Benchmarking the Simplification of Dutch Municipal Text

Abstract

Text simplification (TS) makes written information more accessible to all people, especially those with cognitive or language impairments. Despite much progress in TS due to advances in NLP technology, the bottleneck issue of lack of data for low-resource languages persists. Dutch is one of these languages that lack a monolingual simplification corpus. In this paper, we use English as a pivot language for the simplification of Dutch medical and municipal text. We experiment with augmenting training data and corpus choice for this pivot-based approach. We compare the results to a baseline and an end-to-end LLM approach using the GPT 3.5 Turbo model. Our evaluation shows that, while we can substantially improve the results of the pivot pipeline, the zero-shot end-to-end GPT-based simplification performs better on all metrics. Our work shows how an existing pivot-based pipeline can be improved for simplifying Dutch medical text. Moreover, we provide baselines for the comparison in the domain of Dutch municipal text and make our corresponding evaluation dataset publicly available.

Keywords: Pivot-based text simplification, Dutch municipal text, GPT, Large language models

1. Introduction

Natural language can be difficult to comprehend, especially when the sentences contain terminology or use complex construction. However, such text can be of great importance to all individuals. Some past research showed text simplification (TS) could benefit children (Watanabe et al., 2009) and people with comprehensive problems or language disorders, such as autism (Evans et al., 2014) and dyslexia (Rello et al., 2013). Moreover, simplified texts can be of great use for non-native learners and people with low literacy levels (Shojaeizadeh et al., 2017; Al-Thanyyan and Azmi, 2021). TS systems are developed to reduce the complexity of the text and improve its readability and understandability (Al-Thanyyan and Azmi, 2021). This paper focuses on the simplification of municipal text in Dutch. These are sentences about voting, city policies, taxes, etc. Due to the lack of datasets and research in this domain, there is no benchmark as far as the authors are aware. This research aims to establish the first benchmark for the simplification of text in the domain of Dutch municipal communication.

Existing TS methods are mostly in two categories: rule-based and data-driven approaches. Rule-based methods, which heavily rely on domain expertise or annotated language resources, employ robust rules for simplification, such as splitting compound sentences or converting passive to active voice. Data-driven solutions, including statistical and neural machine translation approaches, allow for one-step simplification of sentences or longer texts. These systems treat complex language as a source language and aim to translate an input sentence to the target, simplified language. Generally, neural solutions are data-intensive and require a monolingual simplification corpus for training. High-resource languages such as English have an abundance of corpora available for translation and simplification. Unfortunately, this is not the case for some low-resource languages where there are no TS corpora, let alone domain-specific ones. Thus, a pivot-based approach can be used as an alternative. The aim is to take advantage of high-resource languages such as English by using them as a pivot in an intermediate step (Evers, 2021). The pipeline's results rely on the performance of each underlying model, for which we need different kinds of training data. Most recently, the use of GPT for TS provides a promising zero-shot approach. For example, it was employed for the simplification of radiology reports with good results (Jeblick et al., 2022). Motivated by these recent advances, we study the following research questions: **RQ1:** Can we improve the results of the pivot approach by augmenting training data or using alternative translation corpora? **RQ2:** How does the pivot pipeline perform when transferred to the domain of Dutch municipal communication? **RQ3:** Can LLMs (such as GPT) outperform the pivot-based approach?

To answer these research questions, we adapt an existing pivot-based approach (Evers, 2021) to the Dutch municipal communication domain and attempt to improve the performance of the models in this approach. We compare its results against the corresponding achievements by the state-of-the-art language model GPT 3.5 Turbo (OpenAI, 2021). As for evaluation, we use Dutch medical text and sentences from municipal letters.

Our main contributions are the following: (1) an adaptation of an existing Dutch medical simplification pipeline for the municipal domain, (2) release of the corresponding Dutch municipal evaluation data, (3) means to improve the performance of the TS approach, and (4) benchmarking and comparing the performance of a pivot-based approach versus an LLM-based approach for TS.[1]

[1] The code, parametric setting, and related materials

2. Related Work

Best-performing TS systems conduct both lexical and syntactic simplifications. Early research on TS systems introduced and compared the usefulness of rule-based systems to data-driven approaches such as statistical and neural machine translation (Bahdanau et al., 2014). To train a system with the latter approach, a monolingual parallel corpus is typically required.

The introduction of the transformer architecture (Vaswani et al., 2017) has lead to reduced training times and computational costs, among other benefits such as the ability to handle longer text sequences, and is used by many state-of-the-art natural language systems nowadays. The transformer architecture also paved the way for large language models. Models such as those from the GPT family are trained on vast amounts of data and can generate text in various languages. Most recently, the chatbot model ChatGPT was employed for the simplification of radiology reports by (Jeblick et al., 2022). This research shows that most radiologists find the simplified reports factually correct and complete without potential harm to the patient.

While the transformer architecture in NLP has improved language models, the problem of scarce data remains. Due to the lack of training data for low-resource languages, many solutions have been proposed to increase the accuracy of translation systems. One example is back-translation which generates additional training data (Sennrich et al., 2015). This approach utilizes a pre-existing machine translation model. A second model is trained in the opposite direction, going from the target language to the source language, and the sentences generated by it are then used to train the original system further. This iterative training approach helps improve the overall performance and effectiveness of the machine translation system. Unfortunately, this method does not apply to our use case as we do not have any Dutch simplification corpora available in the first place with which to back-translate more training data.

One of the alternative architectures for TS was proposed by Evers (2021). The approach, presented in detail in Section 3, uses a pivot method to exploit the resources available in languages with richer data sources (English in this case). The authors also experiment with a zero-shot translation approach for the text simplification task, which requires no explicit parallel corpus between the source and target language, but is instead trained on multiple languages using a shared representation. Although their reported best approach is the zero-shot approach, the pivot approach provides higher-level explainability and control, which are crucial within the municipal domain.

3. Methodology

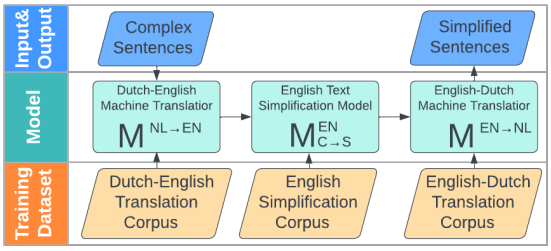

Figure 1: Pivot pipeline for text simplification

As mentioned, our research builds upon the pipeline introduced by Evers (2021). First, we follow the pre-processing and tokenization steps as in the original paper (see Section 4.2 for details). We replicate the pivot-based approach, improve the results, and adapt it to the domain of Dutch municipal communication. The pivot-based pipeline consists of three models and is illustrated in Figure 1. The input data consists of complex Dutch sentences, which are translated to English by the first model, referred to as $M^{NL \rightarrow EN}$. The second model, $M^{EN}_{C \rightarrow S}$, turns these complex English sentences into simple English sentences. Finally, the simplified English sentences are translated back to Dutch by the third model, referred to as $M^{EN \rightarrow NL}$. The final output is evaluated using the SARI (Xu et al., 2016), BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) metrics, which are adopted from the machine translation field, and were also used by Evers (2021).

The success of this approach relies on the choice of corpora and is explained below. The models in the original medical pipeline of Evers (2021) are trained on the specialized medical EMEA corpus (Tiedemann, 2012) for Dutch to English translation, the WikiSimple dataset (Coster and Kauchak, 2011) for English TS, and a medical subset of the OpenSubtitles (Lison and Tiedemann, 2016) for English to Dutch translation.

3.1. Simple Corpus Choice

The work of Evers is based on the assumption that the OpenSubtitles corpus is more suitable for training $M^{EN \rightarrow NL}$ to translate simple English sentences back to Dutch. The assumption is based on the conversational nature of the sentences in the corpus, which should generally be written in a more accessible language. To deter-

are included at https://anonymous.4open.science/r/Text_Simplification-3504.

We make the municipal evaluation dataset publicly available at https://anonymous.4open.science/r/dutch-municipal-text-simplification-F819/complex-simple-sentences/README.md

mine whether this assumption is correct, we compare the pipeline to an almost identical one where the only difference is the training data used for $M^{EN \rightarrow NL}$. Instead of being trained on a medical subset of the OpenSubtitles corpus, as proposed by (Evers, 2021), we train it on the specialized EMEA corpus used for $M^{NL \rightarrow EN}$, only in the opposite direction (English to Dutch).

3.2. Augmenting Train Data

After replicating Evers' pipeline, we noticed poor performance in the initial translation from complex Dutch to complex English. The model trained on the EMEA dataset performed well when translating complex medical terminology but failed to capture more general language that was less common in the training data.

Therefore, to try and improve these results, we experiment with augmenting the domain-specific training data of $M^{NL \rightarrow EN}$ with data from the more general OpenSubtitles corpus. As in-domain data selection has been long shown to improve machine translation systems (Axelrod et al., 2011), we follow Evers' methodology and use a medical subset of OpenSubtitles. This subset is created by extracting OpenSubtitles sentences similar to the specialized domain, and more specifically, to the WikiMed dataset (Van et al., 2020). We experiment with two different approaches to this. The first one, later denoted as MedSubset (TF-IDF), is a TF-IDF-based method which would preserve more sentences containing exact words from the reference corpus. The second one, denoted as MedSubset (BERT), is a BERT-based method which is expected to retain more diverse content that is semantically similar to the reference sentences rather than exact matches. Further details about the data selection are provided in Section 4.1.

3.3. Transferring Pipeline to Municipal Domain

We also conduct experiments to evaluate the transferability of the model to a new domain: municipal text. To adapt the pipeline to the municipal domain, we must select a translation corpus containing domain-specific language and topics. We use the Europarl corpus (Koehn, 2005), as it contains manually translated sentences from European parliamentary proceedings. The simplification model $M^{EN}_{C \rightarrow S}$ is still trained using WikiSimple data. As a baseline for $M^{EN \rightarrow NL}$, we use a model trained on a random sample of 1 million OpenSubtitles sentences. We also perform corpus selection and data augmentation experiments for our municipal pipeline to improve results. The former entails testing whether or not it is more beneficial to use a domain-specific corpus (Europarl) or a more general corpus (OpenSubtitles) to train $M^{EN \rightarrow NL}$ to translate simplified sentences back to Dutch. The latter uses TF-IDF and BERT-based MunSubsets to extend the training data for both the $M^{NL \rightarrow EN}$ and $M^{EN \rightarrow NL}$ directions.

To evaluate the pipelines, a test set is created by extracting simplifications made to municipal letters. These letters are provided by the City of Amsterdam, and the simplifications are performed manually by professional editors. After automatically aligning sentences from the original and simplified letters based on TF-IDF similarity, the final test set consists of 1310 sentences with lexical and syntactic simplifications.

3.4. Zero-shot Simplification

We can use OpenAI's API to leverage the power of GPT 3.5 Turbo for our TS task. While this model was not created with the intention of simplification, LLMs have been shown to be remarkably suited for many NLP tasks (Bubeck et al., 2023) including TS (Feng et al., 2023). We use this model to simplify our evaluation data and compare these results to those of the pivot pipelines. Another model we consider is the zero-shot model described in the work of Evers (2021). This baseline allows us to compare the results of end-to-end TS systems trained without a monolingual Dutch TS corpus.

3.5. Evaluation Metrics

The final output is evaluated by three well-used metrics in the TS community: BLEU, SARI, and METEOR. BLEU compares the predicted simplified sentence against reference outputs and computes the similarity based on n-grams. Another popular metric is SARI, which produces a score based on the weighted average of three components: sentence-level unigram precision, sentence-level unigram recall, and n-gram overlap (Xu et al., 2016). The METEOR metric computes the harmonic mean of precision and recall using exact and stemmed matches between words in the machine-translated output and reference sentences (Banerjee and Lavie, 2005). The metrics are calculated using the following formulas. For BLEU we compute $\text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \cdot \log(p_n)\right)$. Here, BP stands for the brevity penalty, while N is the maximum n-gram order. We use w_n for the weights assigned to n-grams and p_n for the precision of n-grams. As for METEOR, we use $(1-\alpha) \cdot P + \alpha \cdot R \cdot F$. Here we have α as the parameter balancing precision and recall in METEOR. P, R and F are for precision, recall, and their harmonic mean. SARI is computed using $\gamma \cdot A + (1-\gamma) \cdot K$. We use γ as the parameter balancing addition and preservation, A as the number of words in the candidate translation not present in the reference sentences, K as the number of

words in the reference sentences not present in the candidate translation. More discussions about the effectiveness and limit of these metrics are included in Section 6.

Domain	Dataset	Purpose	#Pairs
Encyclopedia	WikiSimple (Coster and Kauchak, 2011)	TS	283K
Subtitles	OpenSubtitles (Lison and Tiedemann, 2016)	MT	1.01M
Medical	EMEA (Tiedemann, 2012)	MT	308K
	WikiMed (Van et al., 2020)	Ref	3.39K
	MedSubset (TF-IDF) (Lison and Tiedemann, 2016)	MT	836K
	MedSubset (BERT) (Lison and Tiedemann, 2016)	MT	379K
	Medical Eval Set (Evers, 2021)	Eval	101
Parliamentary	EuroParl (Koehn, 2005)	MT	1.95M
	Dutch Government Website (European Language Resource Coordination, 2015)	Ref	6.53K
Municipal	MunSubset (TF-IDF) (Lison and Tiedemann, 2016)	MT	559K
	MunSubset (BERT) (Lison and Tiedemann, 2016)	MT	531K
	Municipal Eval Set	Eval	1.31K

Table 1: Datasets used, their purpose and domain, and their number of aligned sentences (Ref for reference for in-domain data selection and Eval for evaluation)

4. Experimental Design

In this section, we describe the datasets used in our work, the extraction of in-domain subsets, the pre- and post-processing of data, as well as some implementation details.

4.1. Data Gathering and Selection

The datasets used in our paper are summarized in Table 1. The OpenSubtitles corpus is a large general corpus containing sentences from various domains. It is further processed to extract in-domain subsets. We extract subsets from it by mapping sentences from the OpenSubtitles corpus to a vector space and extracting the n nearest neighbours of each sentence in an in-domain reference corpus. The dataset used to extract in-domain sentences for the medical subset was WikiMed: a medical corpus previously extracted from the WikiSimple corpus. We use the Dutch Government Website Corpus as a reference to extract municipal sentences. It contains texts published by the Dutch government. We extract a fixed number n of nearest neighbour OpenSubtitles sentences per reference sentence, aiming to obtain one million sentences. Our medical reference corpus contained 3,390 sentences, so n was set to 294, whereas the municipal reference corpus had 6,532 sentences, so n was set to 153.
We experiment with encoding the sentences using both a TF-IDF and BERT-based approach. For the BERT-based approach, we encode sentences using the 'paraphrase-distilroberta-base-v1' model[2]. This model maps sentences to a 768-dimensional dense vector space. In this way, we obtain domain-specific corpora: MedSubset (TF-IDF), MedSubset (BERT), MunSubset (TF-IDF), and MunSubset (BERT) as shown in Table 1.

4.2. Data Preprocessing

We follow preprocessing and tokenization steps similar to those proposed by Evers (2021) The preprocessing script we used, taken from Yasmin Moslem's implementation[3], begins by filtering the data. It removes empty rows, duplicates, and potentially erroneous translations based on sentence length mismatch, as well as HTML elements within the sentences. For tokenization, we employed a fast implementation of byte-pair encoding, YouTokenToMe.[4] Subword models are generated for both the source and target sides of the data and are used to encode and later decode the sentences. The data is then shuffled at the sentence pair level and split into train-test-dev splits to be fed to the models. The size of the test and dev splits are 2,000 sentences for all models. The remaining sentences are used for training.

4.3. Implementation

The models used in the pipelines are transformers implemented using OpenNMT, an open-source ecosystem for neural machine translation (Klein et al., 2017). We train the models on a single GPU using Google Colab.[5]
After training the models, we perform inference on the tokenized test data, and the output of each model is subsequently de-tokenized and passed to the next model in the pipeline.
Additionally, we obtain simplifications for the same evaluation data using the GPT 3.5 Turbo model. We access the model through the OpenAI API[6] and make requests with the prompt: "`Can you simplify the following sentence in Dutch: {sentence}`". Even though we implement a delay in order to respect the API's rate limit, not all requests can be completed. When requests could not be handled, we kept the original complex sentence as the system's output. Some manual postprocessing was performed to fix formatting issues caused by the model's conversational nature.

[2]Model taken from the sentence-transformers library: https://huggingface.co/sentence-transformers/paraphrase-distilroberta-base-v1
[3]https://github.com/ymoslem/MT-Preparation
[4]https://github.com/VKCOM/YouTokenToMe
[5]See the code repository for (hyper-)parameters and other details in the configuration of the models.
[6]https://platform.openai.com/

5. Evaluation

5.1. Medical Pipeline Results

In the medical domain, we compare our results to those by Evers, including their pivot approach and their end-to-end zero-shot approach. We evaluate our results qualitatively and quantitatively.

5.1.1. Automated Evaluation

Our implementation of the baseline pipeline $Pipe^{EMEA}_{MedSubset(BERT)}$ achieved similar SARI and BLEU scores as the results reported by Evers. The relatively small difference could be a product of our aggregation of the scores over multiple runs. We encountered a significant increase in the METEOR score of this pipeline in our implementation. Despite having the largest standard deviation of our experiments, the difference in scores is still too large to be explained by this.

The results of the medical pipelines indicate that both choosing a suitable corpus for translating the simplified sentences back to Dutch, as well as augmenting the domain-specific data increased scores on all metrics.

Our first experiment aimed to test the assumption that the OpenSubtitles corpus provides more suitable training data for translating simple sentences. While it can be true that, because of its narrative tone, the OpenSubtitles sentences are simpler, the results of our experiment indicate that simply using the EMEA dataset on both $M^{NL \rightarrow EN}$ and $M^{EN \rightarrow NL}$ produced better results when compared to any of the pipelines which train $M^{EN \rightarrow NL}$ on an OpenSubtitles-derived dataset alone. We also experimented using the OpenSubtitles dataset alone to train this model in $Pipe^{EMEA}_{OpenSubtitles*}$. However, we found worse results on all metrics. A notable insight from this experiment was the magnitude of the improvements brought by simply using a larger amount of training data, indicating the usefulness of increasing computational resources for the practical application of the pipelines.

Our second experiment, where we augmented the training data of the initial model with medical subsets not only increased word coverage but fluency in general. Additionally, the results show that the BERT-based sampling outperformed the TF-IDF-based sampling on all metrics, most likely due to introducing more diverse phrases and examples.

Combining the approaches of both experiments in the $Pipe^{EMEA+MedSubset(BERT)}_{EMEA+MedSubset(BERT)}$ resulted in the best scores for any combination of $Pipe^{X}_{Y}$. The pipeline trained fully on a combination of the MedSubset (BERT) and EMEA datasets achieved SARI, BLEU, and METEOR scores of 34.14, 15.41, and 40.89, respectively.

5.1.2. Qualitative Analysis

The provided scores do not give insight into the performance of the individual models, only the final results produced by each pipeline. While the scores indicate fluency, grammaticality and similarity to reference sentences, they have many limitations, and thus, we cannot rely on automated evaluation metrics alone to rank our results. Therefore, we analyze some examples of errors and improvements made by the medical pipelines, including the quality of intermediate translations.

One of the main reasons we augmented the in-domain train data is that we noticed errors in the first translation step that were propagated throughout the pipeline. These errors were more often related to general text rather than domain-specific terminology. For example, in the sentence *"Na de Zwarte Dood en de agrarische depressie aan het eind van de 15e eeuw, begon de bevolkingsgroei toe te nemen."* (in English: *After the Black Death and the agricultural depression at the end of the 15th century, population growth began to increase.*), "de 15e eeuw" is translated to "15nd egggg" in the pipeline $Pipe^{EMEA}_{EMEA}$. This error is carried over to the next step and becomes "15nd egg egggg" after simplification, which is eventually translated to "15e eieren" (in English: *15th eggs*). This problem is resolved by providing additional data, as is done e.g. in $Pipe^{EMEA+MedSubset(TF-IDF)}_{EMEA+Medsubset(TF-IDF)}$ where "century" is correctly translated throughout all models with coherent final result.

5.2. Municipal Pipeline Results

Next, we provide benchmark results in the municipal domain and compare the pipeline results with different corpus choices and data augmentation strategies.

5.2.1. Automated Evaluation

The results of the municipal pipelines, displayed in Table 2, showed similar trends to those of the medical pipelines. We can see that scores increase when using the domain-specific MunSubsets for $M^{EN \rightarrow NL}$ rather than a general corpus such as OpenSubtitles. Once again, the BERT-based selection outperformed TF-IDF-based one, suggesting that extracting sentences based on semantic meaning rather than exact lexical matches provides more suitable training data. Surprisingly, the Europarl corpus by itself did not outperform either of the MunSubset corpora in the way that the EMEA corpus outperformed the MedSubsets. One explanation for this could be the similarity of the reference corpora to our domain. The reference sentences used for MunSubsets came from an ELRC dataset, which comprises just over 6500 sentences from texts published by the Dutch gov-

	System	SARI		BLEU		METEOR	
		Mean	SD	Mean	SD	Mean	SD
Medical Pipeline	$Pipe^{EMEA}_{MedSubset(BERT)}$ (Evers, 2021)	33.32		5.28		10.63	
	$Pipe^{EMEA}_{OpenSubtitles*}$	29.55 (31.77)	0.36 (0.72)	5.18 (8.59)	1.00 (1.08)	22.71 (30.31)	1.88 (2.53)
	$Pipe^{EMEA}_{MedSubset(TF-IDF)*}$	30.15 (30.44)	0.52 (0.49)	5.66 (7.13)	0.89 (0.53)	25.45 (25.05)	0.74 (0.60)
	$Pipe^{EMEA}_{MedSubset(BERT)*}$	30.59 (30.83)	0.50 (1.27)	6.82 (6.87)	0.51 (0.57)	27.79 (28.8)	0.55 (2.82)
	$Pipe^{EMEA}_{EMEA}$	32.52	0.79	11.44	1.2	35.27	0.9
	$Pipe^{EMEA}_{EMEA+MedSubset(BERT)}$	32.69	0.16	11.45	0.43	35.49	0.68
	$Pipe^{EMEA+MedSubset(TF-IDF)}_{EMEA+Medsubset(TF-IDF)}$	33.91	0.98	15.15	1.1	40.45	1.57
	$Pipe^{EMEA+MedSubset(BERT)}_{EMEA+MedSubset(BERT)}$	34.14	0.20	15.41	0.83	40.89	0.21
	Zero-Shot Baseline (Evers, 2021)	40.04		**27.19**		31.19	
	GPT 3.5 Turbo	**40.26**	2.73	21.23	2.67	**47.49**	3.56
Municipal Pipeline	$Pipe^{Europarl}_{OpenSubtitles*}$	24.64 (24.13)	0.52 (2.11)	7.72 (6.82)	0.86 (3.12)	29.34 (29.2)	1.74 (1.77)
	$Pipe^{Europarl}_{MunSubset(TF-IDF)*}$	25.66 (25.93)	0.3 (0.15)	9.09 (9.53)	0.57 (0.26)	31.50 (32.38)	1.09 (0.50)
	$Pipe^{Europarl}_{MunSubset(BERT)*}$	27.70 (27.69)	0.06 (0.26)	12.79 (13.1)	0.33 (0.32)	38.26 (37.93)	0.09 (0.31)
	$Pipe^{Europarl}_{Europarl}$	23.57	0.03	6.13	0.28	25.54	0.19
	$Pipe^{Europarl}_{Europarl+MunSubset(BERT)}$	28.7	1.39	14.83	3.00	40.44	3.65
	$Pipe^{Europarl+MunSubset(TF-IDF)}_{Europarl+MunSubset(TF-IDF)}$	29.87	0.24	16.36	0.40	43.26	0.14
	$Pipe^{Europarl+MunSubset(BERT)}_{Europarl+MunSubset(BERT)}$	29.83	0.17	17.12	0.16	43.32	0.27
	GPT 3.5 Turbo	**34.00**	0.59	**22.60**	0.78	**48.63**	0.70

Table 2: Automated evaluation of Medical and Municipal Pipelines.
Pipelines are denoted in the form $Pipe^{X}_{Y}$ where X and Y represent the training data for models $M^{NL \rightarrow EN}$ and $M^{EN \rightarrow NL}$, respectively. For a fair comparison, for rows with * label, we used datasets reduced to 320K sentences for medical pipelines. Similarly, that of municipal pipelines has been reduced to 500K sentences. In the parenthesis are the results when taking the entire datasets in training. The best results are highlighted in bold font. The best results of our approach and the lowest standard deviation scores are highlighted with underlines.

ernment. This domain aligns well with the domain of the test set, which contains sentences from documents published by the City of Amsterdam. In contrast, the medical reference contains fewer sentences (approximately 3,300), which are automatically extracted from the WikiSimple corpus. The combination of fewer examples and those not being as targeted towards the test domain may have contributed to them increasing the evaluation scores by less than the MunSubsets did. For this reason, using only the Europarl corpus, which contains more general governmental data rather than municipal, for the $M^{EN \rightarrow NL}$ may not have outperformed using the MunSubsets.

Similarly to the medical domain, augmenting the Europarl corpus with in-domain subsets of the general corpus brings improvements in all scores. Once again, the pipeline combining both approaches – that is, incorporating the *Europarl+MunSubset(BERT)* training data for both translation models – achieved the highest BLEU and METEOR scores of any combination of $Pipe^{X}_{Y}$. Additionally, it achieves the second-highest SARI score, falling just 0.04 short of the TF-IDF version of the same pipeline.

5.2.2. Qualitative Analysis

One improvement made by incorporating the MunSubset datasets was the coverage of domain-specific terminology. For the sentence *"Stuur je aanvraagformulier dan naar"* (in English: *Send your application form to*), we can see how bolstering the initial translation model with the MunSubset (TF-IDF) data increases performance. When the initial model, $M^{NL \rightarrow EN}$, is trained on the Europarl dataset only, the term "aanvraagformulier" is erroneously translated as "applicant form" instead of "application form". This small error results in the final output producing the word "solicitatieformulier". In the pipeline $Pipe^{Europarl+MunSubset(TF-IDF)}_{Europarl+MunSubset(TF-IDF)}$, where the initial training data is supplemented with the MunSubset data, this initial mistake does not occur. As a result, the term is handled correctly throughout all models in the pipeline. On top of increasing coverage of vocabulary, the general accuracy of simplifications is improved in other ways. For example, the first pipeline also incorrectly introduces a negation into the sentence, which heavily affects the preservation of meaning.

5.3. Numerical Information Preservation

Throughout our results, we noticed errors that arose when translating and dealing with numbers, particularly in cases where numbers had some formatting, for instance, "*3-5%*" This was a recurring concern in our pipelines, and it is a common issue with neural machine translation systems, which has been documented in, for example, the work of Wang et al. (2021). Their work found that state-of-the-art and major commercial systems in both high- and low-resource languages struggle with this. It is important to note that factual information should be preserved, especially given the domains we are dealing with. Mistaken numbers can be crucial for both medical and municipal texts.

As a primitive examination, we manually check numerical data. For the small medical test set, consisting of only 101 sentences, we used all 23 sentences that contain digits. For the larger municipal data (1310 sentences), we randomly selected 50 sentences containing digits and manually compared the source and output. We marked cases as correct where numbers were correctly conveyed (either as digits or text). For numeric correctness in the medical data, GPT achieved an accuracy of 0.88, while the best-performaning pipeline $Pipe^{EMEA+MedSubset(BERT)}_{EMEA+MedSubset(BERT)}$ scored 0.72. For municipal data, GPT achieved an outstanding 0.96 while $Pipe^{Europarl+MunSubset(BERT)}_{Europarl+MunSubset(BERT)}$ gets an accuracy of 0.87. The results indicate that GPT is better at preserving the numerical information present in the source sentences.

5.4. Simplicity Evaluation

The BLEU, SARI, and METEOR metrics measure the similarity between the system's outputs and the reference sentences, but they do not directly evaluate the simplicity of the generated sentences. Therefore, we compare the outputs of our best-performing pivot pipeline and end-to-end models based on their Flesch Reading Ease (FRE) score (Flesch, 1948), which considers both the number of words per sentence and the number of syllables per word. According to the definition, more readable texts score higher on this metric.

On the medical data GPT achieves a score of 57.76 in comparison to 51.84 by the best-performing pipeline $Pipe^{EMEA+MedSubset(BERT)}_{EMEA+MedSubset(BERT)}$. For municipal data, GPT has a remarkable score of 65.20, which is higher than the 63.27 by the best-performing pipeline $Pipe^{Europarl+MunSubset(BERT)}_{Europarl+MunSubset(BERT)}$. On average, GPT's simplifications contain fewer words per sentence and fewer syllables per word than the simplifications made by our pivot pipeline. What is interesting is the degree to which they differ. In the medical domain, the difference between results achieved by GPT and our pivot model is substantially greater than the differences we found in the municipal domain. The notably close reading ease scores in the municipal domain lead us to believe that training the models of the pivot pipeline on a larger amount of data has the potential of outperforming GPT on this aspect.

6. Discussion

In this section, we discuss some of the limitations of the pivot-based pipelines and reflect on the evaluation results regarding our research questions.

6.1. Pivot-based Pipeline

The experiments in this paper can serve as a validation of some of the assumptions made in the work of (Evers, 2021) regarding corpus choices. The results provide insight into RQ1, demonstrating that both augmenting the in-domain training data of the initial model, as well as selecting a more suitable dataset for the final model can improve the simplification results. Overall, the pipeline is highly susceptible to bad results when mistakes are made in the first model ($M^{NL\rightarrow EN}$). Mistakes propagate through the pipeline, resulting in worse and worse translations. These mistakes were amplified by the lack of suitable, domain-specific training data in the final model, translating simple English sentences back to Dutch.

We found that bolstering the in-domain corpora with extracted subsets improved scores. Additionally, despite not having a large initial in-domain corpus, extraction of semantically and lexically similar sentences from a general corpus can substantially improve results. Introducing in-domain subsets always outperformed using a random OpenSubtitles dataset. Our experiments indicate that combining domain-specific data with an in-domain subset of a general corpus increased word coverage, reducing the propagation of certain errors throughout the pipeline.

Regarding RQ2, our results indicate the adaptability of the pipeline to other complex domains. We observed overall similar evaluation results in the metrics after the adaption of the pipeline to the Dutch municipal domain: more concretely, a slight increase in BLEU and METEOR with a decrease in SARI. There are many potential reasons for this. For example, the test set used for the medical domain is significantly smaller in size but with higher quality (101 v.s. 1,310 sentences). Being automatically extracted, sentences in the municipal test set might be incomplete or contain spelling errors, which can add difficulty to simplification but also impact the evaluation scores.

Altogether, the changes in the scores of the pipeline in both domains we observed were similar with regard to the modification of the training data.

This indicates the generalizability of the pipeline and its improvements to different domains.

6.2. Large Language Models

To address RQ3, we performed experiments with the GPT 3.5 Turbo model. We found that its use has many benefits. First of all, it does not require new domain-specific data. Moreover, there is no need for a pivot language. This also reduces the risk of propagating errors through multiple models. Furthermore, there is no additional training needed, which can be time- and resource-saving. GPT has great potential for use in other low-resource languages and tasks, given its universal purpose. Moreover, we noticed that the generated sentences are fluent, which is not always the case for other models.

However, there are multiple downsides to this approach. To the users, GPT is a black box, and the developers would have less control over the system. Also, GPT does not always give exactly identical answers for the same input (unless specified otherwise), which can introduce unwanted variability. Our manual examination found that GPT lacks a fundamental understanding and knowledge of some domain-specific concepts and abbreviations.

6.3. Evaluation Metrics

Despite the wide use of machine translation metrics for text simplification, their suitability for the task has been extensively questioned and criticized, e.g. by Alva-Manchego et al. (2021). The metrics are designed to judge the quality of machine-translated text by comparing the predicted output to one or more reference sentences. This makes our results highly dependent on the available evaluation data. Moreover, higher scores do not always mean that the results are simpler. For example, suitable simplifications that drastically change the vocabulary or structure of the sentence may be punished.

BLEU was sometimes found to have a negative correlation with simplification, especially when scoring sentences whose structure has changed (Sulem et al., 2018). SARI scores are calculated based on the goodness of the three core translation operations; sentence-level addition, deletion, and keep operations. It compares the predicted sentence with the original complex and simplified reference sentences. SARI captures the fluency and paraphrasing of the translation at the sentence level. For instance, if the candidate translation adds or deletes words in a way that aligns with the reference while preserving the original meaning, it would receive a higher SARI score. However, even with flawless simplification, the predicted sentence may not achieve a perfect score if it differs from the reference.

METEOR evaluates the similarity between the candidate translation and the reference translations by aligning the words and considering the exact and stem matches between the predicted and reference translations. In the English implementation of METEOR, unigrams can also be matched according to meaning. This means that the choice of words used for simplification can still be rewarded. Unfortunately, this does not exist in the Dutch implementation of METEOR. Therefore, replacing complex words with synonyms will be punished, regardless of whether the synonym is simpler or not.

7. Conclusion and Future Work

In this paper, we demonstrated how appropriate corpus choices and data augmentation can improve the performance of Evers's pivot-based TS system (RQ1). Regarding RQ2, we adapted the pipeline from the medical domain to the Dutch municipal communication domain. We provided the first TS benchmark for Dutch municipal communication and published our evaluation data. For RQ3, our evaluation showed that GPT can achieve better scores than pivot-based pipelines for Dutch municipal text. We discussed the drawbacks of both approaches and the limits of the metrics used. In the future, some new metrics could be designed to overcome the drawbacks mentioned in Section 6. We could also explore other measures that reflect simplicity. For the municipal domain, it is crucial that the simplified information remains factual so the metrics can cover the preservation of numbers and other factual information (street name, date, opening time).

The main bottleneck of Dutch municipal text simplification is the lack of a corresponding training dataset. The availability of larger Dutch simplification datasets would allow the development of end-to-end models.

Our work shows great potential for using LLMs for TS. Further research with GPT 3.5 Turbo could investigate the effect of fine-tuning the model using the above-mentioned dataset. With its help, a dataset could be constructed semi-automatically.

Finally, both approaches can be used in municipal and medical cases in real life to provide a draft translation, which can ease the work of the editors. If widely used, larger datasets of human-verified simplified sentences are expected to be constructed, which can be used for future research.

8. Bibliographical References

Suha S. Al-Thanyyan and Aqil M. Azmi. 2021. Automated text simplification: A survey. *ACM Comput. Surv.*, 54(2).

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. The (un) suitability of automatic evaluation metrics for text simplification. *Computational Linguistics*, 47(4):861–889.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 355–362.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Richard Evans, Constantin Orasan, and Iustin Dornescu. 2014. An evaluation of syntactic simplification rules for people with autism. Association for Computational Linguistics.

Marloes Evers. 2021. Low-resource neural machine translation for simplification of dutch medical text. Master's thesis, Tilburg University. Available at `http://arno.uvt.nl/show.cgi?fid=158729`.

Yutao Feng, Jipeng Qiang, Yun Li, Yunhao Yuan, and Yi Zhu. 2023. Sentence simplification via large language models. *arXiv preprint arXiv:2302.11957*.

Rudolf Flesch. 1948. *The Art of Readable Writing*. Harper, New York.

Katharina Jeblick, Balthasar Schachtner, Jakob Dexl, Andreas Mittermeier, Anna Theresa Stüber, Johanna Topalis, Tobias Weber, Philipp Wesp, Bastian Sabel, Jens Ricke, and Michael Ingrisch. 2022. Chatgpt makes medicine easy to swallow: An exploratory case study on simplified radiology reports.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

OpenAI. 2021. ChatGPT 3.5.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Luz Rello, Ricardo Baeza-Yates, Stefan Bott, and Horacio Saggion. 2013. Simplify or help? text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, pages 1–10.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Mina Shojaeizadeh, Soussan Djamasbi, Ping Chen, and John Rochford. 2017. Text simplification and pupillometry: an exploratory study. In *Augmented Cognition. Enhancing Cognition and Behavior in Complex Human Environments: 11th International Conference, AC 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings, Part II 11*, pages 65–77. Springer.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Bleu is not suitable for the evaluation of text simplification. *arXiv preprint arXiv:1810.05995*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jun Wang, Chang Xu, Francisco Guzmán, Ahmed El-Kishky, Benjamin Rubinstein, and Trevor Cohn. 2021. As easy as 1, 2, 3: Behavioural testing of NMT systems for numerical translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4711–4717, Online. Association for Computational Linguistics.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro

Pardo, and Sandra Maria Aluísio. 2009. Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM international conference on Design of communication*, pages 29–36.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

9. Language Resource References

William Coster and David Kauchak. 2011. Simple english wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669.

European Language Resource Coordination. 2015. *"Dutch Government Website Corpus". ELRC Data - Tools and Resources for CEF Automated Translation-LOT3*. [link].

Koehn, Philipp. 2005. *Europarl: A parallel corpus for statistical machine translation*.

Lison, Pierre and Tiedemann, Jörg. 2016. *Opendubtitles2016: Extracting large parallel corpora from movie and tv subtitles*. European Language Resources Association.

Jörg Tiedemann. 2012. *Parallel data, tools and interfaces in OPUS.* European Language Resources Association (ELRA).

Van, Hoang and Kauchak, David and Leroy, Gondy. 2020. *AutoMeTS: the autocomplete for medical text simplification*.