP research, (i) fine-tuning the LLMs on the L1 and L2 datasets, and (ii) directly prompting the generative GPT-3.5 model. The results indicate that fine-tuning outperforms zero-shotting on error detection tasks, with higher scores and a much lower post-processing load. We thus conclude that zero-shot prompting does not work efficiently for grammatical error detection.

Considering the top results we observe that for L1 writing products error detection works best when employing BERTje which has been pre-trained on edited data. For L2, on the other hand, XLM-RoBERTa outperforms the monolingual models, which is in line with previous research (Volodina et al. 2023). However, for both L1 and L2, multilingual XLM-RoBERTa scores quite comparably to the best scoring monolingual model for that dataset (BERTje and RobBERT, respectively). The choice in model should thus depend on what the end goal is. If the model is to be solely used for Dutch error detection, it might be best to choose a monolingual Dutch model. Whether the language in the data is more formal or colloquial can then be used to determine to opt for BERTje or RobBERT. If the model is also needed for error detection on other languages, the multilingual XLM-RoBERTa is the better option.

The error analyses demonstrated that the proprietary datasets we had been given access to have both been annotated relatively inconsistently, which directly impacts the fine-tuned models' effectiveness. Therefore, we want to stress the importance of creating representative and consistently annotated Dutch datasets of writing products for future research.

Acknowledgements

We would like to thank Devenyns and Tummers (2013) and the ILT for being able to use their datasets. Additionally, we would like to thank the reviewers for their valuable insights. This work was supported by Ghent University under grant BOF.STG.2022.0012.01 and by the Research Foundation–Flanders under grant number FWO.SPB.2023.0049.01.

References

Allein, Liesbeth, Artuur Leeuwenberg, and Marie-Francine Moens (2020), Automatically correcting dutch pronouns "die" and "dat", *Computational Linguistics in the Netherlands Journal* **10**, pp. 19–36.

Bell, Samuel, Helen Yannakoudakis, and Marek Rei (2019), Context is key: Grammatical error detection with contextual word representations, *arXiv preprint arXiv:1906.06593*.

Biber, Douglas, Tatiana Nekrasova, and Brad Horn (2011), The effectiveness of feedback for l1-english and l2-writing development: a meta analysis, *ETS Research Report Series* **2011** (1), pp. i–99.

Boon, T. and D. Geeraerts (2008), *Van Dale Groot Woordenboek van de Nederlandse Taal*, 14 ed., Van Dale Lexicografie, Utrecht.

Bryant, Christopher, Mariano Felice, Øistein E Andersen, and Ted Briscoe (2019), The bea-2019 shared task on grammatical error correction, *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 52–75.

Bryant, Christopher, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe (2023), Grammatical Error Correction: A Survey of the State of the Art, *Computational Linguistics* **49** (3), pp. 643–701. https://doi.org/10.1162/coli_a_00478.

Colla, Davide, Matteo Delsanto, Elisa Di Nuovo, et al. (2023), Elicode at multiged2023: fine-tuning xlm-roberta for multilingual grammatical error detection, *Linköping Electronic Conference Proceedings*, Vol. 197, Linköping University Electronic Press, pp. 24–34.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov (2020), Unsupervised cross-lingual representation learning at scale, *in* Jurafsky, Dan, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, pp. 8440–8451. https://aclanthology.org/2020.acl-main.747.

Coyne, Steven, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui (2023), Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction, *arXiv preprint arXiv:2303.14342*.

de Vries, Wietse, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim (2019), BERTje: A Dutch BERT Model. http://arxiv.org/abs/1912.09582.

De Wachter, Lieve, Margot D'Hertefelt, and Jordi Heeren (2016), De digitale schrijfhulp nederlands: Een procesgeorienteerde schrijfhulp ter bevordering van schrijfvaardigheid in het hoger onderwijs., *Van Schools tot Scriptie II* p. 49.

Delobelle, Pieter, Thomas Winters, and Bettina Berendt (2020), RobBERT: a Dutch RoBERTa-based Language Model, *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, Online, pp. 3255–3265. https://www.aclweb.org/anthology/2020.findings-emnlp.292.

Delobelle, Pieter, Thomas Winters, and Bettina Berendt (2022), Robbert-2022: Updating a dutch language model to account for evolving language use, *arXiv preprint arXiv:2211.08192*.

Deveneyns, Annelies and José Tummers (2013), Zoek de fout; een foutenclassificatie als aanzet tot gerichte remediëring nederlands in het hoger professioneel onderwijs in vlaanderen, *Levende Talen Tijdschrift* **14** (3), pp. 14–26.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019), Bert: Pre-training of deep bidirectional transformers for language understanding, *Proceedings of naacL-HLT*, Vol. 1, p. 2.

Godwin-Jones, Robert (2022), Partnering with ai: Intelligent writing assistance and instructed language learning, *Language Learning Technology* **26** (2), pp. 5–24, University of Hawaii National Foreign Language Resource Center; Center for Language Technology.

Haeseryn, W. et al. (1997), *Algemene Nederlandse Spraakkunst 2*, Martinus Nijhoff, Groningen.

Herbold, Steffen, Annette Hautli-Janisz, Ute Heuer, Zlata Kikteva, and Alexander Trautsch (2023), A large-scale comparison of human-written versus chatgpt-generated essays, *Scientific Reports*, Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1038/s41598-023-45644-9.

Kaneko, Masahiro and Mamoru Komachi (2019), Multi-head multi-layer attention to deep language representations for grammatical error detection, *Computación y Sistemas* **23** (3), pp. 883–891, Instituto Politécnico Nacional, Centro de Investigación en Computación.

Kang, EunYoung and Zhaohong Han (2015), The efficacy of written corrective feedback in improving l2 written accuracy: A meta-analysis, *The Modern Language Journal* **99** (1), pp. 1–18.

Kasneci, Enkelejda, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. (2023), Chatgpt for good? on opportunities and challenges of large language models for education, *Learning and individual differences* **103**, pp. 102274, Elsevier.

Knill, Kate M, Mark JF Gales, PP Manakul, and AP Caines (2019), Automatic grammatical error detection of non-native spoken learner english, *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 8127–8131.

Le-Hong, Phuong, Thi Minh Huyen Nguyen, et al. (2023), Two neural models for multilingual grammatical error detection, *Swedish Language Technology Conference and NLP4CALL*, pp. 40–44.

Link, Stephanie, Mohaddeseh Mehrzad, and Mohammad Rahimi (2022), Impact of automated writing evaluation on teacher feedback, student revision, and writing improvement, *Computer Assisted Language Learning* **35** (4), pp. 605–634, Routledge.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig (2023), Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3560815.

Min, Bonan, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth (2023), Recent advances in natural language processing via large pre-trained language models: A survey, Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3605943.

Ng, Hwee Tou, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant (2014), The CoNLL-2014 shared task on grammatical error correction, *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, Association for Computational Linguistics, Baltimore, Maryland, pp. 1–14. https://aclanthology.org/W14-1701.

Oostdijk, Nelleke, Martin Reynaert, Véronique Hoste, and Ineke Schuurman (2013), The construction of a 500-million-word reference corpus of contemporary written dutch, *in* Spyns, Peter and Jan Odijk, editors, *Essential Speech and Language Technology for Dutch, Results by the STEVIN-programme*, Theory and Applications of Natural Language Processing, Springer, pp. 219–247. https://doi.org/10.1007/978-3-642-30910-6_13.

Ordelman, Roeland J.F., Franciska M.G. de Jong, Adrianus J. van Hessen, and G.H.W. Hondorp (2007), Twnc: a multifaceted dutch news corpus, *ELRA Newsletter*.

Page, Ellis B. (1966), The imminence of... grading essays by computer, *The Phi Delta Kappan* **47** (5), pp. 238–243, Phi Delta Kappa International. http://www.jstor.org/stable/20371545.

Rei, Marek and Helen Yannakoudakis (2016), Compositional sequence labeling models for error detection in learner writing, *arXiv preprint arXiv:1607.06153*.

Søgaard, Anders (2022), Should we ban English NLP for a year?, *in* Goldberg, Yoav, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, pp. 5254–5260. https://aclanthology.org/2022.emnlp-main.351.

Suárez, Pedro Javier Ortiz, Benoît Sagot, and Laurent Romary (2019), Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures, *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, Leibniz-Institut für Deutsche Sprache.

Ulmer, Dennis, Elisa Bassignana, Max Müller-Eberstein, Daniel Varab, Mike Zhang, Rob van der Goot, Christian Hardmeier, and Barbara Plank (2022), Experimental standards for deep learning in natural language processing research, *in* Goldberg, Yoav, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, pp. 2673–2692. https://aclanthology.org/2022.findings-emnlp.196.

Van Sterkenburg, P. and J. Beeken (2005), *Woordenlijst Nederlandse Taal*, Lannoo, Tielt/Den Haag.

Verlinde, Serge, Lieve De Wachter, An Laffut, Kristin Blanpain, Geert Peeters, Ken Sevenants, and Margot D'Hertefelt (2019), Writing assistants: from word lists to nlp and artificial intelligence, *EUROCALL Conference 2019*, p. 161.

Volodina, Elena, Christopher Bryant, Andrew Caines, Orphée De Clercq, Jennifer-Carmen Frey, Elizaveta Ershova, Alexandr Rosen, and Olga Vinogradova (2023), MultiGED-2023 shared task at NLP4CALL: Multilingual grammatical error detection, *in* Alfter, David, Elena Volodina, Thomas François, Arne Jönsson, and Evelina Rennes, editors, *Proceedings of the 12th Workshop on NLP for Computer Assisted Language Learning*, LiU Electronic Press, Tórshavn, Faroe Islands, pp. 1–16. https://aclanthology.org/2023.nlp4call-1.1.

Yuan, Zheng, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant (2021), Multi-class grammatical error detection for correction: A tale of two systems, *in* Moens, Marie-Francine, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, pp. 8722–8736.

Appendix A. English translations

Nl = Original Dutch sentence
En = English translation

Error category	Example
Incorrect spelling	**Nl:** Dat moet niet *perse*. **En:** That is not necessarily the case.
Incorrect capitalisation	**Nl:** Kortom *Ik* vindt* het asociaal. **En:** In short, I find it antisocial.
Missing capitalisation	**Nl:** Dit meldt *de morgen*. **En:** This is what de morgen reports.
Non-existent lexicon	**Nl:** Het *chat-gedeelde* is maar een deel. **En:** The chat part is only part of it.
Incorrect word usage	**Nl:** Dit kan wel *is* tegenvallen. **En:** This may well disappoint.
Redundant word	**Nl:** Zoals *bijvoorbeeld* bij World of Warcraft. **En:** Like for example in World of Warcraft.
Contamination	**Nl:** Zeker *jongeren* zijn hier zeer kwetsbaar voor. **En:** Young people in particular are vulnerable to this.
Pleonasm	**Nl:** Je kan al je vrienden toevoegen die je *kent*. **En:** You can add all the friends that you know.
Chat language	**Nl:** *Friend requests* worden gedaan*. **En:** Friend requests are being sent.
Loan word	**Nl:** Dit geeft een leuke *touch* aan je profiel. **En:** This gives a nice touch to your profile.
DT-error	**Nl:** De wereld *bied* meer dan vrienden alleen. **En:** The world offers more than friends alone.
Verb congruence	**Nl:** De impact van netwerksites *hebben* gevolgen **En:** The impact of network sites has consequences.
Anaphora	**Nl:** Er zijn niet alleen voordelen aan deze *sites*. **En:** These sites do not only have advantages.
Incorrect link	**Nl:** Om te beginnen wordt *het* snel verslavend. **En:** To start with, it becomes addictive quickly.
New referent	**Nl:** Er zijn ook tal van voordelen aan *deze* sites. **En:** There are also many advantages to these sites.
Incorrect punctuation	**Nl:** Moet iedereen alles weten over iedereen*.* **En:** Does everyone have to know everything about everyone?
Missing punctuation	**Nl:** Als je het niet *hebt* hoor je er niet bij. **En:** If you do not have it you don't belong.

Table 11: Error types present in the L1 dataset with example sentences, including English translations. Errors were not transposed.

Error category	**Example**
Incorrect word position	**Nl:** Morgen *ik kom* niet. **En:** I am not coming tomorrow.
Incorrect spelling	**Nl:** Ik doe het onmiddelijk **En:** I will do it immediately.
Incorrect word usage	**Nl:** Engels is mijn *moedertong*. **En:** English is my mother tongue.
Incorrect grammar	**Nl:** Zij *teleurstelde* mij. **En:** She disappointed me. **Nl:** Dat is een *andere* probleem. **En:** That is a different problem.
Incorrect punctuation	**Nl:** Vind je dat interessant. **En:** Do you find that interesting?
Redundant word	**Nl:** Dat hangt *er*af van het weer. **En:** That depends on the weather.

Table 12: Error types present in the L2 dataset with example sentences, including English translations. Errors were not transposed.