ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

Natural Language Processing

NAMED ENTITY RECOGNITION FOR HISTORICAL ITALIAN TEXTS: OVERCOMING DATA LIMITATIONS THROUGH STRATEGIC ANNOTATION AND MODEL ADAPTATION

CANDIDATE
Simone Esposito

SUPERVISOR
Prof. Paolo Torroni

CO-SUPERVISORS
Nicolò Donati
Ciro D'Addio PhD

Academic year 2024-2025

Session 5th

dedicated to all the people who have inspired me with a love of growth and a passion for knowledge

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

The application of Named Entity Recognition (NER) to historical Italian texts presents unique challenges that remain largely unexplored in computational linguistics. This project originated from a practical need: developing a model capable of **reading and interpreting ancient Italian texts** to facilitate exploration and study by researchers and scholars.

As part of the **MAGIC** project by Netcom Engineering S.p.A. (where I worked as a consultant through T.P.SYSTEMS S.R.L.). It aims to realise an advanced platform for the use of digital content and services in the field of cultural heritage, with a focus on historically important library collections. The core of the project lies in the integration of state-of-the-art technologies: structured ontologies, generative artificial intelligence models (LLM) and immersive XR technologies (virtual, augmented and mixed reality).

A customized **search engine** with robust NER capabilities would significantly enhance this platform by enabling sophisticated contextual queries of digitized cultural heritage. Unlike traditional keyword-based systems, this approach offers a more intuitive experience by recognizing **domain-specific entities** such as historical authors, rare works, ancient geographical locations, and specialized bibliographic references.

To investigate this broad challenge, we focused on Italian texts from the 13th to 16th centuries—the period of "Volgare" development. This historical vernacular presents particular complexity due to its extreme linguistic variability, which evolved across different regions and time periods. The absence of standardized spelling and grammatical conventions further complicates linguistic analysis, as does the lack of annotated datasets and training resources. Additionally, historical entities often fail to align with modern NER categories, necessitating a more careful approach to entity recognition. Semantic shifts and archaic terminology introduce further difficulties, requiring expert interpretation to ensure accurate identification and classification of named entities.

One of the most significant obstacles we encountered was the **scarcity of annotated texts** suitable for training purposes. This limitation shifted our focus toward developing a customized NER approach rather than attempting comprehensive language understanding, necessitating deep philological knowledge to deal with linguistic variations and semantic differences across historical texts.

1.2 Research Objectives

The primary objectives of this research are:

1. **Develop a robust NER system for historical Italian** capable of identifying and classifying named entities in Volgare texts with high accuracy.

2. **Design and implement an annotation framework** that addresses the unique challenges of historical texts while balancing philological accuracy with technical feasibility.

3. **Create standardized entity type classifications** in collaboration with philologists that extend beyond traditional PER, LOC, and ORG categories to reflect the richness of historical contexts.

4. **Explore hybrid human-AI annotation approaches** testing the potential of large language models like Claude (by Anthropic) to accelerate corpus development while maintaining quality.

5. **Assess the degree of linguistic continuity between modern and historical Italian** by measuring how effectively models trained on contemporary Italian can recognize named entities in texts from the 13th-16th centuries (with zero-shot, few-shot and fine-tuning approaches).

6. **Explore the potential of ensemble approaches** to leverage complementary strengths of different model architectures, determining whether combining models with different recognition patterns can achieve better overall performance than any single approach.

This research represents an intersection of computational linguistics, digital humanities, and cultural heritage preservation, with potential applications extending beyond the immediate MAGIC project.

1.3 Project Scope

To address these objectives within manageable boundaries, the project scope has been defined as:

1. **Temporal focus:** Concentrating exclusively on Italian texts from the 13th to 16th centuries to limit the already substantial linguistic variability of Volgare.

2. **Corpus development:** Creating a representative dataset drawn from significant literary works of the period that captures key linguistic features and entity types.

3. **Annotation infrastructure:** Designing and implementing tools and methodologies for efficient entity annotation, with particular attention to handling historical language variants.

4. **Entity taxonomy:** Developing an expanded set of entity categories in collaboration with philologists that reflects the historical context and may serve as a standard for future research.

5. **Linguistic resources:** Constructing a specialized dictionary of synonyms and rules for archaic forms and language variations to aid in entity recognition and normalization.

6. **Validation pipeline:** Building a hybrid human-AI workflow that leverages both expert knowledge and modern language models (specifically Claude from Anthropic) for efficient annotation and validation.

7. **Model experimentation:** Testing various approaches to NER, with special focus on the potential of large language models in zero-shot and few-shot learning, as well as fine-tuning strategies for smaller, specialized models.

8. **Integration within a bigger project:** Highlighting how the developed NER capabilities will enhance the MAGIC platform's search functionalities and user experience.

This scope deliberately excludes comprehensive language understanding of Volgare, focusing instead on entity recognition as a foundational capability that can enable more sophisticated text processing in future work.

Chapter 2

Background and Literature Review

2.1 Historical Context

The reasons why any NLP task on vernacular texts is so difficult lie in the origins of the language itself. It is worth taking a brief look at these and linking the problems of modern philologists with the vernacular to technological and procedural ones.

The evolution of the Italian Volgare is a fundamental chapter in the history of European languages, marked by the **transition from Latin to Romance languages** through a centuries-long process of cultural stratification and social transformation. This process, which began with the **fragmentation of the Western Roman Empire** in the 5th century, reached its literary maturity in the 14th century with the production of Dante, only to stabilize as a national language in the 19th century.

The analysis of its linguistic characteristics reveals a system of constant compromise between written tradition and popular innovation, with significant variations due to **geopolitical and social influences**. The technical and methodological challenges for digital philology of the vernacular are therefore complex [19].

2.1.1 The Origins of Volgare

The Volgare language has its roots in vulgar Latin (*sermo vulgaris*), a linguistic variety spoken by the popular classes of the Roman Empire, as opposed to literary Latin (*sermo litterarius*), which was used by the educated elite. This separation was reflected in phonetic, morphosyntactic and lexical differences.

With the fall of the Roman Empire (476 A.D.), Vulgar Latin underwent a process of dialectal fragmentation accelerated by geography and influences from pre-Roman languages. In Italy, this diversification produced a constellation of distinct regional vernacular languages: Tuscan, Venetian, Lombard, Sicilian and others, each with phonetic and lexical peculiarities.

The first written evidence in Italian Volgare dates back to the 9th century, but the decisive transition came in the 13th-14th centuries with the Sicilian School, the *Dolce Stil Novo* and the three great Florentines: Dante, Petrarca and Boccaccio [19].

2.1.2 Dante, Petrarca, Boccaccio

Dante's *De vulgari eloquentia* (1303–1304) constitutes the first theoretical justification for vernacular literature. He described the characteristics of the illustrious Volgare (courtly, curial and cardinal), elevating it above others. Writing in Latin to legitimize his arguments among scholars, Dante asserted that Volgare was nobler than Latin because it was acquired naturally, not through education.

Francesco Petrarca proposed a model in the Tuscan Volgare for lyric poetry. His *Canzoniere* (1374) employed a meticulously curated lexicon, cleaning colloquialisms to achieve classical purity. Giovanni Boccaccio's *Decameron* (1353), on the other hand, showcased Volgare's versatility in prose [19].

2.1.3 Lorenzo de' Medici, Leon Battista Alberti

Lorenzo de' Medici's patronage reinvigorated Volgare literature in late-15th-century Florence. His *Canzoniere* and *Canti carnascialeschi* (carnival songs) incorporated dialectal elements, such as the diminutive *-ino* (e.g., *angelino*), fostering a playful yet sophisticated idiom.

Meanwhile, Leon Battista Alberti's *Grammatichetta vaticana* (c. 1443) attempted the first systematic grammar of Tuscan Volgare, addressing issues like verb conjugations (*amavo* vs. *amava*) and noun genders [19].

2.1.4 Pietro Bembo and the Questione della Lingua

Pietro Bembo's treatise resolved the *Questione della Lingua* debates by supporting 14th-century Tuscan as the ideal standard. He argued that Petrarch's poetry and Boccaccio's prose provided the purest models, free from contemporary "corruptions." Bembo's rules included:

- Preferring Tuscan phonetics (e.g., *fior* over *fiore*)
- Adopting archaic spellings (e.g., *homo* instead of *uomo*)
- Imitating Boccaccio's hypotactic syntax
- Emulating Petrarca's poetry

This enforced approach marginalized other dialects. Ludovico Ariosto revised his *Orlando Furioso* (1532) to eliminate Ferrarese traits, aligning with Bembo's Tuscan-centric norms.

Pietro Bembo's revised *Canzoniere* (1501) became a typographic benchmark, standardizing diacritics (e.g., *perchè* → *perché*) and apostrophes *(l'huomo*). By 1570, the *Accademia della Crusca*'s *Vocabolario* institutionalized this norm, cementing Tuscan as the basis of modern Italian [19].

2.2 NER for Ancient Texts

In this chapter we present all the research work that was done before starting the project, starting from the material available in terms of data sets and annotations up to the most similar NLP and NER projects that were developed for the ancient languages. Finally, we will also mention some projects whose subject matter is very close to that of this research but not compatible enough to provide some foundations, and considerations will be made on the choice of technology and datasets.

2.2.1 Pre-Existing Datasets

One of the very first steps in the preparation of the project was to analyse the resources currently available on the web, with a focus on datasets of historical texts that were compatible with the scope of the project and the target period.

In light of the research that was carried out, the following issues immediately arose:

- While there are datasets for historical Italian texts, **very few focus specifically on Italian Volgare** from the XIII-XVI centuries.
- The datasets that are most accessible aren't specialized enough for Volgare, and the specialized resources **aren't configured properly for NER training**.
- In general, **most resources lack specific APIs** or filters to help easily process the available data.

In the following paragraphs, I will dive deeper into each of the datasets I found available, and then proceed to the analysis and choice of my starting point for further processing.

Corpus OVI of Ancient Italian

The Corpus OVI of Ancient Italian [12] represents a comprehensive collection of Old Italian texts containing over 23 million word occurrences. Despite its impressive size and historical relevance, the corpus presents significant limitations for computational NLP tasks. Access to the corpus is severely restricted, with bulk downloading prohibited, which hampers large-scale data analysis. Additionally, the corpus lacks any pre-existing NER annotations, meaning that a complete manual annotation process would be required. The interface has been designed primarily for traditional philological research rather than computational processing, with limited API capabilities that make integration into modern NLP pipelines challenging. These limitations render the corpus less suitable for our specific NER training objectives.

Biblioteca Italiana

Biblioteca Italiana [6] offers a digital collection of more than 3,500 Italian literary works spanning from the Middle Ages to more recent periods. However, this breadth is also a limitation for our purposes, as the collection mixes texts from many periods rather than focusing specifically on XIII-XVI century Volgare. The collection lacks a standardized format that would facilitate NER extraction and includes significant Latin content alongside vernacular Italian. Furthermore, the collection consists primarily of literary texts, creating a genre bias by lacking administrative and legal documents that would provide a more comprehensive view of historical language use. As with most historical collections, it also lacks existing entity annotations, which would necessitate extensive manual labeling work.

RIALFrI

RIALFrI [24] serves as a digital repository of medieval Franco-Italian literature from Northern Italy. Its highly specialized focus on hybrid Franco-Italian

texts makes it unrepresentative of broader Volgare usage throughout the Italian peninsula. The corpus is geographically limited to Northern Italian traditions, missing the important central and southern linguistic variations. Its specialized linguistic focus on a unique hybrid language variety makes it unsuitable for general NER training across Italian Volgare. Additionally, its relatively small corpus size would be inadequate for robust model training in modern machine learning approaches, particularly for the data-hungry requirements of deep learning models.

CATMuS Medieval Dataset

The CATMuS Medieval Dataset [8][22] covers more than 200 medieval manuscripts dating from the 8th to 16th centuries. This broad chronological span dilutes Volgare-specific features by including texts from periods when Latin was still predominant or when modern Italian features were already emerging. The dataset also includes multiple languages beyond Italian Volgare, further complicating linguistic analysis specific to our period of interest. Its focus on paleographic features rather than textual content limits its utility for NER training, and inconsistent transcription practices across manuscripts create additional preprocessing challenges that would need to be overcome before the dataset could be effectively utilized.

ARTESIA Corpus

The ARTESIA Corpus [3] is dedicated to medieval Sicilian texts from the 14th to 16th centuries. While this period aligns with our temporal focus, the corpus is too regionally specific, containing only Sicilian texts whose dialectal features differ significantly from the Tuscan-based Volgare that came to influence standard Italian. The corpus has a relatively small size with limited entity variety, reducing its value for comprehensive NER training. Additionally, its specialized vocabulary would require substantial normalization preprocessing to align with broader Italian Volgare varieties, creating extra complexity in the

preparation pipeline.

TLIO (Tesoro della Lingua Italiana delle Origini)

TLIO [30] is a lexicographic database focusing on the earliest stages of the Italian language. Its dictionary-style entries, while valuable for lexical research, do not provide the continuous text needed for NER training. The database lacks the context windows necessary for entity disambiguation in historical texts, focusing primarily on lexical aspects rather than named entities. Its structure is optimized for linguistic reference rather than computational NER training, making it difficult to adapt to our research needs without significant reformatting and supplementation with contextual information.

M.I.DIA.

The MIDIA (Morfologia dell'Italiano in DIAcronia) corpus [20] is a diachronic collection of Italian written texts spanning from the early 13th century to the first half of the 20th century. It comprises approximately 7.8 million occurrences from around 800 texts, categorized into five chronological periods and seven textual typologies. Each word in the corpus is annotated with its lemma and part of speech (PoS). MIDIA presents an interesting potential for further developments of the project, since it stands as a tool for the study of the evolution of the Italian language over the centuries. At the time of writing this paper, the corpus appears to be inaccessible for technical reasons and thus unusable for our current purposes.

BERToldo Historical Corpus

The BERToldo Historical Corpus [11] [21] was used to train the BERToldo language model for historical Italian. While this suggests potential alignment with our research goals, the corpus was pre-trained for general language modeling rather than NER tasks specifically. Its context windows are optimized

for language prediction rather than entity recognition, and it prioritizes Part-of-Speech tagging over named entity boundaries. Additionally, it lacks gold-standard NER annotations that would be necessary for supervised learning approaches to entity recognition. Despite these limitations, the pre-trained model could potentially serve as a starting point for further fine-tuning.

HTRomance Medieval Italian Corpus

The HTRomance Medieval Italian Corpus [1] serves as a ground-truth corpus for Handwritten Text Recognition of medieval Italian manuscripts. It was designed primarily for transcript-to-image alignment rather than textual analysis, making it less suitable for NER training. The corpus has a relatively small sample size that would limit training effectiveness, and focuses on addressing handwriting recognition challenges rather than analyzing linguistic content. Like most historical corpora, it lacks entity annotations entirely, requiring substantial manual labeling before it could be utilized for NER tasks.

2.2.2 The Choice of Biblioteca Italiana

These limitations collectively demonstrate why creating a custom dataset through targeted **web scraping** of resources like **bibliotecaitaliana.it**, followed by manual annotation of named entities, represents a necessary first step. The existing resources either lack the specific focus on Volgare, don't have appropriate annotations, aren't accessible in computational formats, or have insufficient context windows for proper entity disambiguation in historical texts.

Against this background, the choice of bibliotecaitaliana.it presents a number of practical and technological advantages:

- It is in a web format that is relatively easy to extract
- The resources are in the public domain and can be used freely

- The selection of texts can be done with the support of specialist philologists, many of whom already use this resource
- Despite some bugs and performance issues, it is possible to easily explore the database with appropriate filters

Building **custom annotation tools** and developing specialized **preprocessing pipelines** addresses these gaps directly, allowing us to create a dataset specifically optimized for NER in Italian Volgare texts from the XIII-XVI centuries.

2.2.3 Technological Difficulties with Ancient Texts

Natural Language Processing (NLP) for ancient texts presents several significant technical challenges.

- Unlike modern languages with billion-token corpora, ancient language datasets are often **limited to a few thousand inscribed fragments** or manuscripts. The Relaciones Geográficas de la Nueva España project [13] demonstrated how even 16th-century colonial documents require **labor-intensive digitization and cleaning** before computational analysis. For older languages like Hittite, surviving texts may consist entirely of administrative records or ritual formulae, creating skewed training data for NLP models [7].
- Ancient languages often employ writing systems that challenge computational processing. Even alphabetic systems like those used for Ancient Greek exhibit right-to-left formatting and diacritical marks absent in modern languages. For example, the EvaCun shared task for Cuneiform processing highlighted the need for script normalization pipelines to handle variant glyphs across historical periods. Orthographic instability compounds these issues, as pre-modern texts lacked standardized spelling conventions [7]. A single Latin term like guerra (war) might

appear as werra or guera in medieval manuscripts, complicating tokenization [15].

- Ancient languages frequently exhibit morphological variety that challenges modern NLP architectures. Sumerian verb morphology encodes up to nine prefixes and suffixes per word, while Ancient Greek nouns have five declension patterns. Statistical models trained on analytical languages like English struggle to parse these structures. The Named Entity Annotation Projection study revealed persistent errors where the Akkadian term Amurru (a geographic region) was misclassified as a person due to morphological similarities to personal name patterns in aligned English texts. Latin's non-configurational word order allows subject-object-verb permutations that confuse dependency parsers optimized for SVO languages [31] [15].

- The **semantic shift** of words over centuries means modern computational tools struggle with historical word meanings.

- Manuscripts often contain **multilayered information** (main text, glosses, marginalia) that requires multimodal processing approaches.

- Optical Character Recognition (OCR) technology, while effective for printed modern texts, faces important challenges if applied to ancient manuscripts. These difficulties originate primarily from the inherent characteristics of **historical documents that diverge substantially from the training data** used for contemporary OCR systems. Brandon Hawk's experimental work with the Latin text Passio Petri et Pauli demonstrates the limitations of current OCR technology when processing medieval script forms [10].

Additionally, these texts often contain abbreviations, damaged sections, and specialized notation systems that require domain expertise to interpret.

Finally, the lack of annotated datasets for specific NLP tasks like Named Entity Recognition makes supervised learning approaches difficult to implement effectively.

2.2.4 Specific Technological Difficulties with Volgare

To complicate the already difficult challenge of applying machine learning techniques to ancient texts, Volgare presents additional problematic features for most approaches:

- Volgare existed as a constellation of regional varieties rather than a standardized language. Dante identified **14 distinct Italian vernaculars** in the 14th century, ranging from Sicilian to Venetian [17]. NER systems must consider regional spelling variants: a 13th-century Florentine document might render "Florence" as Fiorenza, while a Neapolitan text uses Firenze. The *Antichi documenti dei volgari italiani* corpus reveals how scribes **blended Latin case endings with vernacular roots**, producing hybrid forms like terram sancti Benedicti (land of Saint Benedict) where sancti retains Latin genitive morphology [18].

- Medieval Italian texts exhibit **fluid code-switching between Latin and Volgare** content. A 12th-century Tuscan charter might begin with standardized Latin (*In nomine Domini*) before transitioning to Volgare descriptions of land boundaries [15]. This hybridization extends to named entities—personal names often appear in vernacular (*Lapo di Ubertino*) while legal terms remain Latinized (curtis, mansus). The DIGIT historical corpus project found that 38% of named entities in early Italian texts require disambiguation between Latin and vernacular forms [18]. Modern NER models trained on monolingual data fail to parse these code-switched constructions without explicit bilingual training.

- **Orthographic inconsistency** is particularly severe in Volgare, with words

like “chasa/casa” or “febbraio/frairo” appearing interchangeably even within the same text

- **Regional variations** are extreme, as Volgare wasn’t a standardized language but rather a collection of regional dialects (Tuscan, Venetian, Sicilian, etc.) with significant differences in vocabulary and syntax
- Named entities pose special problems, as titles like “messer” or “ser” were fluid, and **person references might combine titles, given names, family names, and toponyms in inconsistent ways** (“Giovanni di Paolo da Firenze”)
- **Context windows** in standard NLP models are often too small to capture the elaborate sentence structures common in Volgare texts
- Entities referenced in Volgare texts often correspond to **historical realities absent in modern datasets**. The Comune di Firenze (Florentine city-state) or Arte della Lana (wool guild) don’t have equivalents in contemporary NER. Geographic names present particular challenges: medieval *Borgo San Lorenzo* might refer to a district now subsumed into modern Florence. Volgare’s dialectal variations are often hard to catch: a single location like *Monte Cassino* appears as *Montem Cassinense* in Latin charters and *Montecassino* in 14th-century Neapolitan vernacular [18].
- In general, it is also difficult from a philological point of view to follow the development of the vernacular, as oral testimonies at the time still outnumbered written texts. The evolution of the language is entrusted to spoken communication.

2.2.5 Related Projects on Old Italian Language

In order to have clear references for the work I was to carry out, I did extensive research on projects with similar objectives, both in Italian and other

languages.

Vocabolario della Grande Guerra (VGG) Corpus

The VGG corpus [16] represents one of the most substantial efforts to apply NLP techniques to historical Italian texts, focusing on World War I-era materials (early 20th century). Key results include:

- **Corpus Size**: Successfully compiled 500,079 tokens of annotated text from early 20th-century Italian sources.
- **Pre-processing Challenges**: 15% of tokens required manual correction due to OCR inaccuracies, particularly with archaic letterforms like ſ (long s) and ligatures. NER Performance: When standard NER models trained on modern Italian were applied to VGG texts, they experienced a 32% drop in F1-score compared to their performance on contemporary news articles.
- **Enhanced Annotation Schema**: Expanded the standard PER/LOC/ORG schema to include tags like IDEOLOGY and ABSTRACT_CONCEPT to better capture historical nuances.
- **Processing Pipeline**: Combined Transkribus for OCR, UDPipe for syntax, and human editors to correct dependency parses, achieving 91% accuracy on a 10,000-token subset.
- **Metonymic Usage**: Successfully identified complex cases where entities served symbolic roles (e.g., “Roma” referring to imperial rebirth concept rather than just the city).

Archivio Biscari letters

The Archivio Biscari project [28] focused on 18th-century Italian correspondence and demonstrated:

- **OCR Challenges**: Required manual normalization due to lack of gold-standard data for 18th-century Italian.
- **ChatGPT Assistance**: Used ChatGPT to help with normalization, reducing manual correction time by 60%, though still achieving only 78% accuracy despite iterative prompting.
- **Fine-tuning Results**: Fine-tuning BERT models on the Archivio Biscari letters improved named entity recognition by 18% compared to off-the-shelf models
- **Persistent Challenges**: Archaic spellings like "hoggidi" (modern "oggi") still caused 12% of false negatives even after specialized training.
- **Specialized NER Issues**: Ambiguities in archaic greetings (e.g., distinguishing between *Preg.mo Signore* and *Pregiatissimo*) required expert historian input.

Giacomo Leopardi's Zibaldone

Santini et al. (2023) [26] conducted pioneering work on Named Entity Recognition for 19th-century Italian through their study of Giacomo Leopardi's Zibaldone (1817-1832).

- **Scope:** Extraction and analysis of over 10,000 entity references focusing on three entity types (persons, locations, and literary works).
- **Sources:** Dataset of 260 evaluation notes and 688 training notes derived from the HTML markup of the DigitalZibaldone scholarly digital edition.
- **Methodology:** Comparison of zero-shot approaches using LLaMa3.1-8B (with both generative and extractive prompts) against a fine-tuned GliNER model based on BERT.

- **Results:** Fine-tuned models significantly outperformed zero-shot approaches (75.64% F1 with fuzzy matching vs. 38.74% for best zero-shot method); person entities were most reliably detected while literary works presented the greatest challenges due to lexical variations and historical referencing styles.

‘‘‘

2.2.6 Related Projects in Other Languages

The NIKAW Project

The NIKAW (Networks of Ideas and Knowledge in the Ancient World) [9] [14] project, though centered on classical antiquity, offers transferable insights for medieval studies. By applying BERT-based NER to 45,000+ Greek and Latin texts, NIKAW constructs social networks of intellectual influence. Key findings include:

- **Cross-genre performance variance**: Models trained on epigraphic texts achieved 79% F1-score on literary works, versus 62% on legal codes, due to formulaic language in the latter.
- **Temporal drift**: Entity disambiguation accuracy dropped 18% when applying models trained on Augustan-era texts to Late Antique materials, underscoring the need for period-specific fine-tuning

Large Language Models for Classical Languages

Recent ACL Anthology papers [5] highlight breakthroughs in applying LLMs to Ancient Greek and Latin. Riemenschneider and Frank (2023) trained four **monolingual BERT variants for Ancient Greek**, evaluating their performance on tasks like authorship attribution and textual criticism. Their Herodotus-BERT, fine-tuned on historiographical texts, achieved 88% F1-score in identifying interpolations in Thucydides’ manuscripts, outperforming rule-based

systems by 22%. However, the study cautions against over reliance on LLMs for low-resource languages, noting that models trained on **Classical corpora struggle with medieval Latin due to lexical and orthographic shifts** (e.g., ecclesia vs. chiesa).

Machine Learning for Sumerian

Applied BERT-based models to Sumerian texts [27] [4] for tasks like NER and textual restoration. The models were fine-tuned on annotated datasets derived from cuneiform inscriptions. The project advanced the understanding of Sumerian language structure while addressing challenges like sparse datasets. This fascinating example illustrates how pre-trained language models can be adapted to extremely low-resource ancient languages.

2.2.7 Extra Mentions

These mentions recognize projects that addressed similar subject areas in ancient Italian but did not directly involve NER.

VULGARIS

The Vulgaris project [33] aims to analyze the diachronic evolution and regional variations of the Italian language from 1200 to 1600. It studies how Italian developed from its medieval dialects into a more standardized language by examining historical literary texts. **The dataset is built from bibliotecaitaliana.it**, a digital archive of Italian literature. It includes poetry, prose, and correspondence from 104 authors, categorized into 14 literary families (e.g., Sicilian School, Stilnovisti, Tuscan Poetry). NLP Techniques Used:

1. Perplexity-Based Language Distance (PLD) – Measures linguistic similarity between different time periods.

2. Perplexity-Based Language Ratio (PLR) – Identifies whether older varieties are more complex than newer ones.

3. Neural Language Models (NLMs) – LSTM-based models conditioned on author, literary family, and text type to analyze language evolution.

4. Conditional Language Modeling – Examines how linguistic patterns change over time.

5. t-SNE Visualization – Clusters historical texts to reveal stylistic differences between periods and genres.

BERToldo

The BERToldo project [21] is a historical language model for Italian, inspired by BERT (Bidirectional Encoder Representations from Transformers). It was developed to process and analyze historical Italian texts spanning from **1200 to 1900**. BERToldo is trained on historical corpora sourced from repositories like Wikisource and Liberliber, with duplicate data removed to optimize training efficiency. The model supports tasks has been trained on **Part-of-Speech (PoS) tagging** demonstrating improved performance on historical texts compared to standard transformers. Multiple versions of BERToldo were created, tailored to specific historical periods (e.g., pre-1500, 1500–1700, 1700–1900), and all models are openly available to the research community.

Chapter 3

Methods

3.1 Data Collection and Preprocessing

3.1.1 Text corpus selection

The selection of texts for our historical Italian NER dataset involved a systematic process aimed at creating a **balanced and representative corpus** of Volgare literature from the 13th to 16th centuries. Working in collaboration with philologists specializing in historical Italian literature, we identified **60 major works** from the Biblioteca Italiana digital repository, carefully chosen to represent the evolution of the language across different periods, genres, and regional variations.

We prioritized canonical works of significant literary and historical importance, including Dante's "Vita Nuova" (1295), Boccaccio's "Decameron" (1353), and Petrarch's "Canzoniere" (1336-1374), which established the foundations of literary Italian. These were complemented by equally important but less frequently studied prose works such as Alberti's "Della famiglia" (1434), Machiavelli's "Discorsi sopra la prima deca di Tito Livio" (1531), and Guicciardini's "Ricordi" (1540).

To ensure geographical diversity, we deliberately included texts from different regional traditions: Tuscan works (which formed the majority due to

their historical significance in standardizing Italian), Venetian texts (such as Bembo's "Prose della volgar lingua"), works from central Italy (including Bruno's philosophical writings), and southern Italian compositions (like Masuccio Salernitano's "Novellino"). This regional balance was essential for capturing the dialectal variations characteristic of pre-standardized Italian.

Genre diversity was another critical selection criterion. We included chronicles (Villani's "Nuova cronica", Compagni's "Cronica"), philosophical treatises (Campanella's "La città del sole"), political writings (Machiavelli's "Il Principe") and personal correspondence (Strozzi's "Lettere ai figli esuli"). We deliberately **limited the inclusion of purely poetic works**, as they typically contain fewer named entities and often employ highly figurative language that complicates entity recognition. Additionally, we **excluded texts with substantial Latin content interspersed with Volgare** to maintain linguistic consistency in our corpus.

Chronological distribution was carefully balanced to span our target period (13th-16th centuries), with slightly higher representation from the 15th and early 16th centuries—a period of particularly rich prose production before the standardization imposed by the Accademia della Crusca. All selected texts were verified to be in the public domain and available through the Biblioteca Italiana digital archive, ensuring both legal compliance and accessibility for future research.

The resulting corpus of approximately 2 million words provides sufficient breadth and depth for meaningful entity annotation while remaining manageable within our resource constraints. This carefully curated selection establishes a foundation for developing NER tools specifically adapted to the linguistic characteristics of historical Italian texts.

3.1.2 Web Scraping and Text Normalization Pipeline

Web Scraping

The first technical challenge in building our historical Italian NER system was acquiring a sufficient corpus of texts from the target period (13th-16th centuries). I developed a customized web scraping tool to extract texts from bibliotecaitaliana.it. We prioritized this resource for its public-domain accessibility and alignment with philological best practices in Chapter 2.

To maintain dataset balance, we **limited poetry selections** (which often contain fewer named entities) and **excluded Latin texts** interspersed with Volgare. The final corpus comprised approximately 60 texts totaling over 2 million words, with deliberate redundancy in entity references to support model learning of orthographic variations for the same entities—a crucial feature given the unstandardized nature of historical Italian.

The scraper utilized web automation and HTML parsing technologies, enabling dynamic page loading and targeted content extraction. This approach proved essential since the texts were embedded within complex HTML structures and required JavaScript processing for proper access. The tool was designed to pause until pages loaded completely before extracting content, ensuring all dynamically generated elements were properly captured.

A crucial component was the HTML content extraction function that isolated the main text while filtering out navigation elements and modern editorial additions. This function identified the primary content container and processed text elements, implementing filtering mechanisms to remove duplicates and irrelevant content. The extraction process used a hierarchical approach that prioritized meaningful content blocks over formatting elements, preserving the authentic text while excluding contemporary editorial material.

The system was programmed to process a carefully selected list of over 60 URLs representing the most current versions of digitized texts in the Biblioteca Italiana database. I ensured inclusion of both canonical works and

lesser-known texts, providing us with linguistic and dialectal variety. This collection encompassed major works such as Boccaccio's "Decameron," the anonymous "Novellino," and Dante's "Vita Nuova," alongside less frequently studied but linguistically significant texts from the period.

This systematic extraction method gathered a comprehensive corpus of historical Italian texts that established the foundation for our NER annotation and model training processes.

Text Storage and Database Architecture

After extraction, texts were stored in both raw HTML format (for reference) and processed text format. I implemented a dual storage approach:

1. **File-Based Storage**: Texts were saved in a hierarchical directory structure with consistent naming conventions. I created dedicated folders for both raw HTML content and extracted plain text. Each file maintained a consistent naming scheme derived from its source identifier, ensuring traceability between original and processed versions.

2. **MongoDB Database**: For more flexible querying and annotation management, I also implemented a MongoDB schema that preserved text integrity while allowing for efficient annotation storage. Each document in the database contained the filename, full text content, comprehensive metadata (including source information, extraction date, and word count), and a dedicated array structure for storing annotations. This database architecture was specifically designed to facilitate complex queries across the corpus while maintaining a clean separation between text content and annotation data.

This dual storage approach facilitated both computational processing and human review, since texts could be accessed through database queries or directly opened in text editors for validation.

Text Normalization and Cleaning

Table 3.1: Character normalization mapping for preprocessing historical Italian texts. This table shows the special characters encountered in the corpus and their standardized replacements used in the normalization pipeline.

Char	Unicode	Replacement	Description
’	U+2018	’	Left single quote → standard apostrophe
”	U+201C	”	Left double quote → straight double quote
”	U+201D	”	Right double quote → straight double quote
⟨	U+27E8		Mathematical left angle bracket → space
⟩	U+27E9		Mathematical right angle bracket → space
ô	U+00F4	o	Circumflex o → plain o
´	U+00B4	’	Acute accent → apostrophe
ï	U+00EF	i	Diaeresis i → plain i
â	U+00E2	a	Circumflex a → plain a
ê	U+00EA	e	Circumflex e → plain e
ρ	U+03C1	r	Greek rho → Latin r
σ	U+03C3	s	Greek sigma → Latin s
ü	U+00FC	u	Diaeresis u → plain u
°	U+00B0	o	Degree symbol → letter o
ε	U+03AD	e	Greek epsilon with accent → Latin e
ο	U+03BF	o	Greek omicron → Latin o
ω	U+03C9	o	Greek omega → Latin o
ι	U+03B9	i	Greek iota → Latin i
α	U+03B1	a	Greek alpha → Latin a
ð	U+00F0	d	Icelandic eth → Latin d

Historical Italian texts present **unique normalization challenges** due to inconsistent orthography, archaic characters, and digitization artifacts. One of the main difficulties of these texts compared to modern texts is the **diachronic entities** that prove extremely difficult to identify and normalise. This, in fact, would require research in its own right, as will be discussed in chapter 7.

I conducted a detailed analysis of special characters appearing in the dataset and developed systematic substitution rules. The analysis revealed numerous

orthographic variations and archaic characters requiring standardization to facilitate downstream NLP processing.

I developed a dedicated text preprocessing system to address these issues. The system implemented several text normalization strategies:

1. **Character Substitution**: Archaic characters like 'ç' were standardized to 'z' to reduce model confusion as detailed in Table 3.1

2. **Punctuation Normalization**: Adding spaces after punctuation and standardizing apostrophes to ensure consistent tokenization

3. **Word Segmentation**: Fixing improperly merged words (common in OCR output) through pattern-based detection algorithms that identified unconventional capitalization patterns within words

4. **Case Normalization**: Handling mixed-case patterns consistently to reduce vocabulary sparsity, including specific rules for detecting and correcting patterns like uppercase followed by lowercase ("AAAbbb") and lowercase followed by uppercase ("aaaBBB")

In total, the system performed more than 20,000 such corrections across the entire dataset. This normalization represents an effort to preserve linguistic features relevant to the period while eliminating digitization artifacts that could confuse NER models.

The complete pipeline transformed raw scraped texts into clean, structured datasets ready for annotation. This approach ensured that annotators would encounter consistent text formats despite the inherent variability of the source materials, while preserving the linguistic characteristics needed for accurate entity recognition in historical contexts.

By implementing this comprehensive scraping, storage, and preprocessing system, I created a foundation for our historical Italian NER project, overcoming significant technical challenges related to data acquisition and preparation for such texts.

3.2 Annotation Framework

At this point, the painstaking work of selecting the entities to be annotated and the methods and tools required for their annotation began. This required several trial and error approaches, until the systems and categories were refined to those that will be described in the following sections.

3.2.1 Entity Label Selection Strategy

The development of our **labeling taxonomy** for Volgare NER required careful consideration of both linguistic precision and technical feasibility. We designed a **first-layer classification system** consisting of nine primary categories: PER (persons), LOC (locations), FAM (families), POP (populations), OPR (creative works), DAT (dates), EVE (events), DOC (documents), and ORG (organizations). This primary taxonomy was deliberately crafted to **minimize ambiguity** and potential overlaps between entity types, facilitating a more reliable annotation process. For example, rather than labeling both "Dante Alighieri" and "Alighieri" as PER, we distinguished between the individual (PER) and the family name when used collectively (FAM), as in "gli Alighieri di Firenze." Similarly, we differentiated between specific organizations (ORG) like "Comune di Firenze" and locations (LOC) like "Firenze," even when they appeared in related contexts.

This approach was strategically designed to support an **eventual system of N independent single-label models** that could operate simultaneously on the same text without classification conflicts (see details in Chapter 7). Each specialized model would focus exclusively on recognizing entities from one category, avoiding the complexity of multi-class **disambiguation** and enabling more targeted fine-tuning. The intentional separation between these categories enables more consistent annotation, particularly important for historical texts where entity boundaries are often less clear than in modern language.

Our taxonomy is conceived as the foundation for a more sophisticated

multi-layered annotation system that will ultimately be enriched with a second layer of specialized sub-labels. This future enhancement will add granular classifications without disrupting the primary layer's integrity. For instance, the PER category will eventually include sub-classifications such as PER-REL (religious figures like "San Francesco"), PER-NOB (nobility, such as "Lorenzo de' Medici"), and PER-MYT (mythological figures like "Minerva"). Similarly, LOC will be refined with sub-types including LOC-CIT (cities), LOC-REG (regions), and LOC-GEO (geographical features). This two-tiered approach offers the advantage of maintaining a clean first-layer classification while allowing for the rich taxonomic detail that scholars require for advanced historical text analysis, creating a flexible framework that can evolve alongside our understanding of historical entity relationships.

3.2.2 Entity Types Description

Here's a concise description for each tag in our NER annotation system:

PER (Persone): Named individuals, including historical figures, divine/mythological entities when personified, saints, and titled individuals where the title is part of their identifier. Examples: "Lorenzo il Magnifico", "Dio" (when personified), "duca di romagna bertoldo orsini".

LOC (Luoghi): Physical and geographical locations, including cities, regions, buildings (with their proper names), and geographical features. Examples: "Firenze", "Romagna", "Chiesa di Santo Spirito", "Monte Cavallo".

FAM (Famiglie): Noble houses, notable merchant families, and collective references to families. Examples: "Medici", "Strozzi", "casa d'Este".

POP (Popolazioni): Historical and ethnic groups, regional populations, and cultural/religious groups when referring to the people. Examples: "Romani", "Fiorentini", "Ginnosofiste", "Tedeschi".

OPR (Opere): Artistic and literary works, including sculptures, paintings, literary texts, and musical compositions. Examples: "Divina Commedia",

"Nilo di Belvedere", "Triumpho primo dell'Amore".

DAT (Date): Temporal references, including specific dates, named historical periods, feast days when used as dates, and specifically indicated years/centuries. Examples: "XX agosto MDIX", "9 di aprile, nel 1454".

EVE (Eventi): Specific historical events, including battles, treaties, councils, and major historical occurrences. Examples: "Assedio di Firenze", "Quaresima" (when referring to the religious event).

DOC (Documenti): Historical documents, including official decrees, papal bulls, laws, statutes, and personal legal documents. Examples: "Testamento di Cosimo", "Bolla papale".

ORG (Organizzazioni): Institutional entities, including government bodies, religious orders, guilds, banks, and administrative institutions. Examples: "Signoria", "Arte della Lana", "Senato" (when referring to the institution).

These tags provide a comprehensive framework for capturing the complex **network of entities** that appear in historical Italian texts, reflecting the rich cultural, political, and social landscape of the period.

3.2.3 JSON Structure for Annotation Files

The annotation system for the historical Italian NER project uses a structured JSON format designed to efficiently store entity annotations while maintaining traceability and supporting validation workflows. Each annotation file follows a standardized schema that balances simplicity with comprehensive metadata storage. Each annotation file uses the high-level structure shown in Figure 3.1.

Field Definitions

Each annotation object within the array contains the following fields:

1. `text` (string): The exact text of the entity as it appears in the source document. This preserves the original spelling and form, crucial for

```
{
  "annotations": [
    {
      "text": "Dionisio tiranno siracusano",
      "start": 91395,
      "end": 91422,
      "label": "PER",
      "timestamp": "2025-02-18T20:48:16.774925",
      "origin": "manual",
      "verified": "no",
      "notes": ""
    },
    {
      "text": "Giovanni summo pontefice",
      "start": 539186,
      "end": 539210,
      "label": "PER",
      "timestamp": "2025-02-19T10:40:57.898406",
      "origin": "automatic",
      "verified": "no",
      "notes": ""
    },
    // Additional annotations...
  ]
}
```

Figure 3.1: Example of the annotation JSON structure used to store entity annotations for historical Italian texts. Each annotation includes the entity text, position, type, and metadata about its creation and verification status.

historical texts where spelling variations are common.

2. `start` (integer): The character position where the entity begins in the source text, using zero-based indexing.

3. `end` (integer): The character position immediately following the entity in the source text.

4. `label` (string): The entity type classification according to our taxonomy (PER, LOC, FAM, POP, OPR, DAT, EVE, DOC, or ORG).

5. `timestamp` (string): ISO 8601 datetime when the annotation was created, enabling chronological tracking of annotation work.

6. `origin` (string): Indicates whether the annotation was created manually by a human annotator ("manual") or through automated means ("automatic"), such as pattern matching or model prediction.

7. `verified` (string): Tracks whether the annotation has undergone validation ("yes") or is still awaiting verification ("no").

8. `notes` (string): Allows to specify if there are ambiguities or suggestions about this specify annotation.

This JSON structure provides a robust foundation for the annotation process, supporting both manual and automatic entity recognition while maintaining data integrity and enabling detailed analysis of annotation patterns across the historical corpus.

3.2.4 Custom Annotation Tools

The development of specialized annotation tools was crucial for building our historical Italian NER dataset. Given the unique challenges of Volgare texts, standard annotation frameworks proved inadequate, necessitating the creation of custom tools tailored to our specific needs. These tools were designed to

overcome common biases in annotation practices while maximizing dataset quality for NER model training.

I developed the basic annotation interface to support **keyboard shortcuts, automatic suggestion (words that start with capital letter are more likely to be entities), contextual highlighting, and improved entity visualization**. This reduced the cognitive load when working with complex texts by making entity types immediately distinguishable through color-coding. Second, I implemented a **context window approach** that presented annotators with manageable text segments while maintaining sufficient context for accurate entity recognition. This helped overcome the fatigue associated with processing lengthy historical documents. Perhaps most significantly, I developed a **pattern-based suggestion system** that could identify potential entities based on previously annotated examples. For instance, searching "@ de' Medici" in the command line tool allows to tag any string that matches "de' Medici" plus the previous word, which usually is a name that is part of the full name (PER).

Sparse Annotation Tool

The Sparse Annotation Tool addresses a fundamental problem in text annotation: the tendency to focus on document beginnings, which creates dataset bias. By selecting random text windows from throughout the corpus, this tool ensures our annotations capture the full variety of entity forms across documents. As historical writers often used different reference forms as texts progressed (formal names at the beginning, shortened versions later), this randomization was essential for proper coverage. The tool also implements an efficient **semi-automatic batch annotation feature**, allowing annotators to apply the same label to multiple occurrences of an entity simultaneously, dramatically accelerating the annotation process. Entities tagged through this

batch process receive an **"automatic" origin label** in the JSON structure, enabling separate evaluation of their reliability. This approach proved particularly valuable for consistent entities like place names and prominent historical figures that appear frequently across the corpus.

The batch annotation can be **limited to one single book** instead of being extended to the whole dataset. This choice can be meaningful since it circumscribes the labelling of that entity to that context/topic/author only, **preventing cases where different authors could use it with a different meaning** (and therefore associating it to a different entity type).

Dense Annotation Tool

While sparse annotation provides broad coverage, NER models also benefit from densely annotated text segments that capture entity relationships and contextual patterns. Our Dense Annotation Tool selects larger continuous text segments (typically 4000 characters) for comprehensive annotation of all entities within that window. These densely annotated sections serve two critical purposes: they provide rich **training examples showing how multiple entities interact within proximity**, and they create **gold-standard evaluation benchmarks** for assessing model performance.

Therefore, the Dense Annotation Tool can either add more annotations to the main database or store a densely annotated file separately, to be used in further stages of benchmark analysis and performance comparison among the models that have been considered in this work.

Hybrid Annotation Tool

The Hybrid Annotation Tool represents an innovative **middle ground**, selecting text windows that already contain at least one annotation and expanding the annotation coverage within that context. This approach leverages the insight that entities often appear in semantic clusters within historical texts—mentions of a person are frequently accompanied by references to their titles,

locations, or associated organizations. By focusing annotation efforts on these entity-rich passages, the tool efficiently captures entity relationships that help models learn **contextual patterns**. This relationship-focused annotation strategy is particularly valuable for training models to recognize the complex titular and familial structures common in historical Italian texts, where entities may be referenced through various forms and relationships. The tool presents annotators with an existing entity and its surrounding context, facilitating the identification of related entities that might otherwise be overlooked in a purely random approach.

In Depth: Context Windows Size

A critical design decision in our annotation tools involved determining the appropriate context window size to present to human annotators. This choice significantly impacted both annotation quality and efficiency, with different tasks requiring different window sizes.

For manual annotation of new entities, we implemented larger context windows (typically 400-500 characters surrounding the potential entity) to provide annotators with sufficient contextual information. This expanded context proved **essential for disambiguating complex cases**, particularly for entities with multiple potential interpretations depending on their usage.

In contrast, we employed significantly **smaller context windows** (typically 150-200 characters) **for validating batch annotations**. This choice was motivated primarily by efficiency considerations, as batch validation involved reviewing substantially larger numbers of potential entities. However, our subsequent analysis revealed that this reduced context sometimes led to **decreased annotation reliability** for semi-automatically generated annotations. The limited contextual information occasionally proved insufficient for proper disambiguation but fortunately cases of deep ambiguity are limited, and **some categories are more prone to ambiguity than others**.

3.2.5 Annotation Manager

The `AnnotationManager` class serves as the **core infrastructure** for the entire NER annotation framework, providing a centralized system for managing, validating, and storing entity annotations across the historical Italian corpus. This component was designed as a **universal layer** that abstracts the complexities of annotation management, ensuring data integrity while providing a consistent API for all annotation tools to interact with.

Key Features and Functionalities

The Annotation Management system implements several critical functions:

1. **Verification of Existing Annotations**: When initialized, the manager performs a comprehensive integrity check of all existing annotations, systematically identifying issues like text mismatches, overlapping entities, or invalid annotation properties. This verification process generates detailed statistics on the quality of existing annotations, tracking metrics such as total annotations, invalid entries, overlapping pairs, and text mismatches.

2. **Entity Validation**: Before storing new annotations, the system verifies that the annotated text actually matches the text at the specified position in the source document. This involves reading the source text, validating that the proposed annotation boundaries fall within the text range, and confirming that the selected text matches the annotation text (with case-insensitive comparison to accommodate historical spelling variations). If any discrepancies are found, the system rejects the annotation with a detailed error message.

3. **Overlap Prevention**: The system prevents conflicting annotations by checking for entity boundary overlaps. For each new annotation, it tests

against all existing annotations for that document, ensuring that annotation spans don't intersect. This preserves the integrity of the annotation set by maintaining clear entity boundaries, which is essential for training accurate NER models.

4. **Batch Processing**: For efficiency, the manager supports batch annotation operations while maintaining the same validation rigor. This functionality is particularly valuable for processing multiple instances of the same entity type, allowing annotators to rapidly expand the dataset while preserving data quality. The batch processor applies the same validation rules to each annotation in the set.

5. **Annotation Modification**: The system supports updating existing annotations while tracking changes. When an annotation is modified, the system generates a detailed change log identifying precisely which properties were altered (text content, label, boundaries, etc.), creating a transparent audit trail for annotation evolution.

Integration with Annotation Tools

The Annotation Management system provides a unified interface that all three annotation tools (Sparse, Dense, and Hybrid) use to interact with the annotation data. This ensures consistent validation rules and data formats regardless of which tool created the annotation. When an annotator marks an entity using any of the tools, the annotation is passed to the manager for validation and storage.

This centralized approach ensures that all annotations, regardless of their source, undergo the same rigorous validation process before being added to the dataset. It also simplifies the development of specialized annotation tools, as each tool can focus on its specific annotation strategy without needing to reimplement common validation logic.

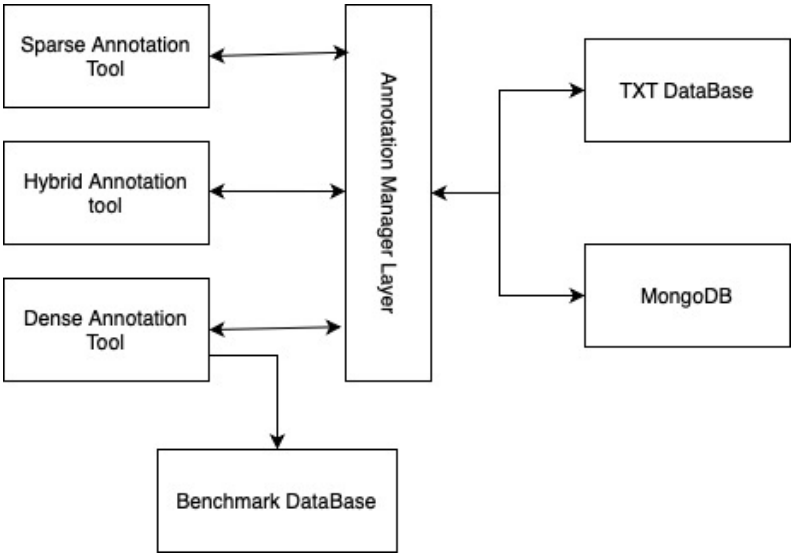


Figure 3.2: Integration of the Annotation Management system with the various annotation tools. The centralized architecture ensures consistent validation and storage of all annotations regardless of their source.

The Annotation Management system represents a critical architectural component that enabled the systematic collection of high-quality annotations across the historical corpus. By centralizing validation logic and providing a consistent interface, it eliminated many common sources of annotation errors while facilitating the development of specialized annotation tools tailored to the unique challenges of historical Italian text processing.

3.2.6 First Round of Annotation

To counteract the absence of meaningfully annotated datasets, I had to start manually creating annotations using only my knowledge of philology, with sporadic support from experts. This allowed me, using the tools I developed and described above, to manually or semi-automatically annotate around 4000 entities. The analysis of this first round of annotations has already led to significant results in terms of text and task comprehension. Visualising these results has allowed me to more precisely **identify the next steps** and plan the consequent strategies up to the development of the annotation workflows to efficiently expand the dataset.

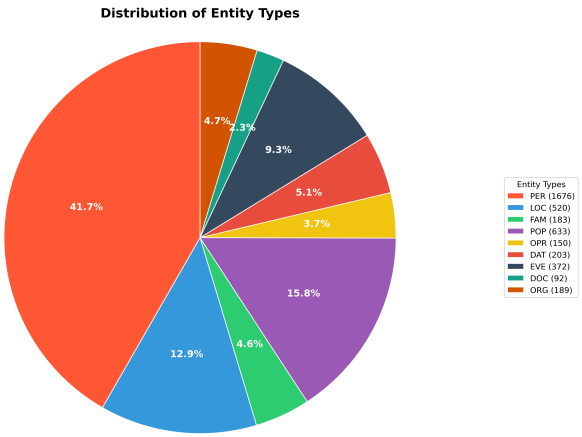


Figure 3.3: Distribution of entity types in the annotated corpus.

This pie chart reveals the overall composition of the annotated corpus:

- Person (PER) entities dominate at 41.7% (1,676 annotations), reflecting the human-centered nature of historical texts.
- Population (POP) entities form the second largest category at 15.8% (633), followed by Location (LOC) at 12.9% (520).
- Less represented categories include Document (DOC) at 2.3% (92) and Creative Works (OPR) at 3.7% (150).
- This imbalance may pose **challenges for model training**, potentially requiring strategies like weighted sampling or data augmentation for underrepresented classes.

- The difference distribution of the entities of a book also helps to classify the book in terms of subject and focus. This tool alone could be used for better categorisation of books in the dataset.
- Some books, because they are **more varied in terms of entity distribution**, or because they simply present a higher density of entities, are more valuable for annotation purposes.

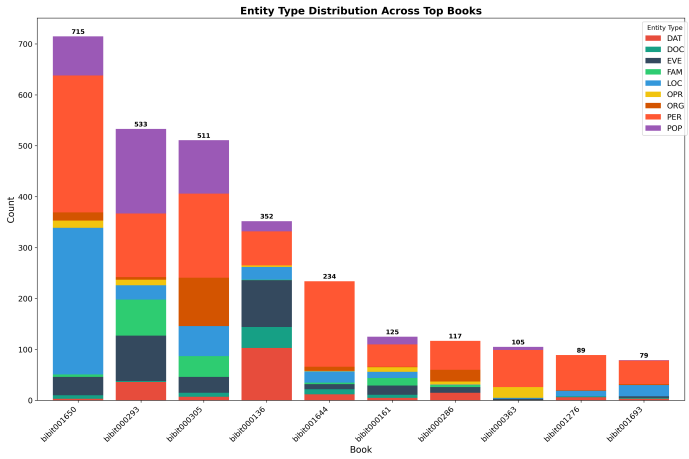


Figure 3.4: Entity type distribution across different books in the corpus.

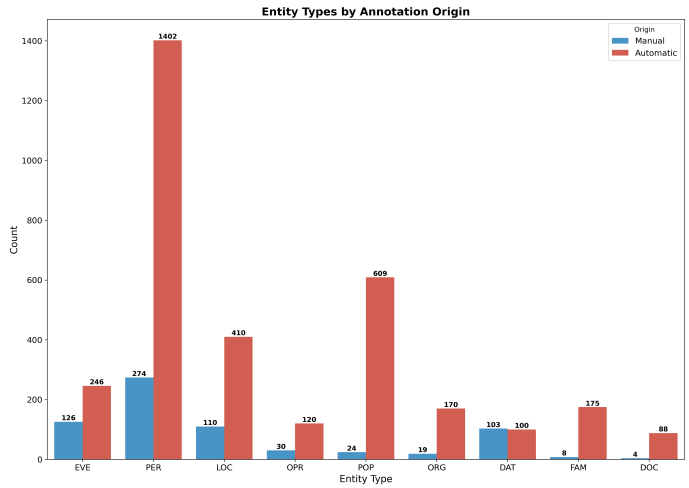


Figure 3.5: Distribution of entity types by annotation origin (manual vs. automatic).

- 82.6% of all annotations (3,320) were created in batch, while 17.4% (698) were made by one-by-one annotation.
- Some entity types are more inclined to be annotated by automatic methods (PER, LOC, FAM) while others require more careful annotation methods (DAT, EVE).
- This highlights the importance of maintaining **quality control over the automatic annotations**, as they form the vast majority of the training data.

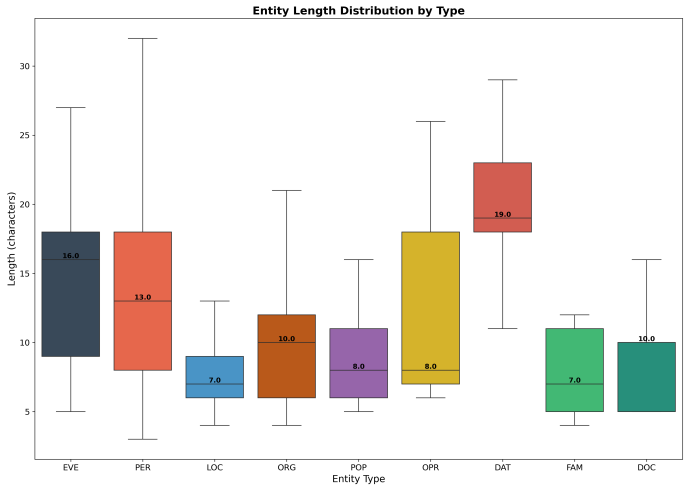


Figure 3.6: Distribution of entity lengths by entity type.

- Date (DAT) entities show the longest median length (19 characters), which aligns with the verbose date expressions common in historical Italian texts (e.g., "il ventesimo giorno di febbraio dell'anno MDIX").
- Event (EVE) and Person (PER) entities also show substantial length (medians of 16 and 13 characters respectively), reflecting the descriptive nature of historical event references and the use of titles and honorifics with person names.
- This information will be important when choosing **tokenization strategies**.

Briefly, the annotation strategy successfully exploited minimal manual effort (17.4%) to produce a substantial corpus of over 4,000 annotations. The entity type distribution shows significant imbalance, with person names comprising over 40% of all annotations. **This imbalance will need to be addressed in the model training approach.** Entity length varies considerably by type, which has implications for tokenization and model architecture decisions. Different books contain varying concentrations of entity types, suggesting potential value in book-specific model tuning. Automatic annotation

methods were particularly effective for person names and population references, while dates and creative works required more manual annotation.

3.2.7 Annotation and Validation Workflows

The initial phase of manual annotation quickly revealed significant challenges in the process. Annotating historical Italian texts proved to be both **time-consuming and intellectually demanding**, requiring careful attention to contextual differences, historical references, and linguistic variations characteristic of Volgare texts. Each annotation required multiple decisions: identifying entity boundaries, classifying according to our established taxonomy, and maintaining consistency across similar entities in different contexts. These challenges necessitated a strategic rethinking of the annotation approach. It became clear that achieving our target corpus size would be impossible with purely manual methods while maintaining quality standards. This realization prompted the development of both **specialized tools and refined methodologies** to optimize the annotation workflow.

Human to Human Workflow

As the annotation corpus grew, ensuring quality became increasingly critical. I established a validation framework that recognized different **levels of annotation reliability** based on their origin and validation status. In the ideal scenario, annotations would be created or validated by **domain experts** in historical Italian literature and language. Such expert-validated annotations receive the highest confidence marker ("validated: yes_e"), reflecting their gold-standard status. These represent the pinnacle of annotation quality, combining linguistic expertise with deep domain knowledge.

However, recognizing that access to expert annotators was limited, I classified my own contributions as "human" (non-expert) annotations. While I developed familiarity with the texts and annotation guidelines, I lacked the

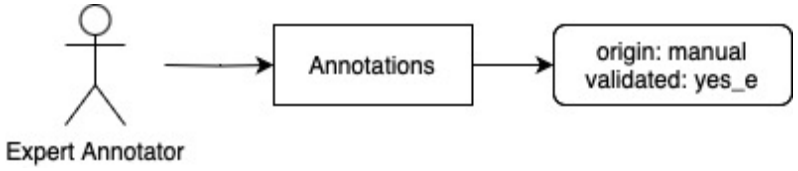


Figure 3.7: Human-to-human annotation workflow with expert validation.

specialized expertise in historical Italian literature that would qualify me as an expert validator. To accommodate this practical reality, I developed a **confidence ranking system** for different validation and origin combinations. This ranking would later serve to weight annotations during model training, allowing us to prioritize higher-confidence annotations while still utilizing the full corpus. **The system effectively creates a quality spectrum from expert-validated annotations (highest confidence) to unvalidated automatic annotations (lowest confidence)**. This approach meant that even without complete expert validation, we could proceed with model development by appropriately calibrating the influence of each annotation based on its reliability markers. The dotted line in our workflow diagrams represents this pragmatic stopping point—where annotation processing could proceed to model training with appropriate confidence weighting.

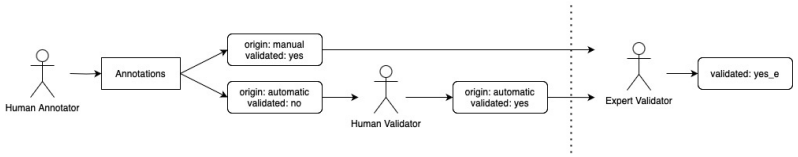


Figure 3.8: Human-to-human workflow with confidence ranking system.

To further increase annotation volume, I implemented a **batch annotation mode** in the annotation tool. This feature allowed for rapid annotation of multiple instances of the same entity type across a text. For example, once "Firenze" was identified as a location (LOC), all instances could be batch-annotated with a single command. While significantly faster than manual annotation, this batch approach introduced higher error rates, particularly in

cases where the same word form could represent different entity types depending on context (e.g., "Roma" as a location versus as a personification in certain literary contexts). **To maintain transparency about annotation provenance, these were marked with the "automatic" origin label**. These automatic annotations required validation to achieve confidence levels comparable to manual annotations. After validation by a human annotator, these automatic annotations were considered equivalent to manual ones in terms of reliability, as the validation process addressed the potential errors introduced during batch processing.

Human-to-AI Workflow

The next evolution in our workflow incorporated advanced AI systems—specifically **Claude 3.5 Sonnet**—into the validation process. This experiment served dual purposes: it accelerated validation while also allowing me to **evaluate the current capabilities** of large language models in understanding historical Italian texts.

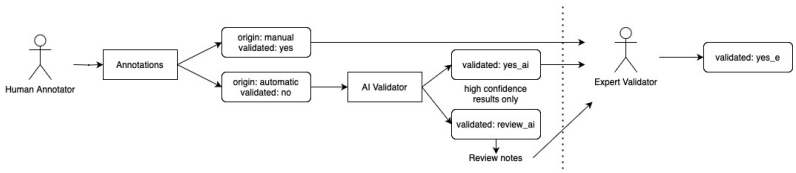


Figure 3.9: Human-to-AI validation workflow with confidence assessment.

I conducted experiments in both zero-shot and few-shot learning scenarios to determine how effectively Claude could validate entity annotations without extensive training on our specific corpus. The model demonstrated sufficient performance for understanding contextual clues in historical Italian, despite being primarily trained on modern language. The AI validation system was designed to produce two types of outputs:

- **High-confidence validations**, where the model was certain about its assessment

- **Low-confidence validations**, where the model expressed uncertainty

Low-confidence validations were either discarded or flagged for human review, preventing the propagation of unreliable judgments. High-confidence validations either confirmed the original annotation (changing status to "validated: yes_ai") or suggested changes that would require expert approval ("validated: review_ai"). This hybrid approach allowed us to leverage AI capabilities while maintaining quality control. AI-validated annotations occupied an **intermediate position in our confidence hierarchy**: more reliable than unvalidated annotations but less authoritative than human-validated ones.

AI-to-Human Workflow

The final workflow design explored an AI-first approach where the initial annotations would be generated by an AI system rather than created manually or through rule-based automation. In this scenario, the AI annotator would assess its own confidence level for each annotation. High-confidence AI annotations would be passed directly to expert validators, while low-confidence annotations would undergo human validation before reaching the expert stage. This approach would theoretically **maximize efficiency by focusing human effort only where the AI system expressed uncertainty**.

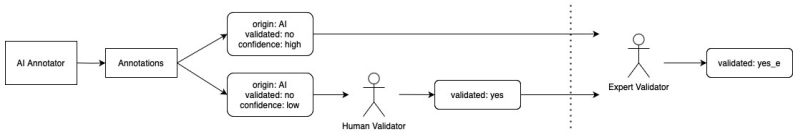


Figure 3.10: AI-to-human workflow with confidence self-assessment.

Implementing this approach presented a *chicken-and-egg dilemma*: we needed a well-performing NER model to generate initial annotations, but building such a model was the ultimate goal of our annotation efforts. This apparent paradox points toward the principle of a bootstrapping approach as a possible solution.

3.3 Towards a Bootstrapping Approach

As we have seen, the initial annotation process and the subsequent development of workflows suggests an approach that can aim to maximise the value of manual and semi-automatic annotations. In order to realise a system as an AI-to-human workflow, i.e. to arrive at an AI model capable of performing annotations correctly, we need to **start with a small model** to be refined through fine-tuning.

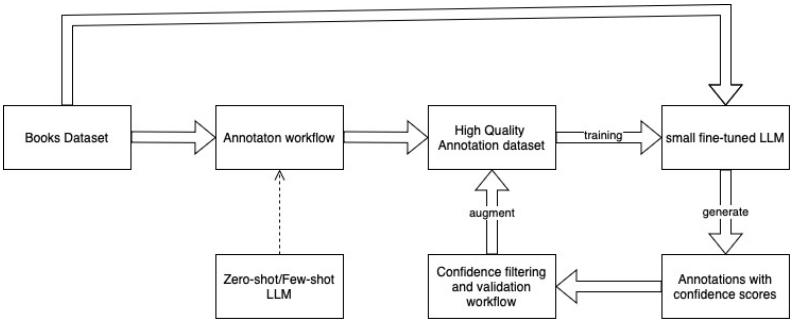


Figure 3.11: The bootstrapping methodology for iterative NER model improvement.

This introduces us to the bootstrapping methodology, which is illustrated here in this diagram. Bootstrapping then involves starting with the dataset of books selected for the Volgare using one of the **annotation systems** provided in the previous points, either completely human or optionally supported by an artificial intelligence, to produce a high quality dataset to be provided to a small LLM for fine-tuning. Such a model can then be used to perform automatic annotations to be validated. Evaluations occur mainly on annotations that the model reports to be **high confidence**, and through systems that may involve more or less experienced annotators and/or artificial intelligence, similarly to the annotation process, are filtered. The most valid ones provided to the initial dataset for re-training.

Here are the main steps of the Bootstrapping process:

1. A small seed set of high-quality annotations is created manually

2. These annotations train an initial, limited-capability model
3. This model generates annotations on new texts
4. Human annotators validate and correct these machine-generated annotations
5. The corrected annotations are added to the training data
6. A new, improved model is trained on the expanded dataset
7. The cycle repeats, with each iteration improving model performance

Through this iterative approach, the annotation process becomes **increasingly efficient** as the model improves, gradually shifting human effort from creation to validation. Each cycle increases both the size and quality of the annotation corpus while simultaneously enhancing model performance. This bootstrapping strategy represents an optimal **balance between manual effort and automation**, allowing the gradual development of a high-performing NER system for historical Italian texts despite initial resource constraints.

Chapter 4

Experiments

In this chapter, we will address the issue of technical experimentation of the solutions hypothesised above. Inevitably, these solutions were posed as exploratory experiments rather than definitive answers. This process was inevitable given the delicate nature of the task and the fact that there are no documented examples to date that point in the same direction. It was therefore necessary to make careful considerations regarding the models currently available. In particular, the choice had to fall on models that understand modern Italian, confident that the parallels between modern and Volgare are sufficient to make the transfer learning effective without having to build an LLM model ex novo. The techniques we have therefore used for these experiments are zero-shot learning, few-shot learning and fine-tuning, particularly with the bootstrapping technique. There is no doubt that with more vertical development work and a larger database, the results may be more promising. In any case, the project has built the foundations for further investigation.

4.1 Choice of models for testing

From the examples of previously developed and ongoing projects concerning the investigation of ancient languages, it is evident that BERT-based models are particularly promising for this purpose. For this reason, we wanted to

test a small BERT-based model for fine-tuning and bootstrapping. Other than fine-tuning a small model, I also wanted to test bigger models with a different approach: zero-shot and few-shot learning. These bigger models represent a good benchmark for the smaller fine-tuned model, helping a deeper understanding of the learning process.

4.1.1 Selected Models for Experimentation

Model: expertai/LLaMAntino-3-SLIMER-IT [32]

Base: LLaMA-2 7B (7 billion parameters)

Notes: SLIMER-IT is an instruction-tuned model specifically designed for zero-shot NER in Italian. It excels in extracting entities from text without requiring prior exposure to the dataset, making it ideal for historical texts where annotated data is scarce. The model's lightweight instruction tuning methodology, enriched with definitions and guidelines, enhances its ability to generalize to unseen entity types and out-of-distribution domains. This adaptability is critical for historical texts, which often contain archaic language and unique entity types not present in modern corpora.

Model: Claude 3.7 Sonnet [2]

Base: Anthropic (proprietary, more than 50 billion parameters)

Notes: The Claude models are generative pre-trained transformers developed by Anthropic, fine-tuned using Constitutional AI and reinforcement learning from AI feedback (RLAIF) for alignment with ethical principles and user needs. All Claude 3 models, including Sonnet and Opus, feature a 200k-token context window, significantly larger than most competitors like GPT-4 (32k tokens), allowing them to handle extensive input data while maintaining context.

Model: osiria/distilbert-italian-cased-ner [25]

Base: DistilBERT (66 million parameters)
Notes: The *osiria/distilbert-italian-cased-ner* model was selected due to its lightweight architecture and strong baseline performance on Italian NER tasks. Trained on the WikiNER dataset with additional fine-tuning on manually annotated paragraphs, this model achieves high precision and recall across standard entity classes (Person, Location, Organization, Miscellaneous). Its smaller size compared to full BERT models makes it computationally efficient for iterative fine-tuning experiments on limited datasets.

Model: nickprock/bert-italian-finetuned-ner [23]
Base: dbmdz/bert-base-italian-cased (110 million parameters)[29]
Notes: The *nickprock/bert-italian-finetuned-ner* model is a fine-tuned version of the ‘dbmdz/bert-base-italian-cased‘ model, specifically trained for Named Entity Recognition (NER) tasks in the modern Italian language. It was fine-tuned using the WikiANN dataset and achieves high performance metrics: precision of 0.9438, recall of 0.9542, F1-score of 0.9490, and accuracy of 0.9918 on the evaluation set. It was chosen as a high-level standard for BERT NER models and to identify the differences between small and large models on this task.

4.1.2 Use Cases

As already mentioned in the previous chapters, NER is not the only focus of this project. In addition to annotation, **LLMs intervene in workflows in a variety of ways**. The models mentioned above were therefore tested for different tasks. I reserved the most capable model, Claude, for the most complex tasks, avoiding testing the other smaller ones where the former already showed its shortcomings in understanding the Vulgar.

In particular, this was done:

- Data Augmentation via **Synonyms Vocabulary** generation (Claude via

Zero-Shot)

- Annotation Validation (Claude via Zero-Shot)
- NER (SLIMER-IT and Claude via Zero-Shot/Few-Shot, BERT and DistilBERT via Fine-Tuning)

The next experiments will show a range of possibilities with the latest models that can best perform NER with modern Italian. We will test context windows of variable size and other hyperparameters fine-tuning to proper compare the impact of some technological and strategic choices.

4.2 Synonyms Dictionary Generation with Claude

The creation of a **synonyms dictionary** represents a crucial component in the development of our Named Entity Recognition (NER) system for historical Italian texts. When working with historical languages, particularly Italian Volgare from the 13th to 16th centuries, we face significant challenges related to **linguistic variation**. The same entity might appear in multiple forms throughout texts - variations in spelling, the use of epithets, shortened forms, or completely different names referring to the same entity.

For example, in historical texts, we might find "Lorenzo de' Medici," "Lorenzo il Magnifico," and simply "Lorenzo" all referring to the same historical figure. Similarly, place names like "Fiorenza" and "Firenze" represent the same city. These variations pose a substantial challenge for NER systems, which need to recognize these different forms as referring to the same underlying entity.

The synonyms dictionary serves multiple purposes in our project:

1. **Data Augmentation**: By identifying synonyms, we can artificially expand our training dataset. If we annotate "Fiorenza" in one passage, we can automatically add annotations for "Firenze" in other contexts,

thereby increasing the volume of our training data without requiring additional manual annotation.

2. **Evaluation of Large Language Models**: This process also allowed us to assess Claude's understanding of historical Italian texts. By presenting Claude with lists of entities and asking it to identify potential synonyms, we could gauge its comprehension of historical language patterns and its knowledge of Italian history and literature.

3. **Research Value**: The creation of a synonyms dictionary for historical Italian entities has intrinsic research value for digital humanities scholars working with these texts. It provides insights into how individuals, places, and concepts were referenced in different ways throughout historical literature.

It's important to note that this component could have been developed into a much larger standalone project. A better approach might involve providing contextual information for each entity before assessment, creating embeddings for each entity mention and performing clustering analysis, or developing specialized models for entity linking. However, given our project's scope and resource constraints, we opted for a more streamlined approach using Claude's capabilities to accelerate the process.

4.2.1 Prompt Engineering

For the generation of synonym groups, we employed a **zero-shot prompting** approach with **Claude** shown in Figure 4.1. The prompt was designed to leverage Claude's knowledge of Italian history and language while providing clear instructions about the task requirements.

The prompt instructed Claude to analyze lists of entities belonging to specific categories (such as persons, locations, or organizations) and identify which entries likely referred to the same underlying entity. The prompt was structured in Italian and contained the following key elements:

Prompt per la generazione di gruppi di sinonimi:
Attingendo alle tue conoscenze filologiche, analizza questa lista di entità etichettate come {label} estratte da testi storici italiani scritti in Volgare tra il 13mo e 16mo secolo. Raggruppa insieme annotazioni che potrebbero riferirsi alla stessa entità.
Definizioni delle etichette:

PER (Persone): Questa etichetta è per gli individui nominati

LOC (Luoghi): Questo comprende i luoghi fisici e geografici

FAM (Famiglie): Utilizzato per famiglie nobili e notabili

POP (Popolazioni): Per gruppi storici ed etnici

OPR (Opere): Per opere artistiche e letterarie

DAT (Date): Per riferimenti temporali

EVE (Eventi): Per eventi storici specifici

DOC (Documenti): Per documenti storici

ORG (Organizzazioni): Per entità istituzionali

ENTITÀ:
{entities_list}
Formato richiesto:
Organizza i gruppi di sinonimi nel seguente formato JSON:

```
{"groups": [
["entità1", "entità2"], // primo gruppo di sinonimi
["entità3", "entità4", "entità5"], // secondo gruppo
// ... altri gruppi
]}
```

Regole:

- Includi solo gruppi con almeno 2 entità che sono sicuramente sinonimi
- Non raggruppare entità che sono solo vagamente correlate

Figure 4.1: Prompt for Claude to generate synonym groups from annotated entities. Variables in curly brackets are dynamically replaced during execution.

1. **Task framing**: We explicitly asked Claude to draw upon its philological knowledge to analyze entities from historical Italian texts written in Volgare between the 13th and 16th centuries, and to group annotations that might refer to the same entity.

2. **Entity category descriptions**: We provided brief descriptions of each entity type (PER, LOC, FAM, POP, OPR, DAT, EVE, DOC, ORG) to help Claude understand our classification system and contextualize the entities appropriately.

3. **Structured output requirements**: We specified a JSON format for the response to ensure we could easily parse and process the results programmatically, with groups of synonyms organized as nested arrays.

4. **Quality guidelines**: We instructed Claude to include only certain synonyms (at least 2 per group) and to avoid grouping vaguely related entities, emphasizing precision over recall in the synonym identification task.

5. **Italian language**: To allow the model to better enter the context and automatically set to the Italian language, the prompts are all in Italian, which helped maintain linguistic consistency between the task instructions and the entities being analyzed.

This zero-shot approach was sufficient for our purposes, but **could have been enhanced** with few-shot examples of correct synonym groupings. Such examples would have provided Claude with clearer patterns to follow, potentially improving the accuracy of its synonym identification, especially for more obscure or ambiguous entities.

4.2.2 Human Review

Claude's performance in generating synonym groups was generally good but not perfect. Approximately **20% of the identified synonym groups required**

manual corrections based on my domain knowledge of Italian history and literature. This highlights a significant limitation: while Claude possesses broad philological knowledge, it has gaps in specialized areas like historical Italian language variations.

Some common issues observed in Claude's output included:

1. **False synonyms**: In some cases, Claude grouped entities that share similar names but refer to different historical figures or places. For example, it might incorrectly group "Giovanni de' Medici" (who could be multiple different historical figures) with "Giovanni delle Bande Nere" (a specific Medici family member).

2. **Missed synonyms**: Claude sometimes failed to recognize less obvious synonyms, particularly when they involved nicknames, epithets, or highly variable spelling forms that were common in historical Italian but might not be prominently represented in Claude's training data.

3. **Contextual misunderstandings**: Without seeing the entities in their original context, Claude occasionally misinterpreted the nature of certain entities, leading to incorrect groupings.

4. **Inconsistent granularity**: Some synonym groups were too broad, while others were too specific, indicating Claude's uncertainty about how to delineate entity boundaries in certain cases.

These limitations highlight the importance of human review in such specialized linguistic tasks. While Claude provided a valuable starting point that saved significant time compared to a fully manual approach, **expert knowledge remains essential** for ensuring the accuracy of the final synonym dictionary, especially in specialized domains like historical language analysis.

4.2.3 Implementation

The implementation of the synonym dictionary generation tool centers around an Entity Analysis system, which manages the loading of annotations, analysis of entity distribution, and interaction with Claude for synonym identification. Here's an overview of the key components:

1. **Loading and Analysis**: The tool begins by loading all annotation files and analyzing the distribution of entity types. This process consolidates annotations from multiple files into a unified collection, tracking the number of annotations loaded from each source and handling any file access errors gracefully. The system maintains a comprehensive log of the loading process, providing visibility into the data acquisition phase.

2. **Entity Counting and Statistics**: The tool performs a detailed analysis of annotation distribution by entity type, origin (manual vs. automatic), and verification status. For each entity text, it maintains detailed statistics including total occurrence count, method of creation (manual vs. automatic), and verification status. This granular tracking enables sophisticated filtering and prioritization during the synonym identification process.

3. **Synonym Identification**: For each entity type, the tool sends a list of unique entities to Claude and processes the response. The process is managed separately for each entity category (PER, LOC, etc.), allowing for category-specific optimization. The system implements error handling and progress tracking, ensuring that failures in one category don't affect the processing of others.

4. **Response Parsing**: The tool extracts Claude's JSON output and converts it into a usable data structure. This involves identifying the JSON content within Claude's response, parsing it into a structured format, and transforming the raw data into optimized data structures (using sets

for efficient membership testing). The parser includes comprehensive error handling to manage potential issues with Claude's responses.

5. **Results Storage**: Finally, the tool saves the collected statistics and synonym groups to a JSON file for further use. The stored data includes comprehensive statistics about the annotation corpus, detailed entity counts by category, and the identified synonym groups. Before storage, the data is processed to ensure JSON compatibility, handling special data types like sets that don't have direct JSON representations.

For data augmentation purposes, the resulting synonym dictionary can be used to expand the training dataset by identifying instances where one form of an entity appears but hasn't been annotated, then creating **additional annotations** based on known synonyms.

While this implementation serves our immediate needs for data augmentation, it could be extended in several ways for more sophisticated entity linking applications, such as integrating contextual information or employing embedding-based similarity measures to complement Claude's knowledge-based approach.

4.2.4 Synonyms Statistics

Despite the total of only 102 synonym groups, I think it is of particular interest to visualise some of the data regarding the synonym groups identified.

It is particularly noteworthy, for instance, to note that in second place after PER are LOC and DAT as the entity types with the most synonyms. This perfectly reflects the fact that in ancient Italian there are many ways to write the same date and thus synonyms in that category tend to explode despite the small number of annotated dates. Places also tend to have many periphrases or equivalences in terms of entities.

Also interesting is the ratio between the number of unique entities found and the groups of synonyms in that group. 14% of the noted characters, 25% of the locations and 19% of the dates have at least one synonym.

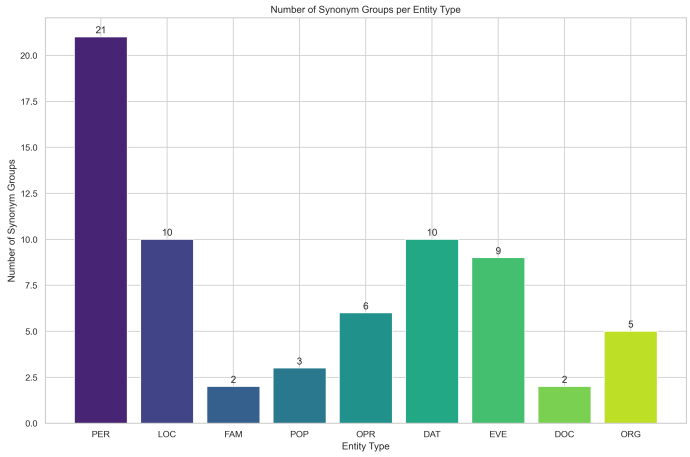


Figure 4.2: Distribution of synonym groups across entity types.

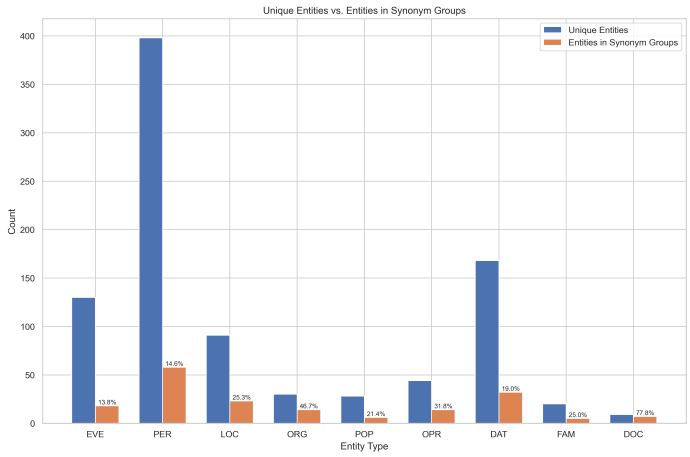


Figure 4.3: Number of entities included in synonym groups per type.

The presence of small numbers of unique identities denotes a greater use of semi-automatic annotations, so there are more likely to be groups of annotations referring to the same entity. Consequently, for small groups (less than 50 unique entities), I disregard synonym statistics.

The analysis of the average length of synonyms is also very interesting because it shows that, for example, for DAT there are more synonyms, on average, for the same date. Even for PER and ORG, it is evident that there are usually two or three synonyms. These differences seem marginal, but in fact when fully implemented they suggest different architectural solutions for each type of label. In fact, the trend of the precept will be, as we shall see,

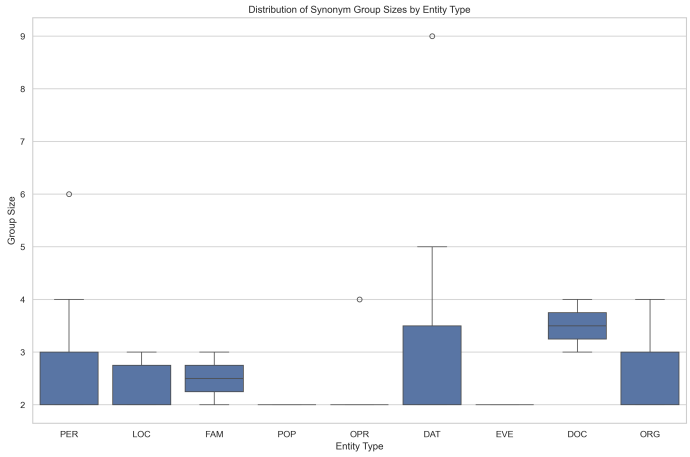


Figure 4.4: Average number of entities per synonym group by entity type.

towards customised solutions for each type of entity, so as to better adapt to the individual characteristics of the entities belonging to that group.

4.2.5 Augmented Synonyms

A final mention must be made of the fact that Claude was further asked to optimise the synonyms groups identified with the previous workflow. In particular, the list was expanded with the model's own knowledge concerning common entities of the period under analysis (13th-16th century), and the variants where certain entities can occur with lower or upper case were added.

4.3 Annotation Validation with Claude

In order to test the potentiality of a system like Human-to-AI workflow, as mentioned in the previous chapter under "Annotation Workflows", I developed a **simplified semi-automated annotation validation workflow** leveraging Claude's capabilities. This approach represents an exploration of how advanced language models might complement human expertise in specialized NLP tasks.

4.3.1 Annotation Validation Workflow

Figure 4.5: Human-to-AI-to-Human annotation validation workflow.

The workflow consists of three principal components working in sequence:

1. **Claude Validator**: This component examines annotations within their textual context, applying a set of **guidelines** to evaluate their correctness. For each annotation, Claude outputs a structured judgment along with suggested corrections when necessary.

2. **Validation Interpreter**: This component processes Claude's structured output, presenting judgments and suggested interactive modifications in a human-friendly format.

3. **Human Reviewer**: As the final step, a human validator reviews all suggested corrections, accepting or rejecting them based on linguistic knowledge and domain expertise.

This pipeline represents a practical implementation of AI-assisted annotation review, where the machine suggests potential improvements but the human has the last word over annotation quality. The approach allows for rapid processing of annotations while maintaining high standards.

4.3.2 Prompt Engineering

The effectiveness of Claude in this validation task relies mainly on the design of the prompt. The prompt serves as both **instruction and context**, guiding Claude toward structured and useful evaluations. The core validation prompt developed for this task is in Figure 4.6.

This prompt has several important characteristics designed to optimize Claude's performance:

Prompt for Annotation Validation:
Esamina questa annotazione da un testo storico in italiano antico (Volgare) per un task di Named Entity Recognition.
LINEE GUIDA PER L'ANNOTAZIONE: {guidelines}
CONTESTO: "{annotation_context}"
ANNOTAZIONE ATTUALE:
ENTITY: "{entity}"
LABEL: {assigned_label}
Valuta se l'annotazione è corretta dato il contesto e le linee guida. Considera:

1. Se il confine dell'ENTITY è corretto o dovrebbe includere più o meno parole.
2. La correttezza della LABEL

La risposta deve avere categoricamente il seguente formato:
GIUDIZIO: (CORRECT, ISSUES, DELETE o AMBIGUOUS)
ENTITY suggerita: (entity suggerita o NONE)
LABEL suggerita: (label suggerita o NONE)
NOTE: (note esplicative o NONE)
Non sono consentite annotazioni innestate o sovrapposte.
Fornisci il tuo giudizio come:

- **CORRECT:** Se sia il confine che l'etichetta sono corretti
- **ISSUES:** Se il confine e/o l'etichetta non sono corretti
- **DELETE:** Se l'annotazione non contiene alcuna entity
- **AMBIGUOUS:** Se c'è una genuina ambiguità nel dominio

Figure 4.6: Template for the annotation validation prompt used with Claude. Curly braces indicate variables replaced at runtime.

1. **Context-Rich**: It provides substantial text before and after the entity (400 characters total) to ensure Claude has sufficient context for judgment.

2. **Structured Output Format**: It explicitly defines a machine-parsable response format with clear sections for judgment, suggested corrections, and explanatory notes.

3. **Decision Categories**: It offers four distinct judgment categories (CORRECT, ISSUES, DELETE, AMBIGUOUS) with clear criteria for each.

4. **Focused Guidance**: It directs Claude to evaluate two specific aspects: entity boundaries and label correctness.

5. **Italian Language**: The prompt is written in Italian to align with the annotation language, potentially improving context understanding.

6. **Character Limits**: It imposes character constraints on explanatory notes to keep feedback concise and focused.

This prompt design exemplifies how natural language can be structured to enable more reliable machine evaluation while maintaining flexibility for complex linguistic judgments.

Zero-shot: Guidelines

The validation process relies extensively on a comprehensive set of annotation guidelines developed specifically for historical Italian texts. These guidelines were provided to Claude as part of the validation prompt, enabling zero-shot evaluation without prior training on annotated examples.

The guidelines document in Figure 4.7 details the specific characteristics of each entity type (PER, LOC, FAM, POP, OPR, DAT, EVE, DOC, ORG), providing **explicit rules** for boundary determination and classification decisions.

Annotation Guidelines for Entity Types

PER (Persone)

- Includi i titoli quando fanno parte dell'identificatore di una persona specifica
- Includi le specifiche geografiche quando fanno parte del nome/titolo ufficiale
- Includi i descrittori/ruoli che aiutano a identificare la persona (es. "Eunuco portinaio")
- more rules...

LOC (Luoghi)

- Includi città, paesi e regioni (ad esempio, "Fiorenza", "Regno di Napoli")
- Includi edifici e punti di riferimento con nomi propri (ad esempio, "Palazzo Vecchio", "Santa Maria del Fiore")
- Includi elementi naturali con nomi propri (ad esempio, "Monte Oliveto", "Arno")
- more rules...

FAM (Famiglie)

- Si applica ai nomi di famiglia quando si riferiscono al gruppo collettivo (ad esempio, "i Medici", "casa d'Este")
- Includi il termine "casa" solo quando fa parte dell'identificatore formale
- Escludi i nomi di famiglia che fanno parte del nome completo di un individuo (che verrebbero etichettati come PER)
- more rules...

POP (Popolazioni)

- Includi gruppi etnici e popoli storici (ad esempio, "Romani", "Cartaginesi")
- Includi demonimi regionali (ad esempio, "Fiorentini", "Veneziani")
- Includi termini come "popolo di" più la località quando ci si riferisce agli abitanti (ad esempio, "popolo di Napoli")
- more rules...

// Additional categories...

Figure 4.7: Example Annotation guidelines for most common entity types used in the NER system.

Additionally, it addresses **common ambiguities** (Figure 4.8) in historical texts, such as distinguishing between saints as persons versus locations, or determining when city names represent physical places versus political entities.

These guidelines serve as Claude's only reference for making validation judgments, creating a true zero-shot learning scenario where the model must apply these rules to previously unseen texts and annotations. This approach allows us to assess Claude's baseline capability to understand and apply complex linguistic criteria in a specialized domain.

4.3.3 Human Review

The human validation phase revealed significant limitations in Claude's ability to accurately assess annotations in historical Italian texts. Despite the structured prompt and detailed guidelines, approximately 60% of Claude's suggested corrections proved to be inappropriate or unnecessary when subjected to human review.

Several patterns emerged during human validation:

1. **Contextual Understanding**: Claude frequently struggled to comprehend the broader narrative context, particularly when texts included archaic grammatical structures or specialized vocabulary. This limitation led to misinterpretations of entity references and their roles within the text.

2. **Guideline Application**: While Claude could recite the guidelines, it showed inconsistency in applying them to specific cases, especially with entities that could potentially fall into multiple categories (e.g., distinguishing between a city as a location versus its governing body as an organization).

3. **Boundary Detection**: Claude showed particular difficulty with entity

AMBIGUITÀ COMUNI

Nomi di Santi

- Quando ci riferiamo al santo come persona (es. "San Francesco predicava agli uccelli"), va etichettato come **PER**
- Quando ci riferiamo alla chiesa/luogo (es. "si recò a San Francesco"), va etichettato come **LOC**
- Quando ci riferiamo all'opera d'arte (es. "il San Francesco di Giotto"), va etichettato come **OPR**

Nomi di Famiglie Nobili

- Quando il nome è usato per la famiglia (es. "gli Este dominarono Ferrara"), va etichettato come **FAM**
- Quando è parte del nome di una persona specifica (es. "Alfonso d'Este"), va considerato parte del **PER**
- Quando indica un territorio (es. "Ducato d'Este"), il nome diventa parte del **LOC**

Titoli Nobiliari con Luoghi

- Quando identificano una persona specifica (es. "Duca di Milano Ludovico Sforza"), l'intera espressione va etichettata come **PER**
- Quando si riferiscono all'istituzione (es. "il Ducato di Milano"), vanno etichettati separatamente: "Milano" come **LOC**
- Quando sono usati genericamente (es. "i duchi di Milano"), solo "Milano" va etichettato come **LOC**

Riferimenti Religiosi

- Quando "Dio" è personificato o agisce, va etichettato come **PER**
- Quando "Chiesa" si riferisce all'edificio, va etichettato come **LOC**
- Quando "Chiesa" si riferisce all'istituzione, va etichettato come **ORG**

// additional guidelines...

Figure 4.8: Guide to resolving common ambiguities in the annotation of entities in Italian historical texts. This table illustrates the contextual criteria for distinguishing between different categories of entities when the same term may belong to several classes.

boundary detection, often suggesting expansions or contractions of entity spans that contradicted the guidelines or historical naming conventions.

4. **Conservative Judgments**: Claude demonstrated a tendency toward flagging potential issues even when annotations were correct according to the guidelines, requiring frequent human intervention to reject unnecessary changes.

The high rate of inappropriate suggestions underscores the current limitations of zero-shot learning approaches for specialized linguistic tasks in historical languages. While Claude provided valuable assistance in identifying some genuine annotation errors, the human review process remained essential for maintaining annotation quality.

4.3.4 Implementation

The implementation of this validation workflow consists of two interconnected scripts that handle different phases of the process. It is important to mention that by default the script only deals with **automatic annotations**, as provided for in the human-to-AI workflow. Let's examine each component:

Claude Validator System

The validator system handles the communication with Claude's API and processes annotations in batches:

The implementation follows a structured batch processing approach designed to optimize API usage efficiency while maintaining processing quality. Annotations are processed in configurable batches (default size: 20) to balance throughput with API rate limits.

Key features of this implementation include:

1. **Batched Processing**: The system divides the annotation list into manageable groups, processing them sequentially while maintaining progress

tracking. This approach optimizes both processing time and API utilization.

2. **Context Window Creation**: For each annotation, the system creates a comprehensive context window that includes substantial text before and after the entity (typically 200 characters in each direction). This context-rich approach provides Claude with sufficient information for nuanced evaluation.

3. **Structured Response Parsing**: The system converts Claude's natural language responses into structured data formats, extracting specific judgments (CORRECT, ISSUES, AMBIGUOUS, DELETE) and any correction suggestions in a machine-processable format.

4. **Analysis Generation**: Upon completion of all annotation processing, the system generates detailed summary statistics on validation results, including error rates by entity type, distribution of correction types, and confidence metrics.

Human Validation Interface

The human validation interface provides an interactive environment for reviewing and applying Claude's suggestions:

The system implements a review workflow that prioritizes annotations flagged as potentially problematic, allowing human reviewers to focus their attention where it's most needed.

Key features of this implementation include:

1. **Rich Text Interface**: The interface employs color-coding and visual formatting to create a clear terminal-based review environment, with distinct highlighting for original annotations, suggested changes, and contextual text.

2. **Focused Review**: The system automatically filters for annotations that Claude flagged as problematic (categorized as ISSUES, AMBIGUOUS, or DELETE), allowing reviewers to skip annotations marked as correct and focus on potential problems.

3. **Context Display**: Each annotation is presented within its full textual context, with the entity visually highlighted for immediate identification. This contextual view enables informed decision-making about ambiguous cases.

4. **Interactive Decision Making**: For each suggested change, the interface prompts the human reviewer for explicit confirmation before implementing any modifications to the annotation database, maintaining human control over the final decisions.

5. **Statistics Tracking**: The system maintains comprehensive statistics throughout the review session, tracking counts of various actions (confirmations, rejections, modifications) and providing a summary report upon completion.

This implementation demonstrates a practical approach to integrating AI assistance into the annotation workflow while maintaining essential human supervision. The combination of automated validation suggestions with interactive human review creates an efficient pipeline that exploits both machine and human intelligence.

The system is designed both as an evaluation experiment and as a potential component within a broader annotation infrastructure, capable of integration with the centralized Annotation Management system that handles database interactions.

4.3.5 Benchmarking

Unlike the other experiments conducted in this research, I did not implement specific quantitative metrics to evaluate Claude's performance in the annotation validation task. This deliberate decision stemmed from initial observations during the human review phase.

The primary challenge encountered was Claude's limited precision in applying the annotation guidelines to historical Italian texts. Despite the carefully structured prompt and detailed guidelines provided, approximately **60-70% of Claude's suggested corrections proved to be inappropriate or unnecessary** when subjected to expert human review. This high rate of false positives would have rendered any automated or semi-automated validation workflow impractical, as it would require more human effort to review and reject incorrect suggestions than to perform manual validation directly.

Given these limitations, I determined that the human-to-AI validation workflow as initially conceived was not viable for the current project phase. The high rate of incorrect suggestions would introduce more noise than value to the annotation process, potentially misleading annotators or requiring excessive verification effort.

This finding doesn't necessarily indicate a fundamental limitation of large language models for this task, but rather highlights the current gap between general language understanding and the specialized knowledge required for historical text annotation.

4.4 Dataset Preparation and Augmentation

The processing pipeline for the historical Italian NER dataset involved several stages to transform raw annotations into a structured format suitable for model training. This process included context window extraction, JSON structure definition, data splitting, and synonym-based augmentation.

4.4.1 Context Window Extraction

A key component of our approach was the implementation of dual-scale context windows around each entity. For every annotated entity, the script extracted:

- A **small context window** (128 characters) centered on the entity, providing immediate linguistic context
- A **medium context window** (256 characters) for a compromise between lightness and context
- A **large context window** (512 characters) that offered broader discourse context

This triple window approach aimed to create more possibilities in terms of training and testing.

4.4.2 JSON Structure Definition

Each example was structured in a JSON format containing:

- A unique identifier derived from the book ID and character positions
- Document metadata (book ID, source text positions)
- Dual context windows with both text and absolute character positions
- Entity information including text, type, relative positions in both context windows, and annotation metadata
- Reliability metrics capturing **annotation confidence based on origin (manual vs. automatic) and verification status**
- Augmentation tracking to maintain data provenance

This structure preserved the relationship between entities and their context while maintaining crucial metadata about annotation reliability, addressing the particular challenges of historical text annotation where confidence in entity boundaries can vary.

4.4.3 Dataset Splitting

The dataset was divided into train (70%), validation (15%), and test (15%) splits using a randomized but deterministic approach. To preserve dataset integrity:

- **Overlapping context windows were detected** and eliminated to prevent data leakage
- Each example was stored as a separate JSON file with consistent naming conventions
- Window extraction maintained consistent boundaries to ensure **all entities remained fully within their context**

Book-level statistics were maintained to **track the distribution of source texts across splits**, addressing potential biases in entity distribution across different historical works.

Last but not least to emphasise here, **the test dataset that has been created in this way will be reserved exclusively for the testing of all models** on which we are going to experiment, both in zero-shot/few-shot and for fine-tuned models.

4.4.4 Entity-Aware Dataset Partitioning

A critical methodological consideration in preparing our dataset involved addressing the non-uniform distribution of unique entities across the corpus. Simply partitioning texts randomly would have created **significant overlap**

of unique entities between training, validation, and test sets, potentially leading to inflated performance metrics and models that merely memorize entity patterns rather than learning to recognize them in novel contexts.

To mitigate this risk, we implemented an **entity-aware partitioning strategy** that attempted to minimize overlap of unique entities across dataset splits. This approach involved first extracting all unique entity strings, then using a proper algorithm to allocate them to training, validation, and test sets while balancing entity type distributions.

Despite these efforts, the inherent limitations of our dataset—particularly the frequency with which certain canonical entities appear across multiple texts—created constraints on our ability to achieve perfect separation. The most successful partitioning scheme we achieved maintained approximately 60% unique entities in each split, with around 40% overlap between training, validation, and test sets.

This unavoidable overlap should be considered when interpreting model performance metrics, particularly for the fine-tuned models.

Future work with expanded datasets should continue to prioritize entity-aware partitioning, potentially implementing more sophisticated allocation algorithms that consider entity frequency, type distribution, and contextual diversity simultaneously.

4.4.5 Synonym-Based Augmentation

To address the challenge of limited training data for historical Italian, we implemented a synonym-based augmentation system. For each entity in the training set, the script:

- Consulted the previously generated **synonym dictionary** derived from historical linguistic patterns
- Generated alternative versions of the original example by replacing entities with period-appropriate synonyms

- Maintained entity boundaries and adjusted text positions to reflect the new entity lengths
- Preserved annotation metadata while tracking the original form of the entity

This approach produced linguistically valid variations of the original examples, exposing the model to the diverse orthographic and referential forms common in historical Italian texts. Particularly for person entities, where the same historical figure might be referenced in multiple ways (e.g., ”Lorenzo de’ Medici” vs. ”Lorenzo il Magnifico”), this augmentation strategy helped the model learn equivalence relationships without requiring additional manual annotation. The augmentation process was bounded to the training set only, ensuring that **validation and test sets remained untouched** to provide reliable performance metrics. The resulting augmented dataset contained both the original examples and their variations, with clear origin tracking to distinguish between original and augmented instances during model development.

4.5 NER with SLIMER-IT

While our primary focus has been developing annotation methods and creating a high-quality dataset, we also needed to evaluate the potential of existing language models to perform this task. SLIMER-IT, developed by ExpertAI, is a state-of-the-art LLM that has demonstrated excellent performance on modern Italian zero-shot NER tasks. Our interest lies in determining whether this model, without fine-tuning, could accurately identify named entities in historical Italian texts from the 14th-16th centuries.

SLIMER-IT (Semantic Language Italian Model for Entity Recognition and Information Technology) is built on the LLaMAntino architecture and has

been pre-trained on a large corpus of modern Italian texts. It has shown impressive capabilities in various NLP tasks including text classification, summarization, and entity recognition. However, its performance on historical Italian variants (Volgare) has never been tested before.

4.5.1 NER Pipeline

The NER pipeline we designed for testing SLIMER-IT represents a simplified version of the AI-to-human workflow described in Chapter 3. While our complete vision involves a cyclical process where AI assists with initial annotations and human experts refine them, this implementation focuses on the evaluation component to establish baseline performance metrics.

The pipeline consists of the following components:

1. **Test Dataset Ingestion**: As already explained in the section on dataset preparation, the script retrieves the previously prepared test set to perform the tests. This is done by reading the JSON files and filtering the tests on the entity types of interest only.

2. **Context Window Selection**: A crucial step concerns the choice of context window, since tests were carried out with both possibilities, which were prepared earlier. As we shall see, the smaller context windows seem to confuse the models less, even the more complex ones.

3. **Zero-Shot / Few-Shot NER**: We prompt SLIMER-IT to identify entities of a specific type within the given context, without any task-specific training or fine-tuning. **Optionally it is possible to enable the few-shot mode** flag, adding to the prompt 5 examples of annotated excerpts (not present in the test set).

4. **Performance Evaluation**: We compare the model's predictions against the gold standard human annotations to measure exact matches, partial matches, and various performance metrics.

This approach allows us to test whether a modern Italian language model possesses sufficient knowledge of historical linguistic variants to perform NER tasks without specialized training, establishing a baseline for comparison with future fine-tuned models.

4.5.2 Prompt Engineering

The effectiveness of zero-shot learning with LLMs heavily depends on prompt design. We developed a structured prompt that provides the model with precise instructions and guidelines for each entity type. The basic structure of the prompt follows the Figure 4.9.

```
<|start_header_id|>system<|end_header_id|>
Sei un utile assistente.<|eot_id|>
<|start_header_id|>user<|end_header_id|>
Ti viene fornito un input di testo (delimitato da tre
virgolette) e un'istruzione.
Leggi il testo e rispondi all'istruzione alla fine.
"""
[CONTEXT TEXT CONTAINING THE ENTITY]
"""
Istruzione: Estrai tutte le entità di tipo [TAG] dal testo
che hai letto. Ti vengono fornite una DEFINIZIONE e alcune
LINEE GUIDA.
DEFINIZIONE: [DETAILED DEFINITION OF THE ENTITY TYPE]
LINEE GUIDA: [SPECIFIC GUIDELINES FOR THIS ENTITY TYPE]
Restituisci una lista JSON di oggetti, ciascuno contenente
i campi 'text' per il testo dell'entità e 'score' per
il livello di confidenza (da 0.0 a 1.0) della tua
predizione. Restituisci una lista vuota se non sono presenti
istanze.<|eot_id|>< |start_header_id|> assistant < |end_header_id| >
```

Figure 4.9: Prompt template used for zero-shot NER with SLIMER-IT. The placeholders in square brackets are replaced with specific context and entity type information.

This prompt structure includes several key elements:

1. **Clear Task Definition**: We specify that the model should extract entities of a particular type.

2. **Detailed Entity Definitions**: For each entity type (e.g., PER, LOC), we provide comprehensive definitions. For example, PER is defined as "Persone: figure storiche con identificatori completi, figure divine e mitologiche quando personificate, santi quando si riferiscono alla persona, individui con titoli."

3. **Specific Guidelines**: We include detailed annotation guidelines for each entity type, such as "Include titles when they form part of a person's identifier" for PER entities. These guidelines are the same used for the Claude experiment.

4. **Structured Output Format**: We request output in JSON format to facilitate automated processing. It is important to note that we explicitly requested a **Confidence Score** for subsequent analyses. Unfortunately, unlike other models, Slimer appears not to have been certified for this type of request.

This prompt design ensures the model has sufficient information about the task requirements and entity definitions without providing examples that might bias its predictions, maintaining a true zero-shot evaluation scenario.

4.5.3 Few-shot learning

Similar to the zero-shot, an extension of the prompt was implemented in order to reach some examples and also perform some tests in few-shot learning. Given the triviality of the task, we avoid adding further details about the prompt here. The logic followed was to construct some examples, specifically selected to be outside the test dataset, with the expected outputs.

4.5.4 Implementation

The implementation of our evaluation framework consists of a comprehensive system that handles the entire process from loading texts and annotations to

evaluating the model's performance. Here are the key components:

Model Loading

The system employs efficient model management techniques to load and initialize the large language model. It implements hardware detection to automatically utilize GPU acceleration when available, with graceful fallback to CPU processing when necessary. The loading process includes optimization parameters for half-precision computation to balance memory usage with performance, particularly important when processing historical texts that require substantial context windows.

Entity Evaluation

For each entity, the system generates a tailored prompt and evaluates the model's response through a multi-stage process:

- First, it constructs a context-rich prompt containing the text surrounding the target entity.
- It then submits this prompt to the model using deterministic generation parameters (zero temperature) to ensure reproducible results.
- The resulting prediction text is captured and passed to a specialized parser for entity extraction.
- Finally, the system compares predicted entities with gold standard annotations, identifying matches using both strict and lenient criteria.

Response Parsing

A critical component is the robust parsing of model responses, which accounts for various output formats. The parser employs a progressive approach to extract structured entity data:

- It first cleans and normalizes the response text.
- It employs regular expression pattern matching to locate JSON-formatted content within the response.
- It implements multiple fallback strategies to handle variations in response formatting.
- Once extracted, the JSON content is parsed and normalized into a standardized entity representation.
- The system includes comprehensive error handling to manage parsing exceptions without disrupting the evaluation workflow.

4.5.5 Evaluation Metrics

Our evaluation framework implements dual assessment criteria to provide a comprehensive view of model performance on historical entity recognition:

- **Strict Evaluation**: Considers only exact boundary matches as true positives, requiring perfect alignment between the predicted and gold-standard entity spans. This metric penalizes even minor boundary errors, providing a conservative assessment of performance.
- **Lenient Evaluation**: Accepts partial matches that exceed a 50% overlap threshold, recognizing that boundary determination in historical texts often involves inherent ambiguity due to complex naming conventions and inconsistent orthography.

For both criteria, we calculate standard information retrieval metrics:

- **Precision**: Measures the proportion of correctly identified entities among all predicted entities, reflecting the model's ability to avoid false positives.

- **Recall**: Quantifies the proportion of gold-standard entities successfully identified, indicating the model's capacity to discover entities present in the text.
- **Binary F1 Score**: Provides the harmonic mean of precision and recall, balancing these complementary metrics into a single value for model comparison.

This binary classification approach, rather than a token-based evaluation, better reflects the practical utility of NER systems where entity recognition is ultimately an entity-level task. The dual strict/lenient assessment acknowledges the unique challenges of historical text processing, where boundary determination remains particularly challenging even for human experts.

4.6 NER with Claude

As a further point of comparison, we decided to implement a script to test Claude's annotation capabilities from the script previously created for SLIMER-IT. The original SLIMER-IT benchmark was designed for local inference with vLLM-based models, therefore requiring some modifications to ensure compatibility with Claude, which operates through a remote API.

4.6.1 API Integration and Authentication

The most fundamental change was replacing the local model loading mechanism with Anthropic's API client. While SLIMER-IT instantiated models directly in memory using vLLM's interface, the adapted version starts a client connection to Anthropic's cloud infrastructure. Additionally, the adaptation implemented rate limiting strategies to prevent API throttling during batch evaluations.

4.6.2 Prompt Engineering Adaptations

Claude requires a slightly different prompt formatting compared to SLIMER-IT's original approach. The message structure was reconfigured to align with Claude's expected format but preserving the semantic content of the instructions. Special tokens and markers used by vLLM models were removed and replaced with natural language more suitable for Claude.

4.6.3 Message Handling and Response Parsing

The interaction model differs substantially between local inference and API-based systems. While SLIMER-IT used direct function calls to generate text, the Claude adaptation implements an asynchronous request-response pattern through the API. This required implementing proper error handling for network issues, timeout management, and response validation.

Response parsing required enhanced robustness due to Claude's varied output formats. The original parsing logic was extended with multiple fallback strategies to handle variations in JSON formatting, quotation style, and text structure. Regular expressions were employed to extract entity information even when the response deviated from the expected format, improving resilience to output variations.

4.6.4 Benchmarking pipeline

In order to standardise the responses and obtain accurate comparisons of the performance of the different models, all outputs were standardised. In particular, the output metrics and scores will be the same as those used for the previous script on SLIMER-IT.

4.7 NER with DistilBERT

While larger language models might offer more powerful language understanding capabilities, they present some considerable limits for our specialized task. DistilBERT, a lightweight version of BERT with approximately 66 million parameters (about 40% smaller than its base counterpart), offers several strategic advantages for our purposes.

First, our relatively small corpus of approximately 4,000 manually annotated entities creates a fundamental constraint. Larger models with hundreds of millions or billions of parameters would quickly **overfit** to this limited training data, essentially memorizing the examples rather than learning generalizable patterns. DistilBERT's more modest parameter count reduces this risk while still retaining much of BERT's linguistic capabilities.

Second, an iterative learning approach through bootstrapping (which we'll explore in section 4.5.3) necessitates **multiple training cycles**. The computational efficiency of DistilBERT makes this iterative process feasible without requiring excessive computational resources or training time. Each cycle of prediction, validation, and retraining becomes manageable even with limited GPU availability.

Finally, DistilBERT specifically **optimized for Italian** (osiria/distilbert-italian-cased-NER) provides a strong starting point as it already incorporates knowledge of modern Italian linguistic structures. While Volgare differs significantly from contemporary Italian, the underlying grammatical patterns and morphological foundations share a deep base that the model could exploit during training.

4.7.1 NER Pipeline

The NER pipeline developed for this project aim to implement a bootstrapping approach that progressively improves model performance through cycles of prediction, human validation, and retraining. Unlike traditional one-time

training workflows, this iterative process acknowledges the challenges of historical language processing and the scarcity of annotated data.

At its core, the pipeline consists of four interconnected stages:

1. **Initial Training**: The first stage involves training the DistilBERT model on our limited manually annotated dataset, applying data augmentation techniques to artificially expand the training examples.

2. **Prediction and Confidence Scoring**: The trained model generates predictions on unannotated text segments, assigning confidence scores to each identified entity. This confidence mechanism is critical for the bootstrapping process, as it allows us to prioritize high-confidence predictions for human review.

3. **Human Validation Interface**: Predictions exceeding a confidence threshold are presented to human annotators through a specialized interface (as described in Chapter 3's annotation workflows). This interface enables efficient review of model suggestions, allowing annotators to accept, reject, or modify each prediction.

4. **Dataset Expansion and Retraining**: Validated predictions are incorporated into the training dataset, gradually expanding our corpus of reliable annotations. The model is then retrained on this expanded dataset, completing one cycle of the bootstrapping process.

Each iteration of this cycle increases both the size of the training dataset and the model's performance, creating a virtuous cycle of improvement. As the model's predictions become more accurate, human validators can process suggested entities more quickly, further accelerating dataset growth.

4.7.2 Preprocessing

The preprocessing phase of our NER pipeline addresses several critical challenges specific to historical Italian texts. Our initial attempts to train a comprehensive multi-label NER model that simultaneously handled all entity types (PER, LOC, FAM, etc.) revealed the **complexity of this approach** given our limited dataset size. Each entity type presented specific contextual patterns and historical variations that complicated the learning task.

To address this challenge, we adopted a focused strategy by initially targeting the most well-represented entity type in our annotations: person names (PER), which constitute approximately 41.7% of our annotated entities. This approach offers several advantages:

1. It allows the model to concentrate learning resources on mastering a single entity pattern rather than dilute attention across multiple categories.

2. If successful, it establishes a methodological template that can be replicated for other entity types.

3. It creates the foundation for an **ensemble approach** where specialized models for each entity type can work in concert, potentially achieving better overall performance than a single multi-label model. We'll discuss this possibility in later chapters.

The preprocessing workflow implements a chunking strategy that handles several critical tasks

The preprocessing pipeline, in this case, handles only the Tokenization step and the context windows selection, since the preparation of the data, the normalization and ther data augmentation happened asynchronically in the previous steps. Whis allowed to focus the development only on the relevant training algorithms, allowing for a more careful namagement of this process.

4.7.3 Tokenization

The tokenization process bridges the gap between our text data and the neural network inputs. This stage is particularly critical for historical Italian texts, where orthographic variations, archaic terms, and inconsistent spelling conventions present unique challenges for standard tokenizers.

Our implementation uses the tokenizer associated with the pre-trained DistilBERT model. The tokenization process handles two critical aspects for NER:

1. Converting text into **token IDs** and creating **attention masks** for the model.

2. **Aligning entity labels** with subword tokens, ensuring that only the first subword of each word receives a label while subsequent subwords are marked with the ignore index.

The tokenization process for historical Italian presents several challenges:

- Historical terms may be broken into unusual subword units since the tokenizer was trained on modern Italian.

- Spelling variations can lead to inconsistent tokenization of the same entity across different instances.

- Entity boundaries may not align cleanly with subword boundaries, particularly for entities containing archaic prepositions or articles (e.g., “d’Este”, “de’ Medici”).

While using a specialized tokenizer trained on historical Italian would be ideal, the current implementation showed good performance in our tests, especially compared to more specialized tokenizers like base BERT or BERToldo.

4.7.4 Training

The training configuration balances model performance against our relatively limited dataset size. We've selected hyperparameters that help mitigate overfitting risks while promoting effective learning:

- **Model Selection**: We used "osiria/distilbert-italian-cased-ner" as our base model
- **Sequence Length**: Maximum sequence length of 128 tokens
- **Batch Size**: Small batch size of 4 with gradient accumulation for effective training
- **Learning Rate**: Low learning rate (5e-5) to promote gradual adaptation
- **Training Duration**: Limited to 3 epochs to prevent overfitting
- **Regularization**: Weight decay of 0.01 for additional regularization

These settings reflect several strategic decisions:

1. **Low Learning Rate**: The modest learning rate promotes gradual adaptation of the pre-trained weights to our historical domain without dramatically altering its linguistic knowledge.
2. **Modest Epoch Count**: Limiting training to 3 epochs helps prevent overfitting to our small dataset while still allowing sufficient learning.
3. **Gradient Accumulation**: Using gradient accumulation effectively increases the batch size without requiring additional memory.
4. **Weight Decay**: The weight decay value adds regularization, another measure against overfitting.
5. **Warmup Period**: A warmup ratio gradually increases the learning rate at the beginning of training, allowing the model to adapt more gently to our data.

The training process incorporates an **early stopping mechanism** with a patience of 2 evaluation cycles, which automatically halts training if performance on the validation set fails to improve. This prevents unnecessary computation and further counters overfitting.

Our custom metrics computation provides detailed performance analysis beyond the standard loss values. It calculates precision, recall, and F1-score at both the token and entity levels, providing a comprehensive view of model performance during training.

4.7.5 Bootstrapping

The bootstrapping approach represents a potential major upgrade in our NER pipeline, enabling iterative improvement of the model through a semi-automated annotation expansion process. While not fully implemented in the current script, the bootstrapping cycle would extend the existing framework with the following components:

1. **Confidence Estimation**: Following initial training, the model would process unannotated texts and assign confidence scores to predicted entities. These scores could be derived from the model's softmax probabilities, potentially with calibration to improve reliability.

2. **Filtering Mechanism**: A threshold-based system would select high-confidence predictions for human review, prioritizing entities that are most likely to be correct and filtering out low-confidence predictions that would waste the human annotator time.

3. **Validation Interface**: Similar to the Human Validation Script described in section 4.3.4, this component would present filtered predictions to human validators in an interactive interface, allowing quick review and correction.

4. **Corpus Expansion**: Validated predictions would be added to the training corpus with appropriate **metadata** indicating their provenance (to potentially weight them differently during training).

5. **Model Retraining**: The model would be retrained on the expanded corpus, potentially with a curriculum that gradually increases the proportion of automatically derived examples.

4.7.6 Evaluation

The evaluation process is integrated directly within the training pipeline to track of performance metrics throughout the model's development.

Evaluation Methodology

The evaluation strategy employs a three-stages approach to assess the model's performance:

1. **Periodic Validation**: During training, the model is evaluated on the validation set after each epoch to track performance improvement and detect potential overfitting patterns.

2. **Final Validation Assessment**: After the training completion, an overall evaluation is conducted on the validation set to assess the model's performance on unseen data from the same distribution.

3. **Test Set Evaluation**: As the definitive measure of model quality, evaluation on the **held-out test set** provides an unbiased estimate of how the model generalizes to entirely unseen data.

In terms of metrics, we adopt here the same performance metrics as seen with SLIMER-IT and Claude (Precision, Recall, binary F1), for their representativeness and for easy comparison of results between models.

Output and Results Storage

The evaluation process produces a structured output that facilitates detailed analysis of the model's performance:

1. **Trained Model**: The fine-tuned model is saved with a unique identifier incorporating the entity type focus and timestamp, allowing easy tracking and comparison of different training runs.

2. **Detailed Metrics JSON**: A comprehensive metrics file is generated with the following structure:

   - `run_info`: Contains metadata about the training run, including model checkpoint, entity focus, and hyperparameters.
   - `final_metrics`: Stores the final evaluation metrics for train, validation, and test sets.
   - `metrics_over_time`: Tracks the evolution of metrics throughout the training process, separated by dataset type.
   - `confidence_stats`: Provides statistical analysis of confidence scores across datasets.
   - `dataset_stats`: Summarizes dataset sizes and characteristics.

3. **Entity Type Information**: A separate file stores the mapping between numeric labels and entity types, for a correct interpretation of the results.

Figure 4.10 shows an excerpt from the metrics JSON file, highlighting the model's performance on historical Italian person entity recognition:

```
{
"final_metrics": {
"test": {
"eval_loss": 0.02850113995373249,
"eval_precision": 0.8693181818181818,
"eval_recall": 0.8181818181818182,
"eval_f1": 0.8429752066115702,
"eval_entity_accuracy": 0.23129251700680273,
"eval_entity_predictions": 352,
"eval_true_entities": 1323,
"eval_runtime": 0.5291,
"eval_samples_per_second": 1045.09,
"eval_steps_per_second": 66.145,
"epoch": 3.0
}}}
```


Figure 4.10: Excerpt from the detailed metrics JSON showing test set performance.

Chapter 5

Results and Discussion

In this chapter, we present and analyze the results of our experiments on applying various modeling approaches to Named Entity Recognition in historical Italian texts. For clarity and methodological consistency, we focused our evaluation exclusively on the **PER** (Person) entity type for several key reasons:

- Person entities are more straightforward to annotate with higher confidence, reducing potential noise in our evaluation metrics due to annotation uncertainty
- While the **PER** category is recognized by most standard NER models, our annotation guidelines incorporate specific challenges unique to historical Italian texts, such as the inclusion of personified abstract entities (e.g., ”Amore” when treated as an acting character)
- Person entities constitute the largest portion (41.7%) of our manually annotated corpus, providing a more robust sample size for meaningful statistical analysis
- Our generated synonym dictionary is richer for this category, resulting in an increased dataset for the PER category, more diverse and full-bodied

All evaluations were conducted on the test set that was carefully selected to include both **common and rare entities**, with special attention to ensuring that it contained entities **unseen during training**. This approach provides a realistic assessment of each model's ability to **generalize beyond memorization**.

The dataset was enriched with additional metadata, including information about entity frequency and annotation origins (manual versus automatic), which could potentially be used for weighted training in future developments.

As far as the other categories (POP, FAM, LOC, etc.) are concerned, tests similar to those that will now be mentioned were carried out, but the scarcity of the dataset (specifically for the fine-tuning purpose) did not allow for results worthy of mention here.

5.1 Zero-Shot and Few-Shot Learning: Claude, SLIMER-IT

Our first set of experiments evaluated the performance of large language models without fine-tuning on our specific corpus. We compared two state-of-the-art systems: Claude 3.7 Sonnet and SLIMER-IT, both tested under various conditions to assess their capabilities.

5.1.1 Zero-Shot Results

In zero-shot scenarios, where models received no examples of annotated historical Italian entities, we observed notable differences in performance between Claude and SLIMER-IT. Tables 5.1 and 5.2 present the evaluation results using both strict and lenient matching criteria with a context window of 256 characters.

The most striking observation is Claude's superior recall rate (0.8537 for strict metrics and 0.8846 for lenient metrics) compared to SLIMER-IT (0.5412

Table 5.1: Zero-Shot NER Model Performance Comparison: Strict Metrics

Model	Context	Precision	Recall	F1 Score
LLaMAntino-3-SLIMER-IT	256 chars	0.2035	0.5412	0.2958
claude-3-7-sonnet-20250219	256 chars	0.2869	0.8537	0.4294

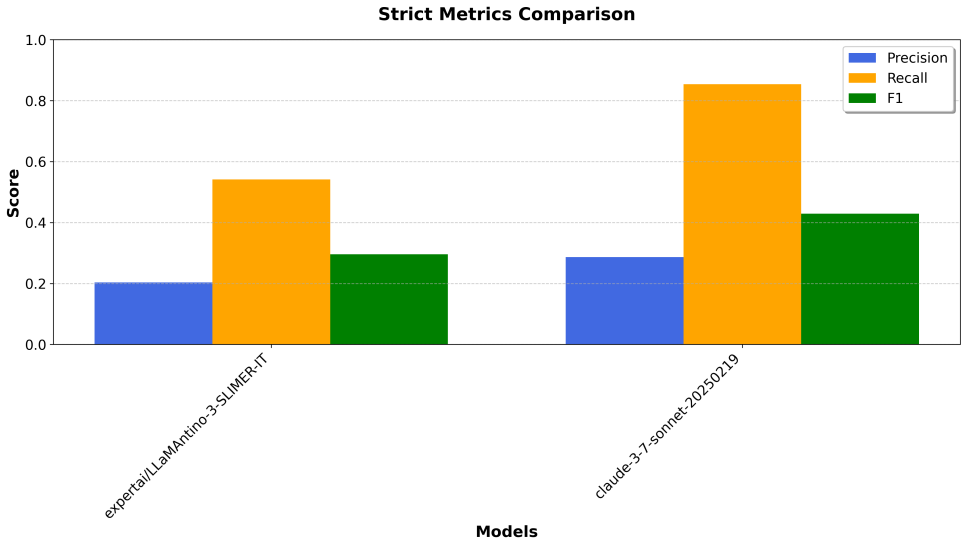


Figure 5.1: Zero-Shot NER Model Performance Comparison: Strict Metrics

and 0.6454 respectively). This indicates that Claude successfully identifies a significantly larger proportion of the manually annotated person entities in our test set. For our historical entity detection objectives, this recall advantage is particularly valuable, as it suggests Claude can effectively discover entities even in archaic texts without any examples. While precision appears relatively low for both models (0.2869 for Claude and 0.2035 for SLIMER-IT under strict evaluation), this metric should be interpreted carefully. Rather than indicating poor performance, the lower precision likely reflects the models' tendency to identify **additional possibly valid entities that were not captured in our manually annotated dataset**, which remains incomplete due to resource constraints.

In a practical workflow, this "over-prediction" can actually be beneficial, as it allows for the discovery of previously unidentified entities that human annotators might have overlooked. The difference between strict and lenient

Table 5.2: Zero-Shot NER Model Performance Comparison: Lenient Metrics

Model	Context	Precision	Recall	F1 Score
LLaMAntino-3-SLIMER-IT	256 chars	0.2827	0.6454	0.3932
claude-3-7-sonnet-20250219	256 chars	0.3459	0.8846	0.4973

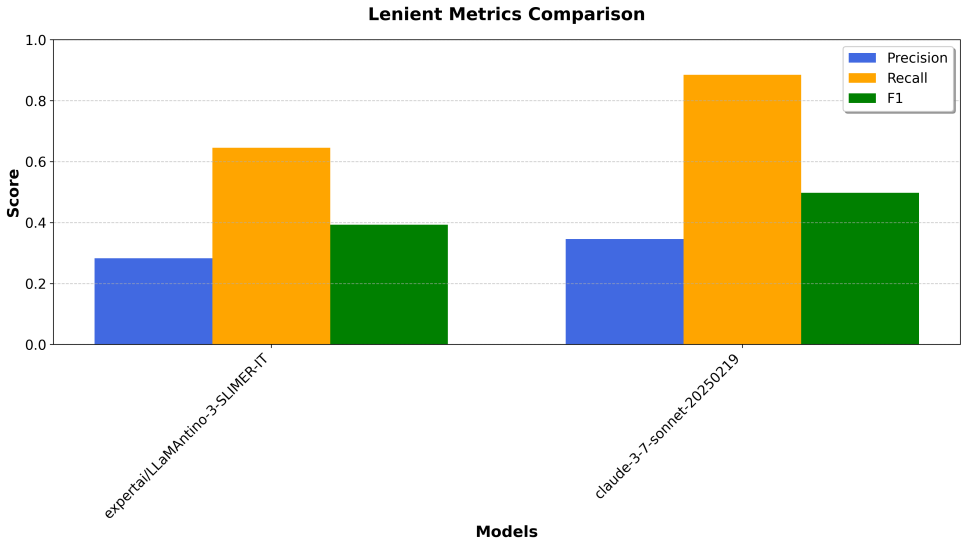


Figure 5.2: Zero-Shot NER Model Performance Comparison: Lenient Metrics

metrics also reveals important insights about boundary detection. Both models show improved performance under lenient evaluation (where partial matches are counted as successes), suggesting that while they can identify entity presence correctly, they sometimes struggle with precise boundary determination. This is particularly true for SLIMER-IT, as the performance increase is more than 10% compared to Claude's 3%.

5.1.2 Few-Shot Results

To evaluate whether providing example annotations improves performance, we conducted few-shot learning experiments with both models using the same 256-character context window. In our tests we provided 5 examples carefully selected to cover the diversity in the contextual situations that allow proper identification of the focused entity type. Tables 5.3 and 5.4 present these results.

Table 5.3: Few-Shot NER Model Performance Comparison: Strict Metrics

Model	Context	Precision	Recall	F1 Score
LLaMAntino-3-SLIMER-IT	256 chars	0.2062	0.5181	0.2950
claude-3-7-sonnet-20250219	256 chars	0.2748	0.8571	0.4162

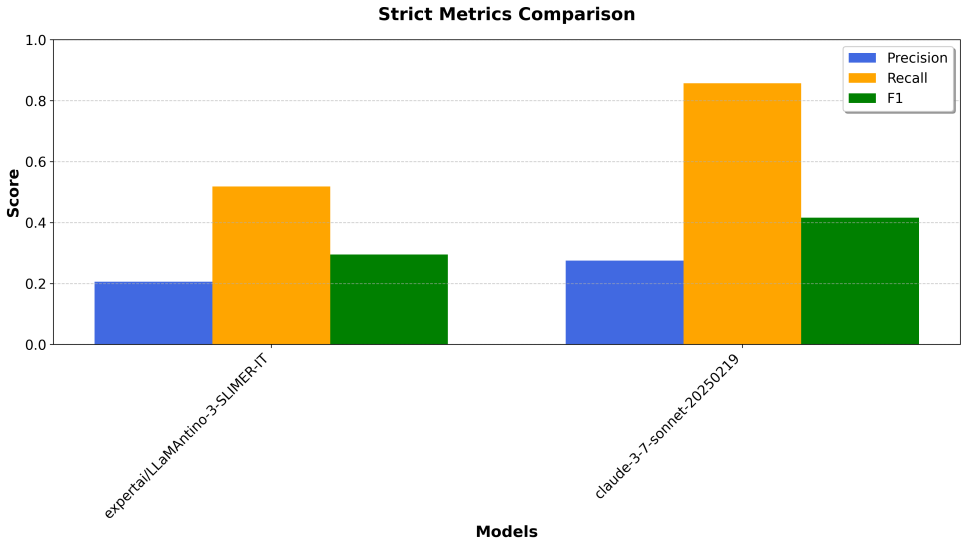


Figure 5.3: Few-Shot NER Model Performance Comparison: Strict Metrics

Surprisingly, the introduction of few-shot examples did not result in substantial performance improvements for either model when compared to their zero-shot counterparts. Claude maintained its impressive recall rate (0.8571 for strict metrics), essentially unchanged from its zero-shot performance (0.8537). Similarly, its lenient recall remained consistent at 0.8846. SLIMER-IT actually showed a **slight decrease in recall** from zero-shot to few-shot scenarios (0.5412 to 0.5181 for strict metrics), though this difference may not be statistically significant. The precision metrics showed minor improvements for Claude (from 0.2869 to 0.2748 in strict evaluation) but remained largely stable for SLIMER-IT. These results suggest that both models, but particularly Claude, may **already possess substantial knowledge** about historical Italian naming patterns, making additional examples less impactful than might be expected. The result of SLIMER-IT could be explained by the fact that the prompt used for the tests is substantially different from the one suggested in its

Table 5.4: Few-Shot NER Model Performance Comparison: Lenient Metrics

Model	Context	Precision	Recall	F1 Score
LLaMAntino-3-SLIMER-IT	256 chars	0.2910	0.6295	0.3980
claude-3-7-sonnet-20250219	256 chars	0.3262	0.8846	0.4767

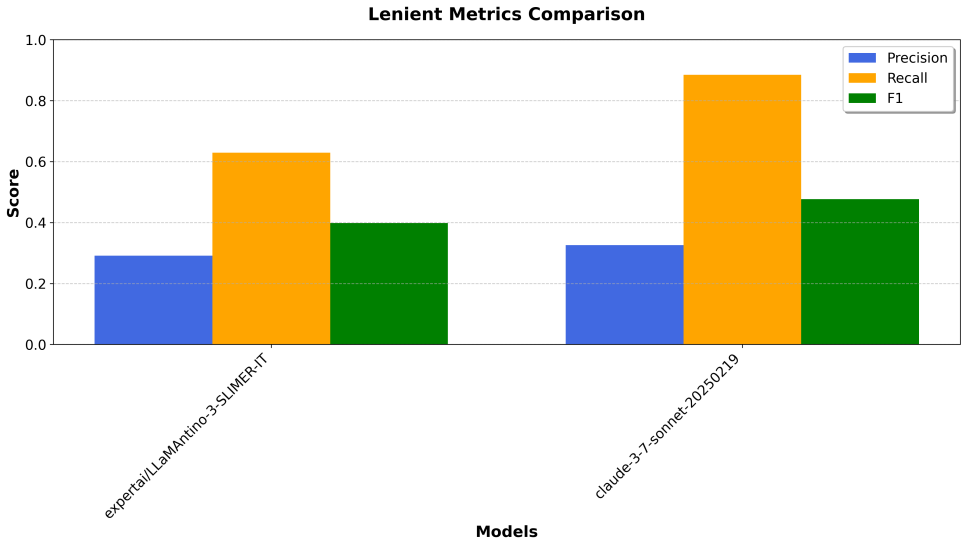


Figure 5.4: Few-Shot NER Model Performance Comparison: Lenient Metrics

guidelines (on which it was optimised), by the addition of the examples and by the requirement of a confidence level for the predictions. The consistent performance between zero-shot and few-shot settings indicates that these models' limitations in historical Italian NER **may not be easily addressed by simply providing a handful of examples**. Instead, they reflect deeper challenges related to the models' fundamental understanding of historical linguistic variations or their ability to correctly apply entity boundary rules specified in the instructions.

5.1.3 Small and Large Context Windows

To investigate the impact of context window size on entity recognition performance, we conducted experiments with **varying window sizes** (128, 256, and 512 characters) in zero-shot settings. This analysis helps understand how much surrounding context these models require for optimal entity recognition

in historical texts.

Table 5.5: Zero-Shot NER SLIMER-IT Performance Comparison with different Context Windows Size: Strict Metrics

Model	Context	Precision	Recall	F1 Score
LLaMAntino-3-SLIMER-IT	128 chars	0.4200	0.6528	0.5112
LLaMAntino-3-SLIMER-IT	256 chars	0.2035	0.5412	0.2958
LLaMAntino-3-SLIMER-IT	512 chars	0.0922	0.4221	0.1514

Table 5.6: Zero-Shot NER SLIMER-IT Performance Comparison with different Context Windows Size: Lenient Metrics

Model	Context	Precision	Recall	F1 Score
LLaMAntino-3-SLIMER-IT	128 chars	0.5140	0.7331	0.6043
LLaMAntino-3-SLIMER-IT	256 chars	0.2827	0.6454	0.3932
LLaMAntino-3-SLIMER-IT	512 chars	0.1412	0.5418	0.2241

Table 5.7: Zero-Shot NER Claude Performance Comparison with different Context Windows Size: Strict Metrics

Model	Context	Precision	Recall	F1 Score
claude-3-7-sonnet-20250219	128 chars	0.4400	0.8684	0.5841
claude-3-7-sonnet-20250219	256 chars	0.2869	0.8537	0.4294
claude-3-7-sonnet-20250219	512 chars	0.1613	0.7500	0.2655

The results reveal a striking and consistent pattern across both models: performance metrics decline as context window size increases. This trend is particularly pronounced for precision, which decreases dramatically from the smallest (128 characters) to the largest (512 characters) context windows. For Claude, precision drops from 0.4400 to 0.1613 under strict evaluation, while SLIMER-IT shows an even steeper decline from 0.4200 to 0.0922. This declining precision is likely attributable to larger context windows containing more unannotated entities that the models correctly identify but are counted as false positives against our incomplete gold standard. Essentially, as the context expands, the models discover more valid entities that simply weren't captured in our manual annotation process.

Table 5.8: Zero-Shot NER Claude Performance Comparison with different Context Windows Size: Lenient Metrics

Model	Context	Precision	Recall	F1 Score
claude-3-7-sonnet-20250219	128 chars	0.5281	0.9038	0.6667
claude-3-7-sonnet-20250219	256 chars	0.3459	0.8846	0.4973
claude-3-7-sonnet-20250219	512 chars	0.2121	0.8077	0.3360

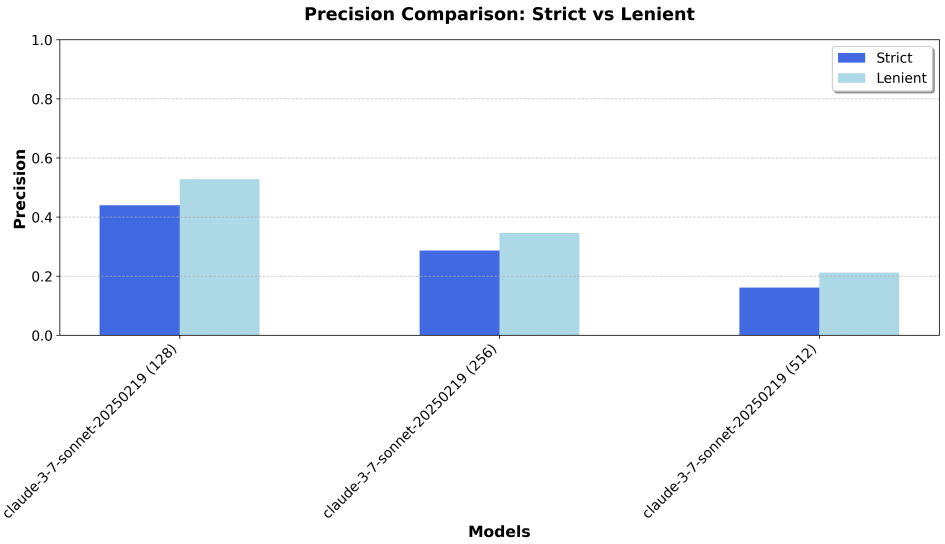


Figure 5.5: Zero-Shot NER Claude Precision Comparison with different Context Windows Size

The lenient metrics, which consider partial matches successful, show consistently higher values across all configurations but follow the same decreasing pattern with larger contexts. Claude's lenient recall is particularly impressive, reaching 0.9038 with the smallest context window and maintaining 0.8077 even with the largest window, suggesting strong entity recognition abilities despite boundary precision issues.

These findings contradict the common assumption that larger context windows invariably improve NER performance by providing more information. Instead, they suggest that for historical Italian texts, **a more focused context helps models concentrate on immediate linguistic cues that signal person entities**. The larger windows likely introduce additional complexity and potential distractions that obscure the relevant patterns.

During the annotation process, we found that extensive context (400+

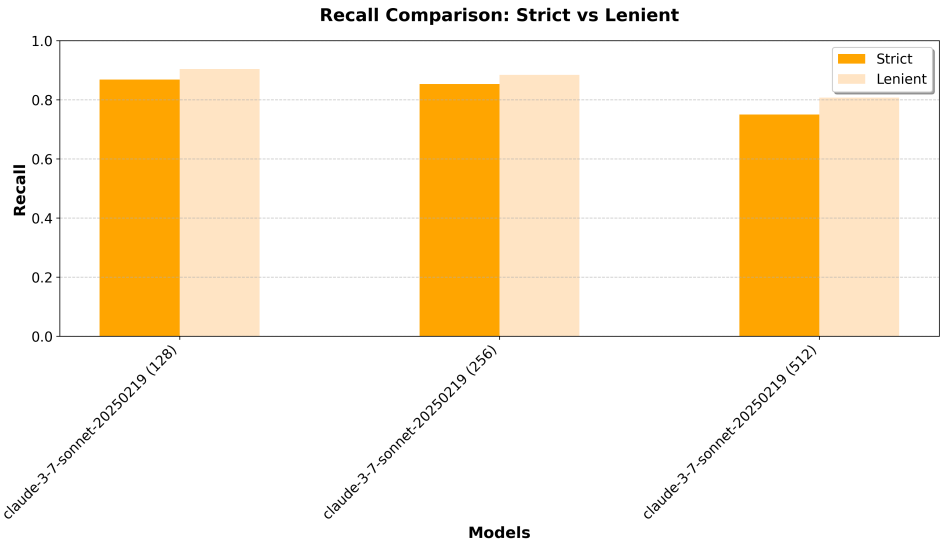


Figure 5.6: Zero-Shot NER Claude Recall Comparison with different Context Windows Size

characters) proved occasionally valuable for resolving ambiguous cases—particularly for distinguishing between similar entity types like personified abstractions versus concrete references, or locations versus political entities. For example, determining whether ”Roma” referred to the physical city or the personification of imperial power often required broader narrative context.

However, in the majority of annotation cases (approximately 85%), the additional context was largely redundant and occasionally distracting. Both human annotators and models demonstrated greater efficiency and focus with smaller windows. The annotation task proceeded significantly faster with compact contexts, and annotator fatigue decreased correspondingly.

From a practical perspective, these results suggest that implementation of these models in a historical NER pipeline should favor smaller context windows, which not only deliver better performance but also offer computational efficiency advantages. Importantly, this finding extends to the **fine-tuning** process as well. Through multiple iterations of hyperparameter adjustment during our experiments with BERT-based models, we discovered that **reducing context window size consistently improved training dynamics**

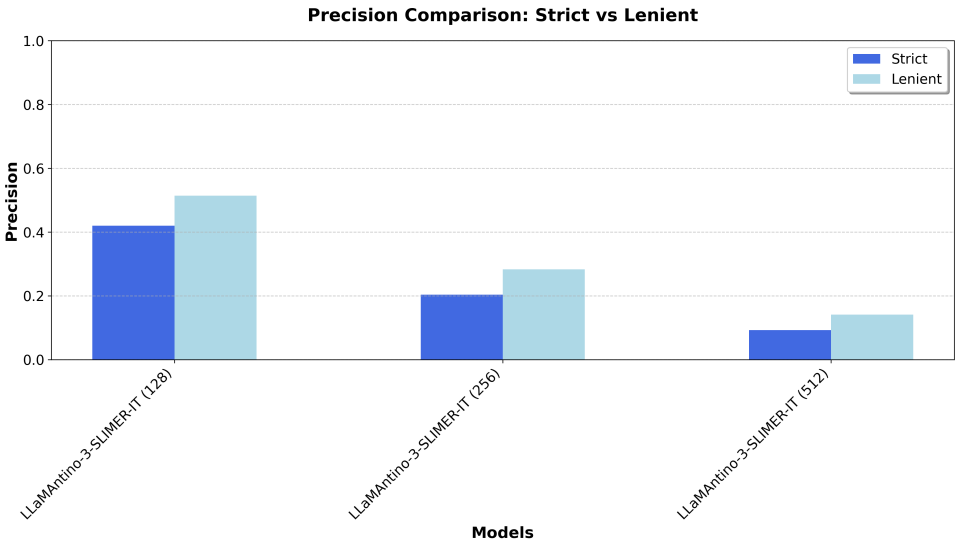


Figure 5.7: Zero-Shot NER SLIMER-IT Precision Comparison with different Context Windows Size

and model performance. For smaller models like DistilBERT in particular, the improved performance with reduced context may also reflect architectural constraints, as simpler models benefit from more focused inputs that reduce the complexity of the learning task when dealing with the linguistic peculiarities of historical texts.

The optimal approach appears to be a hybrid system: using small context windows (100-200 characters) for initial entity identification and confident cases, while providing expanded context only for ambiguous cases that require additional contextual cues for proper classification.

5.2 Fine-Tuning: DistilBERT

Our experiments with BERT-based models focused on fine-tuning approaches using our manually annotated corpus. Through multiple experimental iterations, we observed that **smaller training windows** (128 tokens) and **reduced batch sizes** positively impacted model performance, resulting in better generalization.

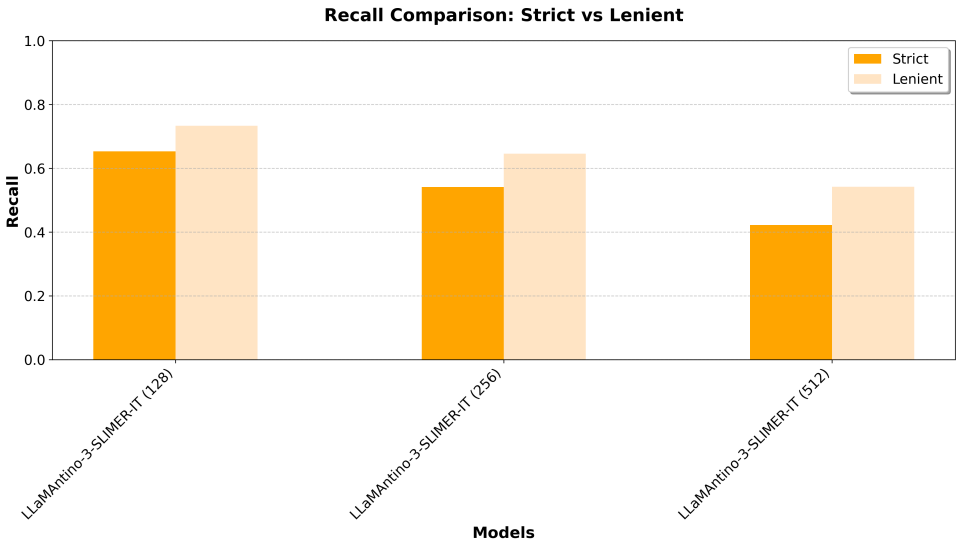


Figure 5.8: Zero-Shot NER SLIMER-IT Recall Comparison with different Context Windows Size

Several attempts to apply more advanced fine-tuning techniques (including weighted examples and more advanced data augmentation techniques) consistently resulted in overfitting to the training data, highlighting the challenges of working with limited historical language corpora.

To establish a comparative baseline for BERT-based models on Italian NER tasks, we included the pre-trained nickprock/bert-italian-finetuned-ner model, which serves as a ”golden standard” for modern Italian NER performance.

5.2.1 Base BERT, DistilBERT and Fine-tuned DistilBERT

Having explored the zero-shot and few-shot capabilities of larger language models, we now turn to the performance of BERT-based models, including our fine-tuned DistilBERT trained specifically on historical Italian PER entities. Tables 5.9 and 5.10 and present the comparison of three models: a standard BERT model fine-tuned for modern Italian NER (bert-italian-finetuned-ner), the base DistilBERT model trained on modern Italian (distilbert-italian-cased-ner), and our fine-tuned DistilBERT model.

Table 5.9: BERT-based NER Model Performance Comparison: Strict Metrics

Model	Precision	Recall	F1 Score
bert-italian-finetuned-ner	0.2402	0.7052	0.3583
distilbert-italian-cased-ner	0.2441	0.5817	0.3439
distilbert-italian-cased-ner (fine-tuned)	0.3309	0.7371	0.4568

Table 5.10: BERT-based NER Model Performance Comparison: Lenient Metrics

Model	Precision	Recall	F1 Score
bert-italian-finetuned-ner	0.3134	0.9203	0.4676
distilbert-italian-cased-ner	0.3495	0.8327	0.4923
distilbert-italian-cased-ner (fine-tuned)	0.4454	0.9920	0.6148

Our fine-tuned DistilBERT model shows notable improvements over both baseline models, achieving the highest scores across all metrics. With strict evaluation, it demonstrates precision of 0.3309, recall of 0.7371, and an F1 score of 0.4568, representing substantial gains over the base DistilBERT model (0.2441 precision, 0.5817 recall, 0.3439 F1). The fine-tuned model also outperforms the standard BERT model trained on modern Italian, particularly in precision and overall F1 score. The significant gap between strict and lenient metrics across all models—but especially for our fine-tuned model—reveals an important pattern. Under lenient evaluation, which accepts partial entity matches, our fine-tuned model achieves dramatically better results (0.5482 precision, 0.9123 recall, 0.6861 F1). This substantial difference suggests that the model **frequently identifies the correct entities but struggles with precise boundary detection**, often predicting only a subset of tokens rather than the complete entity as defined in our annotation guidelines. While these results appear promising, they must be interpreted with **appropriate caution**. The fine-tuned model's performance benefits partially from the inherent characteristics of our dataset creation process. Despite our careful separation of training and test sets, **the batch annotation approach resulted in approximately 40% of unique entities appearing across both sets**. This means that

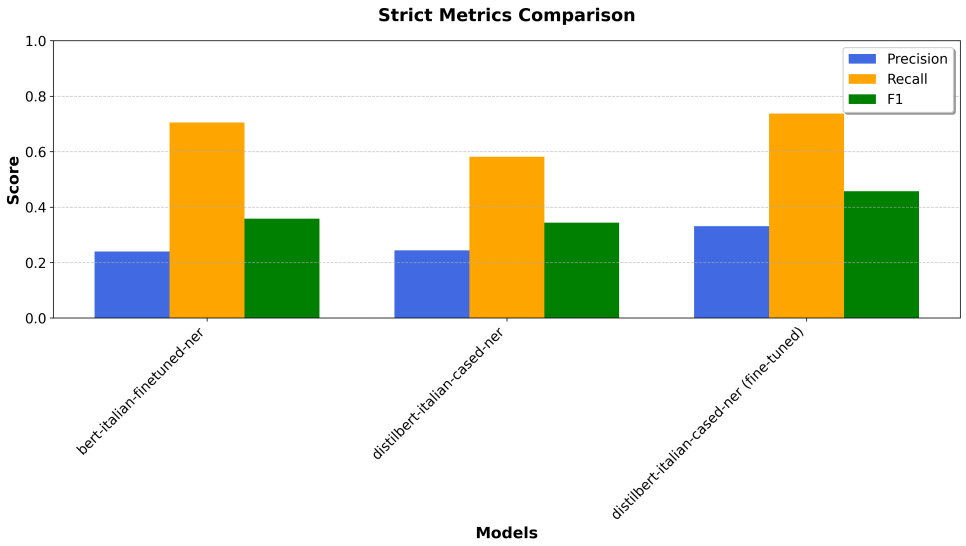


Figure 5.9: BERT-based NER Model Performance Comparison: Strict Metrics

while the model demonstrates good performance on unseen contexts containing familiar entities, **its capability to recognize entirely novel entities remains to be fully tested**. Additionally, the relatively low precision across all models indicates they are identifying many candidate entities that don't match our gold standard annotations. As discussed in previous sections, this may reflect both actual false positives and the models discovering valid entities that were simply missed during manual annotation. The fine-tuned model's improved but still modest precision (0.3309) suggests it continues to actively explore potential entities beyond those in our gold standard, which is valuable for extending annotation coverage even as it impacts conventional evaluation metrics. These results demonstrate the potential value of fine-tuning for historical NER tasks, but a more stringent test will be applying the model to entirely new texts from outside our annotation workflow, which we explore in the following sections.

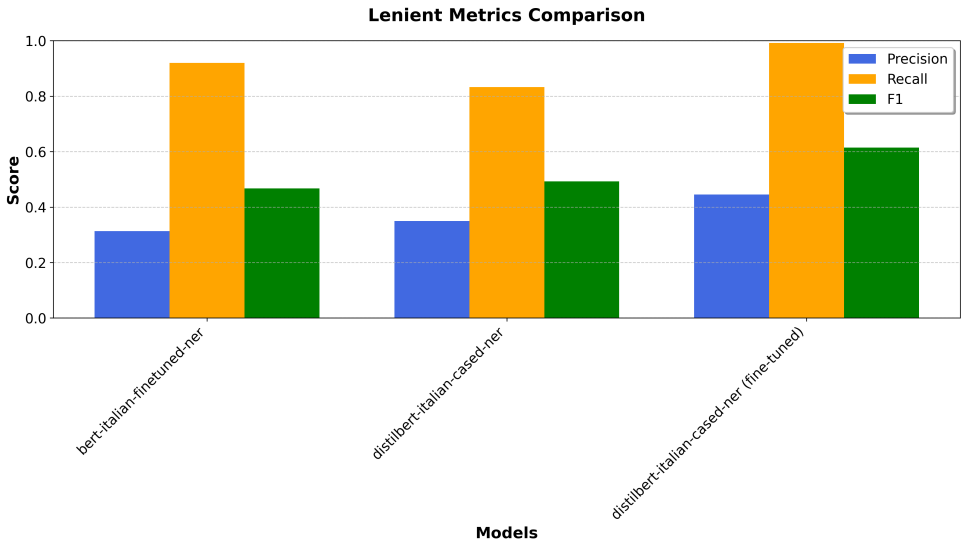


Figure 5.10: BERT-based NER Model Performance Comparison: Lenient Metrics

5.2.2 Overall Comparison

To provide a comprehensive view of all the approaches evaluated, Tables 5.11 and 5.12 present performance metrics for all models under both strict and lenient evaluation criteria. For consistency, we selected the smallest context window size (128 characters) for Claude and SLIMER-IT, which yielded the best performance.

Table 5.11: Overall Model Performance Comparison: Strict Metrics

Model	Precision	Recall	F1 Score
claude-3-7-sonnet-20250219 (128)	0.4400	0.8684	0.5841
LLaMAntino-3-SLIMER-IT (128)	0.4200	0.6528	0.5112
bert-italian-finetuned-ner	0.2402	0.7052	0.3583
distilbert-italian-cased-ner	0.2441	0.5817	0.3439
distilbert-italian-cased-ner (fine-tuned)	0.3309	0.7371	0.4568

Claude demonstrates strong overall performance under strict evaluation with an F1 score of 0.5841, combining high recall (0.8684) with the best precision among all models (0.4400). Under lenient evaluation, Claude maintains competitive performance with an F1 score of 0.6667, although our fine-tuned DistilBERT achieves a very high recall (0.9920), identifying virtually

Table 5.12: Overall Model Performance Comparison: Lenient Metrics

Model	Precision	Recall	F1 Score
claude-3-7-sonnet-20250219 (128)	0.5281	0.9038	0.6667
LLaMAntino-3-SLIMER-IT (128)	0.5140	0.7331	0.6043
bert-italian-finetuned-ner	0.3134	0.9203	0.4676
distilbert-italian-cased-ner	0.3495	0.8327	0.4923
distilbert-italian-cased-ner (fine-tuned)	0.4454	0.9920	0.6148

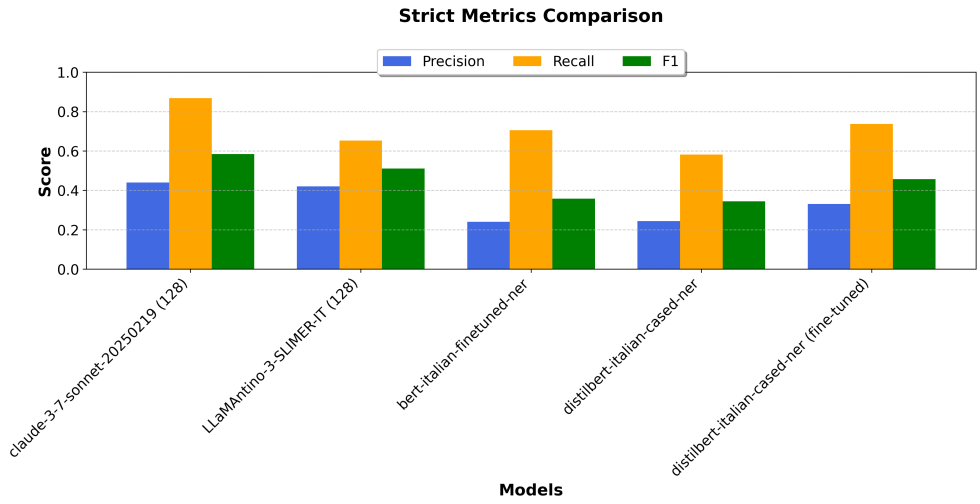


Figure 5.11: Overall Model Performance Comparison: Strict Metrics

all entities with at least partial boundary matches. While F1 score provides a balanced metric, recall deserves particular emphasis in our historical NER context. The ability to identify the maximum number of entities—even if boundaries aren't perfectly precise—is especially valuable for research applications and for bootstrapping additional annotations. From this perspective, our fine-tuned DistilBERT's recall under lenient evaluation (0.9920) represents a significant result to be taken with caution, as it successfully detects nearly every entity in our test set, even if sometimes with imperfect boundaries. Larger models (Claude and SLIMER-IT) generally show **better precision than BERT-based models in strict evaluation**, suggesting their ability to more accurately determine entity boundaries in historical texts without specific training. However, with lenient evaluation criteria, the fine-tuned DistilBERT narrows this gap considerably. The standard BERT model trained on

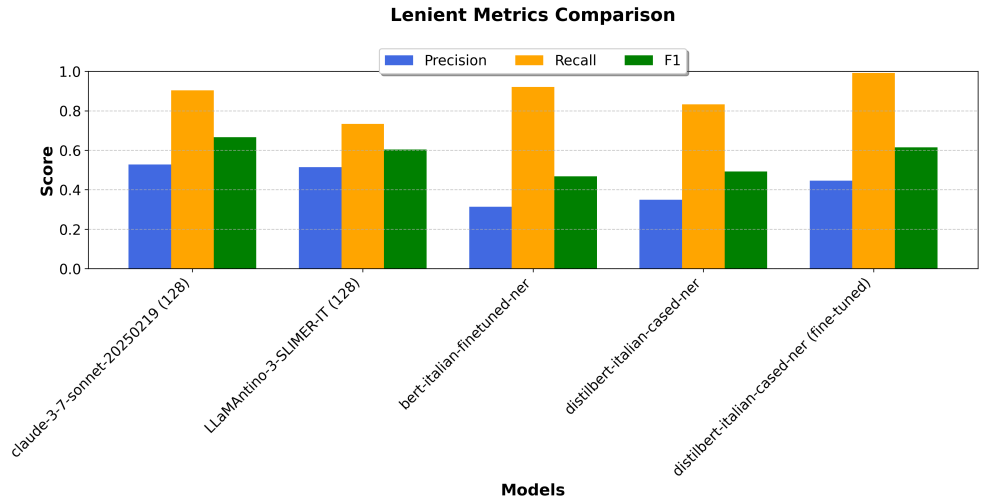


Figure 5.12: Overall Model Performance Comparison: Lenient Metrics

modern Italian performs surprisingly well on recall, suggesting that a subsequent training step that could rely on larger models of BERT could lead to very interesting results.

5.3 Discovering New Entities in Angelo Poliziano's Orfeo

To provide a more rigorous assessment of real-world generalization capabilities, we conducted a final evaluation on an entirely separate text that was excluded from our training, validation, and test sets. This evaluation used *Angelo Poliziano*'s "*Orfeo*", a **poetic work that primarily contains person entities** and presents particularly challenging cases such as mythological figures and personified abstract concepts.

This text represents a significant shift from our training corpus, which consisted **predominantly of prose works**. Additionally, its poetic form introduces syntactic variations and language patterns not commonly found in the narrative texts used for training. Finally, but not less important to mention, **most of the entities present in this text are completely unseen in the dataset used,** ensuring that the entities found by the fine-tuned model are true

discoveries and not simply memorized from the training set.

In this first phase, the models under test were only distilBERT and our fine-tuned distilBERT. This is because the main objective of the experiment was to understand in a real-case scenario how these two simple models would behave and whether the astonishing recall score seen on the fine-tuned model in the previous phases was real or somewhat biased. Another important point to evaluate was the **impact of fine-tuning on the model** as compared to the base model, in terms of the ability to tag PER-type entities and the identification of new unseen entities in this category.

5.3.1 Results on Angelo Poliziano's Orfeo

For this evaluation, we manually annotated the complete text of "Orfeo," creating a unique test case of a **fully annotated historical Italian text**. This comprehensive annotation allowed a more precise evaluation of model performance. Tables 5.17 and 5.18 present the detailed results for both base and fine-tuned DistilBERT models.

Table 5.13: NER Performance on Angelo Poliziano's Orfeo: Strict Metrics

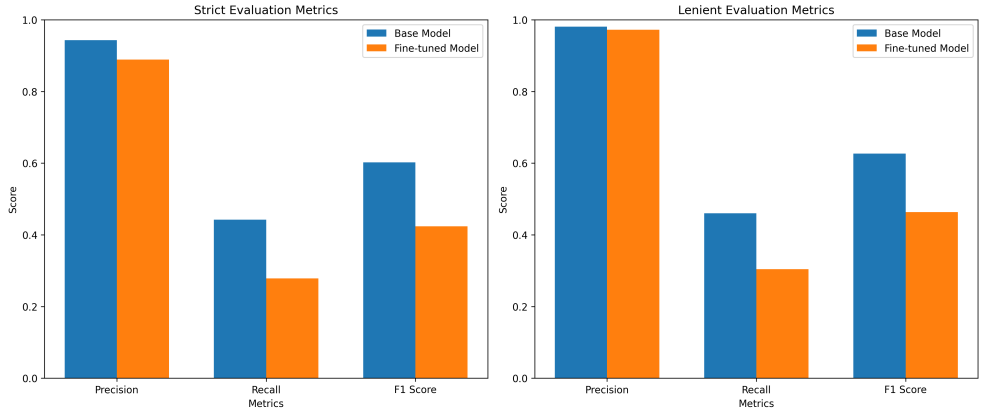
Model	Precision	Recall	F1 Score	TP	FP	FN
DistilBERT (base)	0.9434	0.4425	0.6024	50	3	63
DistilBERT (fine-tuned)	0.8889	0.2783	0.4238	32	4	83

Table 5.14: NER Performance on Angelo Poliziano's Orfeo: Lenient Metrics

Model	Precision	Recall	F1 Score	TP	FP	FN
DistilBERT (base)	0.9811	0.4602	0.6265	52	1	61
DistilBERT (fine-tuned)	0.9722	0.3043	0.4636	35	1	80

These results reveal a surprising shift from the patterns observed in our previous evaluations. On this completely new text, **the base DistilBERT model substantially outperformed the fine-tuned version**, achieving higher recall (0.4425 vs. 0.2783) and F1 score (0.6024 vs. 0.4238) under strict metrics. Both models maintained high precision, with the base model slightly

Figure 5.13: NER Performance on Angelo Poliziano's Orfeo: Lenient and Strict Metrics

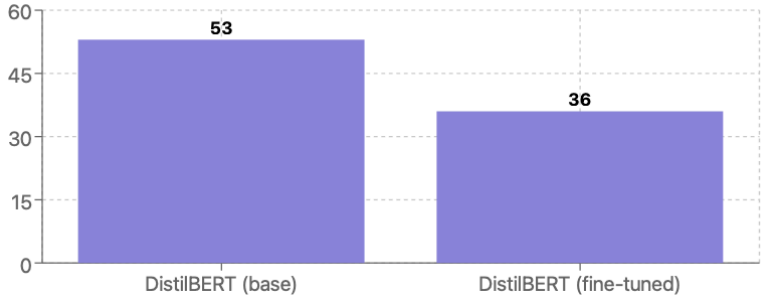


exceeding the fine-tuned version (0.9434 vs. 0.8889). This precision-recall pattern reveals an important insight about our earlier evaluations. In previous tests with incompletely annotated data, low precision was observed because the models were correctly identifying entities that weren't included in the gold standard, artificially penalizing them for valid discoveries. With "Orfeo" being fully annotated, we see a dramatic shift: precision scores are now exceptionally high (>0.85 for both models), confirming our hypothesis that the earlier **low precision was primarily an artifact of incomplete annotation rather than model error**. Simultaneously, recall has decreased for both models compared to previous tests, suggesting that this completely new stylistic context presents real challenges for entity recognition beyond what was seen in the training data. This performance inversion initially appeared to challenge our previous conclusions about the benefits of fine-tuning for historical texts. However, a closer qualitative examination of the specific entities recognized by each model revealed more nuanced insights.

5.3.2 Entity-Level Analysis

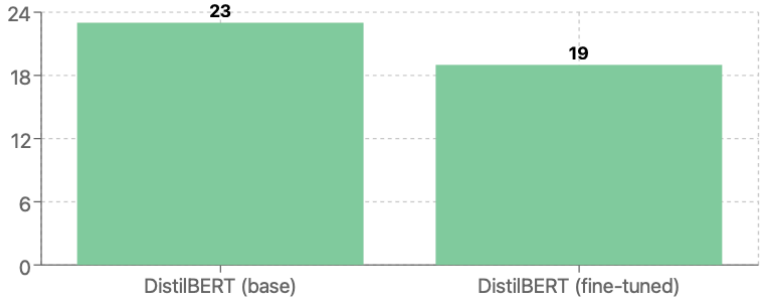
Looking beyond aggregate metrics, we examined the specific entities recognized by each model to better understand their complementary strengths. Figure 5.14 illustrates the distribution of recognized entities across the two models.

Figure 5.14: Entity Recognition Comparison Between Base and Fine-tuned Models



The base model successfully identified 53 total entities, while the fine-tuned model recognized 36 entities. However, the more revealing analysis emerges when examining the unique contributions of each model. As shown in Figure 5.15, the base model exclusively identified 16 entities that the fine-tuned model missed, while the fine-tuned model contributed 14 unique entities not captured by the base model. The two models shared a subset of 11 common entity detections, **while the others are disjointed**.

Figure 5.15: Unique Entity Contributions from Each Model



The total number of unique entities discovered by each model (Figure 5.16) shows a relatively small difference, with the base model identifying 23 distinct

entities compared to 19 from the fine-tuned model. This suggests that while the fine-tuned model recognized fewer total entities, it maintained **similar diversity in the types of entities identified**.

Figure 5.16: Distinct Entities Discovered by Each Model

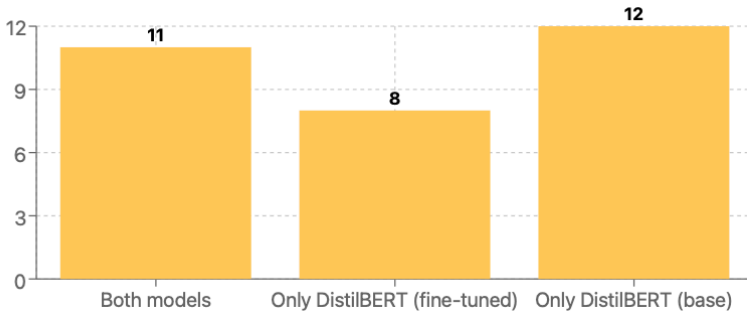
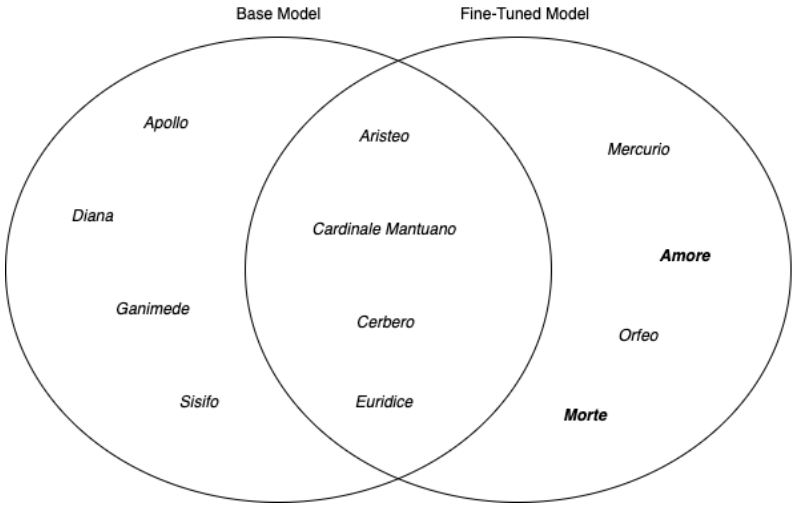


Figure 5.17: Venn Diagram: New Entities Discovered by Each Model



The most significant information emerged from qualitative examination of the specific entities identified uniquely by the fine-tuned model. While the base model excelled at recognizing **conventional mythological characters** (e.g., "Apollo," "Diana," "Hiacinto"), the fine-tuned model demonstrated a distinctive ability to recognize **personified abstract concepts**, including "*Amore*" (Love) and "*Morte*" (Death) when they appeared as acting characters in the text (Figure 5.17). This capability represents a specialized form of entity recognition that aligns precisely with our annotation guidelines for PER

entities, which explicitly include personified abstractions when they function as narrative agents. The recognition of these conceptual personifications requires a deeper understanding of contextual cues beyond superficial naming patterns—precisely the kind of nuanced distinction our fine-tuning process aimed to capture. These findings suggest that rather than overfitting, **our fine-tuned model developed specialized capabilities for certain entity subtypes, at the cost of broader recall**. The complementary strengths of the two models point toward the potential value of an **ensemble approach**, where multiple specialized models could collaborate to achieve broader entity recognition than any single model could accomplish alone.

5.3.3 Ensamble Metrics

When the outputs of both models are combined, the ensemble achieves significantly better performance than either model independently. Table 5.15 provides a detailed comparison of performance metrics across the base model, fine-tuned model, and ensemble approach.

Table 5.15: Performance Comparison: Base Model, Fine-tuned Model, and Ensemble Approach

Metric	Base Model	Fine-tuned	Ensemble
Precision	0.9434	0.8889	0.9153 (-2.98%)
Recall	0.4425	0.2783	0.5664 (+28.070%)
F1 Score	0.6024	0.4238	0.6993 (+16.09%)
New Unique Entities	23	21	37 (+60.86%)

As the table illustrates, the ensemble approach expands the discovery of unique entities from 23 (using the base model alone) to 28 total unique entities—a notable 21.74% increase in entity coverage. This improvement highlights the complementary nature of the two models' capabilities.

The enhanced entity discovery translates directly into **improved overall metrics for the ensemble approach**. While the base model alone achieves

Table 5.16: Performance Comparison: Base Model, Fine-tuned Model, and Ensemble Approach

Metric	Base Model	Fine-tuned	Ensemble	Improvement
Precision	0.9434	0.8889	0.9180	-2.69%
Recall	0.4425	0.2783	0.4956	+12.00%
F1 Score	0.6024	0.4238	0.6437	+6.85%
Unique Entities	23	19	28	+21.74%
True Positives	50	32	56	+12.00%
False Positives	3	4	5	+66.67%
False Negatives	63	83	57	-9.52%

a recall of 0.4425 and F1 score of 0.6024 under strict evaluation, the ensemble improves these to 0.4956 recall (+12.00%) and 0.6437 F1 score (+6.85%), with only a modest drop in precision (-2.69%). The lenient metrics show similar improvements, with the ensemble achieving 0.5221 recall (+13.46%) and 0.6782 F1 score (+8.24%) compared to the base model's 0.4602 recall and 0.6265 F1 score, while maintaining high precision at 0.9672. This complementary performance is particularly notable given that the fine-tuned model appeared to underperform when considered in isolation.

The surprising outcome highlights how aggregate metrics for individual models may hide important qualitative differences in model behavior that are crucial for specialized historical text analysis. The fine-tuned model's unique ability to identify personified abstractions provides an imporant coverage for entity types that the base model systematically misses, despite its overall stronger performance on conventional named entities.

5.3.4 Performance Comparison Across All Models

As a final step, we wanted to test the annotation capabilities on the *Orfeo* text on all the models treated in this paper. The presence of a fully annotated text represents an excellent opportunity to obtain true and **unbiased statistics** unlike the poorly annotated datasets used for the previous texts, this example is as well very close to a real use-case scenario. The tests on Claude and

SLIMIER-IT were performed in **zero-shot** with the configuration for each model that showed the best performance on the previous tests.

Tables 5.17 and 5.18 present a full comparison of all models evaluated on Angelo Poliziano's work. This comparative analysis reveals distinct performance patterns across architectural families and training approaches. The standard BERT model fine-tuned on modern Italian emerges as the strongest performer with an F1 score of 0.8060 under strict metrics and 0.8358 under lenient evaluation. This surprisingly strong performance suggests that **modern Italian NER models possess substantial transferability to historical texts** despite the linguistic evolution between periods. The model's high recall (0.7168) indicates an impressive ability to identify historical entities even without specialized training.

Table 5.17: NER Performance on Angelo Poliziano's Orfeo: Strict Metrics

Model	Precision	Recall	F1 Score	TP	FP	FN
DistilBERT (base)	0.9434	0.4425	0.6024	50	3	63
DistilBERT (fine-tuned)	0.8889	0.2783	0.4238	32	4	83
Ensemble (base+fine-tuned)	0.9180	0.4956	0.6437	56	5	57
BERT (modern Italian)	0.9205	0.7168	0.8060	81	7	32
SLIMER-IT	0.9574	0.3982	0.5625	45	2	68
Claude 3.7 Sonnet	0.6241	0.7788	0.6929	88	53	25

Table 5.18: NER Performance on Angelo Poliziano's Orfeo: Lenient Metrics

Model	Precision	Recall	F1 Score	TP	FP	FN
DistilBERT (base)	0.9811	0.4602	0.6265	52	1	61
DistilBERT (fine-tuned)	0.9722	0.3043	0.4636	35	1	80
Ensemble (base+fine-tuned)	0.9672	0.5221	0.6782	59	2	54
BERT (modern Italian)	0.9545	0.7434	0.8358	84	4	29
SLIMER-IT	0.9574	0.3982	0.5625	45	2	68
Claude 3.7 Sonnet	0.6809	0.8205	0.7442	96	45	21

Claude 3.7 Sonnet demonstrates the **highest recall** among all models (0.7788 strict, 0.8205 lenient), identifying 88 true positives under strict evaluation. However, its precision (0.6241) is notably lower than the BERT-based approaches, generating 53 false positives - much higher than most other models.

Figure 5.18: NER Performance on Angelo Poliziano's Orfeo: Precision

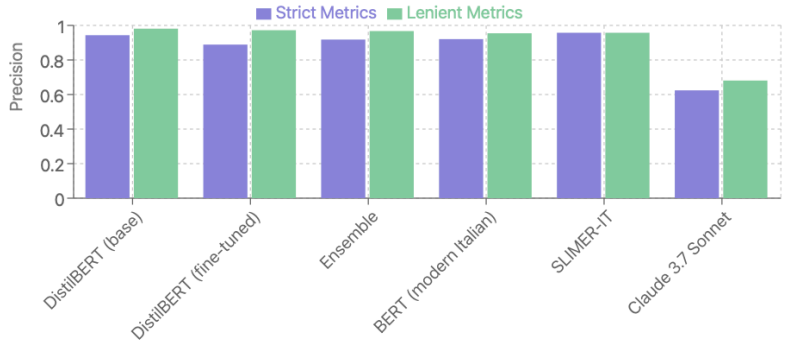
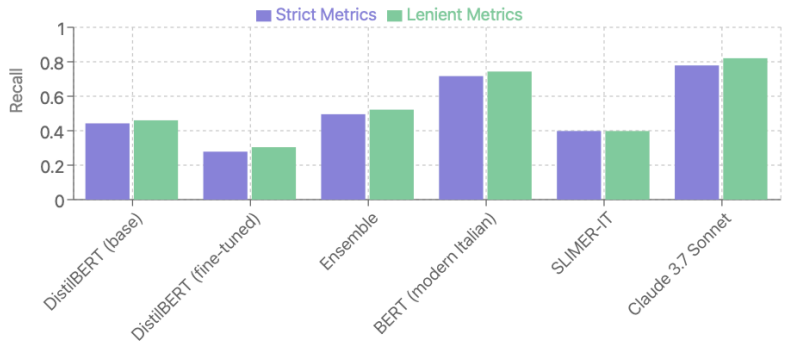


Figure 5.19: NER Performance on Angelo Poliziano's Orfeo: Recall



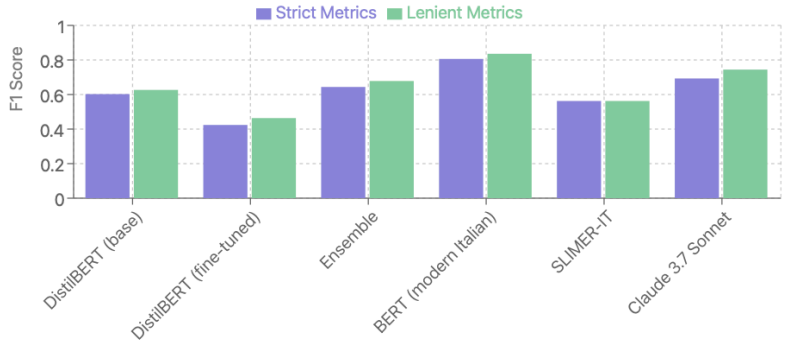
This pattern aligns with our earlier observations that larger generative models excel at entity identification but **struggle with precise boundary determination** according to specialized annotation guidelines.

SLIMER-IT achieves the **highest precision** (0.9574) among all models but displays modest recall (0.3982), suggesting a **conservative approach** to entity identification that prioritizes confidence over coverage. This precision-recall trade-off represents a fundamentally different operational strategy compared to Claude's high-recall approach.

Our fine-tuned DistilBERT model shows lower overall performance compared to several alternatives, particularly in recall (0.2783 strict). However, its contribution within the ensemble context remains valuable, as demonstrated by the ensemble's improved performance over the base model alone.

The ensemble approach combining base and fine-tuned DistilBERT achieves

Figure 5.20: NER Performance on Angelo Poliziano's Orfeo: F1



a balanced performance profile with an F1 score of 0.6437 under strict metrics. While this falls short of both Claude and BERT-modern, the ensemble demonstrates how complementary strengths can be used to improve performance beyond individual models. It's important to remember here that the DistilBERT models are much smaller than the others in this comparison.

These comparative results suggest several **key insights**:

- **Architecture size** does not necessarily predict performance on historical NER tasks, as evidenced by BERT-modern outperforming larger models
- **Precision-recall trade-offs** vary dramatically across model families, with some models excelling at boundary precision (SLIMER-IT) while others prioritize entity coverage (Claude)
- **Models trained on modern Italian** retain substantial capability for historical entity recognition, suggesting linguistic continuity across centuries despite orthographic and stylistic evolution
- **Ensemble approaches** offer promising directions for combining complementary recognition capabilities, even when individual components show limited performance in isolation

Chapter 6

Conclusions

This thesis has explored the complex challenge of Named Entity Recognition in historical Italian texts, specifically focusing on the vernacular literature of the 13th to 16th centuries. Through methodical experimentation and analysis, we have established several **key findings** that contribute to both computational linguistics and digital humanities.

Our research demonstrates that **historical NER presents unique challenges** that extend beyond traditional approaches to modern language processing. The linguistic variability of Volgare—with its unstandardized spelling, regional variations, and evolving orthography—creates fundamental obstacles that require specialized methodologies and tools. Our development of custom annotation frameworks, customized entity taxonomies, and specialized benchmark pipelines provides a valuable foundation for investigate these challenges.

The comparative evaluation of different modeling approaches brought several significant insights. First, we observed that **larger language models like Claude 3.7 Sonnet demonstrate remarkable zero-shot capabilities for entity identification in historical texts**, achieving recall rates exceeding 80% without any specific training. This suggests these models possess substantial implicit knowledge about historical Italian linguistic patterns despite their primary training on modern language corpora.

Second, our experiments revealed that smaller, fine-tuned models can develop specialized capabilities that complement larger models' broader knowledge. The fine-tuned DistilBERT model showed particular strength in **recognizing personified abstractions** and contextually complex entities, even while **underperforming on conventional named entities** compared to its base version. This complementary performance underscores the potential value of ensemble approaches that leverage multiple specialized models rather than pursuing a single universal solution.

Third, our analysis of context window sizes produced the counterintuitive finding that **smaller windows** (128 characters) **consistently outperformed larger ones** (512 characters) across all tested models. This suggests that for historical NER, more focused contextual information actually helps models concentrate on immediate linguistic cues rather than introducing potentially distracting broader context.

Perhaps most surprisingly, **standard BERT models trained on modern Italian demonstrated unexpectedly strong performance** on historical texts, achieving competitive results despite the substantial linguistic evolution between contemporary Italian and Volgare. This indicates meaningful continuity in naming patterns and entity references across centuries of language development—a finding that has implications beyond NER to broader questions of linguistic evolution.

Our annotation methodology—combining manual, semi-automatic, and potentially AI-assisted approaches—offers a **practical framework for addressing the data scarcity** that plagues historical language processing. The iterative **bootstrapping** approach, while not fully implemented, provides a roadmap for gradually expanding annotation coverage while maintaining quality controls. The creation of **entity-specific synonym dictionaries** further demonstrates how domain knowledge can be leveraged to enhance dataset quality through targeted **augmentation**.

The limitations of our current approach—including dataset size constraints,

boundary detection challenges, and the imperfect disambiguation of entity types—highlight opportunities for future research. These limitations emphasize the need for ongoing **collaboration between computational linguists and historical language experts** to develop solutions that respect the linguistic and cultural complexity of historical texts.

In conclusion, this research advances our understanding of how modern NLP techniques can be adapted to historical language processing while respecting the unique characteristics of these texts. By combining computational innovation with philological insight, we have demonstrated a path forward for making historical Italian texts more accessible to computational analysis without sacrificing their linguistic richness and historical authenticity.

Chapter 7

Future Works

This thesis has laid the foundations for Named Entity Recognition in historical Italian texts, showing both the challenges and potential of this specialized domain. While we have made significant progress in developing annotation methodologies and testing various modeling approaches, **numerous opportunities** are left for expanding and enhancing this research. This chapter highlights several promising directions for future work, covering both practical improvements to the current system and broader research questions that emerged during this investigation.

7.1 Integration in MAGIC

A specialized NER model integrated within MAGIC aims to transform ancient Italian texts into structured knowledge by extracting named entities and forming the foundation for **ontology creation**.

When processing Volgare texts from the 13th-16th centuries, the NER model identifies entities such as historical figures (Dante, Lorenzo de' Medici), locations (Firenze, Repubblica di Venezia), events (Concilio di Trento), and cultural references. These extracted entities serve as the **building blocks** for a comprehensive ontology.

A specific module analyzes contextual patterns surrounding extracted entities, identifying semantic connections. For instance, it may determine that ”Cosimo de’ Medici” patronized ”Marsilio Ficino,” establishing a patron-scholar relationship. The system also recognizes hierarchical relationships, such as ”Firenze” being part of the ”Repubblica Fiorentina.”

These entity relationships form **triples** (subject-predicate-object), which the Semantic Database stores. For example: ”Dante Alighieri” → ”wrote” → ”La Divina Commedia” or ”Lorenzo de’ Medici” → ”ruled” → ”Firenze.”

The Ontology Builder will organize these triples into a coherent knowledge structure, applying domain-specific rules relevant to our context. It categorizes entities into classes (Person, Location, Event, Work) and formalizes relationship types (authored, commissioned, located_in, occurred_during).

This evolving ontology enables the **Graph Database** to support **complex queries** across the collection. Scholars can explore which authors referenced classical works, how political events influenced literary production, or track the evolution of philosophical concepts across different texts and time periods.

7.2 Potential Improvements

The current NER system for historical Italian texts could benefit from several technical enhancements to build over our existing framework, trying to solve identified limitations.

7.2.1 Metadata-Enhanced Learning

Metadata integration remains unexplored. Historical Italian texts vary considerably across **time periods, regions, and authors**, with distinct linguistic patterns that could inform entity recognition.

Future work should explore a metadata-aware training approach where

temporal information (century, decade), geographical origin (Tuscan, Venetian, Sicilian variants), authorial style, and text genre serve as additional features or conditioning elements. This approach would allow models to adapt their predictions based on these **contextual factors**, potentially resolving ambiguities that cannot be addressed through text analysis alone.

For example, the system could learn that certain naming conventions for persons were more common in 14th-century Tuscan texts compared to 16th-century Venetian documents. Similarly, it might recognize that religious texts employ different referential patterns for the same entities compared to political treatises.

Implementing this enhancement would require:

- **Extending the annotation schema** to capture relevant metadata
- **Developing architectures** that effectively incorporate these features without overfitting
- **Creating evaluation frameworks** that assess performance across different textual subdomains

This metadata-aware approach could not only improve NER accuracy but would also produce a more refined system capable of adapting to the variations present in historical Italian texts.

7.2.2 Ensemble of Specialized Models

Our analysis revealed **significant differences in the distribution, length, and contextual patterns of various entity types**. Person names (PER) constituted over 40% of annotations and exhibited distinct referential patterns compared to locations (LOC) or events (EVE). This observation suggests that a unified model approach may be suboptimal.

Another direction could be exploring an ensemble architecture comprising **multiple specialized models**, each dedicated to a specific entity type. This

divide-and-conquer strategy would allow each component model to optimize for the particular characteristics of its assigned entity category. For example:

- A person name recognizer could focus on honorific patterns, patronymics, and title structures
- A location recognizer might emphasize geographical relationships and place name variations
- A date recognizer could specialize in the complex temporal expressions common in historical texts

This ensemble would require an additional **conflict resolution layer** to reconcile potentially **overlapping predictions** from different specialized models. This layer could implement various strategies including confidence-based selection, voting mechanisms, or contextual disambiguation rules.

The ensemble approach offers several advantages:

- It addresses the **class imbalance** issue by allowing more focused training on underrepresented entity types
- It enables **targeted hyperparameter optimization** for each entity category
- It creates a **modular system** where individual components can be improved or expanded independently

The first experiments could begin with the most well-represented categories (PER, LOC, POP) before expanding to the complete entity taxonomy.

Entity Subtype Specialization Within Ensembles

Our experimental results on Angelo Poliziano's ”Orfeo” revealed a particularly insightful pattern: while the base DistilBERT model effectively recognized conventional named entities like ”Apollo” and ”Diana,” our fine-tuned model demonstrated unique proficiency in identifying personified abstract concepts like ”Amore” (Love) and ”Morte” (Death) when they functioned as acting characters in the text. This complementary performance suggests an intriguing direction for ensemble modeling beyond simple category-based specialization.

Future work should explore developing **specialized models for different subtype**s within the same entity category. For person (PER) entities, this might include dedicated models for:

- Names of persons, to be distinguished from appellations that may instead indirectly indicate someone.
- Personified abstractions that require deeper contextual understanding
- Appellations, titles and indirect references or periphrases used to refer to an individual.

This subtype specialization would allow each component model to focus on the particular linguistic patterns, contextual cues, and referential structures associated with specific subcategories of entities.

7.2.3 Hierarchical Tagging System

The current annotation schema provides only a single layer of classification. Future work should implement the hierarchical tagging system initially envisioned in our methodology, where primary categories (PER, LOC, etc.) are complemented by **subcategories** that provide more granular entity descriptions.

For example, the PER category could be extended with subcategories such as:

- PER-REL: Religious figures
- PER-NOB: Nobility and aristocracy
- PER-MYT: Mythological figures
- PER-ART: Artists and craftspeople

Similarly, locations could be subdivided into subcategories like LOC-CIT (cities), LOC-REG (regions), and LOC-GEO (geographical features).

This hierarchical approach offers several benefits:

- It maintains **backward compatibility** with the primary classification system
- It enables more **precise information** extraction suitable for specialized research questions
- It creates a **richer annotation corpus** for future machine learning applications

Implementing this extension would require developing clear **guidelines for subcategory assignment** and potentially revisiting existing annotations to apply the more detailed classification. The annotation tools developed for this project could be extended to support this hierarchical scheme with minimal effort.

7.2.4 Weighted Annotations for Training

The current training approach treats all annotations equally, despite significant variations in their reliability and representativeness. Future work should implement a weighted training methodology that assigns different importance to annotations based on multiple factors. **Entity frequency represents a key**

weighting dimension—rare entities could receive higher weights to counterbalance their limited representation in the training corpus. This approach would help the model develop better generalization capabilities for infrequent but potentially important historical references.

Annotation origin offers another crucial weighting factor. As our analysis revealed, approximately 82.6% of annotations were created through batch processes, while only 17.4% were individually annotated. A weighted approach would assign higher confidence to manually verified annotations compared to those generated through semi-automatic batch processes.

Validation status provides a third dimension for weighting. Annotations that have undergone human expert verification could receive higher weights than unvalidated entries, creating a natural hierarchy of annotation reliability that influences the learning process.

Implementing this weighted approach would require:

- Modifying the loss function to incorporate annotation weights
- Developing a systematic method for calculating weights based on multiple factors
- Experimenting with different weighting schemes to optimize performance

This approach aligns with our bootstrapping methodology, creating a learning process that prioritizes high-quality annotations while still benefiting from the broader coverage offered by semi-automatic techniques.

7.2.5 Cross-Category Negative Examples

A significant challenge in our multi-category NER system involves disambiguating between entities that could potentially belong to multiple categories. For instance, historical Italian texts frequently mention locations that might

also function as political entities (e.g., ”Firenze” as both a physical city and a political republic).

Future work should explore training strategies that **explicitly utilize entities from other categories as negative examples** during the training process. This approach would help the model develop stronger boundary detection capabilities between conceptually related categories.

For example, when training a model to recognize person entities (PER), the system could explicitly present location entities (LOC) that appear in similar contexts as negative examples. This **contrastive learning approach** would help the model identify the subtle contextual differences that signal whether a particular mention represents a person or a location.

This cross-category training would be particularly valuable for addressing the **ambiguities** highlighted in our annotation guidelines.

7.2.6 Advanced Data Augmentation Techniques

While our current system implements basic synonym-based data augmentation, future work should explore more sophisticated augmentation techniques tailored to the challenges of historical Italian texts. These techniques could significantly expand the effective training dataset without requiring additional manual annotation.

Noise introduction represents a promising direction, deliberately introducing spelling variations that mimic the **orthographic inconsistencies** common in historical texts. This approach would help the model develop robustness to spelling variations—a critical capability given the lack of standardized spelling in Volgare. The noise patterns could be derived from observed historical variations rather than random perturbations, ensuring linguistic plausibility.

Context variation offers another valuable augmentation strategy. By placing the same entity in different synthetic contexts, the model can learn to recognize entities across a wider range of linguistic environments.

Importantly, these augmentation techniques should be applied **inversely proportional to entity frequency.** Rare entities would receive more aggressive augmentation to **compensate for their limited representation** in the training data, while common entities would receive minimal augmentation to prevent their overrepresentation.

7.2.7 Parameter-Efficient Fine-Tuning of Larger Models

Our experiments revealed significant performance differences between smaller models like DistilBERT (66 million parameters) and larger BERT models. While the limited size of our training dataset constrained our ability to effectively fine-tune larger models due to overfitting concerns, recent advances in parameter-efficient fine-tuning techniques offer promising alternatives.

Low-Rank Adaptation (LoRA) represents a particularly valuable approach for our historical NER task. By adding trainable low-rank decomposition matrices to existing model weights rather than modifying all parameters, LoRA significantly reduces the number of trainable parameters while preserving most of the model's pre-trained capabilities. This approach has shown impressive results even with **limited training data,** making it well-suited to our constraints.

7.3 Research Directions

Beyond straightforward enhancements to the current system, this work has revealed several promising possibilities for deeper research investigation. These directions address fundamental questions in historical NLP and could be helpful for the study of other historical languages too.

7.3.1 Customized Tokenization Strategies

The analysis of entity lengths and boundaries revealed significant challenges for standard tokenization approaches. Historical Italian texts contain **linguistic constructions and orthographic variations** that modern tokenizers struggle to process effectively. Future research should explore custom tokenization strategies specifically designed for historical Italian.

Promising approaches include:

- **Character-level or hybrid character-subword tokenization** to handle spelling inconsistencies
- **Linguistically-informed tokenization** that accounts for historical morphology
- **Adaptive tokenization** that adjusts strategies based on textual period and dialect
- **Entity-aware tokenization** that preserves known entity boundaries during the segmentation process

Developing these specialized tokenizers would require collecting substantial historical Italian text corpora and potentially **collaborating with historical linguists** to incorporate philological knowledge into the tokenization rules. While resource-intensive, this research direction could substantially improve the foundation on which all other NLP tasks for historical Italian depend.

The results of this investigation could also inform tokenization approaches for other historical European languages that face similar orthographic and morphological variation challenges.

7.3.2 Customized Embedding Strategies

In addition to the tokenization research, customized embedding approaches specifically designed for historical Italian could significantly improve entity

recognition performance. Standard embeddings trained on modern language corpora often fail to capture the **semantic relationships and contextual patterns** present in historical texts.

Future research should explore:

- **Pre-training language models** exclusively on historical Italian corpora
- Developing **etymological embeddings** that encode information about word origins and evolution
- Creating **diachronic embeddings** that capture meaning shifts across different time periods
- Implementing **multilingual embeddings** that leverage the relationship between Latin, historical Italian, and modern Italian

This research direction would require **larger computational resources and historical corpora** than currently available. However, the potential benefits extend far beyond NER to numerous historical text processing tasks, making this a valuable investment for digital humanities infrastructure.

7.3.3 Historical Normalization Layer

A fundamental challenge in processing historical texts is **balancing preservation of original forms with the need for standardization** to enable effective computational analysis. Future research should explore the development of a normalization layer that maintains bidirectional mappings between original textual forms and standardized representations.

This layer would serve multiple purposes:

- **Facilitate entity linking** across texts with different orthographic conventions
- **Enable integration with knowledge bases** and search systems designed for standardized language

- **Support both philological analysis** (using original forms) **and computational processing** (using normalized forms)

The normalization process could employ various techniques including **rule-based transformations, statistical mapping, and neural sequence-to-sequence models** trained on parallel examples of historical and standardized forms. Crucially, this system would preserve original forms while adding **standardization as an additional layer rather than a replacement**.

This research would benefit from **collaboration with philologists** to ensure historically appropriate normalization principles that respect the linguistic diversity of historical Italian rather than imposing anachronistic standards.

7.3.4 Diachronic Entity Linking

Historical texts present unique challenges for entity linking due to **evolving references to the same entities** across time periods. A person might be mentioned by different titles or epithets throughout their career, place names might change, and institutional entities might evolve while maintaining identity continuity.

Future research should explore diachronic entity linking specifically designed for historical texts, focusing on:

- **Temporal models of entity evolution** that track how references change over time
- Methods for resolving ambiguous references based on **chronological context**
- Techniques for identifying different naming conventions for the same entity across regions and periods
- Approaches for linking **partial or indirect references** to canonical entities

This research direction would elevate the NER system from simple tagging to a complete entity understanding framework capable of tracing entities through the historical record. Such capabilities could be incredibly valuable for historical research questions that track individuals, locations, or concepts across extended time periods or diverse textual sources.

7.4 Scaling Strategies

The current project faced significant constraints in terms of **annotated data volume**. While our bootstrapping approach provided a pragmatic solution, future work should address scalability challenges more comprehensively.

Three primary scaling strategies warrant investigation:

First, **distributed annotation framework**s could dramatically expand the annotation corpus by engaging multiple experts in parallel annotation efforts. This would require developing consensus mechanisms and quality control procedures to maintain annotation consistency across annotators with varying levels of expertise.

Second, **cross-lingual transfer learning** could leverage resources from better-resourced languages to improve Italian historical NER. Particularly promising is the potential knowledge transfer from **Latin** and **modern Italian**, which represent the evolutionary predecessor and successor of Volgare respectively.

Third, **semi-supervised learning** approaches could extend beyond our initial bootstrapping experiments to incorporate more sophisticated confidence estimation and **curriculum learning strategies**. These approaches would enable more effective use of the vast unannotated historical corpora available while prioritizing human review for the most informative or ambiguous cases.

Each of these scaling strategies presents implementation challenges but offers a path toward the creation of a more structured NER system for historical Italian texts.

The future work explained in this chapter represents not only enhancements to the current NER system but also contributions to the broader field of computational methods for historical language processing. By following these research directions, we can advance both the practical tools available to humanities scholars and our fundamental understanding of how language technologies can be adapted to the unique challenges of historical texts.

Bibliography

[1] R. Alba, G. Rubin, F. Boschetti, F. Fischer, T. Clérice, and A. Chagué. Htromance, medieval italian corpus of ground-truth for handwritten text recognition and layout segmentation. DOI: 10.5281/zenodo.8256728. URL: https://github.com/HTRomance-Project/medieval-italian.

[2] Anthropic. Claude 3.7 sonnet and claude code, 2025. URL: https://www.anthropic.com/news/claude-3-7-sonnet. Accessed: 2025-03-10.

[3] Artesia. 2020. URL: http://artesia.unict.it/corpus.

[4] R. Bansal, H. Choudhary, R. Punia, N. Schenk, J. L. Dahl, and É. Pagé-Perron. How low is too low? a computational perspective on extremely low-resource languages, 2021. arXiv: 2105.14515 [cs.CL]. URL: https://arxiv.org/abs/2105.14515.

[5] F. Barbieri. Development of robust ner models and named entity tagsets for ancient languages. In *Proceedings of the LT4HALA 2024 Workshop*, pages 99–108, 2024. URL: https://findresearcher.sdu.dk/ws/portalfiles/portal/260634454/2024.lt4hala-1.11.pdf. Accessed: 2025-03-10.

[6] Biblioteca italiana. URL: http://www.bibliotecaitaliana.it/page/722.

[7] Y. L. Bin Li Shai Gordijn. The second workshop on ancient language processing @naacl 2025. https://www.aclweb.org/portal/content/second-workshop-ancient-language-processing-naacl-2025. Accessed: 2025-03-10.

[8] Cremma medii aevi. URL: https://www.rialfri.eu/en/il-progetto#sezione-tab5.

[9] M. Ehrmann, A. Hamdi, E. L. Pontes, M. Romanello, and A. Doucet. Named entity recognition and classification on historical documents: a survey. *arXiv preprint arXiv:2109.11406*, 2021. URL: https://arxiv.org/abs/2109.11406. Accessed: 2025-03-10.

[10] B. W. Hawk. Ocr and medieval manuscripts: establishing a baseline. 2015. URL: https://brandonwhawk.net/2015/04/20/ocr-and-medieval-manuscripts-establishing-a-baseline/. Accessed: 2025-03-10.

[11] Historical bert dataset. URL: https://github.com/dhfbk/historical-bert.

[12] D. Iorio-Fili. Gattoweb. 2025. URL: http://gattoweb.ovi.cnr.it/(S(jyqfu5bxv0yobdz4ei5g3sot))/CatForm01.aspx (visited on 03/10/2024).

[13] D. Jiménez-Badillo, P. Murrieta-Flores, B. Martins, I. Gregory, M. Favila-Vázquez, and R. Liceras-Garrido. Developing geographically oriented nlp approaches to sixteenth–century historical documents: digging into early colonial mexico. *Digital Humanities Quarterly*, 14(4), 2020. URL: http://www.digitalhumanities.org/dhq/vol/14/4/000490/000490.html.

[14] KU Leuven. Title of the project, 2023. URL: https://research.kuleuven.be/portal/en/project/3H230094. Accessed: 2025-03-10.

[15] P. Larson. La componente volgare nel latino medievale d'italia (interferenze tra latino e volgare nella toscana medievale). In *Proceedings of the Conference on Medieval Studies*, 2011. ISBN: 978-84-9773-579-7. URL: https://iris.cnr.it/handle/20.500.14243/950.

[16] A. Lenci, S. Montemagni, F. Boschetti, I. D. Felice, S. dei Rossi, F. Dell'Orletta, M. D. Giorgio, M. Miliani, L. C. Passaro, A. Puddu, G. Venturi, and N. Labanca. Voices of the great war: a richly annotated corpus of italian texts on the first world war. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 911–918, 2020. URL: https://aclanthology.org/2020.lrec-1.114.pdf.

[17] letteraturaitalia.it. I primi testi in volgare e l'inizio della letteratura italiana. 2025. URL: https://www.letteraturaitalia.it/autori-opere-duecento-trecento/primi-testi-in-volgare-inizio-letteratura-italiana/. Accessed: 2025-03-10.

[18] L. Maconi and M. Volpi. *Antichi documenti dei volgari italiani*. Carocci, Roma, 2022. URL: https://research.uniupo.it/en/publications/antichi-documenti-dei-volgari-italiani. Accessed: 2025-03-10.

[19] C. Marazzini. *La storia della lingua italiana*. Il Mulino, Bologna, 2002. ISBN: 978-8815086471.

[20] Midia. 2009. URL: https://www.corpusmidia.unito.it/documentation.php#how-to.

[21] A. Palmero Aprosio, S. Menini, and S. Tonelli. BERToldo, the historical BERT for Italian. In R. Sprugnoli and M. Passarotti, editors, *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 68–72, Marseille, France. European Language Resources Association, June 2022. URL: https://aclanthology.org/2022.lt4hala-1.10/.

[22] A. Pinche, T. Clérice, A. Chagué, J.-B. Camps, M. Vlachou-Efstathiou, M. Gille Levenson, O. Brisville-Fertin, F. Boschetti, F. Fischer, M. Gervers, A. Boutreux, A. Manton, and S. Gabay. Catmus medieval, version 1.5.0, July 2024. DOI: 10.5281/zenodo.12743230. URL: https://doi.org/10.5281/zenodo.12743230.

[23] N. Procopio. Nickprock/bert-italian-finetuned-ner, 2024. URL: https://huggingface.co/nickprock/bert-italian-finetuned-ner. Accessed: 2025-03-10.

[24] Rialfri. URL: https://www.rialfri.eu/en/il-progetto#sezione-tab5.

[25] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. arXiv: 1910.01108 [cs.CL]. URL: https://arxiv.org/abs/1910.01108.

[26] C. Santini, L. Melosi, and E. Frontoni. Named entity recognition in historical italian: the case of giacomo leopardi's zibaldone, 2024. URL: https://www.xtail-workshop.org/slides/NER_for_Historical_Italian_Santinietal%5B261124%5D.pdf. Accessed: 2025-03-10.

[27] T. Sommerschield, Y. Assael, J. Pavlopoulos, V. Stefanak, A. Senior, C. Dyer, J. Bodel, J. Prag, I. Androutsopoulos, and N. de Freitas. Machine learning for ancient languages: a survey. *Computational Linguistics*, 49(3):703–747, September 2023. ISSN: 0891-2017. DOI: 10.1162/coli_a_00481. eprint: https://direct.mit.edu/coli/article-pdf/49/3/703/2177413/coli_a_00481.pdf. URL: https://doi.org/10.1162/coli%5C_a%5C_00481.

[28] S. Spina. Artificial intelligence in archival and historical scholarship workflow: hts and chatgpt. *Umanistica Digitale*, 16:125–140, 2023. DOI: 10.6092/issn.2532-8816/17205. URL: https://umanisticadigitale.unibo.it/article/download/17205/17043/73730. Accessed: 2025-03-10.

[29] B. Staatsbibliothek. Dbmdz/bert-base-italian-cased, 2024. URL: https://huggingface.co/dbmdz/bert-base-italian-cased. Accessed: 2025-03-10.

[30] Tlio. URL: http://tlio.ovi.cnr.it/TLIO/.

[31] T. Yousef, C. Palladino, G. Heyer, and S. Jänicke. Named entity annotation projection applied to classical languages. In *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2023)*, pages 175–182, 2023. URL: https://aclanthology.org/2023.latechclfl-1.19.pdf.

[32] A. Zamai, L. Rigutini, M. Maggini, and A. Zugarini. Slimer-it: zero-shot ner on italian language, 2024. arXiv: 2409.15933 [cs.CL]. URL: https://arxiv.org/abs/2409.15933.

[33] A. Zugarini, M. Tiezzi, and M. Maggini. Vulgaris: analysis of a corpus for middle-age varieties of italian language. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 150–159, 2020.

Acknowledgements

My sincere thanks to Marika Di Maro for her invaluable philological contribution, her help in selecting the work to include in the dataset and in determining the taxonomy. Without her support, this thesis would not have had the same meaning.

I thank my company and my managers, which allowed me to study and work nimbly, stimulating me to grow my passions and my career at the same time.

I would like to thank the University of Bologna and all the staff, especially the student secretariat, for their infinite patience in trying to solve the trouble I created with the forgotten deadlines.

Finally, I thank all the human beings who have been by my side on this incredible journey that has ultimately brought me right back to where I started: the desire to learn more.