
Master Thesis

# Multitask Learning of Semantic Role Labeling and Named Entity Recognition for domain-specific documents from the Dutch East-India Company archives

Hannah Goossens

Supervisor Stella Verkijk & Piek Vossen
$2^{nd}$ reader Lucia Donatelli

*a thesis submitted in fulfillment of the requirements for the degree of*

**MA Linguistics**
(Text Mining)

**Vrije Universiteit Amsterdam**

Computational Linguistics and Text-Mining Lab
Department of Language and Communication
Faculty of Humanities

| | |
|---|---|
| Date | June 27, 2025 |
| Student number | 2678984 |
| Word count | 28 450 |

# Abstract

This thesis investigates the effect of Multitask Learning (MTL) on Semantic Role Labeling (SRL) using annotated documents from the Dutch East-India Company (VOC) archives, written in Early Modern Dutch. Several Transformer-based models are evaluated to determine whether an MTL framework, jointly training for SRL and Named Entity Recognition and Classification (NERC), improves SRL performance compared to single-task finetuning for SRL. The findings reveal that certain models, in this case multilingual BERT, can benefit from an MTL configuration, while a domain-specific pre-trained model (GloBERTise) achieves the highest SRL scores despite not benefiting from MTL. The findings highlight the challenges of working with a small, domain-specific, and structurally complex dataset and the difficulties in optimizing the MTL approach. The research emphasizes the need for alternative modeling approaches, that is, different strategies or architectures beyond the basic MTL implementation explored in this thesis, and a more substantial annotated corpus to advance SRL in this domain.

# Declaration of Authorship

I, author, declare that this thesis, titled *title* and the work presented in it are my own. I confirm that:

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

Date: <date>

Signed: <student signature>

# Acknowledgments

# List of Figures

# List of Tables

# Contents

# Chapter 1

# Introduction

In this chapter, the thesis project will be introduced. The project is part of the larger project (GLOBALISE) of building a search engine for the Dutch East-India (VOC) company archives, which is explained further in Section 1.1. This thesis project tries to enable the extraction of arguments of events through semantic role labeling, thereby supporting event-centered retrieval functionalities. This is described in more detail in Section 1.2. The extraction of event arguments facilitates the ability to query the archives to, for example, track the comings and goings of a particular VOC ship. The challenges that arise with working with the VOC archives are explained in Section 1.3. To address these challenges and move toward the broader goal, multitask learning is of NERC and SRL is implemented. The approach is motivated by the proven effectiveness of multitask learning in settings with limited labeled data. Further details on this approaches are provided in Section 1.4.

## 1.1 The GLOBALISE project

This thesis is part of the GLOBALISE project by the Huygens Institute. This project is centered around the UNESCO Memory of the World-listed Dutch East India Company (VOC) archives. Note, throughout this thesis, the abbreviation 'VOC' will further be used throughout this thesis to refer to the Dutch East-India Company. In the 17th and 18th centuries, the VOC was the largest trade company of the world, establishing a vast colonial empire by controlling global maritime shipping routes through settlements across Asia. Its influence extended beyond economic power; through practices such as slavery and warfare, the VOC exerted significant colonial dominance. As a result, their legacy not only shaped Dutch history, but also deeply impacted the histories of many Asian regions.

The VOC archives preserve this complex and far-reaching history through an extensive collection of handwritten documents in Early-Modern Dutch. However, the substantial size of these handwritten documents, written in an outdated form of the Dutch language and only accessible by manually reading the (scans of) documents, makes the information not that easily accessible. As a consequence, forming a complete understanding of this past is difficult and critical insights risk being overlooked.

To address this, the GLOBALISE project is set up with the goal to develop an easily accessible research platform for the VOC archives. By applying language technology, the project digitizes the documents to extract and label relevant information in these documents. Language models are fine-tuned to interpret and represent this information.

In due course, this can enable the reconstruction of multifaceted, and accurate historical narratives.

## 1.2 Information extraction

The GLOBALISE project focuses on extracting named entities, events, and their participants from the documents. Named entities refer to specific semantic categories, such as person or location, which provides concrete and identifiable references to real-world entities, making understanding and analysis of the text easier. Additionally, events denote specific occurrences or incidents, which typically concerns specific participants. This information can help to reconstruct what was happening at a given moment in time and who was involved. To extract and label this relevant information, the project focuses on several tasks, each targeting a specific type of information. This approach aims to organize the data into structured and comprehensible formats.
So far, the primary focus has been on Named Entity Recognition and Classification (NERC) and Event Recognition (ER). Although NERC and ER are established and well-studied tasks within the area of NLP, given the highly domain-specific nature of the VOC data, these tasks need to be adapted accordingly. In particular, the structure and content of the VOC data differ significantly from those of modern datasets, such as those based on Wikipedia or contemporary news articles. Hence, to investigate NERC and ER, the project established domain-specific named entity tags and created a new VOC-specific event ontology that defines identifiable events and their participants within the data.

Building on this foundation, Semantic Role Labeling (SRL) is the next logical task to investigate. SRL aims to identify the core components, also known as event arguments or semantic roles, of an event, which express the who, when, where, and what of an event. This is achieved by determining which words fulfill these meaningful roles in the context of a specific event in a text. This capability is essential for accessing detailed event-specific information within the VOC archives.

## 1.3 Challenges

As mentioned in Section 1.1, the VOC documents are written in Early-Modern Dutch, which presents several challenges. The VOC texts lack standardized spelling and structure, with varying handwriting causing difficulty in digitizing the documents. Moreover, long, unstructured paragraphs create unclear sentence boundaries and long-range references, making information extraction tasks particularly difficult. Moreover, manually extracting and annotating this data is time-consuming, and requires domain-specific expertise, which means the current research works with a limited amount of labeled data.
Current results in ER and NERC for the VOC data reveal that language models struggle to adapt to the domain through fine-tuning, likely due in part to the scarcity of labeled data. A promising approach to address this issue is multitask learning, which enables a model to share and transfer knowledge across related tasks, potentially improving overall performance.

## 1.4 Multitask learning

Multitask learning; fine-tuning a model on multiple tasks simultaneously, has been shown to enhance model performance by leveraging shared knowledge across tasks, especially in contexts with limited annotated data (Marasović and Frank, 2017). Ikhwantri et al. (2018) apply this approach by using NERC as the auxiliary task of SRL for the method of multitask active learning. NERC is a prominent task in information extraction, helping to identify entities such as participants and locations (Abreu and Oliveira, 2018), which often overlap semantically with the labels of SRL. To illustrate, in the VOC data, the named entities *'Ship-type'*, *'Location'*, and *'Date'* can fulfill the roles of *'Agent'*, *'Location'*, and *'Time'* of an *'Arriving'* event. By incorporating these types of associations into a model, this thesis project aims to contribute to the development of systems that can effectively process and retrieve information from the VOC archives.

## 1.5 Research objective

This background and research objective have led to the formulation of the following problem definition and corresponding specific sub-questions:

**Research Question**

When working in a low-resource, domain-specific scenario, does multitask learning with NERC and SRL data increase performance of pre-trained Transformer models, compared to solely fine-tuning on SRL?

**Sub Questions**

1) Does MTL with NERC and SRL data improve SRL performance for texts from the VOC archives, written in Early-Modern Dutch?

2) Which pre-trained Transformer model performs best for the task of SRL with and without MTL?

These sub-questions aim to evaluate the impact of multitask learning on semantic role labeling performance in a low-resource, domain-specific setting. Additionally, they assess the suitability of different pre-trained Transformer models for processing VOC texts. These insights can aid in the development of robust tools for domain-specific archival research.

In summary, the research objective aims to guide a structured investigation of how multitask learning can enhance semantic role labeling with limited annotated data. By assessing the role of NERC as a supporting task and comparing different transformer-based language models, this thesis seeks to provide empirical insights into the use of language technologies for text processing, particularly to improve event-centered information retrieval within the VOC archives.

# Chapter 2

# Related Work

This chapter will give an overview of the most relevant literature in the areas of named entity recognition, event extraction, and semantic role labeling. It also provides a reasoning and description of the model architectures that are used. Furthermore, this chapter surveys the discourse around multitask learning to highlight the reason for the choice of using this approach.

## 2.1 Natural Language Processing and Information Extraction

When working with text data, Natural Language Processing (NLP) has shown to be useful in a number of ways. The goal of NLP is to allow computer systems to simulate the way humans read and understand language (Beysolow II, 2018). This means, for example, for a system to be able to extract the sentiment from a text (Gunasekaran, 2023), or to produce answers to specific questions about a text (Arbaaeen and Shah, 2020). The field of NLP is quickly advancing, using AI techniques (Beysolow II, 2018). To obtain information from text data, NLP can be implemented for Information Extraction (IE). As written texts contain many types of information, different tasks exist to obtain this information. IE is essentially an overarching term of various tasks that all deal with a specific type of information (Wang et al., 2024). The GLOBALISE project focuses on the IE tasks of Named Entity Recognition and Classification (NERC), Event Recognition (ER), and Semantic Role Labeling (SRL).

## 2.2 NERC

Named Entity Recognition and Classification (NERC) is an important NLP task which has been an active area of study for many years (Sharnagat, 2014). NERC entails the identification of words or phrases that refer to predefined semantic categories, like location or person. NERC is used as the base of various NLP applications and is a useful instrument in the case of information extraction (Li et al., 2020). Categories of entities can be split into two types: generic and domain-specific. Entities such as person or location can be seen as more generic entities that will be referred to in general data. However, in the biological domain, for instance, genes and enzymes are entities of interest (Li et al., 2020). In the case of the VOC archives, there are also

domain-specific entities that convey important information and need to be specified. Apart from person, location, and organization, the entities religion and ship are also part of the entities in the VOC texts (Arnoult et al., 2021). Therefore, general models or general datasets of NERC cannot be used to train the models for the VOC data. Arnoult et al. (2021) created a new Dutch NERC dataset based on the VOC documents. Both monolingual (BERTje, RobBERT) and multilingual models (mBERT, XLM-R) were fine-tuned on NERC with this dataset. Their results show that on average, the multilingual models reach higher performance, though when faced with data containing more semantic variance, the monolingual models perform relatively better.

## 2.3 Event Recognition

Another relevant task for information extraction is Event Recognition (ER). ER provides a structured way of handling the stream of information in text, by identifying mentions of events in text. The Automatic Content Extraction (ACE) dataset, which was created with the goal of motivating research in information extraction, is frequently used to train and build models for general event recognition ((Chen et al., 2019); (Yang et al., 2019)). The ACE characterizes an event as "a specific occurrence involving participants" Doddington et al. (2004). To recognize an instance of an event, triggers, also called anchor words, are used. These anchor words often take on the parts of speech of a verb or a noun Doddington et al. (2004). The ACE's general definition of events is, however, not necessarily sufficient when implementing ER for specific data (Ghidalia et al., 2024).
In the domain-specific case of the VOC data, the Globalise project defines the event classes based on three themes: ship movement, trade and (geo) political/social relations. This is done to keep the task of information extraction relevant and manageable (Verkijk and Vossen, 2023). A way to specify what exactly qualifies as an event, so that these can be identified in the data, is by establishing an ontology. Computational ontologies offer a structured method for defining and organizing key concepts within a specific domain, enabling structured data formatting that supports machine learning applications (Ghidalia et al., 2024). The base of an ontology is a taxonomy, which concerns a classification of concepts. Relations are defined for these concepts, where specific entities can hold a certain relation with each other (Studer and Staab, 2004). Thus, an ontology is a structured organization of the principal concepts or objects that occur in a certain domain, including all objects, connections, and acts necessary to characterize the objects and identify their actions (Chen et al., 2020). Existing ontologies can be utilized, such as the more broad ontologies: the Suggested Upper Merged Ontology (SUMO) (Niles and Pease, 2001) and FrameNet (Baker et al., 1998). However, these attempt to represent the entire world with extremely fine-grained detail, which can reduce flexibility and ease of understanding. Therefore, a new VOC-specific event ontology was designed (Verkijk and Vossen, 2023). This ontology lists all the events that are relevant for the dataset used in this thesis, together with the event arguments. These event arguments will be discussed in more detail in Section 2.4.
The VOC-specific event ontology supports the task of event reconstruction, which is designated as an NLP task in the GLOBALISE project. The task of Event Reconstruction is based on Event Recognition, as described in the previous paragraph. However, in addition to explicit mentions of events, this task focuses on implicit events as well. Furthermore, next to implied mentions related to an explicit mention, Event Recon-

struction also extracts implied events that are not directly linked to a mention but can be logically inferred from one. More specifically, these events can be deduced as a logical consequence of a specific referenced event. The ontology enables this reasoning to extract and model these explicit and implied events. The ontology will be described in more detail in Chapter 3. In the paper by Verkijk et al. (2024) the annotation process of the VOC data based on the ontology and analysis of the results are described. This annotation process consisted of the annotation of all types of events and its arguments, according to the concepts defined in the event ontology. The paper also describes how two language models are fine-tuned on event trigger detection using VOC data, serving as baseline experiments for VOC event search automation. The models implemented are XLM-R, a multilingual RoBERTa model (Conneau et al., 2019), and GysBERT, a Dutch Historical Language model (Manjavacas and Fonteyn, 2022). The results indicate both models can be applicable for different objectives. However, the research is ongoing and state-of-the-art results, compared to contemporary English event recognition ((Hong et al., 2018);(Li et al., 2022)), are not yet established. However, as reported by Hong et al. (2018), outside of English event recognition, performance can drop significantly when applied to unfamiliar or slightly shifted domains. As the GLOBALISE project works with data written in Early-Modern Dutch, this also comes with a higher level of syntactic and linguistic complexity, which is further explained in Chapter 3. Though these findings may explain the increased the difficulty of achieving similar performance levels in this project, improvement is still possible. The ontology and baseline experiments provide resources and information for further research in event reconstruction and serves as the foundation of the accompanying task of semantic role labeling, i.e. identifying the event arguments, which is the focus task of this thesis.

## 2.4 Semantic role labeling

Revisiting the ACE definition of an event, an event is stated to be accompanied by participants (Doddington et al., 2004). Apart from participants, events can also have properties or attributes, which together are called event arguments. In general, these arguments are the who, where, when, and with what of an event. The task of detecting these arguments can be accomplished by performing Semantic Role Labeling (SRL), which is an NLP task which involves the identification of roles that words represent in a given sentence. For a given predicate, which describes a certain event, SRL determines the semantic roles that the arguments of a predicate fulfill in that event (Jurafsky & Martin, 2024).

There are many different roles an argument can portray. A set of commonly agreed upon roles is shown in Figure 2.1. However, a formal definition or universal set of these roles is not defined, due to the fact that not every characteristic of a role is displayed by the word representing that role. For instance, an *AGENT* is usually characterized as animate, volitional, sentient, causal, but not all of these properties apply in every case. Therefore, different ways to create a specific set of these semantic roles are constructed. One method involves defining semantic roles in relation to a particular verb or a set of verbs or nouns that share similar meanings. Another approach is to establish a set of broad semantic roles which generalize over the specified thematic roles, also known as proto-roles. Two lexical resources that are generally utilized and which employ the two described directions are FrameNet (Baker et al., 1998) and PropBank (Kingsbury and Palmer, 2002). FrameNet works with frames; semantic schemes which semantic

| Thematic Role | Definition |
|---|---|
| AGENT | The volitional causer of an event |
| EXPERIENCER | The experiencer of an event |
| FORCE | The non-volitional causer of the event |
| THEME | The participant most directly affected by an event |
| RESULT | The end product of an event |
| CONTENT | The proposition or content of a propositional event |
| INSTRUMENT | An instrument used in an event |
| BENEFICIARY | The beneficiary of an event |
| SOURCE | The origin of the object of a transfer event |
| GOAL | The destination of an object of a transfer event |

Figure 2.1: General thematic role examples
Adapted from Jurafsky & Martin (2025)

roles can be specific to. This is a semantics-based strategy. PropBank is syntax-based, where the arguments are grounded in the syntactic structure of the sentence. PropBank applies both the method of roles that are specific to verbs as well as proto-roles. The semantic roles are based on an individual verb sense. Every sense of a verb (the different meanings a verb can portray) holds a separate group of roles (Jurafsky and Martin, 2025).
As discussed in Section 2.3, in addition to the events, the event arguments are also represented in the VOC-specific event ontology. These arguments are defined based on the linguistic reasoning of PropBank regarding the recyclability of roles, meaning they are suitable for any sentence describing a given event. However, the arugments are defined within semantic concepts such as *AGENT* and *PATIENT*, regardless of the syntax of a sentence. The event ontology thus provides an accurate reconstruction of all events and the accompanying arguments that are relevant in the VOC data. This ontology was used for the manual annotation process of VOC data. This resulted in an annotated dataset of both events and arguments, which provides the possibility of training models on semantic role labeling as well. However, no models have yet been trained and fine-tuned using the VOC data for semantic role labeling. A more in-depth description of these annotated data, that is used for fine-tuning in this thesis, will be provided in Chapter 3

## 2.5 Multitask learning

As mentioned, the tasks of Event Recognition (ER), Semantic Role Labeling (SRL), and Named Entity Recognition and Classification (NERC) are fundamental tasks in information extraction (Wang et al., 2024). However, they are often isolated in structure, datasets, and models (Zhu et al., 2023). This means that enhancing a model for a specific task is usually done separately, where a separate model is trained for each task, each using a distinct labeled dataset. These deep learning techniques reach high performance on various supervised learning tasks. Supervised learning is the method of training an algorithm using labeled data (Nasteski, 2017). However, although these very high performing systems have been developed for tasks like ER, SRL, and NERC, these systems often do not work on a different dataset or domain for the same task. This is due to the fact that, conventionally, vast amounts of labeled datasets are re-

quired to obtain comparable results for any domain. This makes it difficult to apply in cases where a large, labeled dataset is hard to obtain, i.e., time consuming, expensive, requiring expertise. This prevents the application of deep learning algorithms in low-resource settings or to novel tasks (Chen et al., 2023). In the case of the VOC data, the annotation process is ongoing, and extraordinary complex due to language, age, and context that are less well-known, resulting in limited availability of labeled data.
Research investigating this issue adopts different approaches to reduce the need for large labeled datasets, with strategies such as semi-supervised learning, unsupervised pre-training, and data augmentation (Chen et al., 2023). Another approach consists of boosting model performance by leveraging knowledge from multiple tasks, mitigating the need for vast amounts of data. This technique is called Multitask Learning (MTL). MTL is a paradigm of knowledge transfer aimed at improving a model's ability to generalize (Caruana, 1998). Rather than training tasks independently, Multitask Learning allows a model to develop a shared internal representation, enabling the transfer of inductive knowledge across tasks. By drawing on task-specific information from multiple sources, MTL introduces helpful inductive biases that guide the model's learning process. These biases facilitate more effective learning by guiding the model toward representations that are beneficial across tasks and improve the model's ability to handle unseen data (Caruana, 1998). Before deep learning, MTL revealed improved results when implemented for decision tree structures, k-nearest neighbour models, and Bayesian multitask learning (Caruana, 1998). With the uprising of deep learning, MTL is applied in broad and domain-specific settings, and in various fields, like NLP (Collobert and Weston (2008); Søgaard and Goldberg (2016)).
As for the information extraction tasks, various studies have combined SRL and NERC, illustrating how insights from these tasks can enhance each other. In particular, NERC can be implemented to identify for example participants and locations (Abreu and Oliveira, 2018) which share semantic information with various labels of SRL. For example, the named entities of *DATE*, *LOCATION*, and *PERSON*, can fulfill the roles of *TIME*, *LOCATION*, and *AGENT* or *PATIENT* in a specific text. Named Entity labels can also be used as a feature for SRL (Jurafsky & Martin, 2025) and Xiang and Li (2021) have incorporated entities in an event/participant detection task. This shared informativeness gives way to possible MTL strategies.
When looking at NERC and SRL for MTL, there are multiple ways to approach this. One method, like the study by Crichton et al. (2017), implements single-task and multitask learning for several models. Multitask learning was applied using multiple datasets where every dataset designates a task. The performance of the multitask models compared with that of single-task models shows that on average multitask learning results in better performance. Moreover, the results indicate that multitask learning is advantageous for small datasets as the performance of multitask models increased when dataset size decreased. A different approach is that of Marasović and Frank (2017), who worked with limited annotated training data for the task of labeling of opinion holders and targets (ORL). They tackle this issue by using SRL data to inform the less-represented task of ORL by combining them in a multitask learning (MTL) scheme. They apply multiple MTL techniques which vary in how many layers of their neural model are shared between tasks. The fully-shared model, where all parameters are shared apart from the output layer, and the hierarchical model, where higher-level tasks build on representations learned from lower-level tasks, but not vice versa, result in notable improvements. Similar to this, Ikhwantri et al. (2018) approach the task of

SRL with a multitask active learning method by adding NERC as an auxiliary task. This framework also aims to reduce the necessity of a large dataset and to leverage information from NERC to aid in the task of SRL. The results show how the multitask active learning approach outperforms single-task active learning.
There is thus enough work that lays out the benefits of Multitask Learning (MTL), particularly in scenarios with limited labeled data. However, the specific combinations of language models, domains, and tasks explored in existing studies differ from the framework of this thesis. While, as described, studies focus on SRL and NERC, they frequently do so using different model architectures, apply MTL for different domains, or use multiple distinct datasets (Marasović and Frank, 2017); Collobert and Weston (2008); Zhao et al. (2019); Peng et al., (2020); Crichton et al. (2017); Chen et al., (2024)).

## 2.6 Transformer models

As mentioned in Chapter 1, this thesis focuses on Transformer-based models. More specifically, the models that are implemented are BERT- and RoBERTa-based language models. Since, in recent years, state-of-the-art performance on information extraction tasks such as event recognition, semantic role labeling, and named entity recognition has primarily been achieved by BERT- and RoBERTa-based models ((Min et al., 2023); (Pakhale, 2023); (Chen et al., 2025).
BERT is short for 'Bidirectional Encoder Representations from Transformers' (Devlin et al., 2019). A Transformer model is a type of neural network framework created to work with sequential information. Unlike earlier methods like Recurrent Neural Networks (RNNs) or Long Short-Term Memory networks (LSTMs), Transformers completely remove the use of recurrence, i.e. where sequences are processed one element at a time and the information of previous input is used for current input. Instead, it uses a mechanism known as self-attention to handle the input all at once Vaswani et al. (2017). Together with the bidirectional architecture, where both the left and right contexts are incorporated when processing text, the model can establish extensive representations by means of pre-training with unlabeled text data, also called self supervised learning. This approach lets the model take the full context of a sequence into account simultaneously, which makes it better at recognizing and learning relationships between distant elements within the sequence. A language model is usually build based on specific learning objectives. These objectives define specific challenges a model is designed to solve at the beginning of its development. The learning/training objectives play a vital role in establishing a solid understanding of language patterns Alajrami and Aletras (2022). The BERT model uses two pre-training objectives: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In the MLM objective, certain tokens from an input text are arbitrarily masked. The goal is for the model to predict the correct token for this masked word, enabling the pre-training of a deep bidirectional model. The second objective, NSP, is implemented with the goal of training a model that recognizes sentence relationships. In this task, the model is shown sentence pairs, which are taken from a large text corpus. The model has to decide whether the second sentence is a logical follow-up based on the first sentence. This training then lets the model capture relationships and connections between textual input. This design enables the pre-trained BERT model to be easily adapted for specific tasks, also known as

the process of fine-tuning, by adding one extra output layer. Typically, this involves appending a single linear layer that maps BERT's output to the set of task-specific labels, with input and output aligned accordingly. This makes the model highly suitable to be fine-tuned to a variety of NLP applications, like NERC and SRL (Devlin et al., 2019).

Liu et al. (2019) created RoBERTa (Robustly optimized BERT approach), based on the studies of BERT. RoBERTa is a refinement of the BERT architecture. The approach for optimizing BERT consists of elongating the training process and using larger batches across more data. Larger batches and sequences reduce uncertainty and boosts task-accuracy. Furthermore, the Next Sentence Prediction (NSP) task is removed as a pre-training objective. The Masked Language Modeling (MLM) is still used, but some changes were made to the objective, by means of dynamic masking. For the MLM objective in BERT, the masking is static, which means the same words are repeatedly masked during training. This may cause the model to memorize these specific positions and patterns, instead of developing broader, more generalizable language understanding. The dynamic masking that is implemented in RoBERTa models tries to encourage the ability to actually grasp the more robust relationships in language by masking different (subsets of) words in each epoch. This results in state-of-the-art performance across different tasks (Liu et al., 2019).

## 2.7 Conclusion

The MTL implementation in this thesis, using Transformer-based models on a domain-specific VOC dataset annotated with both SRL and NERC labels, draws from the positive findings across these prior works described in Section 4.3.2. Although none of the studies directly replicate this configuration, they each support aspects of the design choices made here. This thesis, therefore, combines these elements to form a novel framework that investigates the impact of MTL on SRL using NERC as an auxiliary task in the context of the VOC archives implemented for BERT-, and RoBERTa-based models.

# Chapter 3

# Data

In this chapter, a detailed description of the annotated dataset, developed by the GLOBALISE project, is provided. It outlines the selection criteria and annotation process applied during dataset construction. In addition, an overview of the content of the data set is described, including the label categories and their distribution.

## 3.1 Data collection

For this project, data annotated by the GLOBALISE project is used. As described, the goal of the GLOBALISE project is to create software that enables researchers to search through the archives of the Dutch East-India Company (VOC). They focus on a specific corpus in the archives, namely the collection of Overgekomen Brieven en Papieren (OBP) or Received Letters and Papers. This collection contains the Generale Missiven (General Missives), which report narratives of historical events, along with accounts of the societal structures, governance, economic systems, and environmental conditions in that period. These reports are the core series of VOC correspondence and official papers sent from Batavia, currently Jakarta in Indonesia, to the Dutch Republic. The accounts cover the period 1610-1796 and consist of about 7 million pages. For the overall research objective of GLOBALISE, an annotated dataset had to be created. To achieve this, a selection of pages was taken from the described collection, ensuring variety in years and types of documents (letters, journals, notes, missives) spanning the years 1618 - 1781. The Handwritten Text Recognition (HTR) tool Loghi is used to digitize the data (see `https://github.com/knaw-huc/loghi`). This specific collection of documents comprises the VOC dataset that will further be referred to in this thesis.

### 3.1.1 Sentence boundaries

In the time of the VOC, the way of using writing systems differed in some ways from how we use it now and there were no spelling conventions or rules that everyone adhered to. Also, the corpus is written by many different people with varying handwritings over the years (and per person). Due to this variety in writing style and lack of structure, the output by the HTR system, i.e. the data to work with, can be noisy. Of main importance is the fact that the texts are written in long paragraphs with no clear starting and endpoints. This leads to unclear sentence boundaries, which results in

long-range dependencies in the texts: words sometimes refer to an entity mentioned pages back, and event arguments can belong to an event introduced much earlier in the document. Hence, references are not as clear as in modern text. Therefore, a decision had to be made on how to handle the texts to interpret them for information extraction. Instead of sentence boundaries, the documents are organized by working with text regions. These regions consist of paragraphs which are based on the layout of the pages. For example, if a large empty space is present on the page, this is interpreted as the end of a region. This variety and structure, or lack thereof, adds to the difficulty working with this specific data.

## 3.2 Annotation process

Preliminary experiments implementing existing tools and resources for an automatic annotation process led to unfavorable results (Verkijk et al., 2024). Therefore, the decision was made to design a new annotation scheme, which accurately reflected the extraction of relevant information as determined by the research objective of GLOBALISE. The event ontology that was designed specifically for working with the VOC data, as mentioned in Chapter 2, outlines the relevant information to be extracted from the data. Therefore, the new annotation scheme was guided by this event ontology. During the annotation process, the annotators, expert historians, identified event triggers as well as the arguments (semantic roles) of these events, following the guidelines specified in an event wiki (See Wiki at `https://github.com/globalise-huygens/nlp-event-detection/wiki`), based on the event ontology. As described in Chapter 2, the event ontology can be divided into three themes: ship movement, trade, and (geo)political/social relations. Both dynamic events (actions or processes), e.g. 'Buying', and static events (states or conditions), e.g. 'HavingInPossession', are captured. The shift from one state to another is treated as a logical consequence that can be deduced from the presence of a dynamic event. Challenges with annotating implicit references are resolved by using two types of references: 'isOfType' for explicit mentions, and 'evokes' for implicit ones. For example, the trigger *'left his job'* isOfType *'LeavingAnOrganization'* with direct reference. However, from the trigger *'fired'*, it can be inferred that someone will leave the organization. This therefore 'evokes' the connection between the trigger and the event class. In addition, annotation guidelines were also created for the named entities in these documents (see Wiki at `https://github.com/globalise-huygens/annotation/blob/main/guidelines/ner-guidelines.md`). These were token-based annotations based on the pre-defined named entity classes, see Table 3.2.
There were multiple annotation rounds with eight annotators. The first round was individual, which led to poor results. There was a low agreement in detecting event triggers, where annotators often missed triggers that others did annotate. Therefore, the second annotation round introduced a 'check-task' in which annotators reviewed the annotations of the others. Here, agreement went up, as the annotators agreed with event triggers they themselves initially missed. Based on these improved results, the following annotation rounds were carried out in four teams of two, opening up room for discussion and correction.

## 3.3 The annotated dataset

The annotation process resulted in the gold annotations of events, participants, and entities. The annotated dataset currently consists of 16 documents. These documents differ greatly from each other with respect to size and class distribution. The number of text regions in each document range from 12 to 296, with between 1178 to 78129 tokens. Moreover, not every document contains every class or with the same frequency. There is also a difference between the number of meaningful SRL and NE labels that are present in the documents. On average, there are 6.7 times as many meaningful NE tags as SRL tags in each document. This difference in label frequency highlights the contrast between the two tasks in terms of difficulty and learnability. NE labels occur more frequently in a text region, indicating the fewer constraints on what can be labeled as a named entity. Moreover, the higher number of label instances provides the model with more training signals. This supports the common view that named entity recognition and classification (NERC) is a simpler information extraction task compared to Semantic Role Labeling (SRL). Therefore, we expect that NERC can function as an auxiliary task of SRL, serving as the lower-level task that supports and informs the more complex, higher-level task of SRL, as demonstrated by Marosivc & Frank (2018).

### 3.3.1 The labels

Given the noisy and fragmented nature of the raw data, the ontology for event arguments emphasizes on intuitive, broadly applicable roles that can be assigned to most events. As shown in Table 3.1, there are 11 categories. Most roles correspond to general roles of semantic role labeling, such as *AGENT*, *PATIENT*, or *INSTRUMENT*. Some roles are more specific, such as *AGENTPATIENT*, which is an entity that is both the causer and undergoer of an event. For example in a BeginningContractualAgreement event, where *AGENTPATIENT* is the person or organization who begins an agreement described in a contract. Every role and its explanation are provided in Figure 1 in the Appendix.

| Participants | Spatial | Temporal |
|---|---|---|
| Agent | Location | Time |
| Patient | Source | |
| AgentPatient | Target | |
| Beneficiary | Path | |
| Cargo | | |
| Instrument | | |

Table 3.1: Semantic roles / event arguments
Note: adapted from (Verkijk et al., 2024)

The roles can be divided into two categories: the participants, which are essential to the semantics of an event, and satellites, which are not necessary but do provide

useful information about an event. Participants of an event can be both animate or inanimate entities. Looking at Table 3.1, the satellites are usually the temporal and spatial entities, which report the where and when of an event. However, in certain cases, location and time are essential arguments, namely in the case of translocation events, e.g., *'Leaving'* or *'Voyage'*. Moreover, it is not the case that one instance of an argument class exists per event instance. Any argument can exist more than once for a specific event. For instance, in a *'Trade'* event, there are two agents, namely the two entities who are trading something with each other.

There are 15 named entity categories, see Table 3.2. These consist of more general named entities, like person or location, but also domain specific ones, such as ship, ship-type, ethno-religious attributes, and (civic) status. The complete list of definitions and examples is provided in Figure 2 in the Appendix.

| Persons | Commodities | Ships | Else |
|---|---|---|---|
| PER_NAME | CMTY_NAME | SHIP | LOC_NAME |
| PRF | CMTY_QUAL | SHIP_TYPE | LOC_ADJ |
| STATUS | CMTY_QUANT | | ETH_REL |
| PER_ATTR | | | ORG |
| | | | DATE |
| | | | DOC |

Table 3.2: NERC labels

### 3.3.2 Overlap NE and SRL labels

As mentioned in Chapter 2, based on the semantic similarity that exists between the Named Entity (NE) and Semantic Role (SR) labels, the NE labels can provide relevant information for detecting the SR classes. As there are more NE classes than SR classes and the NE labels appear more frequently in the data, not every NE class overlaps with an SR class. However, there are informative classes across the two tasks. This is illustrated in Figure 3.1, showing the classes with more than 15% overlap between an SR and an NE class. To identify these overlaps, all tokens that both represented a semantic role and a named entity in a text region were extracted. The BIO prefixes (*B-* and *I-*) were removed to simplify the labels. Additionally, the total frequency of each semantic role and named entity class was calculated. The proportion of overlap for each pair was then computed by dividing the number of times a named entity occurred within a specific semantic role by the total number of times that semantic role occurred in the dataset. These percentages indicate how strongly certain named entities are associated with specific semantic roles. The code can be found in this github (). As shown, the largest overlap is between *TIME* and *DATE*, for which more than half of the instances overlap. Moreover, the NE class of *LOC_NAME* is informative for multiple classes. The participant classes *AGENT* and *PATIENT* also reveal overlap with informative NE labels. However, most overlap seems to occur for satellites, i.e. the spatial and temporal classes. These are also less frequent in the data, which could

result in an advantage of the multitask learning setup. That is, the overlap with the NE labels could provide additional information that is otherwise missing in the SRL data.

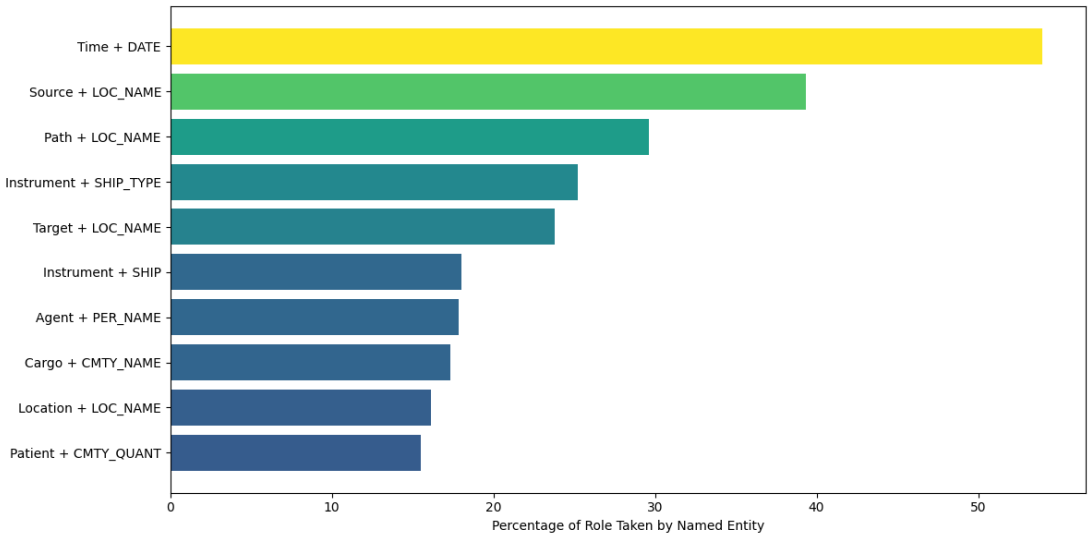



Figure 3.1: Top 10 percentual overlap of SRL and NE classes

### 3.3.3 BIO tagging

An instance of a semantic role or named entity can span across multiple tokens. For example, the three tokens in '*on December 10th*' form the single semantic role mention of *TIME* for a specific event. To structure these spans, the labels are tagged using the BIO (Beginning, Inside, Outside) format. This means that each token in a span, i.e. the collection of tokens that all belong to the same semantic role instance, is provided with a label based on their position. The tokens that are part of a semantic role receive *B-* (Beginning) when the token is the first of the span, *I-* (Inside) for tokens inside the span, and the label *O* (Outside) is given to tokens that are not part of a labeled span. For example, the name '*Cornelis Geuertsz*' consists of two tokens and takes the argument *PATIENT* in a specific region. '*Cornelis*' is tagged with *B-Patient*, and '*Geuertsz*' with *I-Patient*. Semantic roles that only consist of one token also receive the *B-* label. As a result, the set of semantic roles comprises 23 classes in total (including the *O* label). Similarly, the named entity tags now include 31 distinct labels. The use of the BIO tagging scheme also has implications for the evaluation of the predictions. That is, when the correct class is predicted, but not the correct BIO tag, e.g. *I-Agent* instead of *B-Agent*, this is valued as an incorrect prediction.

### 3.3.4 Duplicate text regions

As mentioned in Section 3.1.1, the documents are divided into text regions and not sentence boundaries. These text regions often contain several event instances. The annotated documents therefore contain many duplicate text regions. The semantic role annotation goes per predicate, meaning that only one predicate is focused on at a time. So, for each predicate in a text region, the region is duplicated, where in each

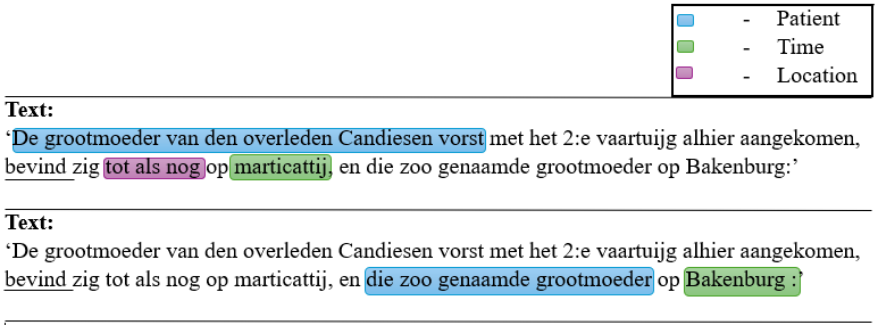


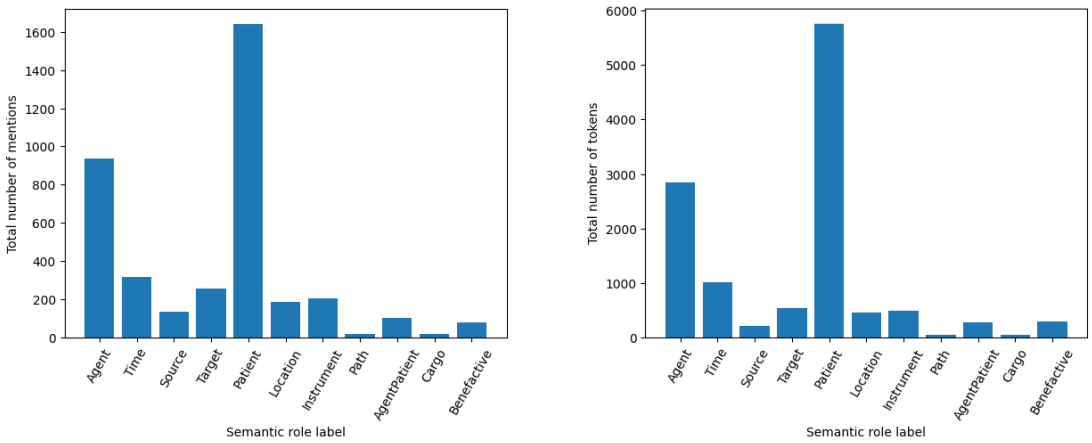

Figure 3.2: Duplicate text regions with same predicate but different roles
*Transliteration: 'The grandmother of the deceased Candese king with the 2nd ship here arrived, is located to date still on marticattij, and this so called grandmother on BakenBurg:'

duplication only one predicate is annotated. Otherwise, this would lead to confusion as to which semantic role belongs to which event. Moreover, a single predicate can be associated with multiple mentions, all fulfilling the same semantic role. For example, an event might involve two distinct entities in the role of *PATIENT*, which also results in a duplicate text. This is illustrated in Figure 3.2, the event '*bevind*' (being in a place) points to two references that both separately take on the role of *PATIENT*, with different satellite roles accompanying them.

### 3.3.5 Label distribution

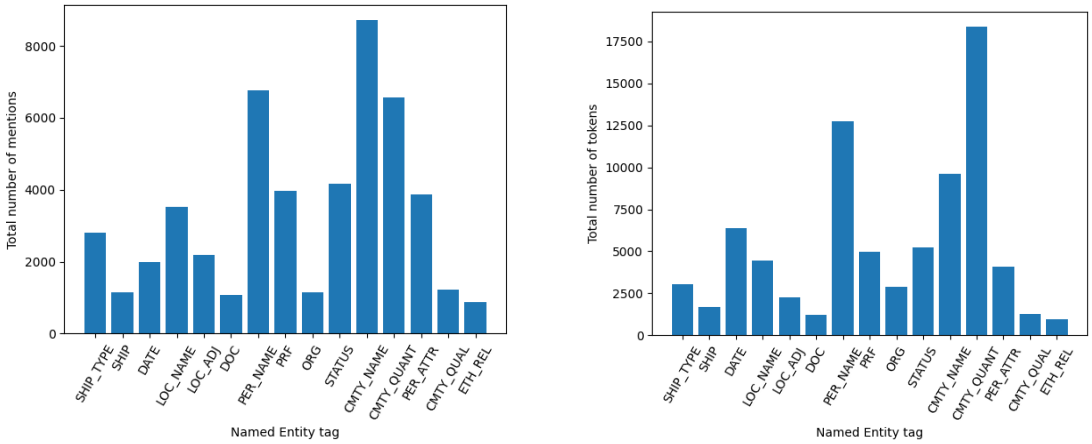
As shown in Figures 3.3 and 3.4, there is a significant class imbalance in both the Semantic Role Labeling (SRL) and Named Entity Recognition and Classification (NERC) data. Figure 3.3a and 3.4b display the number of event mentions per class. Here, a mention refers to a complete span from the beginning (*B-*) to the end (*I-*) of a labeled sequence, counted as a single mention of a role. Figure 3.3b and 3.4b illustrate the total number of tokens across all mentions per class.
The distribution of the SRL labels in both Figure 3.3a and 3.3b illustrate the large class imbalance, where the category of *PATIENT* is much more frequent than the other categories. Moreover, the categories *PATH* and *CARGO* are extremely scarce. The imbalanced distribution of the named entity classes is also illustrated in both Figure 3.4a and 3.4b, where *CMTY_QUANT* and *PER_NAME* are more frequent in the data than, for example, *ETH_REL*. This assymmetry in the distribution of both the SRL and NERC classes can result in a difference in a model's ability to represent certain classes, where some classes will be learned more accurately than others. This should be taken into account when analyzing the results.
Additionally, although both Figure 3.3a and 3.3b show a similar distribution, the absolute numbers differ substantially. For example, the class *PATIENT* contains around 1600 mentions but almost 6000 tokens. This indicates that the span of each mention often consists of multiple tokens. The distribution per class when formatted with the BIO tag reveals this is the case for certain classes, especially for the *PATIENT* class (see Figure 3a in the Appendix). This can make it more difficult for the model to predict an entire span correctly. Conversely, the difference in absolute numbers of Figure 3.4a and 3.4b suggests that the named entity mentions in general consist of shorter spans

than those of the semantic roles. As for the distribution of NERC classes when divided by the BIO-tag scheme, most roles consist of short spans, except for the classes *DATE* and *CMTY_QUANT* that do have a larger *I-* class (see Figure 3b in the Appendix).




(a) SRL mention distribution per class (b) Token distribution per SRL class

Figure 3.3: Distributions SRL classes




(a) NE mention distribution per class (b) Token distribution per NE class

Figure 3.4: Distribution NE classes

# Chapter 4

# Methodology

This chapter lays out a detailed description of the models implemented, the experimental design, the necessary pre- and post-processing steps, and the fine-tuning and multitask learning setup.

## 4.1 Overview of the models

Building on approaches discussed in previous literature (see Chapter 2), this research focuses on two multilingual models: multilingual BERT and XLM-RoBERTa. In addition, a new domain-specific model has become available: GloBERTise. These three Transformer-based models are applied within the experimental design of this thesis.

### 4.1.1 Multilingual BERT, XLM-R, and GloBERTise

As mentioned above, the specific Transformer models implemented are Multilingual BERT, XLM-RoBERTa, and GloBERTise.
Multilingual BERT (mBERT), a BERT-based model, is pre-trained on a substantial corpus containing multilingual data, consisting of monolingual Wikipedia corpora of a 104 languages, which are selected based on those with the most extensive Wikipedia content (Devlin et al., 2019).
The RoBERTa based models implemented in this research are XLM-RoBERTa and GloBERTise. XLM models are cross-lingual language models, with a transformer architecture that is similar to BERT Conneau and Lample (2019). XLM-RoBERTa (XLM-R) is a RoBERTa-based multilingual model, following the XLM approach but with a Roberta-based architecture Conneau et al. (2020). Both XLM and XLM-R share the goal of enabling cross-lingual transfer by learning language-agnostic representations through unsupervised pre-training. Their central training objective is Masked Language Modeling (MLM). XLM-R differs from XLM by removing translation-based objectives such as Translation Language Modeling (TLM) and significantly expanding the number of languages and training data. XLM-R is trained on a hundred languages and where the initial XLM-models are pre-trained on Wikipedia data, XLM-R is pre-trained using a larger CommonCrawl corpus in addition to Wikipedia (Wenzek et al., 2019).
The second RoBERTa-based model is GloBERTise which is pre-trained on a large collection of unlabeled data from the Dutch-East India Company (VOC) archives, consisting of around 7000 documents from the period 1610 - 1796. This is thus a domain-specific pre-trained model on data from the same (VOC) archives and therefore pre-trained on

data with similar structure and content as the fine-tuning VOC dataset described in Chapter 3.

## 4.2 Research design

The experimental design consists of three parts. Both fine-tuning of Semantic Role Labeling (SRL) and fine-tuning for Named Entity Recognition and Classification (NERC) are implemented as well as a multitask learning configuration of both NERC and SRL. These implementations are described in more detail in Section 4.3 and 4.3.1.

1. Fine-tuning for SRL

First, the pre-trained models are fine-tuned for SRL using the VOC dataset. Here the models are trained and evaluated to learn the representations of the arguments of the events.

2. Fine-tuning for NERC

The pre-trained models are also fine-tuned for NERC using the same VOC dataset. The models are trained and evaluated to recognize and label the instances of the named entities in the text.

3. Multitask Learning for SRL and NERC

The pre-trained models are set up for Multitask Learning (MTL) of both NERC and SRL, using the VOC data. An adjusted version of the single task fine-tuning setup is implemented, to allow the model to train on both tasks simultaneously to enable the sharing of parameters from both tasks and evaluate model performance. See Section 4.3.3 for a more detailed description of the MTL framework.

### 4.2.1 Train and test data

Due to the limited data available, there is no separate training, validation, and test set. Therefore, a 16-fold cross-validation method will be used to train and test the models. Cross-validation is a useful method in this context, as splitting the dataset into distinct training and test sets (e.g. an 80-20 percent split; 80 percent of the data used for training and 20 percent for testing) can introduce randomness into the evaluation. The test set might, by chance, be unusually easy or difficult, leading to an unreliable estimate of model performance. Instead, cross-validation ensures that all data points are used for both training and testing across different iterations, providing a more stable and generalizable performance estimate, even when working with a small dataset (Berrar, 2019). With this method, the data is split into groups and this number of groups decides the number of folds. In each fold, a different group acts as the test data, while the rest of the group acts as the training data. The average results of all fold can then be calculated to determine model performance.

The dataset consists of 16 annotated documents. The cross-validation is done using each document as the test set once, leading to 16-fold cross-validation. As previously shown in Figure 3.3, the SRL classes in the VOC data are not equally distributed across the dataset. Therefore, the method of folding per document gives insight into the representation of classes in each document and what this means for model performance,

which also provides useful information for further research using this data.

### 4.2.2 General pre-processing steps

Several pre-processing steps were necessary regarding the input data before starting each training and testing process. Most importantly, formatting the dataset to fit the input requirements of the BERT- and RoBERTa-based models. This concerns label composition, tokenization, and managing special tokens.

#### Label encoding

The labels of NERC and SRL are both categorical and in the form of text. They consist of labels like *AGENT*, *PATIENT* for SRL and *PERSON*, *LOCATION* for NERC. A necessary step in setting up the task is transforming these labels to numerical labels, as the models work with numerical input for classification tasks. This is done using a LabelEncoder, which maps a textual category to a number, in the form of a dictionary. This mapping covers the whole dataset, enabling interpretation of the labels by the model.

#### Tokenization

Transformer models are pre-trained with tokenizers, in the case of BERT and RoBERTa this is a sub-word tokenizer. Before pre-training, subword tokenization is a technique by which words are broken down into subwords based on their frequency in the pre-training dataset. These subwords form a fixed vocabulary which the model uses during training. BERT is pre-trained with the WordPiece tokenizer. The WordPiece tokenizer builds its vocabulary by starting from single-character tokens and gradually adding frequently occurring subword combinations. The merging is focused on maximizing the model's ability to learn language patterns. RoBERTa is also trained with a subword tokenizer, namely Byte Pair Encoding (BPE). The objective is similar to WordPiece but differs in the way it approaches building the vocabulary and the merging of subwords. The BPE vocabulary is created by looking at the frequency of symbol pairs, starting with characters and merging until the most frequent pairs to create longer subword units. This tokenizer does not regard prediction performance (Schmidt et al., 2024).
Once created, the tokenizer and vocabulary are used to encode the training input. This approach enables the models to handle unknown or rare words by piecing them together from familiar subwords in the vocabulary. This combination of subwords allows for the production of meaningful representations of known and unknown words.
Additionally, both tokenizers make use of special tokens. For BERT these tokens are marked with '[ ]' and for RoBERTa with '<>'. They use special tokens like '[CLS]' and '<s >' at the beginning of a sequence and '[SEP]' and '</s>' to indicate the end of a sequence or to seperate sequence pairs. All special tokens are mapped to -100. This is done for ignoring padding during loss computation.
Pre-training with a subword tokenizer results in subword-level predictions. The eventual output is, however, expected to consist of word-level predictions. Therefore, the subwords are mapped to the corresponding label of the complete word they form together. These mappings are used to achieve word-level predictions. The way this is done is further discussed in 4.4.1. Moreover, as the data contains large chunks of text,

the token length was set to the max length of 512, to capture the largest chunks of text possible for the model.

**Predicate augmenting**

As the task of semantic role labeling deals with the relationship between a specific predicate and the accompanying arguments that can be identified, it is important to deal with this relationship with regards to the input. This task does not deal with predicate / event trigger detection and the predicate is therefore already given in the data. To encode this information for the input, it must be identified which token represents the predicate, also called the event trigger. As presented in the paper by Khandelwal and Sawant (2020), there are multiple options to handle this. One method is to replace the predicate token with a special token, and another is to augment the data by adding a special token before / after the predicate token. Based on the results by Khandelwal and Sawant (2020), the latter method of augmenting the predicate with a special token is employed. The input is augmented by adding a special token before the predicate token: '[PRED]'. An example of what an input sentence looks like is shown in Figure 4.1. The special token is added to the list of special tokens for the tokenizer, so that it is dealt with the same way as other special tokens like [SEP].

[ '[CLS]', 'The', 'ship', '[PRED]', 'arrived', 'at', 'the', 'dock', '.', '[SEP]' ]

Figure 4.1: Example of tokenized input sequence for BERT

## 4.3 Hyperparameter tuning

After the pre-processing steps, one more step was necessary before the pre-trained models could be implemented for each experiment described in Section 4.2. That is, hyperparameter tuning was performed for single-task fine-tuning on Semantic Role Labeling (SRL). Hyperparameter tuning involves identifying the most effective values of key parameters to achieve optimal results from the fine-tuned model. The parameters that are optimized are the learning rate and the number of epochs.
The learning rate is a setting that decides how the model adjusts when it learns from errors, by controlling the step size taken to reduce loss during training. If the learning rate is too low, the model updates its parameters too slowly, leading to a slow learning process and delaying convergence, i.e. achieving a stable learning state. A learning rate that is too high lets the model jump past the best values, creating an unstable training process and inability to minimize the loss (Wilson & Martinez, 2001). A loss function measures how accurately a model makes predictions by evaluating the differences between the model's predictions and the actual output. The loss is conveyed through a numerical value, where a high value indicates worse performance. This value guides the training algorithm in model optimization, by adjusting the model parameters to minimize error and improve accuracy. For Transformer models the loss function is Cross-Entropy Loss, also named log loss, which is commonly applied for classification tasks. This function measures the mismatch between the model's predicted probability distribution and the true labels.
The number of epochs specifies how many times the learning algorithm processes the

entire dataset during training. A higher number of epochs can allow a model to pick up more complex structures in the data, however, too many epochs can lead to overfitting on the training data (Afaq & Rao, 2020).
Using a subset of the data, and a three-fold cross-validation for efficiency, the different models were trained and tested using different learning rate and epoch settings. The loss was calculated but led to different best settings per model. Therefore, F1 scores were also investigated and based on this outcome, the settings were defined. These settings were then used throughout the entire training process of each experimental design to ensure consistency and reproducibility.

### 4.3.1 Experiment configurations

With the obtained parameter settings for the learning rate (5e-5) and number of epochs (30), the pre-trained models; multilingual BERT, XLM-RoBERTa, and GloBERTise, were ready to be implemented for the three experiments. The model fine-tuning was conducted on the Snellius high-performance computing (HPC) cluster, utilizing a single GPU.

### 4.3.2 Multitask learning

The framework of the multitask learning task that is implemented in this thesis involves a shared encoder. A shared encoder, instead of a separate encoder for each task, is shared across tasks, allowing for more coherent and integrated representation learning. This enables a model to share parameters and to capture common patterns and relationships in the data Roy et al. (2025). The shared encoder processes the input text and generates contextualized token representations, meaning a unique embedding is produced for each token in a sentence, taking into account the surrounding words. This encoder is then paired with two output layers, also called classification heads. Each classification head is implemented as a linear layer that maps the encoder's output to task-specific labels Hu and Singh (2021). For this project that meant creating a model with two classification heads, one for SRL and one for NERC. Previous research implementing multitask learning utilizes different models or several different tasks. As discussed in Chapter 2, the exact multitask framework for this thesis is not directly found in many studies. Therefore, it was difficult to find substantial information on the correct configuration for this specific multitask implementation. These limitations will be discussed in more detail in Chapter 7.

### 4.3.3 The multitask setup

To establish the multitask configuration, some changes needed to be made to the single-task fine-tuning setup. Multiple custom functions had to be initiated to realize multitask learning. The multitask training setup uses a custom data structure creating task-specific datasets that also return an identifier indicating which task the data belongs to. This ensures that during training, the model applies the correct output layer for each task. Each task also has its own data loader; the tool that handles the process of feeding data to the model by dividing the dataset into batches so the model can process multiple examples at the same time. Each batch also includes the task identifier. A combined iterator samples batches from the two tasks in a balanced and randomized way, allowing the model to alternate between tasks during training. To initiate this, a

modified training routine handles the multitask learning by directing each batch to the correct output layer based on the task identifier. The code can be found in this github (`https://github.com/globalise-huygens/nlp-semantic-role-labeling`) and is adapted from (`https://medium.com/@shahrukhx01/multi-task-learning-with-transformers-part-1-multi-prediction-heads-b7001cf014bf`).

## 4.4 Post processing steps

### 4.4.1 Word level predictions

As mentioned, BERT and RoBERTa models are both pre-trained with an, albeit different, subword tokenizer, meaning text regions are not only split into word-level tokens, but these word-level tokens can also be split into subwords, depending on their existence in the model's vocabulary. The predictions of the model are therefore subword-level predictions and not word-level predictions, meaning that there can be multiple predictions for one word. For an accurate representation of the results, these sub-word level predictions need to be transformed to word-level predictions. This is done by removing padding and iterating over the word.ids, the predictions, and the labels. For each unique word.id the predictions are aggregated in one list, from which the majority label is derived using bincount. In case of a tie, the first label encountered is chosen.

### 4.4.2 Evaluation

The evaluation metrics to analyze the performance of each model consist of the average precision, recall, and F1 scores of all folds. Precision measures the correctness of the instances that the model predicts as a certain class, for example, how many labels that the model classified as *B-Agent* are actually *B-Agent*. Recall illustrates the ability to recognize the instances of a class in the text, so of all the *B-Agent* instances that are present in a text, how many did the model identify. The F1-score portrays the harmonic mean of these two metrics.
The 16-fold per document cross-validation method that is used for training and testing results in varying distributions of instances of each class present in the test data. As mentioned in Chapter 3, the documents differ greatly with respect to size and class representation. Moreover, some classes are not present in every document. This leads to a support of zero for classification as well as zero for precision, recall, and F1 scores for those classes. To account for this, only the classes that have a support of more than zero are included in the final calculation for the evaluation metric scores. Note, classes with a support of more than zero that still receive precision, recall, and F1 scores of zero are included in the calculation. The individual folds of the cross-validation will also be investigated in the error analysis.
Additionally, the evaluation is word-level based. This means that the individual word-level predictions are evaluated, not the entire span. This is a less strict evaluation method than span-level evaluation, where a prediction is only correct if the entire span is labeled correctly. However, since the labels are tagged using the BIO-format, the predictions are penalized when they are not labeled with the correct BIO tag. This evaluation method gives insight into what goes wrong and right for separate words inside a span but also takes span boundaries into account.

Hence, the impact of multitask learning is assessed by looking at the difference in performance between single-task fine-tuning of SRL and the performance of SRL after

multitask learning of SRL and NERC per model. If multitask learning results in higher scores for a model, this will indicate that there is an impact of the method of multitask learning.

# Chapter 5

# Results

This chapter will lay out an extensive evaluation of the models fine-tuned for only Semantic Role Labeling (SRL) and Named Entity Recognition and Classification (NERC) on the one hand, and Multitask Learning (MTL) of SRL and NERC, on the other hand. The evaluation will provide an answer the main research question: 'Is multitask learning beneficial for a semantic role labeling task with the domain-specific VOC data, and which language model is best suited for this task?

## 5.1 Evaluation metrics

As discussed in the previous chapter, the performance is assessed by a word-level evaluation, separating spans based on *B-* (beginning) and *I-* (inside) tokens. The evaluation metrics consist of precision, recall, and F1-score. The model performance in this chapter is represented by the macro average performance of these three metrics. The macro average computes mean performance across all classes, assigning equal weight to each class regardless of its frequency in the test set. Note that the average F1 score presented is not derived as the harmonic mean of the average precision and recall. The F1-scores per class are calculated this way. The macro average F1 score, however, is calculated as the mean of these F1 scores per class. This is the standard computation method in sklearn (Pedregosa et al., 2011). Consequently, macro-average F1 scores can result in a value lower than both the average precision and the recall.
Furthermore, to assess whether the observed differences between models or experimental setups are statistically significant, a paired-samples t-test is conducted. A paired sampled t-test evaluates whether the mean difference between two related measurements within the same group significantly deviates from zero. It does so by computing the difference for each paired observation and evaluating whether the average of these differences is statistically significant. In this context, the test compares either the same model across the two different experimental settings or the two different models on the same task. A statistically significant difference is indicated when $p < 0.05$.

## 5.2 Evaluation of fine-tuning on Semantic Role Labeling

First, the multilingual BERT (mBERT), XLM-RoBERTa (XLM-R), and GloBERTise model, the first being a BERT-based model and the other two RoBERTa-based ones, were fine-tuned on the single task of Semantic Role Labeling (SRL). The results of these models are compared to determine the best performing model for this task. As

illustrated in Table 5.1, each model demonstrates low performance on SRL. In comparison, Transformer models fine-tuned on English SRL tasks ((Li et al., 2019); (Zhang et al., 2020)) and the model BERTje fine-tuned on Dutch SRL (De Vries et al., 2019) achieve F1-scores of around 0.85. As mentioned in Chapter 2 and 3, achieving similar performance on the current dataset may be challenging due to the differences in language, structure and dataset size. For all three models, precision scores exceed recall scores, suggesting that the models are more cautious in what they label and struggle more with identifying relevant instances of semantic roles in the data. However, precision also falls short and leaves room for further enhancement. The results show that the XLM-R model significantly outperforms mBERT ($p < 0.05$), but the GloBERTise model outperforms both mBERT and XLM-R in all metrics. This is a significant difference ($p < 0.05$), which indicates that a RoBERTa-based model, but more specifically a model pre-trained on domain-specific data (GloBERTise) yields the best results.

## 5.3 Evaluation of Multitask Learning

The following experiment used a Multitask Learning (MTL) framework that included both Named Entity Recognition and Classification (NERC) and Semantic Role Labeling (SRL). Both the BERT and RoBERTa models are employed for this task, given that the results from the single-task fine-tuning indicated considerable room for improvement. The results in Table 5.1 reveal that the models continue to exhibit poor performance on the SRL task, with an F1-score of 0.28 for mBERT, 0.28 for XLM-R, and 0.33 for GloBERTise. Recall remains lower than precision, indicating that the issue of identifying the relevant instances of semantic roles in the texts persists. Nonetheless, precision also remains suboptimal and requires substantial improvement.

| | **mBERT** | | | **XLM-R** | | | **gloBERTise** | | |
|---|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| **Evaluation of Single Task Fine-tuning** | | | | | | | | | |
| Average | 0.29 | 0.26 | 0.25 | 0.30 | 0.29 | 0.28 | 0.37 | 0.33 | 0.31 |
| **Evaluation of Multitask Fine-tuning** | | | | | | | | | |
| Average | 0.34 | 0.29 | 0.28 | 0.32 | 0.30 | 0.28 | 0.36 | 0.33 | 0.33 |

Table 5.1: Classification Performance for SRL using mBERT, XLM-R, and gloBERTise

## 5.4 Single task versus multitask learning

To address the main research question, an analysis of the results of the two experiments is essential to assess whether Multitask Learning (MTL) improves Semantic Role Labeling (SRL) performance. The difference in scores when looking at the two different setups (single-task versus multitask) presented in Table 5.1, shows there is no significant increase in the performance of SRL for both the XLM-Roberta model and the GloBERTise model ($p > 0.05$). This indicates these models do not improve from the MTL framework. However, there is an increase in performance of the BERT model (mBERT) between single-task and multitask fine-tuning ($p < 0.05$). This indicates that mBERT does benefit from this multitask learning implementation, where it

now achieves similar scores as the XLM-R model. Nevertheless, as illustrated in Table 5.1, GloBERTise still outperforms the multitask trained BERT and the XLM-R model significantly ($p < 0.05$) without an increase in performance in the MTL setup for the GloBERTise model.

## 5.5 NER scores

As shown, the best performing model overall is GloBERTise, however, this model did not benefit from the Multitask Learning (MTL) framework. To investigate the possible reasons for these results, the other task in the configuration can be assessed, i.e. Named Entity Recognition and Classification (NERC). In an MTL design, the issue of overfitting can arise (Chen et al., 2018). More specifically, there is a risk that the model overfits to the comparatively simpler task, i.e. NERC in this case, thereby impeding learning on the more complex objective (SRL). To investigate this assumption, the results of solely training for NERC and multitask learning of SRL and NERC are illustrated in Table 5.2. These scores show that GloBERTise outperforms XLM-R and mBERT in this task as well, both for the single-task and multitask learning framework. Moreover, all three models do not benefit from multitask learning in this set-up for NERC. This suggests that SRL information does not improve NERC performance in this context. This is in line with the explanation in Chapter 2, where it is described how a 'lower-level' task, in this case NERC, is informative for higher-level task, in this case SRL. Furthermore, there is no sign of overfitting as the single-task and multitask results are almost identical. Lastly, the results of NERC are significantly higher than for SRL (a difference of 0.30 in F1-score), indicating the models can learn these representations more easily, which is in line with the idea that NERC is a simpler NLP task compared to SRL. Still, earlier NERC experiments of Arnoult et al. (2021) using similar VOC data achieve F1 scores of above 0.80, which is significantly higher than the results obtained in this study. However, the dataset used by Arnoult et al. (2021) contained fewer named entity classes and a larger overall volume of data. The individual NE classes that are shared across both datasets achieved comparable scores, but a detailed class-level comparison falls outside the scope of this thesis. Nevertheless, these findings indicate that, similar to SRL, as discussed further in Section 5.6.2, there is considerable variation in how accurately individual classes are identified, and that limited data availability continues to pose a significant challenge.

| | **mBERT** | | | **XLM-R** | | | **gloBERTise** | | |
|---|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| **Evaluation of Single Task Fine-tuning** | | | | | | | | | |
| Macro avg | 0.55 | 0.49 | 0.49 | 0.59 | 0.54 | 0.53 | 0.65 | 0.63 | 0.61 |
| **Evaluation of Multitask Fine-tuning** | | | | | | | | | |
| Macro avg | 0.55 | 0.50 | 0.50 | 0.58 | 0.52 | 0.52 | 0.66 | 0.63 | 0.61 |

Table 5.2: Classification Performance for NER using mBERT, XLM-R, and gloBERTise

## 5.6 GloBERTise performance

Based on the evaluations of single-task and multi-task learning, the GloBERTise model emerges as the best performing model. To take a closer look into this performance, the following sections will provide an analysis of the predictions of the GloBERTise model, looking at the performance in the separate training/testing folds and on the different semantic role classes.

### 5.6.1 Performance analysis per fold

The overall performance of the GloBERTise model does not reveal a benefit from multitask learning. However, the performance is based on the average results of a 16-fold cross validation. As discussed in Chapter 3, the dataset comprises documents that vary considerably in both size and content, which may contribute to performance fluctuations across different folds. Figure 5.1 presents the F1 scores of the individual folds of the GloBERTise model under both task designs. The results show that in 9 of the test documents, the multitask model outperforms the single-task model. In the 7 remaining folds, however, the single-task model performs better. Moreover, the differences in many of the instances are relatively small and tend to cancel each other out when averaged, leading to an overall non-significant difference in performance between the two designs. In a few folds, either the single-task or the multitask model demonstrate a substantial improvement, with differences in F1 score reaching between 0.10 and 0.14. These magnitudes can be considered meaningful and these outliers suggest that both task designs can be beneficial under certain conditions. Further analysis is required to investigate the underlying causes of these differences in performance. This is addressed in more detail in Chapter 6.

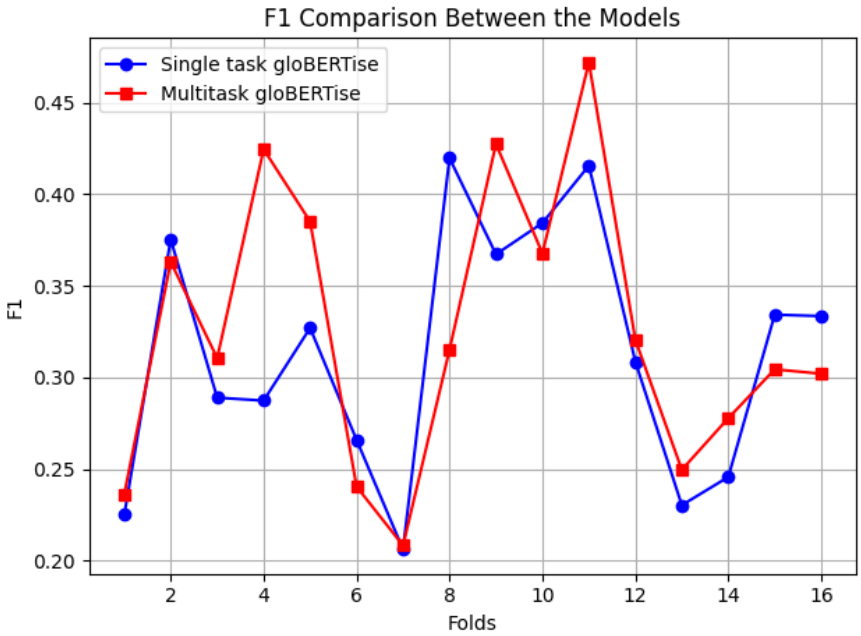



Figure 5.1: Individual test set comparison for gloBERTise on SRL (multi vs. single)

### 5.6.2 Class evaluation

Next to differences in scores across the different folds, the dataset also contains 23 distinct semantic role classes. These consist of 11 distinct classes divided by *B-* (beginning) and *I-* (inside) labels plus an *O* label for non-semantic role tokens, as explained in Chapter 3. It is therefore also crucial to examine model performance at the level of individual classes. As previously discussed, these categories differ in the frequency with which they occur in the annotated data, which may lead to varying levels of performance. The matrix in Figure 5.2 and Figure 5.3 present a visualization of the aggregated confusion matrices across all documents for the single-task model and multitask model respectively. The most prominent issue observed in both models is their overfitting to the *O* class, where semantic role instances are incorrectly labeled as non-role tokens. This is evidenced by the dark blue column representing predicted *O* labels, which indicates that both models frequently classify tokens as *O* (i.e. not part of any semantic role) when they actually correspond to semantic role labels. This misclassification is a major source of error and represents the models' generally lower recall scores - which reflects a model's ability to detect relevant instances of a given class. Furthermore, the models seems to confuse certain classes. These classes are often semantically similar, such as *AGENT* and *BENEFACTIVE*, *PATIENT* and *AGENTPATIENT*, and *PATH* and *TARGET*. This indicates that the model did not learn the detailed distinctions between the semantic role classes well enough.
Additionally, some classes are better predicted than others. Classes that are well represented in the data, such as *B-* and *I-Patient* are relatively well predicted. However, classes that have a lower frequency in the training data can also be relatively well predicted, such as *B-Source* and *B-* and *I-Target*. In contrast, classes such as *B-Agent-Patient*, *PATH*, and *CARGO* are barely identified by the models. As *PATH* and *CARGO* are the most infrequent classes in the data, their low recognition score can be attributed to a lack of representation in the data. However, higher frequency does not necessarily determine higher accuracy of classifications. Some low-frequency roles, which showed overlap with NE classes, see Figure 3.1, were predicted relatively well. However, as Figure 5.3 shows, the results between the single-task and multitask model are almost identical. Hence, these results cannot be attributed to information shared by the NERC task. The specific class errors will be analyzed in more detail in the next chapter.

## 5.7 Conclusions

These results reveal how all three models overall exhibit low performance on the semantic role labeling task, with the highest precision of 0.37, the highest recall of 0.33 and the highest F1-score of 0.33. This indicates that all models were limited in their ability to represent the data accurately. Still, the results do show that the GloBERTise model, i.e. a domain-specific pre-trained RoBERTa model, performs best.
Furthermore, when comparing multitask learning with single task fine-tuning, an interesting result shows up. Where both RoBERTa-based models overall do not benefit from this method, mBERT does improve on all metrics. This indicates that NERC can work as an auxiliary task to boost SRL performance for certain models, in this case a BERT-based model, but not for others, i.e. the XLM-R and GloBERTise model in this context. Still, the GloBERTise model shows to be superior in all configurations when

implemented for the task of SRL using the VOC dataset.
Moreover, the NERC scores reveal that there exists no sign of overfitting to one of the tasks and mostly highlight the difference in complexity between SRL and NERC task, with SRL being a more difficult task. Additionally, when inspecting the separate folds of the training/testing of the GloBERTise model, in some cases either the single-task or the multitask model reveals significantly higher results. This suggests that for some documents one model is better fitted than the other. A closer inspection of the performance of the individual SRL classes also indicates that the models often miss semantic role tokens, and confuse semantically similar classes. Moreover, certain classes are more often correctly predicted than others. Although it may account for some classes, the varying levels of representation do not suffice as an explanation for the varying levels of performance and the overall low performance across the classes.

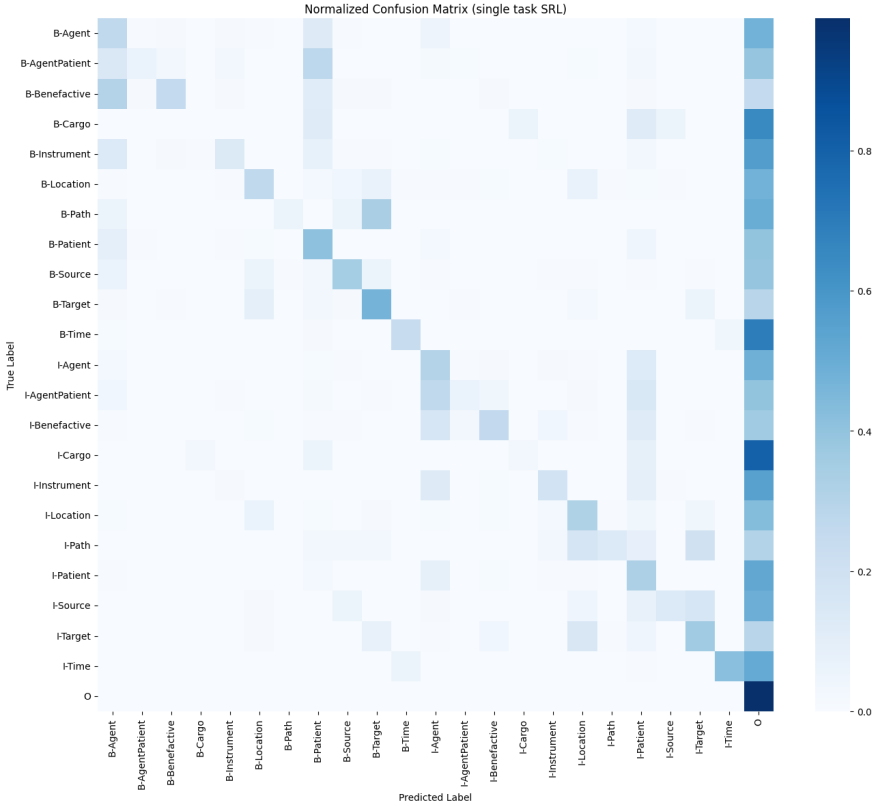



Figure 5.2: Single-task model - aggregated confusion matrix individual classes across all folds for GloBERTise on SRL

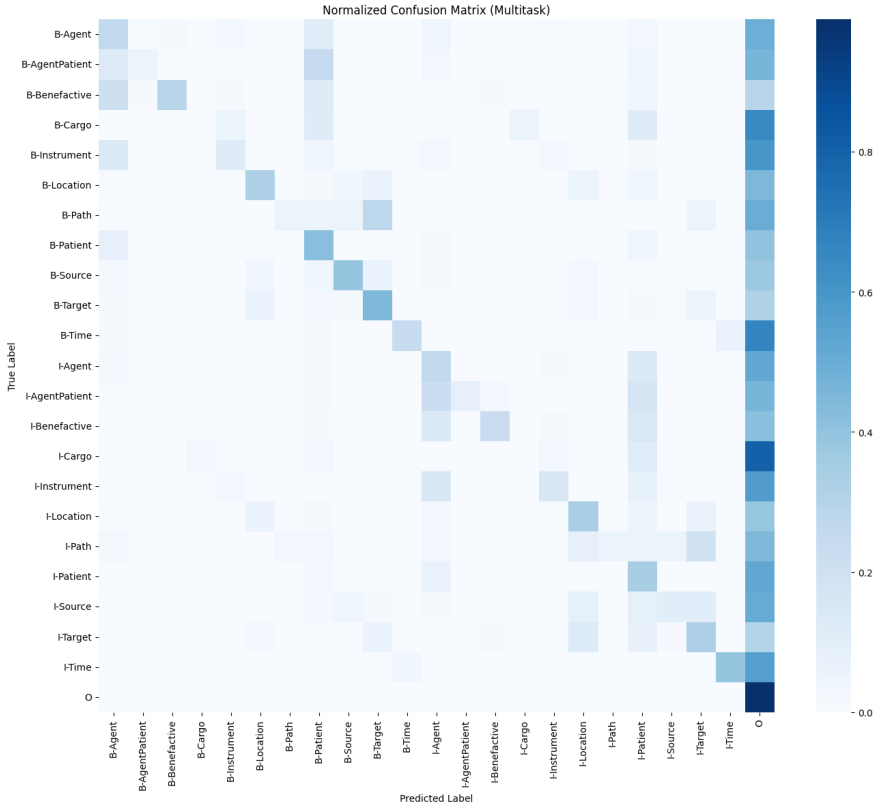



Figure 5.3: Multitask model -aggregated confusion matrix individual classes across all folds for GloBERTise on SRL

# Chapter 6

# Error Analysis

This chapter will provide a detailed analysis on model behavior on specific test documents to better understand performance discrepancies between the single-task and multitask versions of the GloBERTise model. As discussed in the Chapter 5, the overall performance of both the single-task and multitask GloBERTise implementations did not differ significantly. However, the individual folds revealed notable variation, with some favoring the multitask model and others the single-task model. To explore the causes for these differences, this chapter focuses on the two folds with the most substantial difference in performance (based on F1-score) between the single-task and multitask GloBERTise model.

## 6.1 Document Performance

As exhibited in Table 6.1, the document from the year 1747 resulted in a higher performance by the multitask model, with a 0.13 increase in F1-score over the single-task model. Conversely, the document from the year 1679 yielded better results by the single task model, which outperformed the multitask model by 0.10 in F1-score. Interestingly, both documents exhibit a higher average recall than average precision, in contrast to the macro-level scores across all documents as discussed in Chapter 5. This may be due to document-specific characteristics such as token distribution or label frequency.

| | **Single Task Fine-tuning (gloBERTise)** | | | **Multitask Learning (GloBERTise)** | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 |
| **Fold 4 – Doc 1747** | 0.28 | 0.35 | 0.29 | 0.42 | 0.47 | 0.42 |
| **Fold 8 – Doc 1679** | 0.50 | 0.46 | 0.42 | 0.31 | 0.36 | 0.32 |

Table 6.1: Specific Test Document Average Performance

## 6.2 Document analysis

To gain a deeper understanding of the behavior of the models, the remainder of this chapter will analyze both documents separately. This is achieved by conducting an error analysis for each of the two documents. The analysis includes an evaluation of per-class performance metrics as well as a close inspection of individual model predictions. In the examination of individual model predictions, both the errors and the correct predictions of semantic role classes are looked at. The correctly predicted *O* instances

are left out of the examination. The errors are divided into 4 categories: false positive, false negative, label confusion, and boundary error. A false positive error occurs when the model predicts a semantic role label when the gold label is *O* (not a semantic role). A false negative occurs when the model predicts *O* when the token actually fulfills a semantic role. The error of label confusion arises when the model predicts the incorrect semantic role of a token. Thus, the model is correct in its identification of a semantic role, however it does not predict the correct role. A boundary error occurs when the model predicts the correct semantic role label, but the incorrect BIO tag label (*B-* or *I-* ), leading to misalignment in span boundaries. In some cases, a prediction is classified with both the incorrect BIO tag and the incorrect semantic role. These errors are classified as a label confusion, as this is seen as a more serious error with respect to the goal of the task. Additionally, each example of an error, represented in the Figures, is provided with a transliteration. Transliteration is used to express the Early-Modern Dutch script through Modern English for readability.

## 6.3 Document from 1747

The multitask GloBERTise model performed better than the single-task one when predicting on the document from 1747. This document contains 23 text regions, with a total of 3512 tokens. As discussed in Section 3, this is one of the smaller documents in the dataset. Examination of class representation reveals that several role types are not represented at all in the document. These classes are *CARGO*, *PATH*, and *SOURCE* where neither the *B-* or *I-* classes are present. These will therefore be excluded from analysis. For the remaining classes, the number of instances is also limited, ranging from 1 to 21 tokens per class. This class sparsity impacts model performance: both models fail to detect several classes, resulting in precision, recall and F1-scores of 0.00. Examples include *B-AgentPatient*, *B-* and *I- Benefactive*, and *I-Location*, which remain undetected. Since there are only a few mentions of these categories, these oversights must be interpreted with caution.

The multitask model, for example, does not recognize any instances of *I-AgentPatient*, whereas the single-task model does. On the other hand, the single-task model does not recognize classes that the multitask model does, such as *B-Target* and *B-* and *I-Time*. While the single-task model scores 0.00 on these classes, the multitask model achieves high recall and precision scores.

A closer inspection reveals that the multitask model achieves perfect scores (F1= 1.00) for some classes, contributing to its higher overall average. However, as mentioned, these scores have to be interpreted with caution. In cases where a class is represented by only one or two tokens, a correct prediction leads to a disproportionately high score, whereas a single error drastically lowers performance. Thus, the improved performance may not generalize beyond this particular document. The limited support for most classes means that minor variations in prediction accuracy can significantly impact evaluation metrics, potentially overstating the model's capabilities on rare classes.

As illustrated in Table 6.2, the single-task and multitask model correctly predict 37 and 38 semantic roles respectively, out of a total of 93 semantic role tokens. This indicates that the actual difference in correctly predicted semantic roles between the two models is minimal, amounting to just one token. The larger discrepancy lies in the types of errors made by each model, particularly in the number of false positives, which do not affect the count of correctly predicted tokens. Furthermore, the models classify fewer

than half of the semantic roles correctly. The classes that are recognized by both models are *AGENT*, *PATIENT*, *LOCATION*, and *INSTRUMENT*. The full classification reports can be found in Figure 1 in the Appendix.

### 6.3.1 Error types

To better understand model behavior in this document, the errors made by the models will now be investigated in terms of the four error types introduced earlier: false positives, false negatives, label confusion, and boundary errors. Table 6.2 presents the frequency and proportion of each error type and the number of correctly identified semantic roles (non *O*) out of all semantic roles in the document for both the single-task and multitask model.

| **Error Type** | **Count (Single-Task)** | **Percentage** | **Count (Multitask)** | **Percentage** |
|---|---|---|---|---|
| False Positive | 43 | 43.43% | 36 | 39.56% |
| False Negative | 24 | 24.24% | 29 | 31.87% |
| Label Confusion | 24 | 24.24% | 22 | 24.18% |
| Boundary Error | 8 | 8.08% | 4 | 4.40% |
| **Total Errors** | **99** | **-** | **91** | **-** |
| **Correct Predictions** | **37 / 93** | **39.78%** | **38 / 93** | **40.86%** |

Table 6.2: SRL Error Types and Correct Predictions for Single-task and Multitask Model on document from 1747

### 6.3.2 False positive

False positives represent the most frequent error type for both models, which aligns with the observed pattern of higher recall than precision. While the single-task model shows a slightly higher rate of false positives, both models make this type of mistake with considerable frequency.

In Figure 6.1, an example of a mistake is shown that the single-task model makes which the multitask model does not. The model predicts two mentions of *B-* and *I-Patient*, where none of the words actually represent a semantic role. These tokens refer to 'de Heeren Zeventien' which are the the bosses of the VOC. References to them are very frequent in the data, hence it makes sense the model mistakenly classifies these tokens as *PATIENT*. This shows the model did learn the fine-grained distinctions between the predicate-argument structures of the different text regions well enough.

In Figure 6.2, both models actually make two types of errors, a boundary error and a false positive. The models predict *I-location*, when there is no preceding *B-* label. The

**Prediction:**
**Text:**
'Aan den wel Edelen groot agtb : heere Julius Valentijn stein van Pollenesse, Raad ordinaris van Nederlands India, gouverneur Ceijlon en directeur Eijlands en dies onderhoorigheeden '

**Falses positive:** B-Patient, I-Patient, B-Patient, I-Patient, I-Patient

**Gold:**
**Text:**
'Aan den wel Edelen groot agtb : heere Julius Valentijn stein van Pollenesse, Raad ordinaris van Nederlands India, gouverneur Ceijlon en directeur Eijlands en dies onderhoorigheeden '

**Labels:** O, O, O, O, O

Figure 6.1: False positive - single-task model
*Transliteration: 'To the noble greatly honorable: sir Julius Valentijn stein van Pollenesse, Council ordinary of Dutch India, governor Ceylon and director of Island and its dependencies'

**Prediction:**
**Text:**
'De grootmoeder van den overleden Candiesen vorst met het 2:e vaartuijg alhier aangekomen, bevind zig tot als nog op marticattij , en die zoo genaamde grootmoeder op BakenBurg '

**False positive:** I-Location

**Gold:**
**Text:**
'De grootmoeder van den overleden Candiesen vorst met het 2:e vaartuijg alhier aangekomen, bevind zig tot als nog op marticattij , en die zoo genaamde grootmoeder op BakenBurg '

**Labels:** O

Figure 6.2: False positive - both models
*Transliteration: 'The grandmother of the deceased Candese king with the 2nd ship here arrived, is located to date still on marticattij, and this so called grandmother on BakenBurg'

false positive error is not completely illogical, as 'Marticattij' can represent a location, as can be seen in Section 6.3.5. This error illustrates how these ambiguities in the data give rise to confusion in the models.

### 6.3.3 False negative

The next error type is false negative. This error type occurs more frequently in the multitask model, suggesting the multitask model more often under-identifies relevant spans, see Table 6.2.

Figure 6.3 illustrates an error where both models fail to identify the role of *TIME*in this text region. As explained in Chapter 3, text regions often appear multiple times in the data, with different predicates or different interpretations leading to different distributions of semantic roles. Therefore, the models might get confused when they learn that these tokens take do not the role of *TIME* in one region but do in the other.

**Prediction:**
**Text:**
'De grootmoeder van den overleden Candiesen vorst met het 2:e vaartuijg alhier aangekomen, bevind zig tot als nog op marticattij , en die zoo genaamde grootmoeder op BakenBurg '

**Falses negative:** O, O, O

**Gold:**
**Text:**
'De grootmoeder van den overleden Candiesen vorst met het 2:e vaartuijg alhier aangekomen, bevind zig tot als nog op marticattij , en die zoo genaamde grootmoeder op BakenBurg '

**Labels:** B-Time, I-Time, I-Time

Figure 6.3: False negative - both models
*Transliteration: 'The grandmother of the deceased Candese king with the 2nd ship here arrived, is located to date still on marticattij, and this so called grandmother on BakenBurg'

**Prediction:**
**Text:**
'De grootmoeder van den overleden Candiesen vorst met het 2:e vaartuijg alhier aangekomen, bevind zig tot als nog op marticattij , en die zoo genaamde grootmoeder op BakenBurg '

**Label confusion + false negative:** B-Agent, I-Agent, O, O, O, O, O

**Gold:**
**Text:**
'De grootmoeder van den overleden Candiesen vorst met het 2:e vaartuijg alhier aangekomen, bevind zig tot als nog op marticattij , en die zoo genaamde grootmoeder op BakenBurg '

**Labels:** B-Patient, I-Patient, I-Patient, I-Patient, I-Patient, I-Patient, I-Patient,

Figure 6.4: False negative and label confusion - both models
*Transliteration: 'The grandmother of the deceased Candese king with the 2nd ship here arrived, is located to date still on marticattij, and this so called grandmother on BakenBurg'

The example in figure 6.4 illustrates how the models fail to capture the complete span of the semantic role mention, missing 5 of the 7 tokens belonging to the semantic role. This suggests the models struggle with identifying the correct spans of the semantic role mentions. The other error in this mention, i.e. the incorrect label of *AGENT*, is further discussed in Section 6.3.4 This illustrates how the model makes multiple types of errors in single role mentions.

### 6.3.4 Label confusion

The error type of label confusion is evenly distributed between the two models. This error suggests that even when the model successfully identifies a token as being semantically relevant, it often struggles to assign the correct role. In several cases, this is due to the model confusing semantically related roles, where the tokens appear in the data multiple times, taking on these different related roles. The model then confuses these appearances, suggesting the model has not learned these distinctions correctly. However, also unrelated roles are confused by the models, indicating an overall lack of

understanding of the classes.

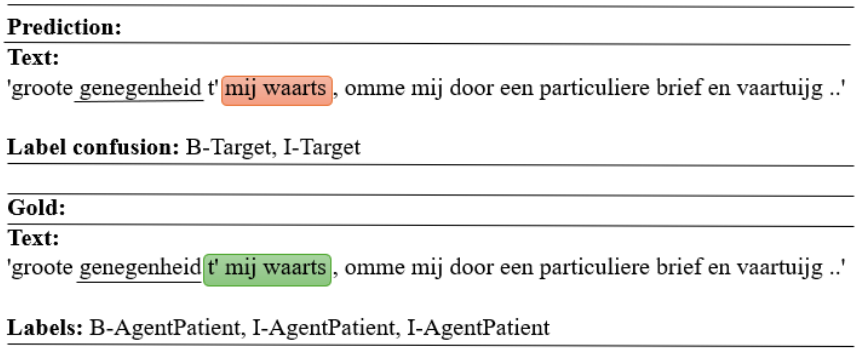



Figure 6.5: Label confusion - both models
*Transliteration: 'great affection towards me, because a particular letter and ship...'

The example in Figure 6.5 illustrates the error where both models predict the incorrect role, i.e., *TARGET* instead of *AGENTPATIENT*. The token *'t mij waarts'* or *towards me* suggests a direction. The mistake illustrates how the model interprets this literally instead metaphorically. Additionally, not only does the model not grasp the correct role, it also does not correctly represent the span, thereby mistaking the boundaries of the role mention.

As mentioned, Figure 6.4 illustrates two error types, false negative and label confusion, see Section 6.3.3. This text exhibits another example of a text region that appears in the data multiple times, where in one case this mention takes on the role of *AGENT*, whereas in this particular case, it takes the role of *PATIENT*. These variations of the same data seem to confuse the models. They do not seem to have learned these fine-grained distinctions, leading to confusion in their predictions.

### 6.3.5 Boundary error

The boundary error type, while the least common type of error, occurs more often in the single-task model. Boundary errors indicate the model struggles to identify the exact boundaries of multi-token semantic role mentions. In many cases, the model correctly recognizes the semantic role, but fails to align the span segmentation accurately.

Figure 6.6 illustrates the same sentence as in Figure 6.3, however, this is a duplicate of this sentence as the predicate refers to different semantic roles. In this case, '*marticattij*' takes the role of *B-Location*. This error is made by both models. It seems as though the models expect a word before it that takes on the role of *B-Location*, though it does not predict this. This can be taken* as a minor error as the models do recognize the word as a location argument.

Figure 6.7 shows an error made only by the single-task model. The model predicts the start of a semantic role mention incorrectly, i.e. starting with '*de*' as *B-Patient*, which results in a boundary error for the next token. '*hofl*' is now classified as I-*AgentPatient* instead of *B-AgentPatient*. The model does recognize the semantic role

**Prediction:**
**Text:**
'De grootmoeder van den overleden Candiesen vorst met het 2:e vaartuijg alhier aangekomen, bevind zig tot als nog op marticattij , en die zoo genaamde grootmoeder op BakenBurg '

**Boundary error:** I-Location

**Gold:**
**Text:**
'De grootmoeder van den overleden Candiesen vorst met het 2:e vaartuijg alhier aangekomen, bevind zig tot als nog op marticattij , en die zoo genaamde grootmoeder op BakenBurg '

**Label:** B-Location

Figure 6.6: Boundary error - both models
*Transliteration: 'The grandmother of the deceased Candese king with the 2nd ship here arrived, is located to date still on marticattij, and this so called grandmother on BakenBurg'

**Prediction:**
**Text:**
'..., en daar ook zal tragten soo veel in al mijn vermogh is zorge voorte dragen/ en de hofl . grooten zoveel respect bewijzen dat deselve zoo vergenoegt vertrekken als die oude grootmoeder welke tot nog op marticattij is'

**Boundary error:** B-AgentPatient, I-AgentPatient, I-AgentPatient, I-AgentPatient

**Gold:**
**Text:**
'..., en daar ook zal tragten soo veel in al mijn vermogh is zorge voorte dragen/ en de hofl . grooten zoveel respect bewijzen dat deselve zoo vergenoegt vertrekken als die oude grootmoeder welke tot nog op marticattij is'

**Labels:** B-AgentPatient, I-AgentPatient, I-AgentPatient

Figure 6.7: Boundary error - single-task model
*Transliteration: '...and so also will try as much as is in my power to care for/and show the court nobles as much respect that they leave as satisfied as that old grandmother who until now is on marticattij'

mention, but does not completely grasp the boundaries, as it includes the article in the role mention which is incorrect in this case. This can again be seen as a minor error, as the main arguments of the semantic role mention are recognized by the model.

### 6.3.6 Conclusion

The error analysis of this document shows how the models exhibit broadly similar error patterns, despite their difference in overall performance. The single-task model appears more prone to false positives and boundary errors, while the multitask model struggles slightly more with false negatives. Overall, the errors also exhibit how the models struggle with multiple error types in one text region or even one semantic role mention. Moreover, as the text regions can appear multiple times in the data, each representing a different predicate and semantic roles, the model seems to confuse these relations. This suggests that the models tend to focus too much on the tokens to predict semantic roles

and too little on the context. Therefore, they struggle with ambiguity. However, it also indicates that the training data may lack sufficient fine-grained distinctions to help the models learn when tokens should or should not be assigned a semantic role. Given the limited representation of many semantic role classes in this document, the discrepancy of average model performance is likely due to this limited support for most classes. This leads to a large impact of minor variations in prediction accuracy which can disproportionately affect aggregate metrics such as F1-score. Therefore, these results should be interpreted with caution.

## 6.4 Document 1679

When using the document from 1679 as a test set, the single-task GloBERTise model yielded higher performance. Similar to the document from 1747, this document is relatively small in size, with only 23 text regions consisting of a total of 3129 tokens. Compared to the previous document, this document contains even fewer classes. More precisely, almost half of the classes are not represented: *CARGO*, *INSTRUMENT*, *LOCATION*, *PATH*, *TIME*, *I-Benefactive*, and *I-Source*. These will therefore be excluded from analysis. For the remaining classes that are represented, token-level support remains limited, ranging from 1 to 27 tokens per class. As a result, both models fail to predict certain roles altogether, resulting in F1-scores of 0.00 for these categories. For instance, both models do not identify any instance of the class *B-* and *I-Target*. Additionally, the multitask model does not identify the class *B-Source*. The single-task model, however, does identify the single instance of *B-Source* in the document and thereby scores 1.00 in precision, recall, and F1, while the multitask model scores 0.00 in all these metrics. As previously discussed in the analysis of Document 1747, such discrepancies highlight the disproportionate impact of small class support on evaluation metrics, where a single error can drastically affect precision, recall, and F1-score.
Moreover, Table 6.3 also shows how the single-task and multitask models correctly predict only 32 and 30 out of 88 semantic role tokens respectively. Similarly to the previous analyzed document, the difference in correctly predicted roles only amounts to two predictions. Furthermore, both models predict less than half of the semantic roles are correctly. The classes that the models do recognize are *AGENT* and *PATIENT*, as well as *BENEFACTIVE*. The full classification reports can be found in Figure 2 in the Appendix.

### 6.4.1 Error types

The specific errors made by the model were further analyzed in more detail using the same four categories of error types: false positive, false negative, label confusion, and boundary error. Table 6.3 presents the distribution of these errors together with the number of correctly identified semantic roles (non $O$) out of all semantic roles in the document across both the single-task and multitask model. The large impact of minor variations on the evaluation metrics as explained in Section 6.4 is further confirmed by the fact that the single-task model actually makes more errors in its predictions but still outperforms the multitask model. This is due to the fact that the single-task model makes more false positive errors, which do not influence the correct semantic role prediction count.

| Error Type | Count (Single-Task) | Percentage | Count (Multitask) | Percentage |
|---|---|---|---|---|
| False Positive | 59 | 51.30% | 46 | 44.23% |
| False Negative | 35 | 30.43% | 35 | 33.65% |
| Label Confusion | 21 | 18.26% | 21 | 20.19% |
| Boundary Error | – | – | 2 | 1.92% |
| **Total Errors** | **115** | **-** | **104** | **-** |
| **Correct Predictions** | **32 / 88** | **36.36%** | **30 / 88** | **34.09%** |

Table 6.3: SRL Error Types and Correct Predictions for Single-Task and Multitask Model on document from 1679

### 6.4.2 False positive

False positives are again the most common error type for both models, which aligns with the earlier observation that recall scores tend to exceed precision. Especially the single-task model struggles with this error type, as more than half of the errors can be appointed to this. However, the multitask model also shows considerable difficulty with this error type, where these mistakes constitute over 44 percent of its errors.

**Prediction:**
**Text:**
'dat voorn : Hisselt insijne bedienige als EerstClercq en secrt = s ons volcomen Contentement heeft gegeven'

**False positive:** B-Agent, I-Benefactive
**Text:**
'dat voorn : Hisselt insijne bedienige als EerstClercq en secrt = s ons volcomen Contentement heeft gegeven'

**Labels:** O, O

Figure 6.8: False positive - both models
*Transliteration: 'that before mentioned Hisselt in his employment as First Clerk and secretary has given us complete contentment'

The error made by both models in Figure 6.8 illustrates both how the models predict semantic roles when the words do not portray any, as well as how they predicts two different roles in one mention. This shows the models do not fully capture the representations of semantic roles, i.e. *B-Agent* and *I-Benefactive* can never follow each other. However, the false positive errors are not unreasonable, as the tokens *'voorn: Hisselt'* mean *'earlier mentioned Hisselt'*, which could be seen as a span that references the same Hisselt, which does represent a semantic role.

The example in Figure 6.9 shows the error of both models in which they predict more tokens belonging to the semantic role mention than is actually the case, leading to two false positive predictions. This indicates the model struggles with the span boundaries of semantic role mentions. The incorrect label in this prediction is further discussed in Section 6.4.4.

**Prediction:**
**Text:**
'bevestight en alsoo sijn Iongste verbandt van drie Iaren den 24:en meij ao 1677 is komen te Experireeren, '

**False positive + label confusion:** B-Patient, I-Patient, I-Patient

**Gold:**
**Text:**
'bevestight en alsoo sijn Iongste verbandt van drie Iaren den 24:en meij ao 1677 is komen te Experireeren, '

**Labels:** B-AgentPatient, O, O

Figure 6.9: False positive and label confusion - both models
*Transliteration: '...confirmed and thus his Newest agreement of three years on the 24th of may year 1677 has come to expire'

### 6.4.3 False negative

The second most common error category is false negative, with both models performing similarly in this regard. This indicates that each model has difficulty identifying certain semantic roles.

**Prediction:**
**Text:**
'soo gaat bij desen sij requeste tot uEd = le Hooge agtb = re over,...'

**False negative + label confusion:** B-Instrument, O, O, O, O, O, O

**Gold:**
**Text:**
'soo gaat bij desen sij requeste tot uEd = le Hooge agtb = re over,...'

**Labels:** B-Patient, I-Patient, I-Patient, I-Patient, I-Patient, I-Patient, I-Patient

Figure 6.10: False negative - both models
*Transliteration: 'so hereby his request to your noble greatly honorable over,...'

In Figure 6.10, the difficulty of the VOC data is illustrated, where multiple symbols have to be identified as *I-Patient*, which both models fail to do. This shows the models struggle with representing the content of the data correctly. It also contains a double error where the models predict the incorrect label for the token they do recognize as belonging to this role mention. This may also add to the models false negative predictions for the other tokens belonging to the role mention.

In Figure 6.11, there is only one semantic role that belongs to the predicate. Both models do not predict any role, suggesting the models may struggle with identifying minimal role structures.

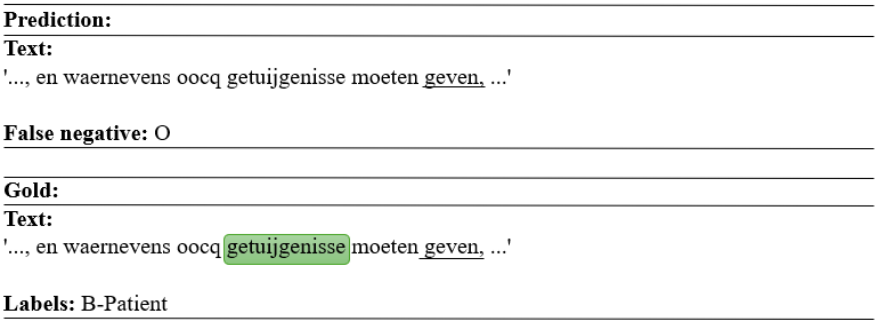



Figure 6.11: False negative - both models
*Transliteration: '... and besides also testimony must give ...'

### 6.4.4 Label confusion

Both models exhibit nearly equal levels of label confusion, indicating a shared challenge in accurately distinguishing between the different semantic roles. In several cases, the predicted label is semantically similar to the gold label, such as *AGENT* and *AGENT-PATIENT*. This type of confusion demonstrates limitations of the models in learning fine-grained semantic distinctions in the training data.

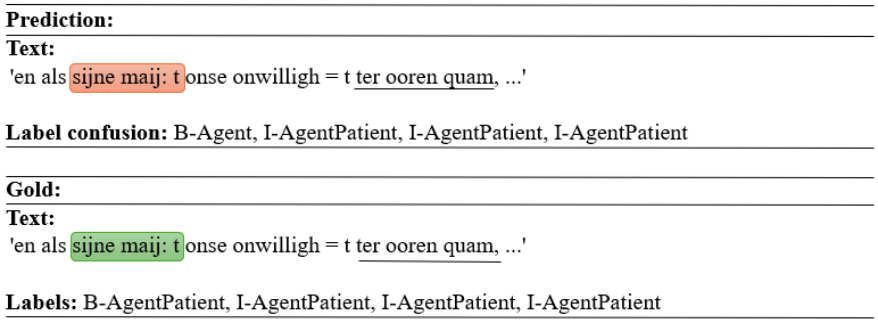



Figure 6.12: Label confusion - multitask model
*Transliteration: 'and if your majesty us unwillingly has heard...'

In Figure 6.12, an error made only by the multitask model is illustrated. It shows how the model confuses the labels *AGENT* and *AGENTPATIENT*, which are semantically related. However, for the tokens inside the span, the model does predict *I-AgentPatient*, showing the models confusion with representing the data correctly.

As mentioned, Figure 6.9, both contains a false positive error and a label confusion error. The label confusion error further shows how both models struggle with the distinction between semantically related labels, in this case *AGENTPATIENT* and *PATIENT*. This illustrates that the distinctions between these classes are not represented well enough.

### 6.4.5 Boundary error

The last type of error is the boundary error, which is not found in the single-task model predictions, and counts for only 2 percent of the mistakes made by the multitask model.

This shows that when the model correctly identifies the correct semantic role, it often also correctly represents the boundary tag.

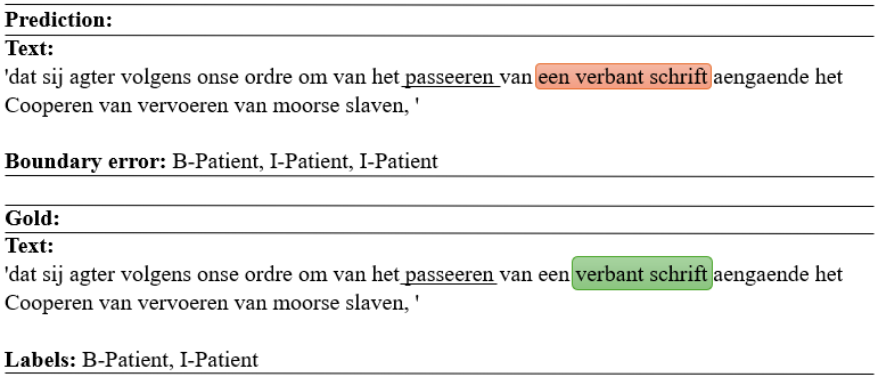



Figure 6.13: Boundary error - multitask model
*Transliteration: 'that they, following our order to pass an agreement regarding the buying and transporting of Moorish slaves '

In Figure 6.13, it is illustrated how the multitask model does not always capture the correct span of the role mention, where it makes mistakes correctly predicting which word is the beginning of the span. In this case, it adds an article (*'een'* or *'a'*) to the span, thereby incorrectly predicting the actual beginning of '*schrift*' with *I-Patient* instead of *B-Patient*. This is similar to the mistake in Figure 6.7, where an incorrectly labeled article leads to a boundary error. Again, the actual argument is correctly predicted as the role of the patient, so it can be seen as a minor error.

### 6.4.6 Conclusion

In summary, both models display similar types of errors in this document, despite the higher overall performance of the single-task model. Where the single-task model shows a greater tendency toward false positives, the multitask model exhibits a slightly higher proportion of boundary errors. The errors also show the models struggle with multiple error types in one text region or even one semantic role mention, indicating an overall limitation in the ability to correctly represent the data. This is in line with the overall low performance scores of both models. As with the previously analyzed document, the limited number of instances per semantic role class mean that small differences in predictions can lead to large shifts in evaluation metrics. Therefore, these findings should be interpreted with care, as they may not generalize beyond this document.

# Chapter 7

# Discussion and Conclusion

## 7.1 Discussion

This thesis aimed to investigate the effect of Multitask Learning (MTL) on the task of Semantic Role Labeling (SRL) using a domain-specific dataset consisting of documents from the Dutch East-India Company (VOC) archives, written in Early-Modern Dutch. The study compared single-task fine-tuning of SRL and Named Entity Recognition and Classification (NERC) with a MTL setup that jointly trained both SRL and NERC, using the same annotated VOC dataset, with annotations for both SRL and NERC.
Three large language models were evaluated: multilingual BERT (mBERT), XLM-Roberta (XLM-R), and GloBERTise, a domain-specific RoBERTa-based model pre-trained on unlabeled data from the same VOC archive. As discussed in Chapter 5, in the single-task fine-tuning for SRL, XLM-R outperformed mBERT and GloBERTise significantly outperformed both mBERT and XLM-R. This highlights the benefit of RoBERTa-architectures and domain-specific pre-training. Notably, GloBERTise significantly outperforms XLM-R, despite being trained on a substantially smaller corpus. This challenges the assumption that larger pre-training datasets always yield better performance.
In the multitask learning setup, mBERT and XLM-R achieved similar scores, while GloBERTise again outperformed both. However, only mBERT showed improvement in SRL performance from MTL. Both RoBERTa models did not benefit from the MTL implementation. This suggests that a BERT-based model like mBERT may benefit more from the MTL paradigm. This could be due to mBERT's smaller pre-training corpus, leaving more capacity to adapt and update its weights. In contrast, XLM-R's extensive pre-training may make it less flexible to adjust to new task-specific information, limiting its ability to update weights effectively for SRL. The GloBERTise model also did not benefit from the shared NERC information, even though GloBERTise was pre-trained on even less data than mBERT As GloBERTise is a domain-specific pre-trained model, it could be that the NERC task did not provide additional information beyond what was already captured during pre-training.
Moreover, the results of NERC show that even the best performing model, i.e. GloBERTise, achieves an F1-score $< 0.70$. This indicates a general difficulty in learning robust NERC representations, which may also limit the models' ability to generalize this information for improving the SRL task. Although mBERT improved from multitask learning, its improved performance still lagged behind the GloBERTise model. Hence, the integrated NERC information was insufficient to justify mBERT's use in future

implementations.
Moreover, performance varied widely across documents, reflecting the lack of structure and varying content of these documents. Smaller documents especially led to unstable results due to low class frequency, where a single correct or incorrect prediction could greatly affect score.
Overall, multitask learning did not reveal clear benefits. The overall higher NERC scores illustrate the inherent complexity of the SRL task in this domain-specific data. These and other difficulties with this research design are further discussed in Section 7.1.1.

### 7.1.1 Limitations

There are several factors that can be considered as limitations to the outcomes of this thesis project. Firstly, the Dutch East-India Company (VOC) dataset presents substantial challenges, particularly for the task of Semantic Role Labeling (SRL). The data contains a lot of variation and inconsistency, both in terms of linguistic structure and content. The data does not adhere to standard sentence boundaries, making it more difficult to separate the text in a way that aligns with typical SRL workflows. Additionally, the presence of duplicate text regions, each annotated differently depending on the focus predicate, add to the complexity. Due to limited training data, the model lacks sufficient exposure to learn how to differentiate between these varying annotations, and no alternative signal is available to resolve the ambiguity. Moreover, the structure of the texts differs from what is typically expected in SRL, with irregular syntax, unconventional phrase constructions, and inconsistent punctuation. These characteristics create a complex domain, which poses difficulties, even for a domain-specific pre-trained language model like GloBERTise. Therefore, accurately applying SRL to this dataset (and domain) is inherently more difficult than in more standard Natural Langue Processing (NLP) settings. In addition to these linguistic and structural obstacles, the dataset is also extremely small. The insubstantial size limits the ability of models to learn robust and generalizable patterns. It also increases the impact of noise, outliers or rare events on performance metrics, reducing the stability of results across different runs or configurations.
Another important limitation lies in the implementation of Multitask Learning (MTL). There are multiple ways to design and implement MTL frameworks, ranging from more simple shared-encoder configurations to more advanced architectures that dynamically balance learning across tasks. The current work implements a relatively simple and straightforward design, which, while functional, may not exploit the full potential benefits of MTL. It proved difficult to find exact or very similar implementations of this research design, constraining the ability to build a more advanced setup. Other work has investigated more advanced MTL techniques, such as gradient normalization (Chen et al., 2018), to address possible limitations like task interference. However, these possible improvements were thus not explored in this study due to limited time, knowledge and resources.
Furthermore, MTL has rarely been applied in the specific context investigated in this project: the combination or SRL and NERC, using transformer-based models on domain-specific historical texts. Most related studies focus on modern languages, other domains, and often utilize multiple datasets to improve generalizability. As a result, the findings from these studies may not be directly applicable to the current research context, and the challenges that are faced in this project may be more domain-specific.

Finally, the training configurations used in this study were optimized only based on the single-task SRL setup. It could be possible that different hyperparameter settings that were tailored specifically to the MTL setup, could have yielded better results. Parameters such as learning rate and epoch size could significantly influence the effectiveness of MTL, potentially limiting the ability to realize the full potential of the MTL architecture. These and other possible improvements are further discussed in Section 7.1.2.

### 7.1.2 Future work

Different directions for future research can be taken based on the findings of this thesis. One area to explore is to enrich the model input with additional information. In the current setup, the only signal provided to the model is the special token '[PRED]', which marks the predicate. Future work could explore the inclusion of more specific semantic cues, such as indicating the event class, e.g *'Translocation'* or *'Buying'*, to indicate which semantic roles are associated with this type of event. For example, *'Translocation'* is associated with an *INSTRUMENT* and *'Buying'* with a *BENEFICIARY* role. This could help the model better anticipate the role types relevant to each event.
Another potential direction involves an alternative approach to Semantic Role Labeling (SRL), given the unique structure and content of the domain-specific data. Rather than relying on text region analysis, which can be problematic due to the lack of clear sentence boundaries and the irregular syntax, future work could consider document-level modeling. This may allow models to improve in capturing contextual relationships across larger textual units, which could lead to more accurate role predictions.
In addition, another aspect that can be investigated is the refinement of the Multitask Learning setup (MTL). The current study implemented a relatively simple shared-encoder framework, however, a more advanced configuration could lead to better results for SRL. There are multiple techniques to look into, such as dynamic task weighting, auxiliary loss balancing, and gradient normalization. These methods aim to reduce interference between tasks. Another extension could be to expand the MTL setup by including the task of event and event class recognition as a third task. This is an alternative to feeding this information directly to the model. In doing so, the model could learn event patterns and class distinctions implicitly, which can aid in its ability to detect event arguments. Furthermore, hyperparameter tuning specifically for the MTL setup, instead of those optimized for single-task fine-tuning of SRL, may lead to improved performance.
The results also indicate the importance of data quantity and quality. The combination of a limited amount of labeled data together with variability in the annotated dataset posed clear challenges in this study, especially for a complex task like SRL. The sharing of knowledge by using MTL did not suffice to overcome this limitation. Expanding the dataset would improve the training process of a model and with that its ability to learn more robust and generalizable language representations. As the GLOBALISE project is ongoing, the annotation process of the data is also still in progress. Moreover, possibilities of automatic data augmentation techniques should be explored.
It should also be noted that a relatively strict evaluation method was used in this study, more specifically a word-level evaluation that distinguishes between span boundaries (*B-* versus *I-*) labels. While this allows for fine-grained analysis, it also increases the chances of penalizing minor boundary errors or partial detections of role mentions, which may not significantly affect the overall usefulness of the output. In the broader

context of this research, given the challenges of working with the domain-specific data and the overarching goal of extracting useful information from the data in the VOC archives, the evaluation may be unnecessarily limiting. Future work could consider easing these constraints.

## 7.2 Conclusion

Relating back to the research question and specific subquestions:

**Research Question**

When working in a low-resource, domain-specific scenario, does multitask learning with NERC and SRL data increase performance of pre-trained Transformer models, compared to solely fine-tuning on SRL?

**Sub Questions**

1) Does MTL with NERC and SRL data improve SRL performance for texts from the VOC archives, written in Early-Modern Dutch?

2) Which pre-trained Transformer model performs best for the task of SRL with and without MTL?

These sub-questions aim to evaluate the impact of multitask learning on semantic role labeling performance in a low-resource, domain-specific setting. Additionally, they assess the suitability of different pre-trained Transformer models for processing VOC texts. These insights can aid in the development of robust tools for domain-specific archival research.

The findings suggest that the multilingual BERT model improves Semantic Role Labeling (SRL) performance through the multitask learning implementation compared to the single-task fine-tuning of SRL. However, a domain-specific RoBERTa-based model (GloBERTise) proves to be the best model for the task of SRL utilizing the VOC data. That said, significant improvement is necessary to satisfy the goal of a research interface of the VOC data. To improve this, a multitask learning framework has not shown to be beneficial for this model, suggesting the need for other or improved methods to boost model performance.

# References


C. Abreu and E. Oliveira. Feup at semeval-2018 task 5: An experimental study of a question answering system. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 667–673, 2018.

A. Alajrami and N. Aletras. How does the pre-training objective affect what large language models learn about linguistic properties? *arXiv preprint arXiv:2203.10415*, 2022.

A. Arbaaeen and A. Shah. Natural language processing based question answering techniques: A survey. In *2020 IEEE 7th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pages 1–8. IEEE, 2020.

S. I. Arnoult, L. Petram, and P. Vossen. Batavia asked for advice. pretrained language models for named entity recognition in historical texts. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 21–30, 2021.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, 1998.

D. Berrar. Cross-validation. -, 2019.

T. Beysolow II. *Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing*. Apress, 01 2018. ISBN 978-1-4842-3732-8. doi: 10.1007/978-1-4842-3733-5.

R. Caruana. *Multitask Learning*, pages 95–133. Springer US, Boston, MA, 1998. ISBN 978-1-4615-5529-2. doi: 10.1007/978-1-4615-5529-2_5. URL `https://doi.org/10.1007/978-1-4615-5529-2_5`.

H. Chen, M. Zhang, J. Li, M. Zhang, L. Øvrelid, J. Hajič, and H. Fei. Semantic role labeling: A systematical survey. *arXiv preprint arXiv:2502.08660*, 2025.

J. Chen, D. Tam, C. Raffel, M. Bansal, and D. Yang. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, 11:191–211, 2023.

Y. Chen, T. Chen, S. Ebner, A. S. White, and B. Van Durme. Reading the manual: Event extraction as definition comprehension. *arXiv preprint arXiv:1912.01586*, 2019.

Y. Chen, Z. Ding, Q. Zheng, Y. Qin, R. Huang, and N. Shah. A history and theory of textual event detection and recognition. *IEEE Access*, 8:201371–201392, 2020.

Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR, 2018.

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

A. Conneau and G. Lample. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32, 2019.

A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale, 2020. URL `https://arxiv.org/abs/1911.02116`.

G. Crichton, S. Pyysalo, B. Chiu, and A. Korhonen. A neural network multi-task learning approach to biomedical named entity recognition. *BMC bioinformatics*, 18: 1–14, 2017.

W. De Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim. Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*, 2019.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Strassel, and R. M. Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon, 2004.

S. Ghidalia, O. L. Narsis, A. Bertaux, and C. Nicolle. Combining machine learning and ontology: A systematic literature review. *arXiv preprint arXiv:2401.07744*, 2024.

K. P. Gunasekaran. Exploring sentiment analysis techniques in natural language processing: A comprehensive review. *arXiv preprint arXiv:2305.14842*, 2023.

Y. Hong, W. Zhou, J. Zhang, G. Zhou, and Q. Zhu. Self-regulation: Employing a generative adversarial network to improve event detection. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 515–526, 2018.

R. Hu and A. Singh. Unit: Multimodal multitask learning with a unified transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1439–1449, 2021.

F. Ikhwantri, S. Louvan, K. Kurniawan, B. Abisena, V. Rachman, A. F. Wicaksono, and R. Mahendra. Multi-task active learning for neural semantic role labeling on low resource conversational corpus. *arXiv preprint arXiv:1806.01523*, 2018.

D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. Pearson, 3rd edition, 2025. URL `https://web.stanford.edu/~jurafsky/slp3/`. Online manuscript released January 12, 2025.

A. Khandelwal and S. Sawant. NegBERT: A transfer learning approach for negation detection and scope resolution. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5739–5748, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL `https://aclanthology.org/2020.lrec-1.704/`.

P. R. Kingsbury and M. Palmer. From treebank to propbank. In *LREC*, pages 1989–1993, 2002.

J. Li, A. Sun, J. Han, and C. Li. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70, 2020.

Q. Li, J. Li, J. Sheng, S. Cui, J. Wu, Y. Hei, H. Peng, S. Guo, L. Wang, A. Beheshti, et al. A survey on deep learning event extraction: Approaches and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Z. Li, S. He, H. Zhao, Y. Zhang, Z. Zhang, X. Zhou, and X. Zhou. Dependency or span, end-to-end uniform semantic role labeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6730–6737, 2019.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL `https://arxiv.org/abs/1907.11692`.

E. Manjavacas and L. Fonteyn. Non-parametric word sense disambiguation for historical languages. In *Proceedings of the 2nd international workshop on natural language processing for digital humanities*, pages 123–134, 2022.

A. Marasović and A. Frank. Srl4orl: Improving opinion role labeling using multi-task learning with semantic role labeling. *arXiv preprint arXiv:1711.00768*, 2017.

B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40, 2023.

V. Nasteski. An overview of the supervised machine learning methods. *Horizons. b*, 4 (51-62):56, 2017.

I. Niles and A. Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 2–9, 2001.

K. Pakhale. Comprehensive overview of named entity recognition: Models, domain-specific applications and challenges. *arXiv preprint arXiv:2309.14084*, 2023.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

S. Roy, F. Ogidi, A. Etemad, E. Dolatabadi, and A. Afkanpour. A shared encoder approach to multimodal representation learning. *arXiv preprint arXiv:2503.01654*, 2025.

R. Sharnagat. Named entity recognition: A literature survey. *Center For Indian Language Technology*, 1:1, 2014.

A. Søgaard and Y. Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, 2016.

R. Studer and S. Staab. *Handbook on ontologies*, volume 2. Springer, 2004.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

S. Verkijk and P. Vossen. Sunken ships shan't sail: Ontology design for reconstructing events in the dutch east india company archives. In *CEUR Workshop Proceedings*, page 320. CEUR Workshop Proceedings, 2023.

S. Verkijk, P. Sommerauer, and P. Vossen. Studying language variation considering the re-usability of modern theories, tools and resources for annotating explicit and implicit events in centuries old text. In *Proceedings of the Eleventh Workshop on NLP for Similar Languages, Varieties, and Dialects (VarDial 2024)*, pages 174–187, 2024.

Y. Wang, B. Yu, Y. Liu, and S. Lu. True-uie: Two universal relations unify information extraction tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1863–1876, 2024.

Y. Xiang and C. Li. A trigger-aware multi-task learning for chinese event entity recognition. In S. Li, M. Sun, Y. Liu, H. Wu, L. Kang, W. Che, S. He, and G. Rao, editors, *Chinese Computational Linguistics*, pages 341–354, Cham, 2021. Springer International Publishing. ISBN 978-3-030-84186-7.

S. Yang, D. Feng, L. Qiao, Z. Kan, and D. Li. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 5284–5294, 2019.

Z. Zhang, Y. Wu, H. Zhao, Z. Li, S. Zhang, X. Zhou, and X. Zhou. Semantics-aware bert for language understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9628–9635, 2020.

S. Zhao, T. Liu, S. Zhao, and F. Wang. A neural multi-task learning framework to jointly model medical named entity recognition and normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 817–824, 2019.

T. Zhu, J. Ren, Z. Yu, M. Wu, G. Zhang, X. Qu, W. Chen, Z. Wang, B. Huai, and M. Zhang. Mirror: A universal framework for various information extraction tasks. *arXiv preprint arXiv:2311.05419*, 2023.

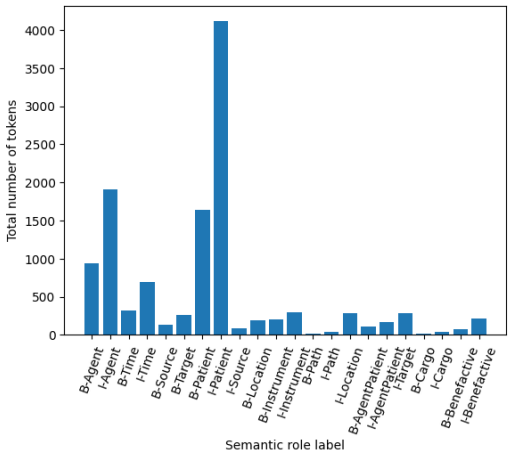
# Appendix

| **Participants** | **Definition** |
|---|---|
| Agent | Actor in an event who initiates, carries out or causes the event* |
| Patient | Undergoer in an event that is usually structurally changed, for instance by experiencing a change of state, location or condition* |
| AgentPatient | Entity that is simultaneously the causer and undergoer of an event |
| Beneficiary | Undergoer of an event that is (potentially) advantaged or disadvantaged by the event* |
| Cargo | Anything transported on a ship |
| Instrument | Undergoer in an event that is manipulated by an agent, and with which an intentional act is performed* |
| **Satellites** | |
| Time | Any indication of time |
| Location | Any indication of place/space |
| Source | Location where any type of Translocation event starts |
| Target | Location that is the (planned) endpoint of any type of Translocation event |
| Path | Location that is stopped at / crossed / encountered during any type of Translocation event |

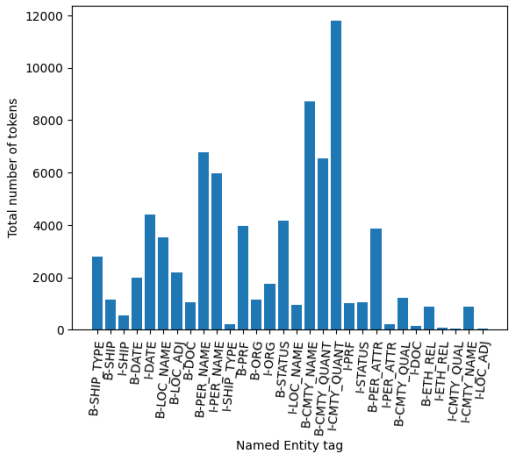
Figure 1: Semantic role classes explanation - adapted from: Event Annotation Guidelines GLOBALISE

| NER label | Description | Example | Related entities |
|---|---|---|---|
| PER_NAME | Name of Person | Tilling | persons |
| PRF | Profession, title | Conincx | persons |
| STATUS | (civic) status | veduwe, slaaf | persons |
| PER_ATTR | other persons attributes (than PER or STATUS) | manspersoon | persons |
| LOC_NAME | Name of Location | Sumatra | locations, polities |
| LOC_ADJ | Derived (adjectival) form of location name | Ternaats | persons, any (through qualification) |
| ETH_REL | ethno-religious appelation or attribute, not derived from location name | Moor, alfoerees | persons, any (through qualification) |
| CMTY_NAME | Name of Commodity | peper | commodities |
| CMTY_QUAL | Commodity qualifier: colors, processing | gemeen gebleekt | commodities |
| CMTY_QUANT | Quantity | 840 pikol | commodities |
| SHIP | Ship name | Abigail | ships |
| SHIP_TYPE | Ship type | jacht | ships |
| ORG | Organisation name | Compagnie, Comptoir | organisations, polities |
| DATE | Date | 14e, ultimo Februari | dates |
| DOC | Document | Jongste, brief | documents |

Figure 2: Named Entity classes explanation - adapted from:`https://github.com/globalise-huygens/annotation/blob/main/guidelines/ner-guidelines.md`




(a) Token distribution per BIO-tagged SRL class




(b) Token distribution per BIO-tagged NER class

Figure 3: Distributions BIO-tagged classes

| **Label** | **Single-task** | | | **Multitask** | | | **Support** |
|---|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 | |
| B-Agent | 0.13 | 0.25 | 0.17 | 0.20 | 0.25 | 0.22 | 8 |
| B-AgentPatient | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 7 |
| B-Benefactive | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1 |
| B-Cargo | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| B-Instrument | 1.00 | 1.00 | 1.00 | 0.50 | 1.00 | 0.67 | 1 |
| B-Location | 0.60 | 0.60 | 0.60 | 0.33 | 0.20 | 0.25 | 5 |
| B-Path | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| B-Patient | 0.33 | 0.42 | 0.37 | 0.43 | 0.75 | 0.55 | 12 |
| B-Source | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| B-Target | 0.33 | 1.00 | 0.50 | 0.00 | 0.00 | 0.00 | 1 |
| B-Time | 1.00 | 0.67 | 0.80 | 0.00 | 0.00 | 0.00 | 3 |
| I-Agent | 0.44 | 0.73 | 0.55 | 0.29 | 0.64 | 0.40 | 11 |
| I-AgentPatient | 0.00 | 0.00 | 0.00 | 0.67 | 0.17 | 0.27 | 12 |
| I-Benefactive | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 3 |
| I-Cargo | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| I-Instrument | 1.00 | 1.00 | 1.00 | 0.67 | 1.00 | 0.80 | 2 |
| I-Location | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1 |
| I-Path | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| I-Patient | 0.45 | 0.62 | 0.52 | 0.36 | 0.62 | 0.46 | 21 |
| I-Source | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| I-Target | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| I-Time | 0.50 | 0.20 | 0.29 | 0.00 | 0.00 | 0.00 | 5 |
| O | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 3419 |
| Accuracy | 0.97 | | | 0.97 | | | 3512 |
| Macro avg | 0.29 | 0.32 | 0.30 | 0.19 | 0.24 | 0.20 | 3512 |
| Weighted avg | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 3512 |

Table 1: Comparison of Single-task and Multitask Classification Performance on the document from 1747 - Fold 4
Note - the macro averages scores are not adjusted to not include the classes with zero support

| **Label** | **Single-task** | | | **Multitask** | | | **Support** |
|---|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 | |
| B-Agent | 0.33 | 0.27 | 0.30 | 0.38 | 0.27 | 0.32 | 11 |
| B-AgentPatient | 0.00 | 0.00 | 0.00 | 1.00 | 0.20 | 0.33 | 5 |
| B-Benefactive | 0.33 | 1.00 | 0.50 | 0.20 | 1.00 | 0.33 | 1 |
| B-Cargo | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| B-Instrument | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| B-Location | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| B-Path | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| B-Patient | 0.41 | 0.47 | 0.44 | 0.37 | 0.47 | 0.41 | 15 |
| B-Source | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 1 |
| B-Target | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1 |
| B-Time | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| I-Agent | 0.33 | 0.17 | 0.22 | 0.35 | 0.33 | 0.34 | 18 |
| I-AgentPatient | 0.60 | 0.60 | 0.60 | 1.00 | 0.40 | 0.57 | 5 |
| I-Benefactive | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| I-Cargo | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| I-Instrument | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| I-Location | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| I-Path | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| I-Patient | 0.37 | 0.48 | 0.42 | 0.27 | 0.41 | 0.32 | 27 |
| I-Source | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| I-Target | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 4 |
| I-Time | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |
| O | 0.99 | 0.98 | 0.99 | 0.99 | 0.98 | 0.98 | 3041 |
| Accuracy | | 0.97 | | | 0.96 | | 3129 |
| Macro avg | 0.15 | 0.17 | 0.15 | 0.24 | 0.22 | 0.20 | 3129 |
| Weighted avg | 0.97 | 0.97 | 0.97 | 0.97 | 0.96 | 0.97 | 3129 |

Table 2: Comparison of Single-task and Multitask Classification Performance on the document from 1679 - Fold 8
Note - the macro averages scores are not adjusted to not include the classes with zero support