Unlocking Domain Knowledge: Model Adaptation for Non-Normative Dutch

Florian Debaene, Aaron Maladry, Pranaydeep Singh, Els Lefever and Véronique Hoste

LT^3, Language and Translation Technology Team, Ghent University
Groot-Brittanniëlaan 45, 9000 Ghent, Belgium

firstname.lastname@ugent.be

Abstract

This study examines the adaptation of transformer models to two non-normative Dutch language variants: early modern Dutch and contemporary social media Dutch. Both share linguistic features that set them apart from standard Dutch, including spelling inconsistencies, semantic shifts and out-of-domain vocabulary. To address this, we explore two domain adaptation techniques to adapt models to these language variants: (1) continued full-model pre-training and (2) training specialized adapters integrated into existing models. We evaluate these adaptation techniques on sentiment and emotion detection in early modern Dutch comedies and farces and on emotion and irony detection in Dutch tweets. Our results show that both adaptation methods significantly improve performance on historical and social media Dutch tasks, with the greatest gains occurring when domain-relevant datasets are used. The effectiveness of model adaptation is task-dependent and sensitive to the selection of pre-training data, emphasizing domain relevance over data quantity for optimizing downstream performance. We hypothesize that contemporary Dutch encoder models already capture informal language but lack historical Dutch exposure, making adaptation more impactful for the latter. Additionally, we compare adapted encoder models to generative decoder models, which are state-of-the-art in many NLP tasks. While generative models fail to match the performance of our adapted models for historical Dutch, fine-tuned generative models outperform adapted models on social media Dutch tasks. This suggests that task-specific fine-tuning remains crucial for effective generative modelling. Finally, we release two pre-training corpora for Dutch encoder adaptation and two novel task-specific datasets for early modern Dutch on Hugging Face.

1. Introduction

Natural Language Processing (NLP) has seen remarkable progress in the last years, particularly due to the development of transformer-based encoding and decoding language models. These breakthroughs have predominantly centered on and benefited well-resourced languages, like English. This leaves low-resourced, under-represented and highly specialized languages or language forms at the periphery. Efforts to address this imbalance have resulted in the development of multilingual models to include less prioritized or dominant languages, like mBART (Tang et al. 2020) and mT5 (Xue et al. 2021), as well as domain-specific models such as BioBERT (Lee et al. 2019), LEGAL-BERT (Chalkidis et al. 2020) or FinBERT (Araci 2019), which improve NLP performance in English biomedical, legal and financial domains, respectively. These models exemplify how domain adaptation can effectively enhance model performance on specialized linguistic settings by leveraging knowledge from a related source domain.

While domain adaptation in NLP is generally associated with adapting to specialized fields such as biomedicine or law, its principles are equally relevant to adapt existing language models to linguistic variation across registers and language varieties, including social media language, historical

language, or dialects. Exhibiting distinct linguistic characteristics, vocabularies, and styles, these language variations can significantly impact model performance. Adapting to such settings involves additional training on corpora that are more representative of the target language variant. However, adaptation remains highly constrained by the usual bottlenecks in NLP research: the availability of machine-readable training data, both in quantity and quality. This challenge is even more significant for non-normative language variants, which often lack sufficient resources and are generally overshadowed by the focus on standard language in NLP. By leveraging the linguistic knowledge embedded in existing models, domain adaptation techniques enable the effective use of limited, specialized datasets for model fine-tuning. Rather than training models from scratch, this approach capitalizes on the general language understanding of pre-trained models and facilitates their adaptation to niche linguistic domains with minimal data requirements. In this way, adaptation serves as a crucial technique for bridging the gap between well-resourced and under-represented language varieties, ensuring broader applicability of NLP models across diverse linguistic landscapes.

For Dutch, a mid-resourced language, adaptation has pushed forward language-specific NLP engineering like for many other languages, with models like BERTje (de Vries et al. 2019) and RobBERT (Delobelle et al. 2020) trained on Dutch corpora outperforming English language models on many tasks. Similarly, recently released Dutch-specific generative models such as GEITje (Vanroy 2024b), Reynaerde (Van den Avenne 2024) and Fietje (Vanroy 2024a) attest the adaptation of generative architectures to Dutch. Despite these successes, Dutch NLP remains constrained by the limited size and scope of available training data. For instance, corpora like OSCAR (Ortiz Su'arez et al. 2019), which are widely used for Dutch model development, predominantly consist of web-crawled resources like news articles, books, and encyclopedic content with a strong emphasis on standard Dutch. While web-crawled data indubitably includes some social media (and possibly historical) content, the exact amount remains unclear, as these corpora lack a specific focus on scraping such data. This ambiguity complicates the assessment of social media data's contribution to model training and, consequently, their effectiveness in non-normative social media contexts. Furthermore, recent measures by social media platforms such as X and Facebook to restrict web scraping and mandate the use of (paid) APIs further limit access to substantial and representative social media datasets.

For historical Dutch, limited data accessibility and a lack of linguistic knowledge present even greater challenges for existing Dutch-specific models, as these factors complicate the processing and interpreting of historical texts. In particular, non-normative language characteristics of historical Dutch become manifest in its spelling variability, out-of-domain vocabulary, semantic shifts, and syntactic inconsistencies. Spelling variations are frequent, with a word like "ik" (EN: I) also appearing as "ick" or "ic". Out-of-domain vocabulary includes terms like "smakwambes", which are no longer present in contemporary Dutch. Semantic shifts add further complexity, as meanings can change drastically over time. For instance, "wijf" historically referred to a "woman" or "wife", but it has contracted a derogatory connotation in contemporary Dutch. Finally, historical Dutch also exhibits syntactic patterns that differ from those of standardized contemporary Dutch. For example, the verb "wachten" (EN: to wait) could take a direct object ("iemand wachten", EN: awaiting someone), whereas contemporary Dutch requires a preposition ("wachten op iemand", EN: waiting for someone). Social media language shares many of these characteristics. Spelling variability appears in the form of creative abbreviations, like "w8" instead of "wacht" (EN: wait). Social media also introduces out-of-domain vocabulary, such as slang, emoji, and borrowed terms, which do not occur in standard Dutch corpora. Moreover, neologisms are adopted significantly faster in social media language than in standardized language, meaning that the domain is more strongly affected by temporal shifts than the language norm. Semantic shifts occur rapidly in social media contexts, mirroring the temporal changes observed in historical Dutch. For instance, an ellipsis (...) has no sentimental or emotional connotation in standard language, but is often used to express frustration on social media by recent generations. In addition, social media texts frequently break grammatical conventions and consist of telegram-style messages, where subjects are omitted.

Both non-normative varieties of Dutch are – to different extents – affected by orthographic variability, distinct vocabularies, syntactic peculiarities and semantic change, all of which can complicate the application of existing Dutch NLP tools. Therefore, we explore the adaptation of Dutch transformers to non-normative historical and social media Dutch language varieties and aim to answer the following research questions:

- **RQ1**: Does expanding background knowledge by adapting Dutch pre-trained encoder models to non-normative language variants improve performance on downstream tasks?
- **RQ2**: What is the impact of pre-training data quantity and data relevance on the adaptation to non-normative Dutch? How much and which type of data do you need before continued model pre-training outperforms adapter-only training?
- **RQ3**: How do adapted encoder models match up against generative decoder models? Do generative models need further adaptation?

We investigate two methodologies for model adaptation: adapter-only training and continued full model pre-training. Since continued model training typically performs better on larger datasets due to the greater number of trainable parameters, we investigate how training resource characteristics, such as data quantity and relevance, influence the point at which adapter-only training is outperformed by continued full-model pre-training. We evaluate adaptation by measuring adapted model performance on downstream tasks for two different domains. Doing so, we can assess how domain and task-specific knowledge affect model performance. More concretely, the tasks we focus on consist of emotion and sentiment detection in early modern Dutch comedies and farces and emotion and irony detection in Dutch tweets. In our evaluation, we account for the two employed adaptation methods and the selection of pre-training corpora. Finally, we compare adapted encoder models to generative large language models (LLMs), testing both established multilingual models like Llama 3 (Grattafiori 2024) and language-specific models like GEITje (Vanroy 2024b). With this research, we contribute to the exploration of domain-specific NLP for Dutch, a language where the full potential of recent methodological advances remains under-explored. By investigating whether adapted encoder models can compete with generative LLMs for highly specific Dutch language variants, we aim to determine if adaptation for transformer-based models proves to be a worthwhile methodology to optimize the performance of existing NLP tools on downstream tasks in non-standard language. This research not only enhances our understanding of parametrization for adjacent but largely out-of-domain language variants, but also provides guidance for the development of future NLP tools for other similar language domains.

2. Related Research

In this Section, we describe related work on adaptation in language modelling. In Section 2.1, we elaborate on key concepts, methods and scenarios in which adaptation has previously been employed in NLP with regards to non-standard language. Then, in Section 2.2 we describe the development of generative AI and the implication for model adaptation. Finally, in Section 2.3 we discuss traditional approaches for adapting data through normalization techniques contrary to model refinement techniques like model adaption.

2.1 Adaptation of Transformer Models

Adaptation in NLP revolves around the transfer of a pre-trained model to two types of knowledge: domain knowledge and task knowledge. Increasing a model's domain knowledge refers to the improvement of its understanding of new language data of interest. This results in the model creating better representations of the linguistic features, style and structure typical for this particular type of

content. Domain adaptation often serves as the foundation for developing models that perform well across a wide variety of use cases, leveraging large-scale unlabelled corpora to enhance performance in related domains (Devlin et al. 2019, Gururangan et al. 2020). In contrast, increasing a model's task knowledge relates to improving its performance at a specific objective the model is supposed to achieve, such as general NLP downstream tasks like sentiment and emotion detection or named entity recognition. This usually involves fine-tuning pre-trained models on labelled data tailored to the task at hand in order to create expert systems. Domain adaptation has been explored across a variety of language contexts, each presenting unique challenges and opportunities for advancing NLP. These scenarios range from broad, general applications of adaptation to highly specialized ones and span languages from high-resourced and extensively researched, such as English, to low(er)-resourced and less commonly studied languages, such as Dutch. Given this study on adapting to historical and social media language, we discuss related work on adaptation accordingly.

Research has advanced NLP for non-standard English in recent years through domain adaptation (Gururangan et al. 2020, Manjavacas and Fonteyn 2022a) and the development of domain-specific models. Examples for the latter include social media English models such as BERTweet (Nguyen et al. 2020), Twitter-RoBERTa (Barbieri et al. 2020) and the multilingual Twitter-XLM-RoBERTa model (Barbieri et al. 2022), and historical English models like HistBERT (Qiu and Xu 2022) and MacBERTh (Manjavacas and Fonteyn 2022b), which have notably improved performance on downstream tasks involving non-normative language variants of English. For Dutch, contrastingly, handling non-normative language is mostly dependent on fine-tuning standard Dutch models like BERTje (de Vries et al. 2019), RobBERT (Delobelle et al. 2020) or multilingual models on downstream tasks. This method enables models to adapt their parameters to the task at hand, enhancing performance by increasing task-specific knowledge. Nevertheless, it does not substantially improve their domain-specific knowledge. Compared to English, no Dutch domain-specific model exclusively focusing on social media language has been developed yet. However, GysBERT (Manjavacas and Fonteyn 2022b), a domain-specific BERT-based model for historical Dutch, was developed using texts from 1500 to 1950, spanning the majority of the machine-readable Dutch literary tradition. Still, GysBERT remains the sole endeavour of its kind for the Dutch language.

Based on related work, three trends are identified. Firstly, (1) new encoder models are created from scratch with specialized tokenizers and are specifically trained for the target domain (Manjavacas and Fonteyn 2022b) or language (Delobelle et al. 2020, de Vries et al. 2019). Although building such models allows for the most specialized tokenization and makes use of all available parameters to model the target domain, there are some downsides to this approach. Due to the limited size of the corpus, generalization may be limited and overall language modelling may fail to capture some nuances by relying too much on memorization. In addition, training models from scratch requires a sufficiently large and qualitative corpus alongside intensive pre-training to learn language modelling from scratch. These resources may not be available for all specialized domains, especially for historical data. The second method (2) takes an already pre-trained monolingual or multilingual model with their corresponding tokenizer, which can already model the language relatively well, and continues pre-training on a new, specialized dataset to adapt all existing weights to represent the language of the domain (Qiu and Xu 2022, Barbieri et al. 2022). Just as the first method, this approach also updates all weights of the base model to create a new domain-adapted model. A third alternative to this, involves (3) training of adapters. This approach shares the same training objectives, unsupervised training through continued pre-training. However, as opposed to training from scratch (1) and continued pre-training (2), the weights of the source model are not updated to the target domain but are kept intact. Instead, a set of new trainable parameters is introduced to learn to map domain-generic to domain-specific embeddings. The MAD-X framework (Pfeiffer et al. 2020), for example, incorporates two invertible adapters into the model: one placed at the embedding layer and another at the output layer. These adapters have parameters that are inverses of each other, allowing them to be updated together efficiently while preserving the integrity of the original model's parameters. In their setup, they create specialized adapters that turn multi-

lingual pre-trained models into language-specific models. Their experiments suggest that training adapters is computationally more efficient than full-model pre-training, as it involves updating only a small set of additional parameters rather than the entire model. Despite using fewer parameters, the performance of adapter-based models on downstream tasks is competitive to that of models undergoing full-model pre-training, particularly for languages that are well-represented in the pre-training corpus (Pfeiffer et al. 2020). Interestingly, other research also found that adapter-training to languages that are poorly represented in the pre-training data can even outperform full-model pre-training (He et al. 2021). This makes them a promising approach for adapting models to under-represented language variants. To our knowledge, these invertible adapters have only been applied for cross-lingual adaptation from multilingual models and monolingual specialization from English to English dialects (Held et al. 2023, Xiao et al. 2023), leaving adapting monolingual Dutch models to intralingual variants, like non-normative (historical or social media) language, as a novel application.

2.2 Generative AI

Expert transformer-based encoder models have dominated the field in the last decade. Nonetheless, the landscape of NLP has been rapidly shifting with the emergence of decoder-only generative large language models (LLMs), such as GPT-3 (Brown 2020), which have demonstrated to be flexible, generalizable and performant across a wide range of tasks without requiring task-specific architectures. These LLMs are trained on massive, diverse corpora, enabling them to generalize to many domains, often with only minimal fine-tuning or prompt engineering (Bommasani 2022). This versatility stems from their extensive pre-training on massive datasets, instruction fine-tuning, and reinforcement learning through human feedback. Tasks can be addressed through zero-shot prompting, where only task instructions and input text are provided, or improved with in-context learning, where labelled examples are included directly in the prompt to guide the model's predictions. In theory, this capability should make these models promising for fully automatic classification tasks, even in under-represented languages like Dutch. These models, however, are still predominantly trained and evaluated on English corpora, leaving a gap in understanding their effectiveness for other languages, including Dutch (Huang et al. 2024, Liu et al. 2024). A recent study has shown that without fine-tuning, generative models like Llama 2 and GPT-3.5-Turbo often produce outputs that are inconsistent and fail to conform to the specific formatting required for classification tasks (De Langhe et al. 2024). These inconsistencies make it difficult to map outputs directly to predefined classification labels, requiring manual intervention to clean or restructure results. This challenge is further compounded by the limited availability of resources for instruction-tuning in Dutch. Generative models are predominantly instruction-tuned using English data or translated datasets. As a result, their ability to generalize effectively to Dutch and follow instructions in the prompt is constrained. Limited to a zero-shot setting with only in-context examples and no additional fine-tuning, these experiments prompt the question of whether fine-tuning could bridge the performance gap.

Recently, the development of Dutch-specific generative models like GEITje (Vanroy 2024b), Fietje (Vanroy 2024a), and Reynaerde (Van den Avenne 2024) has expanded the potential for generative modelling within the Dutch language. However, these models have predominantly been trained on general-domain corpora, such as Wikipedia and web-scraped content, which are largely representative of standardized Dutch. This focus limits their capabilities in handling non-normative language forms, such as historical language or modern day informal language variation commonly seen on social media. Emerging studies (Manjavacas and Fonteyn 2022b) indicate that domain-specific adaptation is critical for addressing the orthographic, lexical, and syntactic idiosyncrasies of such texts—factors often absent from the general corpora used to pre-train these new models.

2.3 From Data Modification to Model Refinement

So far we have discussed adaptation primarily from the perspective of model architectures and pre-training. A different approach for adapting to non-normative language in NLP is through "translating" such language into its standard form, a process often referred to as normalization. While the goal of normalization aligns with model adaptation in its aim to improve model performance on non-standard language, its methods differ significantly: rather than adapting the model to the data, normalization modifies the data to conform to forms the model is already familiar with and can effectively process. Adaptation through normalization typically involves techniques like spelling correction, lexical replacement and syntactic restructuring to align input data with standard language forms. Regarding adapting non-normative Dutch language through normalization, research has focused on normalizing historical language forms to their modern standard counterpart. For example, normalizing historical Dutch texts to modern spelling conventions has been explored in both rule-based (van Cranenburgh and van Noord 2022) and data-driven transformer-based encoding-decoding approaches (Wolters and Van Cranenburgh 2024). Also normalization in Dutch user-generated content (UGC) on social media has previously been explored with rule-based orthographical (Schulz et al. 2016), statistical (De Clercq et al. 2013) and transformer-based sequence-to-sequence methods for lexical normalization (Ashmawy et al. 2023). Normalizing data to align better with pre-existing model expectations might improve performance on semantic tasks such as sentiment, emotion, and irony detection. However, this approach inherently projects standardized perspectives on non-normative language, leading to the loss of linguistic nuance and potential misrepresentations of the authentic context of the original content. Consequently, normalization seems less suitable than model adaptation to support tasks requiring a thorough understanding of the unique semantic and stylistic features of non-standard language. Focusing on refining its knowledge of the characteristics of such language, model adaptation techniques ensure a more representative and context-aware modelling approach. Conclusively, advancements in model architectures, such as adaptation, offer a promising alternative. By leveraging these advancements, model adaptation can unlock new opportunities for tailoring NLP systems to challenging linguistic contexts.

3. Adapting Transformer Models

In this Section, we describe the encoder-only models (3.1), the pre-training methodology (3.2) and data (3.3) used for model adaptation.

3.1 Model Selection

In our experimental setup, we explore the adaptation of two prominent pre-trained encoder models for Dutch: BERTje (de Vries et al. 2019) and RobBERT-v3 (Delobelle and Remy 2023). These BERT-based and RoBERTa-based models are then separately fine-tuned on historical Dutch and social media data using continued full-model pre-training and adapter-heads, and successively evaluated on downstream tasks within each domain. To complement these experiments, we include an additional model tailored to historical Dutch: GysBERT (Manjavacas and Fonteyn 2022b). While GysBERT is trained on data from 1500 to 1950 and covers diverse literary and non-fictional genres, 25% of it originates from DBNL, while the other 75% comes from OCRed historical newspapers, books and journals with varying quality originating from Delpher. Even though efforts were made to select qualitatively enough OCRed texts by using a perplexity model trained on DBNL data, the predominance of such data in GysBERT can be an argument for further adapting it to the dramatic style of 17th-18th century, which is the target language of the downstream tasks for historical Dutch. Further, since GysBERT is specialized for historical contexts, we exclude it from adapting to social media tasks to make a logical comparison between models within their respective domains. Finally,

the impact of adaptation is assessed by comparing the performance of the models in their adapted settings against their baseline results on the respective tasks.

3.2 Methodology

As relayed in Related Work (see Section 2.1), there are three ways of performing adaptation for encoder models: full-model training of new models from scratch, continued training of existing pre-trained models and leveraging adapters for existing pre-trained models. As monolingual models for standardized Dutch are already available, these models contain relevant linguistic knowledge and can serve as a starting point for adapting to non-normative Dutch language. This suggests that pre-training from scratch is likely neither efficient nor required, as it would introduce an unnecessary computational load. In this work, we, therefore, do not experiment with pre-training from scratch and instead we focus on the adaptation of existing models with both continued full-model pre-training and adapter training.

These forms of adaptation work through continuing pre-training using corpora representative of the target domain to capture linguistic idiosyncrasies and contextual nuances specific to the data. For BERT-based models, like BERTje, this process relies on two pre-training objectives: Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). MLM masks a selection of tokens in a sentence and trains the model to predict them based on their surrounding context. NSP, by contrast, determines whether one sentence logically follows another. RoBERTa models, like RobBERT, on the other hand, utilize only MLM, as its creators demonstrated that NSP has minimal impact on downstream task performance (Liu et al. 2019). This finding informed our decision to adopt MLM as the sole objective for continued pre-training across all models, promoting consistency while also reducing computational overhead. MLM is the same objective for both adapter-training and continued full-model pre-training. Continued pre-training involves updating all the weights of the base model. In contrast, adapter training keeps the base model's weights frozen while updating only the additional weights added to the embedding layer and just before the output layer. In our experiments, we keep the training objectives (MLM) and all parameters consistent across the two fine-tuning methods, with the only difference being which and how many weights are updated.

The training process was tailored to reflect the characteristics of our data. The maximum sequence length was reduced to 256 tokens to accelerate training and accommodate the short length of our data, which includes single sentences from historical texts and tweets (the latter being inherently capped at 280 characters). For optimization, we employed AdamW as optimizer (Loshchilov and Hutter 2019) with a learning rate of 1e-4 and a weight decay of 0.01. The batch size was set to 64, distributed across 4 GPUs to maximize efficiency. To monitor training progress, evaluations were conducted every 2500 steps on a held-out development set comprising 1% of the total data. Early stopping was implemented to prevent overfitting, with training halting after three consecutive evaluations without a reduction in development loss, or at a maximum of 25 epochs. Notably, most models, especially those trained on larger datasets, terminated early due to the early stopping criteria, thereby avoiding overfitting while maintaining robust results. A more detailed overview of the model adaptation process and the 22 trained systems for historical and social media Dutch this resulted in can be found in Appendix A, including the reported training steps, evaluation loss and model perplexity.

3.3 Pre-Training Data

3.3.1 Historical Dutch

To adapt language models for historical Dutch, particularly for downstream tasks focusing on early modern Dutch comedies and farces from the EmDComF corpus (Debaene et al. 2024), we introduce the nLit-DBNL corpus. This corpus consists of 4 progressively expanding pre-training datasets, balancing domain specificity and data volume. The datasets are available on Hugging Face under

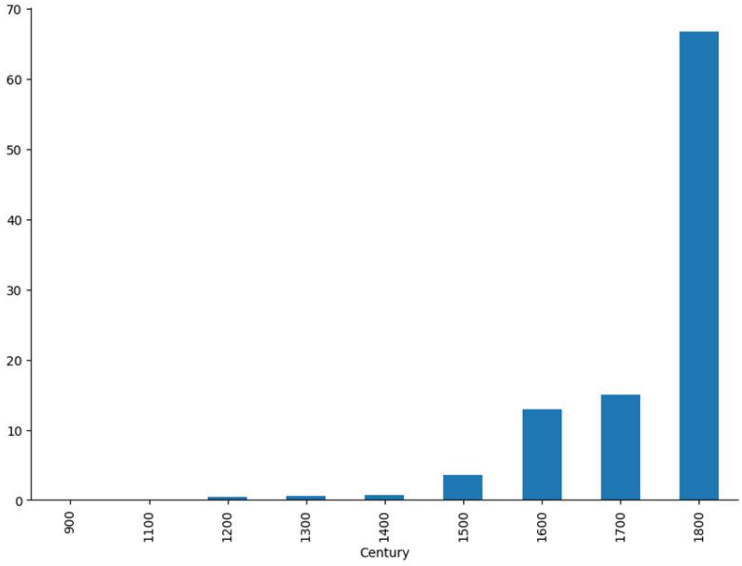


Figure 1: Texts per century (%) from the open-access historical Dutch data provided by DBNL.

LT3/nLit-DBNL. The 4 pre-training datasets are constructed as follows: (1) nLit-EmDComF contains only texts from EmDComF (Debaene et al. 2024), ensuring maximum relevance to the target language domain; (2) nLit-EmDrama expands (1) by including additional contemporaneous drama texts (1600-1800) from the Digitale Bibliotheek voor de Nederlandse Letteren (DBNL); (3) nLit-EMod further expands (2) by incorporating texts from all literary genres from that same period, introducing more stylistic and topical variety while maintaining historical proximity; (4) nLit-DBNL+ includes all available DBNL texts spanning 900-1900, maximizing data volume but also introducing greater linguistic variation and potential noise. The 1600-1800 period was selected for (2) and (3) because it represents a large portion of DBNL's available data (Figure 1), aligns chronologically with EmDComF (1650-1725), and is expected to be linguistically more consistent with early modern Dutch. Expanding beyond this time frame in (4) introduces broader linguistic variation, reflecting the evolution of Dutch but reducing domain relevance to early modern theatre texts. To enhance data quality and consistency, we applied two pre-processing steps. First, OCR post-correction was performed to address transcription errors in EmDComF, where approximately 50% of the texts originate from OCR-scanned sources. We used the best performing mBART sequence-to-sequence correction model, fine-tuned on a parallel OCR-to-gold EmDComF dataset (Debaene et al. 2025). This model, evaluated on a held-out test set (n=15,897), fully corrected 9% of all samples while balancing error reduction and avoiding overgeneration. Second, deduplication was applied to remove identically repeated data points, as these are irrelevant for masked language modelling (MLM), such as repeated character names, structural cues, and titles, ensuring more consistent pre-training data.

Since these datasets span multiple historical periods and genres, their relevance to early modern Dutch theatre varies. The further a dataset expands beyond EmDComF, the less relevant it becomes but the larger it gets. The vocabulary overlap between EmDComF and the other datasets decreases, as shown in Table 1, where nLit-EmDrama retains 62% of EmDComF's vocabulary, nLit-EMod retains 11%, and nLit-DBNL+ retains only 5%, indicating a potential loss of domain specificity and linguistic drift. Similarly, the percentage of overlapping lines in Table 1 reflects data expansion, which may enhance generalization but also introduce noise. The nLit-DBNL+ pre-training corpus is structured to systematically explore these trade-offs. By incrementally expanding the datasets in size while decreasing domain specificity, we aim to assess the impact of domain-specific pre-training

on early modern Dutch theatre tasks and examine when additional data becomes counterproductive. This structured approach provides a nuanced evaluation of data quality versus volume, contributing to historical NLP research for Dutch.

	(1)	(2)	(3)	(4)
(1) nLit-EmDComF	100.0	-	-	-
(2) nLit-EmDrama	61.73	100.0	-	-
(3) nLit-EMod	11.13	18.03	100.0	-
(4) nLit-DBNL+	4.94	8.02	44.42	100.0

	(1)	(2)	(3)	(4)	(5)
(1) nLT-Tweets	100.0	-	-	-	-
(2) nLT-Tweets	74.84	100.0	-	-	-
(3) nLT-Tweets	19.80	26.46	100.0	-	-
(4) nLT-Tweets	8.57	11.45	43.26	100.0	-
(5) nLT-Tweets	3.23	4.31	16.30	37.68	100.0

Table 1: Vocabulary overlap in Historical (left) and Social Media (right) pre-training datasets (%).

Dataset	nLit-DBNL+		nLT-Tweets	
	Lines	%	Lines	%
(1)	367,636	3.08	367,636	0.77
(2)	553,396	4.64	553,396	1.17
(3)	3,647,708	30.56	3,647,708	7.69
(4)	11,937,488	100.00	11,937,488	25.16
(5)	-	-	47,440,915	100.00

Table 2: Overlapping lines in the Historical and Social Media pre-training datasets.

3.3.2 Social Media Dutch

For Dutch social media, we present and release the nLT-Tweets corpus on Hugging Face. This previously unreleased background corpus is constructed using the Twitter (now X) Research API. We collected Dutch tweets identified via the platform's built-in language detection, excluding all retweets to focus on original content. Our approach utilized two lists of search terms: common words and stop words. The list of common words, originally compiled by Hazenberg and Hulstijn (Hazenberg and Hulstijn 1992) to support second-language learning in Dutch, includes 2,000 of the most frequent words in a collection of written Dutch. For stop words, we made use of the predefined list of 340 Dutch stop words in Spacy (Honnibal et al. 2020). By combining these lists, we ensured broad coverage of highly frequent content topics (via common words) while also incorporating general-purpose terms that are not overly topic-specific (via stop words). Given the significant overlap between the two lists, we took additional steps to refine the corpus. Lastly scraped in January 2023, the tweets were deduplicated, normalized by stripping newlines, and anonymized by removing @-mentions. Links to pictures and external websites were also excluded to minimize noise. This process resulted in a final dataset of 47,440,915 unique lines of text (tweets). Contrary to the creation of the historical pre-training corpora, no other metadata was used to further filter out data points more closely related to the domain of the downstream tasks of emotion and irony detection. In accordance with the historical Dutch pre-training datasets, however, we transposed the data size sampling by number of lines to analyse the impact of data quantity on model performance in this domain, which resulted in five pre-training datasets, as seen in Table 2. In this sampling, the distribution of vocabulary overlap between the datasets shows similar tendencies as in the historical pre-training data, but is on average more similar (Table 1). We hypothesize that more similar content is present throughout the sets, which could have an impact on downstream performance after domain adaptation.

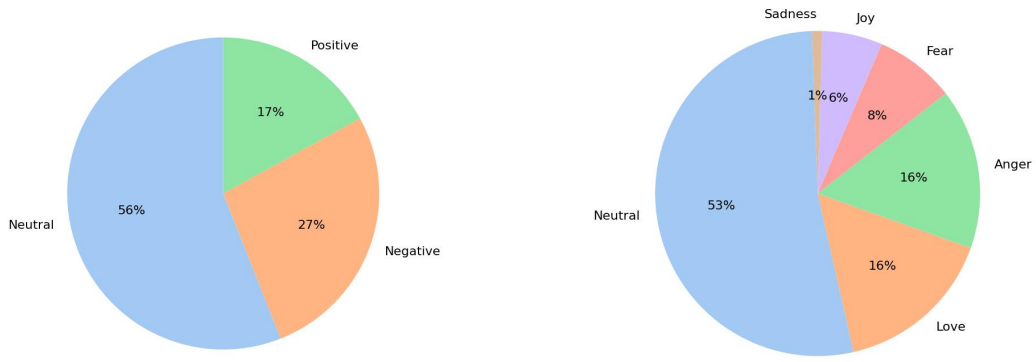
4. Performance on Downstream Tasks

4.1 Task Data

In the following sub-sections, we describe the data used to evaluate our models for downstream tasks. For historical Dutch, we evaluate on detecting sentiment and emotion in early modern Dutch comedies and farces. For social media Dutch, we evaluate on the tasks of irony detection and emotion detection in tweets. As social media data can contain personal data, we made sure to use anonymized data. This involves the same pre-processing as we used for cleaning the social media pre-training corpus.

4.1.1 Sentiment Analysis - Historical

For the historical downstream tasks, we present two datasets for multiclass sentiment and emotion detection in early modern Dutch originating from the EmDComF corpus (Debaene et al. 2024). Out of this corpus, 9 plays were selected (4 comedies and 5 farces) and sequentially annotated at the sentence level after automatic sentence splitting was applied using NLTK (Bird et al. 2009), as plays in EmDComF are unstructured .txt files. This also means that annotators had access to the context in which sentences follow each other. For an overview of the annotated plays, we refer to Appendix B. Sentiment was annotated with fine-grained "very negative", "negative", "neutral", "positive" and "very positive" labels by 3 expert annotators. Measuring inter-annotator agreement (IAA) on a farce consisting of 326 sentences resulted in an α score (Krippendorff 2011) of 0.700. For this work, however, we merge the sentiment labels to three categories, ignoring the strength of the sentiment, which results in an increased α score of 0.774, almost indicating satisfactory agreement ($\geq$0.80). Finally, annotating 9 plays this way resulted in a dataset of 7,332 sentences annotated with sentiment. The sentiment label distribution is shown in Figure 2a. The dataset is split into a train set of 5278 samples (72%), dev set of 587 samples (8%) and test set of 1467 samples (20%) and is available on Hugging Face at floriandebaene/EmDComF_primary-emotion.



(a) Sentiment Analysis (b) Emotion Detection

Figure 2: Label Distributions for Historical Tasks

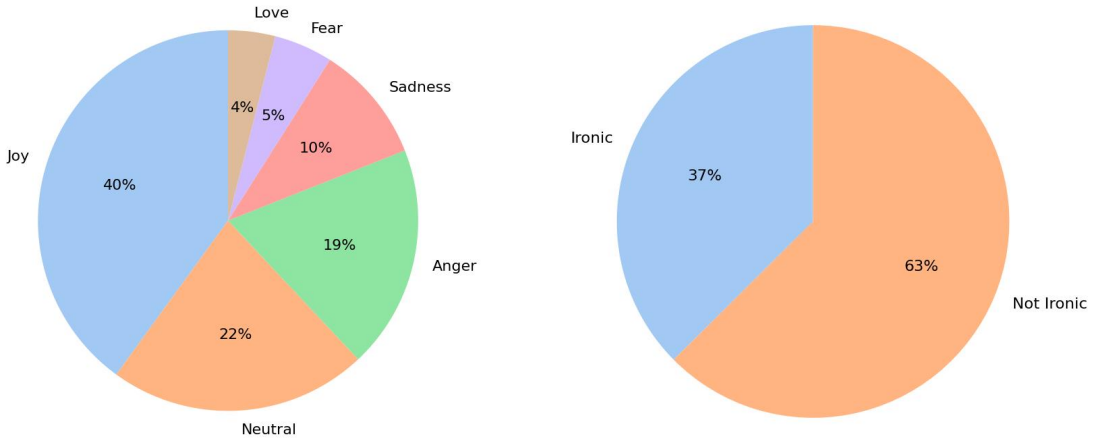
4.1.2 Emotion Detection - Historical

The second downstream task for historical Dutch is multiclass emotion detection. For this task, we used the same 7,332 sentences as we labeled for sentiment detection and enriched these with the following emotion labels: "Love", "Joy", "Fear", "Anger", "Sadness" and "Neutral". Measuring

IAA resulted in a moderate alpha score of 0.773 across all classes. More concretely, satisfactory agreement was found for neutral ($\alpha = 0.812$), moderate agreement for love ($\alpha = 0.788$), joy ($\alpha = 0.738$), fear ($\alpha = 0.732$), sadness ($\alpha = 0.688$) and almost moderate agreement for anger ($\alpha = 0.667$; < 0.67). The final emotion label distribution in the whole dataset is shown in Figure 2b. The data is split into a train set of 5278 samples (72%), dev set of 587 samples (8%) and test set of 1467 samples (20%), available on Hugging Face at floriandebaene/EmDComF_sentiment. In this work, we will establish SOTA on both historical tasks and make the datasets open access.

4.1.3 Emotion Detection - Social Media

The first downstream task in the social media domain is multiclass emotion detection. For this purpose, we utilize the EmotioNL dataset (De Bruyne et al. 2021), which originally consists of 2,000 texts: 1,000 tweets and 1,000 captions from Flemish reality TV-shows. From this dataset, we only make use of the 1,000 tweets, as the other texts already align more with standardized language. This previously unpublished corpus is now publicly available on Hugging Face at LT3/EmotioNL_Tweets. The tweets in this dataset were scraped using emoticons and emojis as search terms to attain a high number of emotional texts. These 1,000 tweets are labelled as "Love", "Joy", "Fear", "Anger", "Sadness" or "Neutral" and follow the class distribution described in Figure 3a. For this task, Cohen's kappa (Cohen 1960) was calculated for 300 tweets, resulting in an overall moderate agreement (κ =0.504) across all classes. In addition, the agreement was also calculated per class, finding a substantial agreement for anger ($\kappa = 0.608$) and sadness ($\kappa = 0.682$) and fair agreement for fear ($\kappa = 0.313$), joy ($\kappa = 0.380$) and love ($\kappa = 0.210$). For the neutral category, a moderate agreement was found ($\kappa = 0.513$). To ensure effective model evaluation, the dataset is split into a training set (700 samples), a development set (100 samples), and a test set (200 samples). The current state-of-the-art for fine-tuning pre-trained models is achieved using RobBERT v2 (Delobelle et al. 2020), which reaches a macro-averaged F1-score of 40% (De Bruyne et al. 2021). However, recent tests with zero-shot generative models have demonstrated improved performance for generative models, with GPT-3.5-turbo attaining 46% F1-score (De Langhe et al. 2024). For this SOTA, 10-fold cross-validation was used during evaluation for RobBERT, meaning that 10 models were trained within a single experimental setup. As we explore many different experimental settings, using 10-fold cross-validation would significantly increase the computational cost of our experiments. Therefore, we use a single held-out test set instead of 10-fold CV and will not be able to compare to this exact score, but we still include fine-tuned RobBERT as one of the systems in our evaluation.



(a) Emotion Detection (b) Irony Detection

Figure 3: Label Distributions for Social Media Tasks

4.1.4 Irony Detection - Social Media

The second downstream task in the social media domain is irony detection. The dataset used for this task is the only available dataset for Dutch irony detection. It was originally collected by scraping tweets on the search terms #irony, #sarcasm and #not (Van Hee et al. 2016). After removing these irony-related search terms, the tweets were annotated by student workers into two categories "ironic" and "not ironic". An agreement study on 200 tweets by 3 annotators affirms substantial agreement for this annotation task with an α score (Krippendorff 2011) of 0.7905. The final dataset comprises 5,566 samples, following the distribution described in Figure 3b. For training and evaluation, the data is split into a training set (4,008 samples, 72%), a development set (445 samples, 8%), and a test set (1,113 samples, 20%).

4.2 Fine-tuning Methodology

4.2.1 Pre-Trained Encoder Models

Transformer encoding models like BERT and RoBERTa can be fine-tuned for classification tasks by replacing the language modelling head with a dedicated classification head. In BERT-based models, this head processes the [CLS] token, a special token that encodes a representation of the entire input sentence, and forwards it through a single layer of neurons to map the representation to the desired output classes. For RoBERTa models, the classification head consists of two layers of neurons, providing an additional level of task-specific modelling that can enhance performance. Once the model maps the sentence meaning to the predefined classes, the output logits are converted to probabilities using a softmax function. The class with the highest probability is then selected as the model's prediction.

During training, both the weights of the pre-trained encoder and the classification head are updated to optimize task-specific performance. Cross-entropy loss is used as the objective function to minimize prediction errors. All models are fine-tuned with a batch size of 16, distributed across 2 GPUs, using AdamW as optimizer with a learning rate of 5e-6, a weight decay of 0.01, and 200 warm-up steps. Fine-tuning is performed for up to 30 epochs, with performance evaluated every 100 steps on a task-specific held-out development set. To prevent overfitting and ensure training efficiency, early stopping is applied when the macro-averaged F1-score on the development set fails to improve for three consecutive evaluations.

In our experiments, we evaluate two types of pre-trained encoder models: (1) models where the weights of the entire encoder are fine-tuned and (2) models where only the adapter weights are trained. Adapters, when added to the pre-trained encoder models, increase the total number of parameters. To ensure a fair comparison between models with and without adapters, we freeze the adapter weights during fine-tuning, updating only the weights of the encoder and classification head. This setup allows us to isolate the impact of the adapters during the pre-training phase while keeping the fine-tuning procedure consistent across both configurations. Alongside the adapted models (with and without adapters), we also evaluate baseline models, which are pre-trained encoders without adapters.

4.2.2 Generative Decoder Models

As detailed in Section 2, generative models often produce outputs that are inconsistent and fail to conform to the specific formatting required for classification tasks. To address these challenges, fine-tuning offers a practical solution for aligning generative models with task-specific requirements. By fine-tuning, we can adapt models to handle Dutch-specific tasks and ensure their outputs conform to predefined formats without requiring extensive manual intervention. However, the large size of generative models presents significant computational challenges for traditional full-model fine-tuning. Given that the modifications needed for many classification tasks are often limited to learning the output format, fine-tuning the entire model is cumbersome, inefficient and unneces-

sarily resource-intensive. To overcome these limitations, we employ Quantized Low-Rank Adapters (QLoRA) (Dettmers et al. 2023), a method that reduces the computational cost and memory requirements of fine-tuning while maintaining high performance. Low-rank adapters approximate updates to a model's weights using rank decomposition, representing the changes (ΔW) as the product of two smaller matrices:

$$\Delta W = A \cdot B$$

Here, A and B are low-rank matrices with significantly fewer parameters than the original weight matrix. By training only these matrices, the majority of the pre-trained parameters remain frozen, significantly reducing the memory and computational demands of fine-tuning. Unlike invertible adapters, which are inserted between attention layers and increase computational overhead, low-rank adapters operate in parallel with the base model layers, ensuring minimal impact on training speed. When combined with quantization, which optimizes weight precision, QLoRA offers a highly efficient solution for adapting large models with minimal performance loss (Dettmers et al. 2023). Fine-tuning was performed with 6 predefined random seeds to strike a balance between reproducibility and fluctuations in model performance due to different initializations.

In our experimental setup, we leverage both in-context learning and adapter-based fine-tuning to enhance the generative models' ability to produce outputs in the desired format. Specifically, we sample an additional labelled example for each label from the training set and embed them into the prompt to leverage in-context examples when training the generative models. Fine-tuning is then performed using LoRA adapters with 4-bit quantization, an alpha value of 16, and a rank of 128 to capture the complexity of tasks such as emotion and irony detection. Although this rank is higher than typical LoRA configurations, it remains computationally efficient compared to training the full embedding matrix of models like RoBERTa, which have an embedding size of 768. All generative models are trained with a learning rate of 2e-5 for five epochs using the full training set, without early stopping, with a batch size of 4 (1 sample per GPU). Gradient accumulation steps are set to 3, meaning that weights are updated every 12 samples. This setup ensures robust training while maintaining computational efficiency.

For our experiments, we train both widely used English language models, such as Llama 3 (Grattafiori 2024), and Dutch-specific models, including GEITje and Fietje (Vanroy 2024b, Vanroy 2024a), and Reynaerde (Van den Avenne 2024). To maximize performance and ensure consistency with task instructions, we select instruction-tuned versions of all models. The language of the prompt also influences generative models' performance. For this study, we prompt all generative models in English to maintain consistency and optimally leverage prior instruction tuning, which often includes tasks such as sentiment analysis, emotion detection, and irony classification performed on English datasets. The prompts for all tasks are included in Appendix C.

4.3 Results

To assess the model performance on downstream tasks, we use macro-averaged F1-score as our primary evaluation metric. While irony detection and sentiment analysis only consider two or three labels, emotion detection includes 5 emotion labels along with a neutral label. Given the class imbalances present in all tasks, we employ macro-averaging to ensure equal weighting across all labels. The reported F1-scores represent the average performance across 6 random seed initializations. Additionally, we provide Δ coefficients, which quantify the statistically estimated improvement over the baseline using a mixed-effects model implemented with statsmodels (Seabold and Perktold 2010). The mixed-effects model accounts for variability across different models and random seeds to provide a more reliable estimate of performance differences. To assess whether these improvements are statistically significant, we include two-tailed p-values alongside the Δ coefficients. These p-values indicate the probability of observing the reported performance differences under the assumption that there is no real effect of model adaptation. Lower p-values suggest stronger evidence that the adaptation methods contribute to meaningful performance gains. We use standard significance

thresholds, namely: $p < 0.05$*, $p < 0.01$**, and $p < 0.001$***, where smaller values denote stronger statistical confidence in the observed improvements. In this experimental setup, we evaluate a total of 588 fine-tuned encoding models (60 baseline models and 528 adapted models). To provide a clear overview of the results, we split up the evaluation to answer our specific research questions and refer to Appendix D for an overview of all detailed scores. First, we discuss whether model adaptation improves performance in general by averaging across pre-training dataset, adaptation method and base models. Secondly, we investigate what role data relevance and data volume plays in the two different adaptation methods by averaging across base models. Thirdly, we explore which models perform best at each task, including both pre-training data size, method and base model in our analysis, and compare this to the performance of the generative models at these tasks.

4.3.1 Model Adaptation versus Baselines

Model adaptation through continued pre-training significantly improves downstream performance in both historical and social media domains. Table 3 presents macro-averaged F1-scores, averaged across random seed initializations, pre-training datasets per domain, and base models. The results indicate that adaptation yields statistically significant improvements in most cases, with adapters generally outperforming continued full-model pre-training (cfmp). Adapters demonstrate more stable performance across tasks, as reflected in their lower standard deviation across tasks compared to cfmp. This stability contributes to their superior average performance for historical emotion detection, social media emotion detection, and social media irony detection, where adaptation has negligible impact but adapters still outperform cfmp. For the **historical domain**, both adaptation methods lead to statistically significant performance gains of nearly 2% F1-score, highlighting the effectiveness of domain-specific pre-training. However, in the **social media domain**, the impact of adaptation is task-dependent. While emotion detection benefits from a substantial and statistically significant increase of over 6% F1-score, irony detection shows minimal and insignificant improvement, with cfmp even performing worse than the baseline. The negative Δ coefficient for cfmp on irony detection suggests that additional pre-training data may have introduced noise rather than meaningful improvements. These findings emphasize the importance of selecting pre-training data that aligns with the linguistic properties of the specific task, rather than assuming broad domain adaptation will yield uniform benefits.

method	task	F1	Δ	std
baseline	emotion	0.4208	-	0.0318
adapter	emotion	**0.4387**	**0.0179***	0.0418
cfmp	emotion	0.4277	0.0069	0.0528
baseline	sentiment	0.6745	-	0.0170
adapter	sentiment	0.6887	0.0142***	0.0203
cfmp	sentiment	**0.6913**	**0.0168*****	0.0276

(a) Historical

task	F1	Δ	std
emotion	0.2927	-	0.0498
emotion	**0.3558**	**0.0631*****	0.0389
emotion	0.3506	0.0579***	0.0404
irony	0.7213	-	0.0122
irony	**0.7219**	**0.0006**	0.0119
irony	0.7129	-0.0084*	0.0111

(b) Social Media

Table 3: Performance comparison of model adaptation methods against baselines on the downstream tasks per domain, averaged across random seeds, datasets and models. F1 represents classification performance. Δ denotes the estimated improvement from a mixed-effects model over the baseline. Standard deviation (std) quantifies variability in performance. Statistical significance levels: $p < 0.05$*, $p < 0.01$**, and $p < 0.001$***.

dataset	emotion				irony			
	adapter		cfmp		adapter		cfmp	
	F1	Δ	F1	Δ	F1	Δ	F1	Δ
1	0.3417	0.0490**	**0.3989**	**0.1062*** **	0.7211	-0.0001	0.7128	-0.0085*
2	0.3501	0.0574***	0.3607	0.0680***	0.7222	0.0009	0.7033	-0.0180***
3	**0.3649**	**0.0722*** **	0.3238	0.0311**	0.7242	0.0029	**0.7277**	**0.0064**
4	0.3622	0.0695***	0.3141	0.0213	0.7167	-0.0046	0.7093	-0.0119***
5	0.3600	0.0673***	0.3555	0.0628***	**0.7252**	**0.0039**	0.7115	-0.0097**

Table 4: Effect of dataset and model adaptation methods on Social Media downstream tasks against the baseline, averaged across random seeds and models. F1 represents classification performance. Δ denotes the estimated improvement from a mixed-effects model over the baseline. Statistical significance levels: $p < 0.05$*, $p < 0.01$**, and $p < 0.001$***.

dataset	emotion				sentiment			
	adapter		cfmp		adapter		cfmp	
	F1	Δ	F1	Δ	F1	Δ	F1	Δ
1	0.4467	0.0259**	0.4330	0.0122	0.6900	0.0155***	0.6985	0.0240***
2	**0.4538**	**0.0330*** **	**0.4444**	**0.0236***	0.6901	0.0156***	**0.6994**	**0.0249*** **
3	0.4297	0.0089	0.4245	0.0037	**0.6904**	**0.0159*** **	0.6858	0.0113**
4	0.4244	0.0036	0.4091	-0.0117	0.6842	0.0097***	0.6814	0.0069*

Table 5: Effect of dataset and model adaptation methods on Historical downstream tasks against the baseline, averaged across random seeds and models. F1 represents classification performance. Δ denotes the estimated improvement from a mixed-effects model over the baseline. Statistical significance levels: $p < 0.05$*, $p < 0.01$**, and $p < 0.001$***.

4.3.2 Impact of Pre-training Datasets

Up to this point, our analysis has focused on model adaptation methods without considering differences in pre-training datasets or base models. Now, we examine the effect of the pre-training datasets while still averaging across base models, aiming to determine the optimal pre-training setting for downstream performance. For the **social media domain** (Table 4), where no pre-training data relevance distinctions were made, we observe task-dependent effects of domain adaptation. Cfmp does not require the largest datasets to achieve optimal performance, as the smaller pre-training datasets often yield better results. For emotion detection, dataset (1) produces the highest cfmp F1-scores, whereas for irony detection dataset (3) achieves best performance. In contrast, adapter-based adaptation does benefit from larger datasets. Adapter models perform best on dataset (3) for emotion detection and dataset (5) for irony detection, suggesting that adapters require more extensive domain exposure to reach peak performance. Interestingly, while the adapters consistently and significantly improve emotion detection, their performance remains on par with the baseline, neither improving nor degrading results. However, cfmp significantly worsens performance for irony detection on all datasets but (3), where the gains remain negligible. These results indicate that domain adaptation for the social media Dutch tasks is task-sensitive, with emotion detection benefiting from adaptation, while irony detection shows little to no improvement. Additionally, cfmp models require less data for effective adaptation, whereas adapter models rely on larger datasets for optimal results.

In the **historical domain** (Table 5), increasing the quantity of pre-training data does not always translate to optimal adaptation. In fact, it can reduce adaptation effectiveness. For cfmp, the best performance is achieved with dataset (2) for both emotion and sentiment analysis, while for adapter

models dataset (2) is optimal for emotion detection, and dataset (3) slightly outperforms both (2) and (1) for sentiment analysis. Dataset (1) emerges as the second-best performer across all historical tasks. Notably, dataset (2) includes the EmDComF corpus and extends it with additional drama texts from 1600-1800, making it the most effective pre-training dataset for both adaptation methods. This suggests that both genre and time-bound data are particularly beneficial for adapting to both tasks set in early modern Dutch comedies and farces, leading to improved downstream performance. Expanding the dataset beyond dataset (2) to create datasets (3) and (4) introduces additional pre-training data from different time periods and genres, which appear irrelevant for emotion detection but does benefit sentiment analysis, albeit to a lesser extent. This suggests that domain relevance generally outweighs dataset size in our adaptation experiments. However, despite the lower results from the largest datasets, all adapted models still generally outperform their non-adapted counterparts, underscoring the effectiveness of adaptation.

task	method	split	model	F1	Δ
emo	baseline	-	gysbert	0.4462	-
emo	adapter	2	gysbert	**0.4974**	0.0512
emo	cfmp	2	gysbert	0.4956	0.0493
sent	baseline	-	gysbert	0.6946	-
sent	adapter	1	gysbert	0.7166	0.0220
sent	cfmp	1	gysbert	**0.7266**	0.0320**

(a) Historical

task	split	model	F1	Δ
emo	-	robbert	0.3297	-
emo	4	robbert	0.3905	0.0608
emo	1	robbert	**0.4135**	0.0838*
irony	-	bertje	0.7264	-
irony	5	robbert	0.7258	-0.0006
irony	3	robbert	**0.7315**	0.0050

(b) Social Media

Table 6: Competitive comparison on the downstream tasks per domain using the best-performing setting for each method, averaged across random seeds. F1 represents classification performance. Δ denotes the estimated improvement from a mixed-effects model over the baseline. Statistical significance levels: $p < 0.05$*, $p < 0.01$**, and $p < 0.001$***. Significance should here be interpreted with caution, as it is based on 6 random seeds.

4.3.3 Best Model Adaptation Configurations

Extending our analysis to consider differences between individual base models, we present the best-performing pre-training configurations in Table 6. These results provide insights into how different adaptation methods interact with specific models across both historical and social media domains. For the **historical domain** (Table 6a), GysBERT consistently emerges as the best-performing model, achieving the highest F1 scores across both adaptation methods. This suggests that GysBERT's initial design and pre-training already provide a strong foundation for historical Dutch NLP tasks, making it better suited for adaptation to early modern Dutch than RobBERT or BERTje. Looking at specific tasks, sentiment analysis achieves optimal performance with dataset (1), yielding a significant F1-score improvement of 3% over baseline GysBERT. This suggests that for historical sentiment analysis, highly curated, domain-specific data is sufficient for effective adaptation. In contrast, emotion detection, a more nuanced and complex task, benefits from a larger dataset (2), which provides a more diverse linguistic context and results in a 5% performance increase over the baseline. However, in this highly competitive comparison where F-1 scores are based on only 6 results, this particular performance increase is not found statistically significant. In the **social media domain** (Table 6b), RobBERT achieves the best results in most scenarios, outperforming BERTje in adapted models. The only exception is irony detection, where BERTje outperforms RobBERT in the baseline. However, after adaptation, RobBERT consistently achieves higher scores for both irony and emotion detection, demonstrating that it benefits more from domain adaptation than BERTje. For emotion detection, cfmp adaptation achieves the highest performance on dataset (1) using RobBERT, resulting in an 8% increase in F1-score compared to the best-performing baseline. This

substantial improvement highlights that cfmp effectively refines the model for detecting emotional nuances in social media texts. For irony detection, the best performance is obtained using cfmp on dataset (3). However, the performance differences for irony detection across different datasets remain minimal, indicating that the choice of pre-training data is less critical for this task.

A fine-grained analysis of pre-training data and base model selection reveals that the optimal adaptation strategy depends on both the task and the model at hand. While adapter training generally performs better on average (Table 3), the highest task-specific performance is typically achieved with cfmp, which enables deeper integration of domain knowledge. Three out of four tasks reach peak performance using cfmp, with historical emotion detection as the only exception, where the difference is minimal. Additionally, performance gaps between adapter training and cfmp for historical emotion detection, sentiment analysis, and irony detection remain within 0.01 F1-score, suggesting that both methods offer competitive results. These results indicate that the effectiveness of adaptation also depends on the model's prior familiarity with the domain. Models like GysBERT for historical Dutch and RobBERT for social media Dutch achieve state-of-the-art performance after continued pre-training, reinforcing the importance of domain familiarity. While adapter-based training provides a modular and computationally efficient approach, cfmp generally delivers superior results, particularly for models with pre-existing domain strengths. Ultimately, the choice between these methods should consider both the model's baseline capabilities and the linguistic complexities of the task.

4.3.4 Adapted Encoders Versus Generative Decoders

Having identified the most effective adaptation strategies approaches for encoder models, we now compare their performance with generative decoder models. Table 7 shows that fine-tuned Llama 3 achieves the highest F1 scores on three out of four tasks, outperforming other models in historical sentiment analysis, social media emotion detection and irony detection. The only exception is historical emotion detection, where fine-tuned Fietje and Reynaerde surpass Llama 3 with F1 scores of 27% and 25.5%, respectively. Across all tasks, fine-tuned generative models significantly outperform their zero-shot counterparts with in-context learning, underscoring the necessity of task-specific fine-tuning for classification tasks in non-normative varieties of Dutch. However, our results also reveal that multilingual English-dominant models like Llama 3 still outperform monolingual Dutch generative models. We hypothesize that this advantage stems from the broader availability of instruction-tuning data for English-based models, whereas the scarcity of such data limits the effectiveness of monolingual Dutch generative models for medium- and low-resource languages.

task	model	F1	
		finetuned	zeroshot
emo	fietje	**0.2669**	0.0105
emo	geitje	0.2339	0.0714
emo	reynaerde	0.2548	0.0529
emo	llama	0.2417	0.1042
sent	fietje	0.3899	0.0160
sent	geitje	0.3673	0.1056
sent	reynaerde	0.3948	0.0556
sent	llama	**0.4913**	0.1331

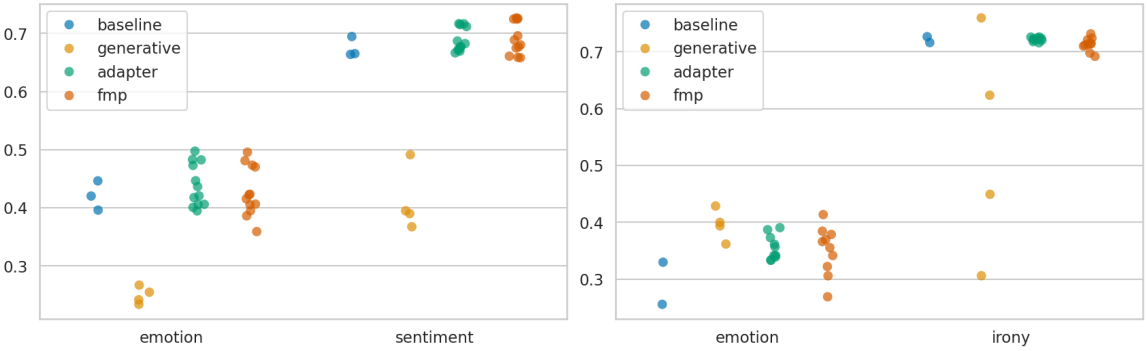
(a) Historical

task	model	F1	
		finetuned	zeroshot
emo	fietje	0.3998	0.0393
emo	geitje	0.3936	0.2652
emo	reynaerde	0.3617	0.1488
emo	llama	**0.4286**	0.3444
irony	fietje	0.4492	0.0090
irony	geitje	0.3060	0.0274
irony	reynaerde	0.6235	0.0322
irony	llama	**0.7595**	0.3969

(b) Social Media

Table 7: Classification Scores for Generative Models

Figure 4 contrasts the performance of the generative models with our adapted encoder models and their baselines. In the **historical domain** (Figure 4a), generative models perform considerably worse than the baseline encoders, highlighting their limited ability to handle early modern Dutch. Conversely, both adaptation methods consistently improve over baseline models for historical sentiment and emotion detection. Among the adaptation methods, adapters demonstrate greater stability with smaller performance fluctuations, whereas cfmp models achieve high scores when sufficient high-quality pre-training data is available. For the **social media domain** (Figure 4b), performance trends vary by task. Emotion detection benefits from model adaptation, with adapted encoders and fine-tuned generative models outperforming baselines. In contrast, irony detection shows minimal differences between baseline and adapted models, suggesting that adaptation has limited impact on this task. Among generative models, monolingual Dutch variants perform poorly across both tasks, whereas fine-tuned Llama 3 emerges as the top-performing model, surpassing adapted encoders with F1 scores of 49.13% for emotion detection and 75.95% for irony detection. The main difference we can identify between the two social media tasks lies in the annotated dataset construction. For irony detection, the tweets were originally collected using irony-related hashtags. Although these hashtags were removed before annotation and were not included in the input for classification models, it is likely that some of the original tweets (including the irony hashtags) were present in the pre-training data of the encoder models (RobBERT and BERTje) and generative models like Llama 3. If these pseudo-labels were available during pre-training, they could artificially inflate model performance, reducing the need for explicit adaptation. This suggests that fair evaluation for irony detection may require a novel dataset that eliminates reliance on pre-existing irony markers such as hashtags.



(a) Historical (b) Social Media

Figure 4: Domain Adaptation Methods versus Generative Models.

5. Conclusion & Future Work

Our study demonstrates that model adaptation significantly improves the performance of pre-trained Dutch encoder models on downstream tasks in both historical and social media domains. Both adapter training and continued full-model pre-training consistently improve results over general baselines. The primary difference between these methods lies in their stability: adapter training produces more consistent results with less fluctuation across different seeds and data sizes, whereas full-model pre-training achieves the highest absolute performance when sufficient and qualitative data and computational resources are available. A key finding of our work is that historical Dutch tasks benefit more from model adaptation than social media tasks across the evaluated models. This likely stems from the nature of pre-training data, as web-scraped corpora such as OSCAR used for Dutch language modelling already include a substantial amount of social media data, whereas historical Dutch remains under-represented. Consequently, model adaptation boosts sentiment analysis

and emotion detection in historical Dutch, whereas in the social media domain, only emotion detection sees significant gains. For irony detection, baseline models already perform strongly, suggesting that existing pre-training data contains sufficient domain-relevant information.

When comparing generative and encoder-based models, our results highlight the limitations of current monolingual Dutch generative models. Even though fine-tuning with in-context learning improves their performance over zero-shot prompting, these models underperform compared to adapted encoder models, particularly in historical Dutch tasks. Despite Llama 3 being an English-centric multilingual model, it outperforms Dutch generative models and, in some cases, even adapted encoder-based models, especially for social media emotion and irony detection. This advantage is likely due to the availability of extensive instruction-tuning data, which remains scarce for monolingual Dutch generative models. However, none of the generative models come close to matching the performance of encoder-based models in historical Dutch, suggesting that the linguistic differences between early modern and contemporary Dutch remain a significant challenge.

Future work should further research domain-specific adaptation strategies across different model architectures, datasets, and languages. While generative models perform well in social media classification, exploring their potential for historical Dutch through improved instruction tuning or domain-specific pre-training could be valuable. Overall, our findings confirm that model adaptation is essential for improving NLP performance in under-represented Dutch language varieties. While adapter-based training offers an efficient and stable approach, continued full-model pre-training remains the most effective strategy when sufficient resources are available. Moreover, while generative models show promise, their effectiveness in low-resource and historical domains remains limited, necessitating further research into improved adaptation techniques.

Limitations

This study does not explore tokenization and full-model pre-training from scratch, which are alternative approaches for developing domain-specific or language-specific models (see Section 2). However, given the wide range of strategies for training models from scratch, this falls beyond the scope of this work. Nevertheless, initializing vocabulary from existing models (Remy et al. 2023) could help address lexical gaps caused by spelling variations and out-of-domain vocabulary in non-normative language. Additionally, due to resource limitations for non-normative Dutch, our analysis focuses on two domains with only two tasks per domain. During task evaluation, we account for label imbalance by reporting macro-averaged F1-scores, but we do not investigate how model adaptation affects performance on individual labels.

Acknowledgments

The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO and the Flemish Government – department EWI. This work was supported by the Research Foundation Flanders (FWO) under grant G032123N and by Ghent University under grant BOF.24Y.2021.0019.01.

References

Araci, Dogu (2019), Finbert: Financial sentiment analysis with pre-trained language models. https://arxiv.org/abs/1908.10063.

Ashmawy, Mohamed, Mohamed Waleed Fakhr, and Fahima A Maghraby (2023), Lexical normalization using generative transformer model (ln-gtm), *International Journal of Computational Intelligence Systems* **16** (1), pp. 183, Springer.

Barbieri, Francesco, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves (2020), TweetEval: Unified benchmark and comparative evaluation for tweet classification, *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, Online, pp. 1644–1650. https://aclanthology.org/2020.findings-emnlp.148.

Barbieri, Francesco, Luis Espinosa Anke, and Jose Camacho-Collados (2022), XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, European Language Resources Association, Marseille, France, pp. 258–266. https://aclanthology.org/2022.lrec-1.27.

Bird, Steven, Ewan Klein, and Edward Loper (2009), *Natural language processing with Python: analyzing text with the natural language toolkit*, O'Reilly Media, Inc.

Bommasani, et al. (2022), On the opportunities and risks of foundation models. https://arxiv.org/abs/2108.07258.

Brown, et al. (2020), Language models are few-shot learners. https://arxiv.org/abs/2005.14165.

Chalkidis, Ilias, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos (2020), Legal-bert: The muppets straight out of law school. https://arxiv.org/abs/2010.02559.

Cohen, Jacob (1960), A coefficient of agreement for nominal scales, *Educational and psychological measurement* **20** (1), pp. 37–46, Sage Publications Sage CA: Thousand Oaks, CA.

De Bruyne, Luna, Orphée De Clercq, and Véronique Hoste (2021), Prospects for dutch emotion detection: Insights from the new emotionl dataset, *Computational Linguistics in the Netherlands Journal* **11**, pp. 231–255.

De Clercq, Orphée, Bart Desmet, Sarah Schulz, Els Lefever, and Véronique Hoste (2013), Normalization of dutch user-generated content, *9th International conference on Recent Advances in Natural Language Processing (RANLP 2013)*, Incoma, pp. 179–188.

De Langhe, Loic, Aaron Maladry, Bram Vanroy, Luna De Bruyne, Pranaydeep Singh, Els Lefever, and Orphée De Clercq (2024), Benchmarking zero-shot text classification for dutch, *Computational Linguistics in the Netherlands Journal* **13**, pp. 63–90.

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.

Debaene, Florian, Aaron Maladry, Els Lefever, and Véronique Hoste (2025), Evaluating transformers for ocr post-correction in early modern dutch theatre, *Proceedings of the 31st International Conference on Computational Linguistic (COLING 2025)*, International Committee on Computational Linguistics.

Debaene, Florian, Kornee van der Haven, and Veronique Hoste (2024), Early Modern Dutch comedies and farces in the spotlight: Introducing EmDComF and its emotion framework, *in* Sprugnoli, Rachele and Marco Passarotti, editors, *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA) @ LREC-COLING-2024*, ELRA and ICCL, Torino, Italia, pp. 144–155. https://aclanthology.org/2024.lt4hala-1.17.

Delobelle, P and F Remy (2023), Robbert-2023: Keeping dutch language models up-to-date at a lower cost thanks to model conversion. https://clin33.uantwerpen.be/abstract/robbert-2023-keeping-dutch-language-models-up-to-date-at-a-lower-cost-thanks-to-model-conversion/.

Delobelle, Pieter, Thomas Winters, and Bettina Berendt (2020), RobBERT: a Dutch RoBERTa-based Language Model, *in* Cohn, Trevor, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, Online, pp. 3255–3265. https://aclanthology.org/2020.findings-emnlp.292.

Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer (2023), Qlora: Efficient finetuning of quantized llms.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019), BERT: Pre-training of deep bidirectional transformers for language understanding, *in* Burstein, Jill, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. https://aclanthology.org/N19-1423.

Grattafiori, et al. (2024), The llama 3 herd of models. https://arxiv.org/abs/2407.21783.

Gururangan, Suchin, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith (2020), Don't stop pretraining: Adapt language models to domains and tasks, *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. 8342–8360. https://aclanthology.org/2020.acl-main.740.

Hazenberg, Suzanne and Jan Hulstijn (1992), Woorden op zicht. woordselectie ten behoeve van het nt2-onderwijs, *Levende Talen Magazine* **79** (467), pp. 2–7.

He, Ruidan, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and Luo Si (2021), On the effectiveness of adapter-based tuning for pretrained language model adaptation, *arXiv preprint arXiv:2106.03164*.

Held, William, Caleb Ziems, and Diyi Yang (2023), TADA : Task agnostic dialect adapters for English, *in* Rogers, Anna, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, Association for Computational Linguistics, Toronto, Canada, pp. 813–824. https://aclanthology.org/2023.findings-acl.51/.

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem, and Adriane Boyd (2020), spaCy: Industrial-strength Natural Language Processing in Python.

Huang, Kaiyu, Fengran Mo, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, Jinan Xu, Jian-Yun Nie, and Yang Liu (2024), A survey on large language models with multilingualism: Recent advances and new frontiers. https://arxiv.org/abs/2405.10936.

Krippendorff, Klaus (2011), Computing krippendorff's alpha-reliability, *Computing* **1**, pp. 25–2011.

Lee, Jinhyuk, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang (2019), Biobert: a pre-trained biomedical language representation model for biomedical text mining, *Bioinformatics* **36** (4), pp. 1234–1240, Oxford University Press (OUP). http://dx.doi.org/10.1093/bioinformatics/btz682.

Liu, Weihao, Ning Wu, Wenbiao Ding, Shining Liang, Ming Gong, and Dongmei Zhang (2024), Towards truthful multilingual large language models: Benchmarking and alignment strategies. https://arxiv.org/abs/2406.14434.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019), Roberta: A robustly optimized bert pretraining approach. https://arxiv.org/abs/1907.11692.

Loshchilov, Ilya and Frank Hutter (2019), Decoupled weight decay regularization. https://arxiv.org/abs/1711.05101.

Manjavacas, Enrique Arevalo and Lauren Fonteyn (2022a), Adapting vs. pre-training language models for historical languages, *Journal of Data Mining & Digital Humanities*. https://jdmdh.episciences.org/9152.

Manjavacas, Enrique Arevalo and Lauren Fonteyn (2022b), Non-parametric word sense disambiguation for historical languages, *in* Hämäläinen, Mika, Khalid Alnajjar, Niko Partanen, and Jack Rueter, editors, *Proceedings of the 2nd International Workshop on Natural Language Processing for Digital Humanities*, Association for Computational Linguistics, Taipei, Taiwan, pp. 123–134. https://aclanthology.org/2022.nlp4dh-1.16.

Nguyen, Dat Quoc, Thanh Vu, and Anh Tuan Nguyen (2020), Bertweet: A pre-trained language model for english tweets, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 9–14.

Ortiz Su'arez, Pedro Javier, Benoit Sagot, and Laurent Romary (2019), Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures, Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, Leibniz-Institut f"ur Deutsche Sprache, Mannheim, pp. 9 – 16. http://nbn-resolving.de/urn:nbn:de:bsz:mh39-90215.

Pfeiffer, Jonas, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder (2020), MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer, *in* Webber, Bonnie, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Online, pp. 7654–7673. https://aclanthology.org/2020.emnlp-main.617.

Qiu, Wenjun and Yang Xu (2022), Histbert: A pre-trained language model for diachronic lexical semantic analysis, *arXiv preprint arXiv:2202.03612*.

Remy, François, Pieter Delobelle, Bettina Berendt, Kris Demuynck, and Thomas Demeester (2023), Tik-to-tok: Translating language models one token at a time: An embedding initialization strategy for efficient language adaptation. https://arxiv.org/abs/2310.03477.

Schulz, Sarah, Guy De Pauw, Orphée De Clercq, Bart Desmet, Veronique Hoste, Walter Daelemans, and Lieve Macken (2016), Multimodular text normalization of dutch user-generated content, *ACM Transactions on Intelligent Systems and Technology (TIST)* **7** (4), pp. 1–22, ACM New York, NY, USA.

Seabold, Skipper and Josef Perktold (2010), Statsmodels: econometric and statistical modeling with python., *SciPy* **7** (1), pp. 92–96.

Tang, Yuqing, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan (2020), Multilingual translation with extensible multilingual pretraining and finetuning. https://arxiv.org/abs/2008.00401.

van Cranenburgh, Andreas and Gertjan van Noord (2022), Openboek: A corpus of literary coreference and entities with an exploration of historical spelling normalization, *Computational Linguistics in the Netherlands Journal* **12**, pp. 235–251.

Van den Avenne, Julien (2024), Reynaerde-7b-instruct model card. https://huggingface.co/ReBatch/Reynaerde-7B-Instruct.

Van Hee, Cynthia, Els Lefever, and Véronique Hoste (2016), Exploring the realization of irony in twitter data, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 1794–1799.

Vanroy, Bram (2024a), Fietje: An open, efficient llm for dutch. https://arxiv.org/abs/2412.15450.

Vanroy, Bram (2024b), Geitje 7b ultra: A conversational model for dutch. https://arxiv.org/abs/2412.04092.

Wolters, Andre and Andreas Van Cranenburgh (2024), Historical dutch spelling normalization with pretrained language models, *Computational Linguistics in the Netherlands Journal* **13**, pp. 147–171. https://clinjournal.org/clinj/article/view/178.

Xiao, Zedian, William Held, Yanchen Liu, and Diyi Yang (2023), Task-agnostic low-rank adapters for unseen English dialects, *in* Bouamor, Houda, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Singapore, pp. 7857–7870. https://aclanthology.org/2023.emnlp-main.487/.

Xue, Linting, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel (2021), mT5: A massively multilingual pre-trained text-to-text transformer, *in* Toutanova, Kristina, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Online, pp. 483–498. https://aclanthology.org/2021.naacl-main.41.

Appendix A. Overview Model Pre-Training

dataset	model	adapter			cfmp		
		steps	eval_loss	perplexity	steps	eval_loss	perplexity
1	gysbert	35,000	2.6283	13.85	15,000	2.7153	15.10
	robbert	42,500	2.4920	12.08	52,500	2.1481	8.57
	bertje	60,000	2.8795	17.80	60,000	2.2191	9.20
2	gysbert	65,000	2.4849	11.99	70,000	2.4210	11.25
	robbert	57,500	2.3594	10.58	95,000	1.8092	6.11
	bertje	45,000	2.9464	19.04	45,000	2.1789	8.84
3	gysbert	80,000	2.7722	15.99	130,000	2.6461	14.10
	robbert	102,500	2.6414	14.03	167,500	1.9526	7.05
	bertje	82,500	3.2162	24.93	80,000	2.3470	10.45
4	gysbert	10,000	3.7720	29.28	10,000	3.2553	25.93
	robbert	107,500	2.8435	17.18	87,500	2.5003	12.19
	bertje	112,500	3.3907	29.69	77,500	2.6872	14.69

Table 8: Model Pre-Training Overview for Historical Dutch.

dataset	model	adapter			cfmp		
		steps	eval_loss	perplexity	steps	eval_loss	perplexity
1	robbert	27,500	2.3960	10.98	12,500	2.6537	14.21
	bertje	27,500	3.0442	20.99	15,000	3.0120	20.33
2	robbert	32,500	2.3537	10.52	10,000	2.6090	13.59
	bertje	82,500	2.9100	18.36	10,000	2.9459	19.03
3	robbert	62,500	2.2664	9.64	10,000	2.5736	13.11
	bertje	77,500	2.8550	17.37	10,000	2.9535	19.17
4	robbert	95,000	2.2544	9.53	10,000	2.5830	13.24
	bertje	62,500	2.8912	18.01	10,000	2.9640	19.38
5	robbert	135,000	2.2320	9.32	10,000	2.5953	13.36
	bertje	140,000	2.8091	16.59	10,000	2.9557	19.21

Table 9: Model Pre-Training Overview for Social Media Dutch.

Appendix B. Annotated Plays from EmDComF

title	year	genre	author
Klucht van de Bedrooge vryer	1672	Farce	Adriaan Boelens
Gulsigheydt	1682	Farce	Willem Ogier
De meid juffrouw	1685	Farce	Pieter de la Croix
Bellamante, ofte de minsiecke juffer	1689	Comedy	Anoniem
De spilpenning of verkwistende vrouw	1693	Comedy	Thomas Asselijn
De klugtige schoenlapper, of de nieuwe hondeslager	1702	Comedy	Cornelius Schrevelius
De bruiloft van Kloris en Roosje	1707	Farce	Dirk Buysero
Quincampoix of de windhandelaars	1720	Comedy	Pieter Langendijk
De driftige minnaars, of arglistige juffrouw	1723	Farce	Adriana van Rijndorp

Appendix C. Prompts for Generative Finetuning

In these prompts, we present the turn switches between user and systems with placeholders. As these tokens are different depending on the base model, we automatically replace them with the correct models during inference. Our prompt examples contain a single example per label, in practice, we made use of one example for each label, meaning 5 examples for emotion detection.

C.1 Historical - Emotion Detection

[System Prompt] You are specialized in recognizing emotions in historical texts. This means that you are able to identify and name the most prominent emotion that is expressed in a text. The possible labels are: 'neutral', 'joy', 'sadness', 'anger', 'fear' and 'desire'.
[User] Which emotion is expressed in this text? ### Text: En alzo verspild men zyn tyd en geld.
[System] ### Label: anger
[User] Which emotion is expressed in this text?
Text: Nu sal ik eerst de Vrede sluiten.
[System] ### Label:

C.2 Historical - Sentiment Analysis

[System Prompt] You are specialized in recognizing sentiment in historical texts. This means that you are able to identify and name the most prominent sentiment that is expressed in a text. The possible labels are: 'positive', 'neutral' and 'negative'.
[User] Which sentiment is expressed in this text? ### Text: En alzo verspild men zyn tyd en geld.
[System] ### Label: negative
[User] Which sentiment is expressed in this text?
Text: Nu sal ik eerst de Vrede sluiten.
[System] ### Label:

C.3 Social Media - Emotion Detection

[System Prompt] You are specialized in recognizing emotions in social media. This means that you are able to identify and name the most prominent emotion that is expressed in a text. The possible labels are: 'neutral', 'joy', 'love', 'sadness', 'anger' and 'fear'.
[User] Which emotion is expressed in this text? ### Text: nog 8 dagen tot ik @user en @user kan zien ;D heb er zo veel zin in
[System] ### Label: joy
[User] Which emotion is expressed in this text?
Text: @user wat is ze mooi geworden, geniet ze!
[System] ### Label:

C.4 Social Media - Irony Detection

[System Prompt] You are specialized in detecting irony and sarcasm in social media. This means that you are not only able to identify verbal irony and sarcasm but also situational irony. The possible labels are: 'This text is genuine and does not contain any irony or sarcasm.' and 'This text contains irony or sarcasm.'
[User] Does this text contain irony or sarcasm? ### Text: Altijd fijn, zon ram volle bus in de vroege ochtend kank wel van geniete
[System] ### Label: This text contains irony or sarcasm.
[User] Does this text contain irony or sarcasm?
Text: Op naar den haag voor een stadswandeling met school. Zoveel zin in! haha
[System] ### Label:

Appendix D. Detailed Fine-Tuning Results

task	method	split	model	F1	std
emo	adapt	1	bertje	0.4364	0.0268
emo	adapt	1	gysbert	0.4831	0.0166
emo	adapt	1	robbert	0.4206	0.0290
emo	cfmp	1	bertje	0.4234	0.0410
emo	cfmp	1	gysbert	0.4702	0.0278
emo	cfmp	1	robbert	0.4053	0.0415
emo	adapt	2	bertje	0.4175	0.0298
emo	adapt	2	gysbert	0.4974	0.0181
emo	adapt	2	robbert	0.4465	0.0257
emo	cfmp	2	bertje	0.4222	0.0205
emo	cfmp	2	gysbert	0.4956	0.0429
emo	cfmp	2	robbert	0.4154	0.0245
emo	adapt	3	bertje	0.4060	0.0101
emo	adapt	3	gysbert	0.4824	0.0182
emo	adapt	3	robbert	0.4007	0.0310
emo	cfmp	3	bertje	0.4063	0.0244
emo	cfmp	3	gysbert	0.4810	0.0150
emo	cfmp	3	robbert	0.3861	0.0425
emo	adapt	4	bertje	0.4056	0.0118
emo	adapt	4	gysbert	0.4729	0.0214
emo	adapt	4	robbert	0.3947	0.0361
emo	cfmp	4	bertje	0.3949	0.0333
emo	cfmp	4	gysbert	0.4733	0.0263
emo	cfmp	4	robbert	0.3591	0.0644
emo	base	-	bertje	0.3960	0.0075
emo	base	-	gysbert	0.4462	0.0421
emo	base	-	robbert	0.4202	0.0099
sent	adapt	1	bertje	0.6760	0.0043
sent	adapt	1	gysbert	0.7166	0.0099
sent	adapt	1	robbert	0.6775	0.0037
sent	cfmp	1	bertje	0.6802	0.0104
sent	cfmp	1	gysbert	0.7266	0.0026
sent	cfmp	1	robbert	0.6887	0.0072
sent	adapt	2	bertje	0.6715	0.0042
sent	adapt	2	gysbert	0.7165	0.0064
sent	adapt	2	robbert	0.6824	0.0048
sent	cfmp	2	bertje	0.6767	0.0095
sent	cfmp	2	gysbert	0.7257	0.0048
sent	cfmp	2	robbert	0.6958	0.0068
sent	adapt	3	bertje	0.6694	0.0060
sent	adapt	3	gysbert	0.7150	0.0076
sent	adapt	3	robbert	0.6869	0.0094
sent	cfmp	3	bertje	0.6581	0.0047
sent	cfmp	3	gysbert	0.7244	0.0069
sent	cfmp	3	robbert	0.6748	0.0146
sent	adapt	4	bertje	0.6665	0.0043
sent	adapt	4	gysbert	0.7117	0.0036
sent	adapt	4	robbert	0.6744	0.0066
sent	cfmp	4	bertje	0.6587	0.0061
sent	cfmp	4	gysbert	0.7246	0.0053
sent	cfmp	4	robbert	0.6608	0.0087
sent	base	-	bertje	0.6638	0.0051
sent	base	-	gysbert	0.6946	0.0118
sent	base	-	robbert	0.6651	0.0096

Table 10: Model Performance on Historical Dutch tasks, averaged across 6 random seeds.

task	method	split	model	F1	std
emo	adapt	1	bertje	0.3416	0.0366
emo	adapt	1	robbert	0.3418	0.0347
emo	cfmp	1	bertje	0.3843	0.0000
emo	cfmp	1	robbert	0.4135	0.0000
emo	adapt	2	bertje	0.3613	0.0398
emo	adapt	2	robbert	0.3390	0.0304
emo	cfmp	2	bertje	0.3660	0.0000
emo	cfmp	2	robbert	0.3554	0.0000
emo	adapt	3	bertje	0.3731	0.0186
emo	adapt	3	robbert	0.3567	0.0192
emo	cfmp	3	bertje	0.2690	0.0000
emo	cfmp	3	robbert	0.3786	0.0000
emo	adapt	4	bertje	0.3339	0.0601
emo	adapt	4	robbert	0.3905	0.0229
emo	cfmp	4	bertje	0.3059	0.0000
emo	cfmp	4	robbert	0.3222	0.0000
emo	adapt	5	bertje	0.3331	0.0461
emo	adapt	5	robbert	0.3870	0.0281
emo	cfmp	5	bertje	0.3695	0.0000
emo	cfmp	5	robbert	0.3416	0.0000
emo	base	-	bertje	0.2557	0.0137
emo	base	-	robbert	0.3297	0.0446
irony	adapt	1	bertje	0.7201	0.0141
irony	adapt	1	robbert	0.7221	0.0177
irony	cfmp	1	bertje	0.7137	0.0000
irony	cfmp	1	robbert	0.7118	0.0000
irony	adapt	2	bertje	0.7210	0.0134
irony	adapt	2	robbert	0.7233	0.0087
irony	cfmp	2	bertje	0.6918	0.0000
irony	cfmp	2	robbert	0.7148	0.0000
irony	adapt	3	bertje	0.7226	0.0125
irony	adapt	3	robbert	0.7257	0.0093
irony	cfmp	3	bertje	0.7239	0.0000
irony	cfmp	3	robbert	0.7315	0.0000
irony	adapt	4	bertje	0.7157	0.0101
irony	adapt	4	robbert	0.7177	0.0106
irony	cfmp	4	bertje	0.7210	0.0000
irony	cfmp	4	robbert	0.6977	0.0000
irony	adapt	5	bertje	0.7245	0.0118
irony	adapt	5	robbert	0.7258	0.0133
irony	cfmp	5	bertje	0.7131	0.0000
irony	cfmp	5	robbert	0.7099	0.0000
irony	base	-	bertje	0.7264	0.0072
irony	base	-	robbert	0.7161	0.0145

Table 11: Model Performance on Social Media Dutch tasks, averaged across 6 random seeds.