From Handwritten Keyword Spotting to Query-Guided Sentence Retrieval in Document Images using Large Language Models

George Voudiotis

Master Thesis

— ♦ —

Ioannina, October 2025

Τμημα Μηχανικων Η/Υ & Πληροφορικης
Πανεπιστημιο Ιωαννινων

Department of Computer Science & Engineering
University of Ioannina

From Handwritten Keyword Spotting to Query-Guided Sentence Retrieval in Document Images using Large Language Models

A Thesis

submitted to the designated

by the Assembly

of the Department of Computer Science and Engineering

Examination Committee

by

George Voudiotis

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN DATA AND COMPUTER SYSTEMS ENGINEERING

WITH SPECIALIZATION

IN DATA SCIENCE AND ENGINEERING

University of Ioannina

School of Engineering

Ioannina 2025

Examining Committee:

- **Christophoros Nikou**, Professor, Department of Computer Science and Engineering, University of Ioannina (Advisor)
- **Konstantinos Blekas**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Lisimachos Paul Kondis**, Professor, Department of Computer Science and Engineering, University of Ioannina

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Christoforos Nikou, of the Department of Computer Science and Engineering, University of Ioannina, for his continuous support, valuable guidance, and trust throughout this research project. I am also deeply thankful to Dr. Angelos P. Giotis, Postdoctoral Researcher at the same department, for his insightful advice and invaluable assistance during the preparation of this thesis.

I would like to extend my heartfelt thanks to my family for their constant love, support, and encouragement throughout my academic journey. I am especially grateful to my girlfriend, Eleni, for her patience, understanding, and unwavering support during this period. Finally, I wish to thank Nefeli E. Sextou and Aristides Panagiotidis-Diktampanis for their kind help and support.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abstract

George Voudiotis, M.Sc. in Data and Computer Systems Engineering, Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, 2025.
From Handwritten Keyword Spotting to Query-Guided Sentence Retrieval in Document Images using Large Language Models.
Advisor: Christophoros Nikou, Professor.

This thesis presents a novel sentence-level retrieval framework that extends traditional handwritten Keyword Spotting (KWS) toward contextual understanding of historical document images. Conventional KWS methods identify isolated word occurrences based solely on visual similarity, without capturing the surrounding linguistic context. In contrast, the proposed system integrates visual retrieval, selective transcription, and reasoning through Large Language Models (LLMs) to reconstruct coherent sentences from handwritten sources.

The pipeline begins with a segmentation-based Seq2Seq KWS model that produces a ranked list of visually similar word images for a given query. For each of the top-k ranked results produced by the baseline keyword spotter, a local neighbourhood is examined around the detected hit. Only the word images within this neighbourhood, determined either by a fixed or dynamically estimated window, are transcribed through Handwritten Text Recognition (HTR) using the Seq2Seq and TrOCR architectures. This selective transcription strategy enables efficient, localized processing instead of full-page transcription. In the first case, a tuneable fixed-size symmetric window determines the sentence length, whereas when the dynamic window approach is concerned, an LLM-based mechanism estimates how many left and right neighbouring words should be included to form a complete sentence. Within the dynamic windowing framework two distinct techniques are explored: (a) few-shot prompting, which infers neighbourhood boundary lengths from the target word alone,

and (b) segment-based prompting, where a short local fragment guides the model's boundary prediction. Both dynamic strategies employ pre-trained (few-shot) and fine-tuned LLaMA 3.2-3B and Mistral 7B models adapted via Low-Rank Adaptation (LoRA) to iteratively refine candidate sentences.

Experimental evaluation on the IAM handwriting dataset demonstrates that dynamic windowing significantly outperforms the fixed-size approach. Using the segment-based strategy, the system achieves BLEU = 77.7% and BERTScore = 85.0% on Seq2Seq transcriptions, confirming its ability to generate syntactically coherent and semantically faithful sentence hypotheses. Additional tests with TrOCR further validate robustness under noisier transcriptions. Although the segment-based configuration incurs higher computational cost, it delivers superior accuracy and contextual completeness compared to both fixed and few-shot strategies.

Overall, this work bridges image-based retrieval and language-based reasoning, introducing a scalable framework for Query-Guided Sentence Retrieval, and demonstrating how document summarization can be approached as a reduced sentence concatenation task derived from reconstructed textual segments. Beyond improving access to historical handwritten archives, it provides a foundation for future multimodal systems combining visual understanding, selective OCR, and generative language modeling.

Εκτεταμενη Περιληψη

Γεώργιος Βουδιώτης, Δ.Μ.Σ. στη Μηχανική Δεδομένων και Υπολογιστικών Συστημάτων, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, 2025.
Από τον Εντοπισμό Λέξεων στην Καθοδηγούμενη από Ερώτημα Ανάκτηση Προτάσεων σε Εικόνες Χειρόγραφων Κειμένων με χρήση Μεγάλων Γλωσσικών Μοντέλων.
Επιβλέπων: Χριστόφορος Νίκου, Καθηγητής.

Η αναζήτηση πληροφοριών σε χειρόγραφα ιστορικά έγγραφα αποτελεί ένα από τα πιο απαιτητικά πεδία της σύγχρονης Υπολογιστικής Όρασης και της Επεξεργασίας Φυσικής Γλώσσας. Παρά την πρόοδο που έχει σημειωθεί στην αυτόματη μεταγραφή και στην αναζήτηση λέξεων (*Keyword Spotting – KWS*), οι υπάρχουσες προσεγγίσεις εντοπίζουν μεμονωμένες λέξεις ή φράσεις βασισμένες αποκλειστικά σε οπτική ομοιότητα, χωρίς να ανακτούν το γλωσσικό πλαίσιο και τα συμφραζόμενα μέσα στα οποία αυτές εμφανίζονται. Ως αποτέλεσμα, ο χρήστης λαμβάνει αποσπασματική πληροφορία που δεν επιτρέπει βαθύτερη κατανόηση του περιεχομένου ή αυτοματοποιημένες διαδικασίες, όπως η περίληψη εγγράφων.

Η παρούσα μεταπτυχιακή διατριβή προτείνει ένα νέο πλαίσιο αναζήτησης σε επίπεδο πρότασης (*sentence-level retrieval framework*), το οποίο επεκτείνει το πρόβλημα του *Keyword Spotting (KWS)* προς τη νοηματική κατανόηση και ανακατασκευή συμφραζομένων. Στόχος είναι η μετατροπή των αποτελεσμάτων ενός παραδοσιακού συστήματος εντοπισμού λέξεων, που συνήθως αποτελούνται από λίστες οπτικά παρόμοιων εμφανίσεων, σε συνεκτικές προτάσεις που αποτυπώνουν το περιεχόμενο και τα συμφραζόμενα του χειρόγραφου κειμένου.

Οι μέθοδοι KWS έχουν γνωρίσει ραγδαία εξέλιξη, ιδιαίτερα μετά την υιοθέτηση βαθιών συνελικτικών και ακολουθιακών αρχιτεκτονικών όπως τα CNN και τα Seq2Seq μοντέλα. Παρόλα αυτά, τα αποτελέσματά τους παραμένουν περιορισμένα σε μεμονωμένες λέξεις ή μικρές φράσεις, αγνοώντας τη σύνταξη και τη συνοχή του κειμένου. Επιπλέον, τα ιστορικά χειρόγραφα χαρακτηρίζονται από υψηλή ποικιλομορφία γραφής, θόρυβο και φθορά λόγω παλαιότητας, γεγονός που επιδεινώνει την απόδοση των συστημάτων πλήρους οπτικής αναγνώρισης (*Handwritten Text Recognition - HTR*). Ως εκ τούτου, απαιτούνται πιο προηγμένες μέθοδοι που συνδυάζουν οπτικά και γλωσσικά χαρακτηριστικά.

Η προτεινόμενη μεθοδολογία βασίζεται σε μια πολυεπίπεδη υπολογιστική δομή (pipeline) που περιλαμβάνει τέσσερα κύρια στάδια: (α) οπτική ανάκτηση λέξεων μέσω ενός μοντέλου Seq2Seq, (β) επιλεκτική μεταγραφή (*Selective HTR*) μόνο στο τοπικό γειτονικό πλαίσιο της ανιχνευμένης λέξης, (γ) δυναμική παραθυροποίηση (*Dynamic Windowing*) με χρήση Μεγάλων Γλωσσικών Μοντέλων (LLMs), και (δ) τελική διόρθωση και σύνθεση πρότασης.

Στο πρώτο στάδιο, το μοντέλο Seq2Seq εκτελεί αναζήτηση τύπου *Query-by-Example (QbE)*, παράγοντας ταξινομημένες λίστες από λέξεις που είναι οπτικά παρόμοιες με το ερώτημα. Στη συνέχεια, το σύστημα μεταγράφει μόνο τις λέξεις που βρίσκονται μέσα σε ένα καθορισμένο παράθυρο γύρω από το αποτέλεσμα, χρησιμοποιώντας τα μοντέλα Seq2Seq και TrOCR. Η επιλεκτική αυτή μεταγραφή μειώνει δραστικά το υπολογιστικό κόστος σε σχέση με την πλήρη μεταγραφή σελίδων.

Το τρίτο στάδιο, και το πιο καινοτόμο, αφορά στη δυναμική εκτίμηση του μεγέθους του παραθύρου με χρήση LLMs. Εξετάζονται δύο στρατηγικές: (α) η μέθοδος *few-shot prompting*, όπου το μοντέλο προβλέπει τον αριθμό των λέξεων αριστερά και δεξιά βασιζόμενο μόνο στη λέξη-στόχο, και (β) η μέθοδος *segment-based prompting*, όπου παρέχεται στο LLM ένα μικρό αποσπασματικό συμφραζόμενο για ακριβέστερη εκτίμηση των ορίων. Χρησιμοποιούνται προεκπαιδευμένα και προσαρμοσμένα μέσω *Low-Rank Adaptation (LoRA)* μοντέλα LLaMA 3.2-3B και Mistral 7B, τα οποία βελτιστοποιούνται για την ανακατασκευή προτάσεων από χειρόγραφα δεδομένα.

Στο τελικό στάδιο, το σύστημα συνθέτει και διορθώνει τις μεταγραμμένες προτάσεις, εξαλείφοντας σφάλματα και εξασφαλίζοντας συντακτική και νοηματική συνοχή. Το αποτέλεσμα είναι η παραγωγή υποψήφιων προτάσεων (*sentence hypotheses*) που αποτυπώνουν με ακρίβεια τα συμφραζόμενα του ερωτήματος, προσφέροντας ένα ενδιάμεσο βήμα προς σημασιολογική αναζήτηση ή αυτόματη περίληψη.

Η αξιολόγηση πραγματοποιήθηκε στο σύνολο δεδομένων IAM Handwriting Database, το οποίο περιλαμβάνει περισσότερες από 100.000 εικόνες λέξεων. Δοκιμάστηκαν τρεις στρατηγικές παραθυροποίησης, σταθερή, *few-shot* και *segment-based*, καθώς και δύο μοντέλα μεταγραφής, Seq2Seq και TrOCR. Τα αποτελέσματα έδειξαν ότι η δυναμική παραθυροποίηση υπερτερεί σαφώς της σταθερής. Η μέθοδος *segment-based* επιτυγχάνει BLEU = 77.7% και BERTScore = 85.0% στις μεταγραφές του Seq2Seq, αποδεικνύοντας ότι το σύστημα μπορεί να παράγει συντακτικά ορθές και νοηματικά συνεκτικές προτάσεις. Παρά το αυξημένο υπολογιστικό κόστος, η συγκεκριμένη προσέγγιση προσφέρει ανώτερη ακρίβεια και πληρότητα συμφραζομένων.

Η εργασία αποδεικνύει ότι ο συνδυασμός οπτικής ανάκτησης και γλωσσικής κατανόησης μπορεί να επεκτείνει δραστικά τις δυνατότητες των συστημάτων KWS. Με τη χρήση Μεγάλων Γλωσσικών Μοντέλων, η αναζήτηση μπορεί να περάσει από το επίπεδο της λέξης στο επίπεδο της πρότασης, προσφέροντας πλουσιότερη και περισσότερο σημασιολογική πρόσβαση σε ιστορικά δεδομένα. Το προτεινόμενο σύστημα μειώνει τον υπολογιστικό φόρτο μέσω επιλεκτικής μεταγραφής, αυξάνει την ακρίβεια ανακατασκευής προτάσεων και θέτει τη βάση για μελλοντική αυτόματη περίληψη εγγράφων.

Μελλοντικές επεκτάσεις περιλαμβάνουν την εφαρμογή της μεθόδου σε πολυγλωσσικά χειρόγραφα σύνολα δεδομένων, την ενοποίηση με *layout analysis* για αναγνώριση άρθρων ή παραγράφων, καθώς και την ανάπτυξη μηχανισμών αυτόματης περίληψης ή εννοιολογικής αναζήτησης βασισμένων σε πολυτροπικές αναπαραστάσεις (όπως CLIP και BLIP). Συνολικά, η διατριβή εισάγει ένα πλήρως λειτουργικό και επεκτάσιμο σύστημα ανάκτησης προτάσεων από χειρόγραφα έγγραφα, το οποίο αξιοποιεί τη δύναμη των Μεγάλων Γλωσσικών Μοντέλων για τη σύνθεση και κατανόηση φυσικής γλώσσας. Η συνεισφορά της είναι διττή: τεχνικά, αποδεικνύει τη βιωσιμότητα της μετάβασης από το KWS στο *sentence-level retrieval*, και επιστημονικά, ανοίγει τον δρόμο για νέα εργαλεία στην ψηφιακή ανθρωπιστική έρευνα, την τεκμηρίωση πολιτιστικής κληρονομιάς και τη σημασιολογική αναζήτηση σε ιστορικά αρχεία.

Chapter 1

Thesis Introduction

Writing has become an integral part of human civilization. Since ancient times, it has served as a means for people to externalize their thoughts and record them on various objects and materials. With the discovery and widespread use of paper, writing evolved into a tool for documenting the elements of daily life. It was used to record historical events and holy texts, to preserve customs and traditions, and to support the rapid growth of literature and education. Writing also enabled the documentation and exchange of scientific ideas and, perhaps most importantly, allowed people to communicate and share knowledge across distances and generations.

Throughout history, every culture has developed its own handwritten forms of expression, producing an immense body of written material. Unfortunately, a significant portion of this heritage has been lost. The preservation of manuscripts depended on laborious manual copying and the periodic replacement of damaged or worn texts. Wars, natural disasters, and the fragility of early storage materials further contributed to the disappearance of countless works and collections.

As human technology has advanced, new tools have emerged to preserve manuscripts through digitization, primarily using high-resolution scanning techniques. However, the need to improve both the digitization and analysis of such texts continues to grow. Despite the involvement of domain experts, manual transcription and analysis of historical manuscripts remain time-consuming and labor-intensive tasks. Therefore, the development of automated techniques for analyzing and extracting information from digitized collections is becoming increasingly essential. Such methods can significantly reduce the resources required to manage the vast and ever-expanding volume of digital archives, while enabling deeper access to the knowledge they contain.

1.1 Introduction to information retrieval in handwritten documents

The digitization of handwritten documents has driven the research community to develop methods capable of retrieving information directly from collections of document images. Two main paradigms have emerged to address this problem: recognition-free retrieval and recognition-based retrieval.

Recognition-free retrieval, commonly referred to as *Keyword Spotting* (KWS) or *Word Spotting* (WS), focuses on locating all occurrences of a query within a collection of handwritten documents without requiring full transcription. KWS methods can be further categorized according to how the retrieval process is implemented. One major distinction is between *segmentation-free* and *segmentation-based* approaches. In segmentation-free methods, word detection is performed directly on full, unsegmented document pages. In contrast, segmentation-based methods operate on word or line images that have been extracted during a preprocessing stage. Segmentation itself can take place at the word level, where each page is divided into individual words, or at the line level, where each page is divided into lines.

KWS methods may also differ based on how the user specifies the query. In the *Query-by-Example* (QbE) scenario, the user provides an image of the target word, and the system searches for visually similar word images within the collection. In the *Query-by-String* (QbS) scenario, the user provides a text string as input, and the system must align textual and visual representations to identify relevant matches.

A final distinction concerns the use of training data. *Learning-based* methods rely on an offline training stage, during which the system learns visual or textual features from annotated examples. In contrast, *learning-free* methods do not require labeled data and instead use handcrafted features or direct similarity measures to perform retrieval.

Recognition-based retrieval follows a different philosophy. Instead of comparing visual representations extracted from document images, it first converts the images into machine-readable text and then performs retrieval using *word recognition* techniques. In this paradigm, *Handwritten Text Recognition* (HTR) is used for handwritten documents, while *Optical Character Recognition* (OCR) is used for printed material. The output of these methods is a transcription, typically represented in ASCII or another textual format. Once transcriptions are obtained, the system can build a dictionary or index of all recognized words, and retrieval is conducted over this textual representation.

Most recognition-based systems depend on supervised learning and therefore require large amounts of annotated data for training. These annotations may be provided at the word level, line level, or even at the character level, depending on the model design. While this approach enables the use of powerful *Natural Language Processing* (NLP) techniques and supports richer forms of retrieval, it also suffers from important limitations. Handwritten documents, especially historical ones, exhibit significant variation in writing style. In addition, physical degradation introduces noise and distortions. These factors often lead to transcription errors, which in turn negatively affect retrieval accuracy. As a result, the overall performance of recognition-based systems is strongly tied to the quality and robustness of the underlying HTR or OCR model.

The material and terminology presented in this section are primarily adapted from the seminal survey by Giotis *et al.* [1]. For an in-depth understanding of the topic, please refer to their original work.

1.2 Motivation and Objectives

Despite significant progress in research on Keyword Spotting (KWS), current systems seem to be reaching a ceiling. Conventional KWS methods rely mainly on visual features, which are only effective for matching a query to common or similar images across a collection of documents. However, the results obtained usually consist of single word-level hits, offering little or no context for how these words are used in the text. Understanding the sentence in which a word appears is of great importance to people dealing with historical manuscripts. Retrieving entire sentences from manuscripts remains a difficult problem.

Since historical manuscripts exhibit variability in writing style and significant degradation of text features over time. This affects the Optical Character Recognition (OCR) and Handwriting Text Recognition (HTR) processes because errors are introduced during transcription, such as incorrect character recognition and incorrect punctuation. Furthermore, the process of transcribing entire collections is time-consuming, especially when only partial information needs to be extracted from the texts. These factors make it difficult to retrieve a sentence containing the detected word.

To address this gap, this thesis proposes a system that bridges the understanding of KWS with sentence-level understanding. The goal is to go beyond simple word detection and enable the retrieval of coherent sentences using the visual feature information of the detected word. You achieve this by integrating visual retrieval with selective transcription with language modeling techniques capable of inferring how many neighboring words, to the left and right of the detected word, should be included to create a complete sentence.

1.3 Problem Definition

Given a Query-by-Example (QbS), where the input is an image of a handwritten word, and the ranked list of retrieved instances returned by a KWS system, the goal is to predict a minimal and linguistically plausible sentence for each of the top-k retrieved instances in the list.

Each retrieved instance corresponds to a specific position within a sequence of handwritten word images. Therefore, the task is to determine the appropriate number of neighboring positions that will be included to the left and right of the detected word in order to reconstruct a coherent and complete sentence. Achieving this requires combining information from HTR systems through selective transcription, together with Large-Language Models (LLMs) that infer sentence boundaries and correct transcription errors introduced during text recognition.

By solving this problem, the proposed approach bridges image-based retrieval with language-based reasoning, moving beyond traditional word detection toward sentence-level comprehension and retrieval in handwritten historical sources. This formulation enables the extraction of meaningful textual information from KWS outputs and represents a step toward more context-aware analysis of digitized manuscript collects.

1.4 Thesis outline

The structure of this thesis is organized as follows.

- Chapter 2 provides an overview of Keyword Spotting (KWS), presenting the fundamental concepts, taxonomy, and recent developments in segmentation-based and segmentation-free approaches.
- Chapter 3 introduces the theoretical background and formulates the sentence retrieval problem, describing the Sequence-to-Sequence (Seq2Seq) baseline and the principles of Large Language Models (LLMs).
- Chapter 4 presents the proposed system pipeline, detailing each component—from visual retrieval, selective transcription, and dynamic windowing to LLM-based correction—along with algorithmic complexity considerations.
- Chapter 5 describes the experimental setup, datasets, evaluation metrics, and ablation studies, as well as the implementation details used to reproduce the results.
- Chapter 6 reports and discusses the quantitative results, analyzes performance–cost trade-offs, and highlights key findings.
- Finally, Chapter 7 concludes the thesis with a summary of contributions and suggestions for future research directions.

Chapter 2

An overview of Word Spotting.

2.1 Keyword Spotting System Architecture

In this section, the stages of a typical Keyword Spotting (KWS) pipeline are described. As illustrated in Figure 2.1, a KWS system operates in two phases: an offline phase and an online phase.

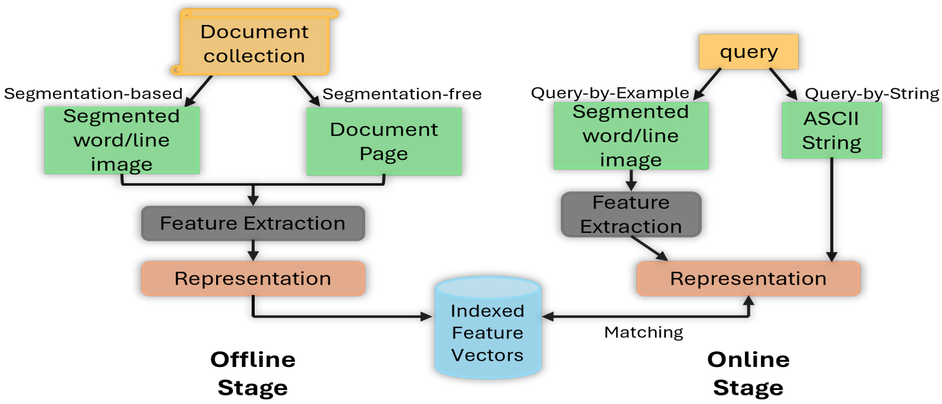


Figure 2.1: The architecture of a general KWS system.

In the offline phase, the system processes the entire collection of document images. Depending on the preprocessing method was applied, the collection may consist of segmented word images, text lines, or even full page images. Each image is transformed into a feature vector that represents its visual content. Feature extraction methods can be divided into learning-free and learning-based approaches. Learning-free methods do not require annotated data and rely on handcrafted features such as Histogram of Oriented Gradients (HoG) or Scale-Invariant Feature Transform (SIFT). In contrast, learning-based methods employ supervised training and typically use deep neural network architectures such as Convolutional Neural Networks (CNNs) to learn discriminative representations. These models produce fixed-length feature vectors that facilitate efficient comparison during retrieval.

The online phase begins when the user submits a query, and the nature of this query determines the retrieval modality. In the Query-by-Example (QbE) setting, the input is a word image selected from the document collection, which means the query must already exist within the dataset. In the Query-by-String (QbS) setting, the user provides an arbitrary text string, allowing greater flexibility since the query does not need to appear in the collection. Regardless of the query type, the system converts the input into a feature representation that lies in the same embedding space as the representations generated in the offline phase. This query vector is then compared with all stored representations, and similarity scores are computed. Finally, the results are sorted in descending order of similarity, producing a ranked list in which the most relevant word images appear at the top.

2.2 Related Works

With the rise of deep learning, neural networks have become the dominant paradigm in keyword spotting. Early CNN-based systems either produced fixed-length embeddings for word images or extracted descriptors from intermediate network activations. These representations were typically compared using Euclidean or cosine distance. However, alternative dissimilarity measures like Bray-Curtis in Sudholt *et al.* [2] were also shown to be effective for pyramid-based representations.

Some approaches framed KWS as a pairwise similarity learning problem. For instance, Zhong *et al.* [3] trained a network on positive and negative word-image pairs and learned to predict similarity scores directly, treating KWS as both classification and regression. Because neural methods require large training sets, data augmentation (affine transformations) and pretraining followed by fine-tuning on target datasets became common strategies. In some cases, such as the work of Sfikas et al. [4], only pretraining was used, and activations of intermediate layers were combined to capture more abstract representations.

A major line of research explored attribute-based embeddings to support both QbE and QbS. Sudholt *et al.* [5] extended their PHOCNet by comparing multiple descriptors such as Pyramidal Histogram of Characters (PHOC), Discrete Cosine Transform of Words (DCToW), and Spatial Pyramid of Characters (SPOC) a multinomial version of PHOC. They further improved KWS by introducing Temporal Pyramid Pooling (TPP), a modification of their former Spatial Pyramid Pooling (SPP) layer [2], to accept input images of variable size. Different loss functions, such as cosine loss, and binary cross-entropy were evaluated for training, and cosine distance was used for final ranking.

To reduce annotation requirements, weakly supervised and synthetic-data-driven KWS methods were introduced. Gurjar *et al.* [6] pre-trained PHOCNet CNN on the synthetic HW-SYNTH / IIIT-HWS dataset [7, 8], then fine-tuned on target datasets, achieving near state-of-the-art performance while using up to 98% fewer real annotations. Al-Rawi *et al.* [9] extended PHOC to a multi-script representation using a ResNet-152 [10] backbone, enabling script-independent word spotting across English, French, German, Arabic, and Bangla via a unified multi-PHOC vector. Wolf *et al.* [11, 12] went further by proposing a completely annotation-free KWS pipeline where CNNs predicted font and slant to generate adapted synthetic data, and pseudo-labels were created and refined via a TPP-PHOCNet model.

Other works focused on cross-modal embedding spaces. Gomez *et al.* [13] first learned a string embedding correlated with Levenshtein distance and then trained an image embedding model [14] to regress into that space, achieving superior performance compared to PHOC and DCToW. Retsinas *et al.* [15] proposed deep descriptors derived from max-pooled convolutional zones to capture unigram/bigram information, while Krishnan *et al.* [16] utilize an extended version of the HWNet system [8], named HWNet v2 [17], which is based on a multi-scale ResNet-34 architectures and joint image and text embedding to support end-to-end word spotting and recognition.

Building upon these developments, Daraee *et al.* [18] enhanced CNN-based keyword spotting by incorporating Monte-Carlo dropout to estimate prediction uncertainty. Multiple stochastic forward passes yield certainty scores, which are used with adaptive thresholds for both query-by-example and query-by-string. This uncertainty-aware design improved retrieval accuracy on several handwritten datasets. Majumder *et al.* [19] proposed a recognition-free method using dynamic time warping on logarithmic word profiles, where multi-view fragments and a voting scheme improved matching without deep models.

To improve feature robustness under limited supervision, Giotis *et al.* [20] introduced an adversarial framework with spatial transformer networks to adapt deep features in weakly supervised keyword spotting. Their Feature Map Adversarial Deformation module deforms intermediate feature maps to generate harder examples, improving robustness to handwriting variation. Using PHOC-based embeddings and minimal fine-tuning, the model achieved performance comparable to supervised state-of-the-art methods. Krishnan *et al.* [21] extended this direction with HWNet v3, an end-to-end label embedding framework that jointly embeds images and text using synthetic data to improve both spotting and recognition.

The reliance on annotated data was further challenged by Wolf *et al.* [22] who addressed the annotation bottleneck by proposing a fully self-training pipeline for handwritten text recognition and keyword spotting. Models are pretrained on synthetic data and iteratively refined using pseudo-labels on real data, with confidence thresholds removing noisy samples. This annotation-free method outperformed traditional learning-free and semi-supervised approaches. Matos *et al.* [23] applied these advancements in a full transcription pipeline for medieval Portuguese manuscripts, combining layout analysis, segmentation, and recognition to reduce manual effort while maintaining high accuracy.

One of the most significant recent developments in HTR is Transformer OCR [24], a transformer-based architecture that combines a Vision Transformer encoder with a Transformer decoder pre-trained on large-scale text corpora and fine-tuned on handwriting datasets such as IAM. TrOCR achieves state-of-the-art character error rates, clearly outperforming traditional models. Importantly, TrOCR is a segmentation-based model that operates directly on isolated word or line images and produces highly accurate transcriptions even in challenging or historical settings.

Finally, a major advancement in segmentation-based keyword spotting was introduced by Retsinas *et al.* [25], who proposed a unified architecture combining a CTC branch for transcription and a Seq2Seq decoder for learning more expressive visual–textual representations. During training, the decoder forces the encoder to align visual features with linguistic structure, while at inference time, only the encoder is used to generate compact word embeddings. This design bridges recognition and retrieval within a single model, allowing it to support both QbE and QbS. To further optimize retrieval, the authors applied binarization with straight-through estimators, producing efficient and discriminative descriptors. This Seq2Seq model achieved state-of-the-art results on multi-writer datasets such as IAM and GW, outperforming PHOCNet, HWNet v2, and other leading methods. Since it unifies recognition and spotting in a single framework and operates directly on segmented word images, Seq2Seq has become the standard baseline in modern segmentation-based KWS.

In this thesis, Seq2Seq is adopted as the primary keyword spotting backbone, forming the visual retrieval foundation upon which sentence-level reconstruction is built.

Chapter 3

Theoretical background and problem formulation

3.1 Baseline Model

Retsinas *et al.* [25] proposed a Sequence-to-Sequence (Seq2Seq) recognition model in which handwritten images as well as text strings are represented as fixed-length vectors. First, they enable efficient KWS. Second, they provide a latent space from which a transcription can be extracted from a given input image.

The proposed architecture consists of five components. A convolutional backbone, a bidirectional Gated Recurrent Unit (GRU) encoder, a bidirectional GRU character encoder, a unidirectional GRU sequence decoder, and an auxiliary Connectionist Temporal Classification (CTC) branch. CTC is only employed during training to accelerate convergence. During inference the model activates specific branches according to the input type.

If the given input is a Query-by-Example (QbE), the image is initially processed by the convolutional backbone, which extracts a sequence of features that captures the spatial patterns. Then, the sequence is compressed by the encoder into a fixed-length vector that contains the visual content of the entire word. This vector can then be used in two ways. The first approach enables retrieval through discriminative descriptors, which can be directly compared with the embeddings of other word images to identify identical or similar words. The other approach determines the transcription using the decoder, which generates the corresponding text character by character.

On the other hand, if the given input is a Query-by-String (QbS), the aim is to retrieve occurrences of that word from a collection of handwritten documents. This can be achieved by using the character encoder to project the string into the embedding space used to represent word images. The resulting vector is then compared with the image vectors to retrieve the word images with the highest similarity. Figure 3.1 illustrates the Seq2Seq model used for transcription generation, as well as its operation in keyword spotting for both QbE and QbS.

Within the scope of this thesis, the Seq2Seq model serves as a necessary component of the proposed information retrieval system, as it extracts text from images of handwritten documents. Its outputs provide information in vector form and also enable keyword spotting.

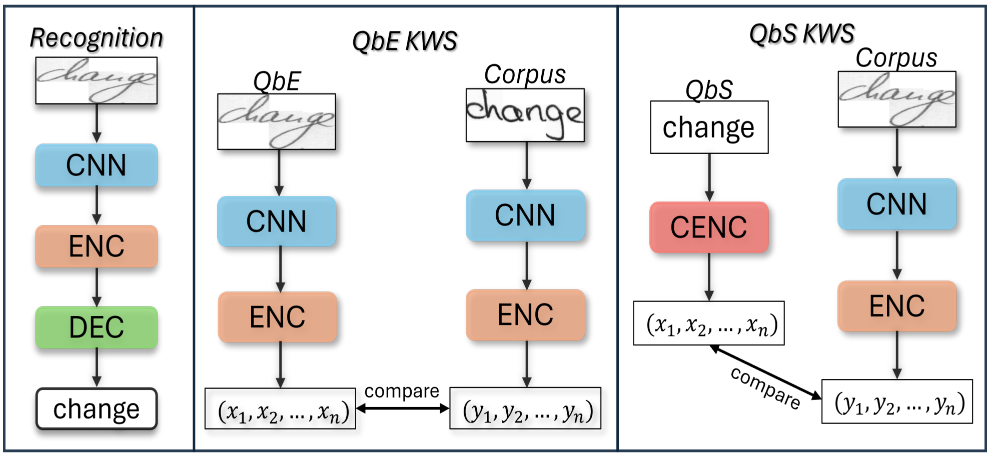


Figure 3.1: Sequence-to-Sequence model.

3.2 Large Language Model

3.2.1 Auto-regressive

Auto-regressive language modeling [26, 27] is the foundation for many LLMs, such as LLaMA [28, 29, 30] and Mistral [31], which learn to forecast the next token in a dataset, based on previous tokens. Typically, given a sequence of tokens $x_1, x_2, \ldots, x_T$, its probability is factored as follows:

$$P(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} P(x_t | x_{<t}) \ , \tag{3.1}$$

where $x_{<t}$ indicates the prefix of tokens prior to position t. The model is trained by minimizing the negative log-likelihood of the observed data:

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} logP_\theta(x_t | x_{<t}) \ , \tag{3.2}$$

where θ represents the model's parameters. This auto-regressive formulation allows models to produce coherent sequences one token at a time.

3.2.2 Context Window

An important point regarding auto-regressive models is the *context window* [32], or as it is otherwise referred to in the literature, *context length*. The context window is responsible for the amount of information that the model can use to predict the next token. For instance, if the context window of a model is equal to 4096, then, according to equation 3.1, the previous 4096 tokens are used regardless of the text size.

In addition, another point about the context length is that the larger this number, the more computationally expensive the model becomes. Due to the complexity of the self-attention mechanism used by transformer models, which scales as $O(n^2)$, the requirements for both memory and computational power increase. Nevertheless, techniques such as the Group-Query Attention (GQA) [33] and the Sliding Window Attention mechanism [34, 35], used in LLaMA and Mistral, respectively, reduce the computational complexity. Therefore, the size of the context length can be increased, allowing the model to receive more information to predict the next token.

3.2.3 Tokenization

Before employing an LLM, the unprocessed text must be converted to a numerical representation through tokenization. SentencePiece [36] is an unsupervised tokenizer that does not require pre-discriminating or language-specific heuristics. It is utilized in both LLaMA and Mistral to split sentences or words into tokens.

The algorithm used by SentencePiece to tokenize words is Byte-Pair Encoding (BPE) [37]. This algorithm iteratively combines the most frequent pairs of characters from a given text until a dictionary of fixed size is achieved. Hence, both frequent morphemes and rare words are recorded. Finally, each token produced is mapped to an embedding vector via an embedding matrix $X \in \mathbb{R}^{n \times d} \subseteq \mathbb{R}^{V \times d}$, where V is the vocabulary size, d is the dimension of the model, and n the length of the sequence. Thus, for each token x_i, the corresponding embedding is given by:

$$x_i = X[x_i] \in \mathbb{R}^d \ . \tag{3.3}$$

To preserve information about word order, adding Positional Encodings [38] to these embeddings is considered essential prior to passing them through the transformer architecture.

3.2.4 Transformer Architecture

Vaswani *et al.* [39] invented the transformer architecture, which serves as the foundation for almost all current LLMs. The self-attention mechanism, referenced to in the literature as "Scaled Dot-Product Attention", is the major innovation in this architecture. This strategy helps the model assess the importance of all tokens in a sequence when representing a specific position.

For a sequence of input representations $X \in \mathbb{R}^{n \times d}$, Q, K and V are computed as:

$$Q = XW^Q,\ K = XW^K,\ V = XW^V \ , \tag{3.4}$$

where W^Q, W^K, $W^V \in \mathbb{R}^{d \times d}$ and d is the dimension of the model, which enables the learning process to be stabilized. Subsequently, the attention output is calculated:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right) \ . \tag{3.5}$$

Multi-Head Attention (MHA) extends the previous idea by projecting the input sequence into multiple subspaces. In this way, attention can be calculated in parallel. Therefore, Equation (3.5) is transformed as follows:

$$MHA(X) = Concat(H_1, H_2, \ldots, H_h)W^O \ , \tag{3.6}$$

with

$$H_i = Attention(XW_i^Q, XW_i^K, XW_i^V) \ , \tag{3.7}$$

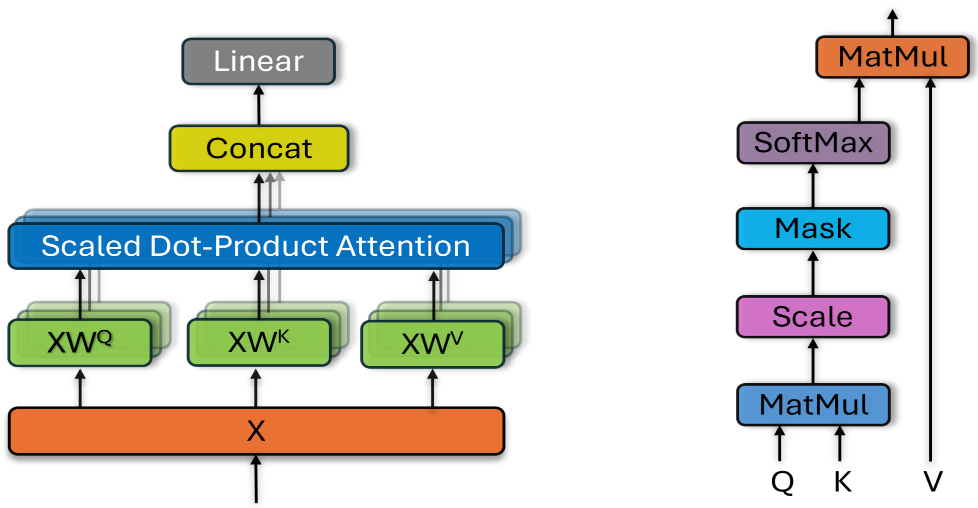


Figure 3.2: Self-Attention Mechanism (left) and Scaled Dot-Product Attention (right)

where W_i^Q and $W_i^K \in \mathbb{R}^{d \times d_k}$, $W_i^V \in \mathbb{R}^{d \times d_v}$, $W^O \in \mathbb{R}^{h * d_v \times d}$ and $d_k = d_v = \frac{d}{h}$, h the number of parallel attention layers.

The original transformer was designed as an encoder-decoder architecture. Nevertheless, the auto-regressive models, such as LLaMA and Mistral, adopt a decoder-only architecture. The principal variation is that causal masking attention is used. Thus, Equation (3.5) is modified as follows:

$$Attention(Q, K, V) = softmax\Big(\frac{QK^T}{\sqrt{d_k}} + M\Big) , \tag{3.8}$$

where M is a mask matrix assigning $-\infty$ to attention scores from future tokens, ensuring that the model cannot peek ahead during prediction. Figure 3.2 provides an overview of the attention mechanism of a decoder-only model.

Although the original transformer architecture is very robust, LLaMA [29, 30] introduced two main contributions that improve the model's efficiency. The first enhancement is the use of Rotary Positional Embeddings (RoPE) [40] instead of the sinusoidal encodings employed in the original architecture. By applying RoPE, the model is able to better process long sequences, including those longer than sequences observed during training.

The second innovation used by LLaMA, aiming for efficiency and long-term reasoning, is GQA [33]. In a typical MHA, the attention weights are calculated with respect to all keys and values for each element of a sequence, making the process computationally demanding. In contrast, GQA organizes the queries into groups and calculates the attention using these groups. This way, the efficiency is improved because attention calculations are limited to within the groups. Improving the efficiency of GQA makes it more scalable to larger sequences and datasets.

Finally, another attention technique, used by the Mistral model [31], is Sliding Window Attention (SWA) [34, 35]. Contrary to the preceding approach, which uses information from all tokens in the sequence at once, this procedure benefits the model by concentrating on a local window of tokens at a time. SWA reduces the quadratic attention complexity to linear with respect to the sliding window size, allowing for more efficient handling of large sequences. Additionally, the sliding window mechanism tolerates a fixed attention span, which reduces the cache size required for inference without compromising the model's quality.

3.3 Problem Formulation

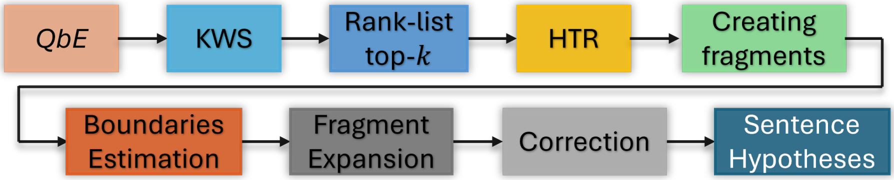
The primary objective of this thesis is to develop a comprehensive system capable of retrieving knowledge from handwritten documents, such as historical texts. The information to be retrieved is represented as sentences based on the visual information provided to the system.

A QbE, which involves an image of a handwritten word, is given to the system as input. From this query image, visual features are extracted and compared against the corpus to identify *hits*, which are positions in a sequence of word images where the query word occurs. Each hit corresponds to a single position within the word image sequence. Therefore, additional context is required to recover a complete sentence. This is achieved by retrieving the left and right neighboring positions around each hit. Determining how many neighbors to include on each side is the central prediction problem addressed in this work.

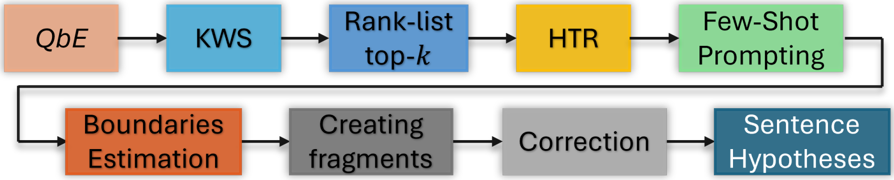
More specifically, for each hit, two values, L_n and R_n, must be estimated. Here, L_n is the number of tokens before the query word and R_n is the number of tokens after it. These values define the boundaries of the sentence hypothesis. The total predicted sentence length is $L = L_n + R_n + 1$, where the additional $+1$ accounts for the query word itself. To solve this problem, two methods are proposed to estimate L_n, R_n for each hit. The methods are described in detail in the following chapter.

Finally, by utilizing the resulting knowledge in conjunction with LLMs, the system retrieves coherent sentence-level information across the entire document collection.

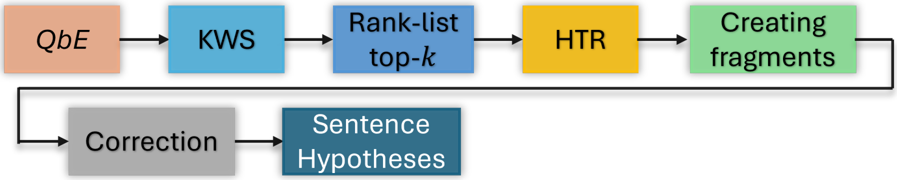
Figure 3.3 illustrates the proposed system end-to-end.



(a) Dynamic windowing segment-based.



(b) Dynamic windowing using few-shot prompting.



(c) Fixed size window.

Figure 3.3: Proposed system pipeline integrating KWS, OCR/HTR, (a) dynamic windowing segment-based, (b) dynamic windowing few-shot prompting, (c) fixed size window, and LLM-based correction for sentence reconstruction from handwritten documents.

Chapter 4

Proposed System

4.1 Pipeline Overview

The sentence retrieval system proposed in this thesis is structured as a pipeline that incrementally transforms the results of KWS into sentence-level retrieval instances and then into coherent sentence hypotheses. This system combines image-based search, selective image transcription from a collection of historical or handwritten documents, and inference using LLMs. Each stage addresses a distinct challenge: locating relevant words in image form, handling OCR/HTR transcription errors, and resolving variable sentence boundaries. Figure 4.1 shows a summary of the workflow.

The process begins with KWS. The Seq2Seq model compares a QbE image of a handwritten word to the document collection, producing a ranked list of matches, and only the top-k most similar hits are retained. Each hit corresponds to a position within the word images sequence. To extract text information, the system leverages OCR/HTR models to selectively transcribe around the hits, rather than transcribing the entire collection. This selective approach reduces computational costs while still preserving the amount of knowledge required for sentence reconstruction.

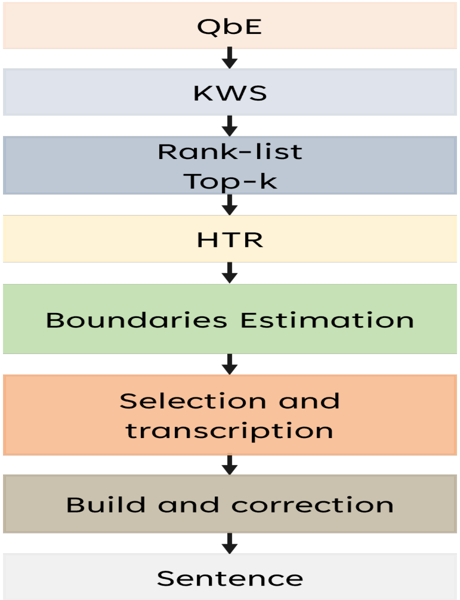


Figure 4.1: Overview for the proposed system.

The next stage estimates the number of neighboring tokens to include on the left (L_n) and right (R_n) of each hit to reconstruct the sentence. Two approaches have been implemented in this procedure, both relying on a pre-trained LLM. The first strategy only requires the transcription of the hit. The second strategy creates a short fragment that includes the neighboring word images of the hit. Both approaches provide boundary predictions that guide the selection of word images for transcription and assembly into sentence-level text.

After the sentence has been generated, two additional options are presented. In the first option, employment of a fine-tuned LLM is included, with a larger context window, allowing for more accurate boundary estimation, leading to an extended initial sentence. In the second option, the initial LLM-derived sentence is preserved without further modification. In either case, the resulting sentence is finally refined by applying the fine-tuned LLM in a corrective mode. This ensures that transcription artifacts are minimized and that the sentence hypothesis is both syntactically coherent and semantically correct.

This multi-stage pipeline provides robust retrieval of sentence-level information from handwritten and historical sources, which can be noisy, by converting isolated KWS hits into complete, well-formed sentence hypotheses.

In summary, the input–output flow between the core components of the system is summarized below. The KWS module receives a query image and produces a ranked list of visually similar word instances within the document collection. Each detected instance, together with its local neighborhood window, forms the input to the Selective HTR module, which outputs transcribed word sequences. These sequences are then passed to the LLM-based reasoning module, which predicts sentence boundaries and refines the textual output into coherent sentence hypotheses.

4.2 Handwritten Text Recognition Normalization

A crucial aspect of the pipeline is the transition from the visual to the textual domain. Two models are employed for text recognition: TrOCR [24] and a Seq2Seq architecture [25]. Both models are well-suited for the current task, as they have demonstrated outstanding results in handwritten image transcription. However, when converting from one space to another, transcription errors are unavoidable, especially in handwritten and historical documents.

A common error is the insertion of inappropriate punctuation within words, which causes fragmented or incorrect tokens. The inconsistent usage of lowercase and capital letters is another issue. To handle these issues, a technique called normalization is employed, which lowercases all characters and removes unwanted punctuation from transcribed words. This ensures that tokens are handled uniformly in subsequent pipeline stages.

A further difficulty arises from character-level misrecognitions, in which visually similar symbols are confused. To correct such cases, an additional post-processing step is implemented. Using the character encoder of the Seq2Seq model, the most similar correctly spelled word is retrieved from a collection of valid words. This correction mechanism effectively reduces noise by replacing incorrectly recognized forms with their closest valid equivalents.

Through this combination of text recognition models and noise reduction procedures, the system ensures that the OCR/HTR results are as accurate as possible. This provides a strong textual basis for LLM-based boundary prediction.

4.3 LLM-Based Dynamic Windowing

The central challenge of the sentence retrieval process is to dynamically determine how many neighboring tokens must be included to the left and to the right of a detected hit in order to reconstruct the sentence. This task is addressed with the help of an LLM, which predicts the boundary values based on the input transcription. Two prompting strategies are employed: *few-shot prompting* with the hit only, and *segment-based prompting* with limited context.

In the first strategy, a pre-trained LLM is provided only with the transcription of the hit word. The prompt is constructed in a few-shot format, where several annotated examples of hits are presented together with their corresponding L_n and R_n values. These examples guide the LLM boundary prediction task for the current hit.

In the second strategy, a fine-tuned LLM (detailed description in Section 5.4) is given a short segment that contains the transcription of the hit together with up to two neighboring transcriptions on each side. The model is prompted to generate m candidate sentences that incorporate this segment. From the generated sentences, the values of L_n and R_n are calculated by counting the number of tokens occurring to the left and right of the hit. When multiple candidates are produced, the final boundary values are obtained by averaging across the predictions.

In both strategies, the predicted values of L_n and R_n are used to select the corresponding neighboring word images from the document sequence. These images are then transcribed and corrected, providing the textual material from which the sentence is assembled.

4.4 Algorithm and Complexity

The proposed sentence retrieval pipeline can be described by the pseudocode in Algorithm 4.1.

The overall process follows a sequential flow where the output of each stage serves as the input to the next. Specifically, the ranked hits produced by the KWS module are passed to the HTR normalization step for selective transcription, whose results provide the textual input for the LLM-based dynamic windowing and final correction modules. This explicit linkage ensures a coherent and reproducible pipeline structure.

Algorithm 4.1 Pseudocode of the proposed end to end system.

Input: Query-by-Example q, document collection D

Output: Final sentence hypotheses H

1: hits = KWS(q, D) //top-k matching positions
2: **for** hit in hits **do**
3: $hit_{trans} \leftarrow get\_transcription(hit)$
4: $hit_{trans} \leftarrow correction(hit)$
5: **if** $strategy = `few\_shot'$ **then**
6: $(L_n,\ R_n) \leftarrow LLM\_FewShot(hit_{trans})$
7: **else if** $strategy = `create'$ **then**
8: $neighbor\_images \leftarrow find\_neighbors(hit, N)$ //hit + N images
9: $images_{trans} \leftarrow correction(neighbor\_images)$
10: $fragment \leftarrow create\_frag(images_{trans})$
11: $candidates \leftarrow LLM\_Create(fragment, m)$
12: $(L_n,\ R_n) \leftarrow average\_boundaries(candidates, hit_{trans})$
13: **end if**
14: $extended\_images \leftarrow (hit, L_n, R_n)$
15: $extended\_images_{trans} \leftarrow get\_transcription(extended\_images)$
16: $extended\_images_{trans} \leftarrow correction(extended\_images_{trans})$
17: $sentence \leftarrow create\_sentence(extended\_images_{trans})$
18: $sentence \leftarrow LLM\_Correct(sentence)$
19: $H \leftarrow H \cup sentence$
20: **end for**

4.4.1 Runtime Characteristics

As far as runtime is concerned, the algorithm depends mainly on three factors. First, KWS (step 1) is dependent on the complexity of the feature extraction. For the Seq2Seq KWS model, the execution time is proportional to the number of word images in the collection. Second, HTR (steps 3 and 15) processes only the hits and their local neighborhoods rather than transcribing the entire document collection. If n is the average neighborhood length, this reduces the cost of transcription from $O(|D|)$ to $O(k \cdot n)$, where k is the top-k matching positions. Third, LLM inference (steps 6, 11 and 18) depends on the model size and the length of the prompt. The few-shot strategy uses shorter prompts, while the segment-based strategy requires additional context and multiple sentence generations (m candidates). These steps dominate the runtime when using large-scale models.

Overall, the runtime can be expressed as $O(KWS + k \cdot L \cdot HTR + k \cdot LLM)$, where KWS scales with the document collection size, L is the total neighborhood size determined by L_n and R_n and LLM reflects the cost of inference under the chosen prompting strategy. The selective design ensures that the cost increases primarily with the number of hits rather than the total length of the collection, making the entire process feasible for large collections of handwritten documents.

Chapter 5

Experimental Setup & Protocols

5.1 Datasets

For the purposes of this work, two datasets were used: the IAM Handwriting Database[1][2] (IAM) and a subset of the English Wikipedia corpus[3]. IAM serves as a standard reference for comparing performance in the field of handwritten documents, while Wikipedia provides clean textual data for fine-tuning language models.

5.1.1 IAM Handwriting Database

The IAM contains 1539 pages of modern, calligraphic, handwritten English text produced by 657 authors. The pages are segmented in three different ways: word-level, line-level and sentence-level, with their corresponding annotations. The heterogeneity caused by the multi-writer setup contributes significantly to the difficulty of the

[1]Available online: `https://fki.tic.heia-fr.ch/databases/iam-handwriting-database`
[2]IAM = Institut für Informatik und Angewandte Mathematik, University of Bern, Bern, Switzerlan
[3]Available online: `https://huggingface.co/datasets/wikimedia/wikipedia`

dataset. The official partition of the database was used, as is common practice in literature [41, 5].

In this thesis, the IAM dataset has a crucial role in multiple stages of the process. First, it provides the training data for the KWS and OCR/HTR models, which are required to detect similar word-images as well as to transcribe visual features in to text. Second, its transcriptions are used to adapt and improve the language models used for sentence correction. Finally, the IAM serves as the main benchmark for evaluating keyword spotting performance, word-image transcription accuracy, and the quality of reconstructed sentence hypotheses.

An overview of the IAM dataset statistics is provided in Table 5.1.

Table 5.1: IAM dataset statistics

Dataset Characteristic	Count	
# images	115320	
# punct. images	14719	
# num. images	454	
# words	100147	
# unique words	12105	
# training images	53841	
# testing images	17616	
# sentences	5677	
# unique sentences	4926	
	Mean	**Std**
sent. length	17.26	11.7

5.1.2 Wikipedia Corpus

In addition to the handwritten data, a large corpus was required to support the language modeling components of the pipeline. For this purpose, a subset of the English Wikipedia dataset was used. From the full collection, the first 1000 articles were extracted to provide a sufficiently large but computationally manageable training set.

The raw texts were pre-processed. As part of the pre-processing steps, all letters were converted to lowercase, non-standard symbols were removed, as well as unnecessary punctuation. The final dataset comprises well-constructed English sentences that cover a wide variety of subjects.

The Wikipedia subset is primarily used to improve language models by providing fluent and coherent sentences. This helps the models increase their ability to transform noisy or incomplete transcriptions into sentences that are syntactically coherent and semantically meaningful.

An overview of the statistics of the English Wikipedia subset is provided in Table 5.2.

Table 5.2: English Wikipedia subset statistics

Dataset Characteristic	**Count**	
# words	38340	
# sentences	41407	
	Mean	**Std**
sent. length	13.544	3.65

5.1.3 Fragment per Sentence Dataset

In addition to the datasets described in Sections 5.1.1 and 5.1.2, two supplementary datasets were created specifically for fine-tuning the LLMs (see Section 5.4). These derived datasets comprise sentence fragments that maintain local syntactic and semantic coherence around a target word.

The construction of these datasets relied on the perplexity metric, a standard measure in information theory that reflects the uncertainty in a sample value from a unknown probability distribution. In language modeling, perplexity expresses the model's prediction uncertainty; the lower the value, the more confident the model is in predicting the next word. Formally, it is defined as:

$$PPL(X) = exp\Big(- \frac{1}{n} \sum_{i=1}^{n} log\big(p(x_i|x_{<i})\big)\Big) \ ,$$

where $p(x_i|x_{<i})$ is the predicted probability of the i^{th} token in sentence $X = (x_1, x_2, \ldots, x_n)$, and n is the total number of tokens.

The dataset creation procedure is as follows. For each unique word in the original corpus, one sentence containing that word was selected at random manner. Around this target word, multiple fragments are generated by varying the number of tokens included the left and right. Each candidate fragment is then scored using perplexity, and the fragment with the lowest score is chosen, since it corresponds to the most fluent and coherent local context. Repeating this procedure for all unique words produced two new collections, each organized in a fragment-per-sentence (FPS) format. Table 5.3 summarizes the statistical information of the newly constructed datasets.

Table 5.3: Fragment per Sentence datasets statistics

	Dataset	Size		Mean	Std
train set	IAM	19205	fragment length	16.54	8.77
test set		4802	fragment length	16.52	8.78
train set	Wikipedia	30657	fragment length	9.24	3.07
test set		7665	fragment length	9.21	3.03

5.2 Evaluation Protocols

5.2.1 Evaluation Strategies

At this point, reference will be made to the scenarios used for evaluating the proposed sentence retrieval system. To evaluate the system's performance, two main experimental scenarios were designed. The key difference between them lies in how the system decides how many neighboring words to include around each retrieved hit when reconstructing a sentence.

In the first scenario, a fixed-size window is used for every hit. For a given value of m, the system retrieves m words to the left and m words to the right of the detected term, creating a sentence segment with a total length of $2m + 1$. This setup offers a simple and consistent way to see how the amount of surrounding context influences both the accuracy and the fluency of the reconstructed sentences.

In the second scenario, the window size is not fixed but is instead adjusted dynamically for each hit. In this case, the system uses pre-trained and fine-tuned LLMs to estimate the most appropriate number of words to include on each side L_n on the left and R_n on the right. These predicted values are then used to select the most relevant neighboring words, allowing the system to build a sentence that fits the local context more naturally.

By comparing these two ways of selecting context (fixed versus dynamic) using the evaluation metrics described in Subsection 5.2.2, it becomes possible to understand which approach is better suited for reconstructing complete and coherent sentences from handwritten documents.

5.2.2 Evaluation Measurements

Evaluating the performance of a KWS system as well as an LLM, requires a combination of quantitative measurements that reflect both the accuracy of information retrieval and semantic adequacy. For KWS, the problem is formulated as detecting the appearance or absence of a keyword given a query [1].

For a given query, *Precision* is defined as the fraction of retrieved instances that are relevant to that query:

$$Precision = \frac{|\{relevant\ instances\} \cap \{retrieved\ instances\}|}{|\{retrieved\ instances\}|} .$$

Recall is the fraction of relevant words that are successfully retrieved:

$$Recall = \frac{|\{relevant\ instances\} \cap \{retrieved\ instances\}|}{|\{relevant\ instances\}|} .$$

The *F1-score* is calculated as the harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} .$$

Precision at rank k (P@k) is defined as the precision calculated exclusively from top-k retrieved results, indicating the dependability of the system's highest-ranked outputs:

$$P@k = \frac{|\{relevant\ instances\} \cap \{k\ retrieved\ instances\}|}{|\{k\ retrieved\ instances\}|} .$$

For a given query *Average Precision* (AP) is defined as:

$$AP = \frac{1}{|\{relevant\ instances\}|} \sum_{k=1}^{n} \big(P@k \times rel(k)\big) ,$$

where $rel(k)$ is an indicator function equal to 1 if the word at rank k is relevant and 0 otherwise. The mean of the *Average Precision* across all queries in a keyword spotting task constitutes the *Mean Average Precision* (mAP).

While all the aforementioned metrics apply solely to the KWS task, this work also leverages LLMs, which can generate varied sentences. Therefore, it is necessary to define appropriate metrics to compare the generated sentences to the reference sentences, considering both textual and semantic similarity.

The BLEU score [42] measures n-grams precision, adjusted with a brevity penalty:

$$BLEU = BP \cdot exp\Big(\sum_{n=1}^{N} w_n log(p_n)\Big)$$

$$\text{where } \ BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \ . \end{cases}$$

Here, c is the number of words in the predicted sentence and r is the number of the reference sentence. According to Papineni *et al.* [42], $N = 4$ and the weigths are uniform $w_n = \frac{1}{N}$.

The *ROUGE-N* [43] emphasizes recall of n-grams and is defined as:

$$ROUGE-N = \frac{\sum\limits_{ref\,n-grams} Count_{match}}{\sum\limits_{ref\,n-grams} Count_{ref}} \ ,$$

where $Count_{match}$ is the number of matching words between the generated sentence and the reference sentence, and $Count_{ref}$ is the total number of words in the reference sentence. *METEOR* [44] extends these methods by incorporating stemming, synonymy and paraphrase matching.

Finally, more recent works demonstrate that embedding-based metrics can capture semantic similarity beyond word overlap. *BERTScore* [45] leverages contextual embeddings to align generated and reference tokens. It is defined as:

$$R_{BERT} = \frac{1}{|x|}\sum_{x_i \in x} \underset{\hat{x}_j \in \hat{x}}{max}(x_i^T \hat{x}_j), \ P_{BERT} = \frac{1}{|\hat{x}|}\sum_{\hat{x}_j \in \hat{x}} \underset{x_i \in x}{max}(x_i^T \hat{x}_j), \ F_{BERT} = 2 \cdot \frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}} \ ,$$

where x and $\hat{x}$ are the vector representation in the embedding space of reference and candidate sentences, respectively. At sentence level, semantic similarity can be computed using the cosine similarity of sentence embeddings:

$$sim(x, \hat{x}) = \frac{x \cdot \hat{x}}{||x||\,||\hat{x}||} \ ,$$

which provides a scalar similarity score between $[-1, 1]$. For metrics that use vectors from the embedding space, various encoder-only LLMs can be employed, such as BERT [46, 47] and RoBERTa [48]. Using different LLMs can result in varying similarity values because each model generates distinct embedding spaces.

5.3 Ablation Techniques

The estimation of neighboring words L_n and R_n, plays a vital role in the proposed framework. Two additional exploratory models were developed to investigate alternative strategies for their prediction. These ablation models were designed to assess whether encoder-only language models could effectively approximate the boundary estimation behavior of the larger generative LLMs described in Section 4.3. Although these models were not ultimately integrated into the final system due to limitations observed during experimentation, they provide valuable insights into the challenges of this stage in the proposed framework.

Before presenting the models, it is necessary to describe the dataset used for their training. The dataset was derived from the FPS-Wikipedia corpus (see Section 5.1.3), augmented with two supervision signals: for each fragment–sentence pair, the required numbers of neighboring words L_n and R_n were computed, representing the number of tokens that must be added to the left and right sides of the fragment to reconstruct the original sentence.

As noted above, both ablation models share a common encoder backbone, the BERT-base[4] [46] model which serves to encode the textual input into contextualized embeddings. BERT-base consists of 12 bidirectional transformer layers, each comprising 12 self-attention heads, and a hidden size of 768, totaling approximately 110 million parameters. The encoder captures progressively richer linguistic abstractions across its depth: the lower layers encode surface and syntactic patterns, the middle layers model contextual relationships, and the later layers represent semantic and task-specific information [49].

Given the relatively small size of the training dataset and the large number of parameters in the base encoder, a parameter-freezing strategy was adopted to reduce over-fitting and computational cost. Specifically, the first 8 layers of the encoder were frozen, while the remaining 4 layers were left trainable, enabling fine-tuning of high-level semantic features for the boundary estimation task. This configuration retained the general linguistic knowledge of the pre-trained encoder while focusing the adaptation on task-relevant representations.

The two ablation variants that employ this configuration are described in the following subsections.

[4] Available online: `https://huggingface.co/google-bert/bert-base-uncased`

5.3.1 Variational Autoencoder and Multilayer Perceptron Predictor

The first ablation model integrates three main components: the encoder-only LLM described earlier, a Variational Autoencoder (VAE) [50] for dimensionality reduction, and a Multilayer Perceptron (MLP) classifier for boundary estimation. The model's objective is to predict the appropriate values of L_n and R_n based on the encoded textual representation.

The VAE module consists of an encoder–decoder pair designed to compress the pooler output of the LLM into a lower-dimensional latent space. The encoder comprises two fully connected layers with dimensions $(768, 256)$ and $(256, 64)$, each followed by a ReLU activation. The resulting latent vector z captures the most important semantic features of the input while reducing redundancy. The decoder mirrors this structure with two linear layers of dimensions $(64, 256)$ and $(256, 768)$, using the same activation function to reconstruct the original embedding space of the LLM's pooled representations.

The latent vector z is then passed to an MLP responsible for predicting the discrete boundary values. The MLP contains two linear layers of dimensions $(64, 32)$ and $(32, n)$, where n denotes the number of output classes corresponding to possible L_n and R_n values. ReLU activation function is applied between layers, and a dropout layer with a probability 0.2 is used for regularization. Finally, softmax activation produces a probability distribution over all candidate boundary values, from which the most likely L_n and R_n are selected. Figure 5.1 illustrates the architecture of the model.

5.3.2 Multilayer Perceptron Predictor

The second model differs from the previous one in that it relies solely on MLP architecture. Its input is the final hidden state of the BERT model. Specifically, the output vector produced by the last bidirectional transformer layer. This vector has dimensions $(m, hidden\_size)$, where m denotes the length of the token sequence.

Since the MLP requires fixed-size input, pooling techniques were applied to transform the vector dimensions from $(m, hidden\_size)$ to $(1, k)$, where k depends on the pooling strategy used. Three pooling approaches were evaluated: max pooling ($k = hidden\_size$), mean pooling ($k = hidden\_size$), and a combined mean–max pooling strategy ($k = 2 * hidden\_size$). These techniques enable the aggregation of

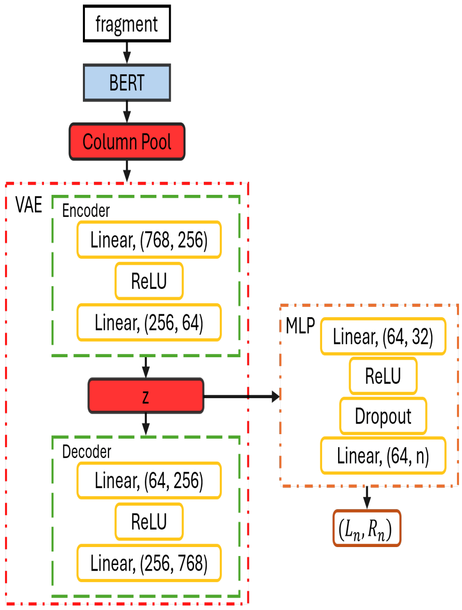


Figure 5.1: Architecture of the VAE-MLP model, which combines BERT embeddings with a VAE for dimensionality reduction and an MLP classifier for predicting L_n and R_n.

contextual token-level information into a single representative vector suitable for classification.

The pooled vector is then passed through an MLP consisting of three linear layers with dimensions $(k, 256)$, $(256, 64)$, and $(64, n)$, where n represents the number of output classes. The network employs ReLU activations, two normalization layers, and a dropout layer with a probability of 0.3 to improve generalization. As in the first model, the final output vector is passed through a softmax function to estimate the most probable values of L_n and R_n. Figure 5.2 illustrates the overall architecture of this model.

5.3.3 Training Procedure

At this stage, the training process of the two previously described models is presented. To ensure a fair comparison, both models were trained under identical experimental conditions. As mentioned earlier in this section, the dataset used was the FPS-Wikipedia corpus, extended with the corresponding L_n and R_n values for each fragment–sentence pair.

During training, the AdamW optimizer was employed with an initial learning rate of 2×10^{-4}. Furthermore, several learning rate schedulers, such as LinearLR,

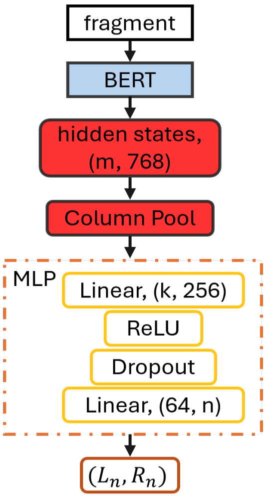


Figure 5.2: Architecture of the MLP model, which applies pooling to the final hidden states of BERT and uses an MLP to predict L_n and R_n.

ExponentialLR, and CosineAnnealingLR, were tested to gradually reduce the learning rate throughout training. Each model was trained for 200 epochs, with evaluation performed after every epoch. The loss function used was weighted cross-entropy, compensating for the class imbalance observed in the dataset.

Despite extensive experimentation, both models exhibited significant overfitting, as shown in Figure 5.3. Various configurations and regularization attempts failed to mitigate this issue. The main conclusion drawn from these results is that a larger and more diverse dataset is required, along with enhanced feature extraction mechanisms capable of capturing richer contextual information from each input fragment.

Figure 5.3: Training and evaluation loss curves for both models (VAE-MLP and MLP). The plots illustrate the overfitting behaviour observed during training.

5.4 Implementation Details

This section describes the implementation details of the proposed system, covering the training of the KWS and HTR models, as well as the adaptation of LLMs for dynamic boundary prediction and sentence correction. Where relevant, deviations from standard practice are highlighted to ensure reproducibility and to provide context for the observed differences with respect to previous work.

The Seq2Seq[5] architecture employed for both KWS and HTR follows the open-source implementation proposed by Retsinas *et al.* [25]. The official IAM split was used for training, as described in Section 5.1.1 and shown in Table 5.1. The model was trained for 160 epochs and evaluated every 5 epochs using thee official test split and the mAP metric. In each evaluation, the Character Error Rate (CER) was also computed. The training configuration followed the optimal settings reported by Retrinas *et al.* [25], which were found to yield the highest mAP performance. Table 5.4 presents the reproduced and reported mAP and CER scores.

Table 5.4: Comparison of the reproduced and reported mAP and CER scores.

Model	**QbE mAP**	**QbS mAP**	**CER**
Seq2Seq (Reproduced)	92.35	96.09	5.4
Seq2Seq (Reported)	91.62	95.5	5.1

As discussed in Section 4.2, the second architecture used for the HTR process is TrOCR. Specifically, the TrOCR-large model[6] was employed, which had already been fine-tuned on the IAM dataset. No additional fine-tuning was performed in this work, as Li *et al.* [24] report that the model achieves a CER of 2.89.

To further reduce the CER where necessary, a post-processing mechanism for correcting misspelled words was implemented. A dictionary containing all unique words from the IAM test set without access to their corresponding images, was constructed and encoded using the character encoder from the Seq2Seq architecture. For each transcribed word generated by either Seq2Seq or TrOCR, its encoded representation was compared with those in the dictionary, and the most similar entry in embedding space was selected as the corrected word. Table 5.5 reports the CER and Word Error Rate (WER) values before and after correction observed on the official IAM test set.

[5] The Seq2Seq source code, is available at: `https://github.com/georgeretsi/Seq2Emb`

[6] Model card: `https://huggingface.co/microsoft/trocr-large-handwritten`

Table 5.5: Comparison of CER and WER on the official IAM test set before and after applying the word-level correction mechanism.

	Before		After	
Model	**CER**	**WER**	**CER**	**WER**
Seq2Seq	5.32	14.8	6.0	11.73
TrOCR	19.9	30.7	22.5	27.3

The estimation of neighboring words L_n and R_n, as described in Section 4.3, was performed using two distinct prompting strategies. The first option, a few-shot prompting approach, used pre-trained LLaMA-3.2-3B[7] and Mistral-7B[8] models. For each top-k transcription retrieved from the KWS stage, a prompt was constructed that included example triples (word, L_n, R_n) drawn heuristically from the Wikipedia corpus (Table 5.7). The prompt was then provided to the LLM (see Figure 5.5a), and the predicted boundary values L_n and R_n were extracted from its response.

The sentence approach, the segment-based strategy, was also implemented using LLaMA-3.2-3B and Mistral-7B, but these models were fine-tuned to improve boundary prediction and sentence correction. Fine-tuning was conducted in two stages using the FPS datasets (see Section 5.1.3). In the first stage, the FPS-Wikipedia corpus was used for 3 epochs; in the second stage, the FPS-IAM dataset was used for 6 epochs. The AdamW optimizer and a linear scheduler were employed, with an initial learning rate of 2×10^{-4}. To reduce the computational cost of fine-tuning, Low-Rank Adaptation (LoRA) [51], a Parameter Efficient Fine-Tuning (PEFT) technique [52, 53], was adopted, with configuration parameters $r = 16$, $\alpha = 32$, and $dropout = 0.05$. Figure 5.4 illustrates the training and evaluation loss observed during the fine-tuning process for both stages and models.

During inference, after a segment of up to 5 words has been created from the neighborhood of the hit word from the collection, ensuring that the hit word itself is included, the corresponding prompt is prepared in the format presented in Figure 5.5b. This prompt is provided as input to the fine-tuned model, following the procedure described in Section 4.3 to determine the new segment boundaries. The retrieved fragment is subsequently passed to the correction model using the prompt format shown in Figure 5.5c. the output of this step constitutes the sentence hypothe-

[7]Model card: `https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct`

[8]Model card: `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3`

ses from the collection.

All components described above were integrated into a unified sentence retrieval pipeline. Each module KWS, HTR, boundary estimation, and sentence correction was implemented as an independent stage, enabling modular experimentation and consistent evaluation. The final system configuration, incorporating the fine-tuning LLMs and post-correction mechanisms, was used in all subsequent experiments and analyses presented in Chapter 6.

To ensure transparency and reproducibility, the main training and fine-tuning hyperparameters used across all experiments are summarized in Table 5.6. For the Seq2Seq and HTR components, the configuration follows the optimal settings reported by Retsinas *et al.* [25], while minor adjustments were applied for convergence stability. Regarding the LLM components, the reported LoRA parameters correspond to the best-performing setup identified during preliminary experiments. These configurations were kept fixed throughout all subsequent evaluations to ensure consistency across models and datasets.

Table 5.6: Main hyperparameters used for model training and fine-tuning.

Parameter	**Seq2Seq / HTR**	**LLMs (LoRA Fine-tuning)**
Optimizer	Adam	AdamW
Learning rate	1×10^{-4}	2×10^{-5}
Batch size	16	8
Epochs	160	3 + 6
LoRA rank (r)	–	16
LoRA α	–	32
Dropout	0.25	0.05
Precision	fp16	fp16

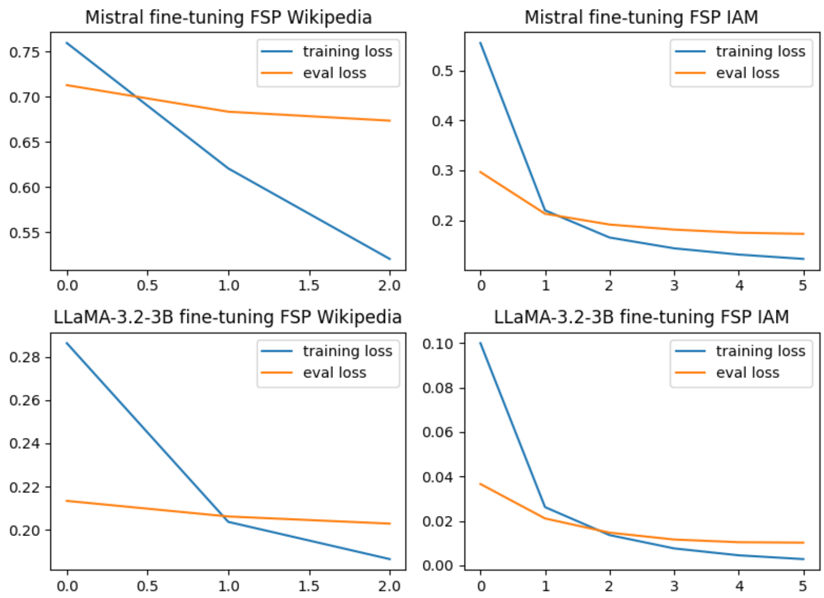


Figure 5.4: Training and evaluation losses from Mistral-7B and LLaMA-3.2-3B LLM during fine-tuning.

Table 5.7: Examples for the few-shot strategy

	Number of Neighbors	
word	L_n	R_n
regeneration	7	5
belongs	3	3
center	6	4
entertain	3	6
appears	4	9
sun	3	5
write	5	5

Figure 5.5: Prompt templates employed in the proposed system. The figure presents the three prompting configurations: (a) few-shot prompting, (b) creation prompting, and (c) correction prompting

(a) Few-Shot Prompting Template

Word: word

Number of Neighbors:

left: L_n, right: R_n

(b) Sentence creation prompt template

You are a helpful assistant.

Write a grammatically correct and natural sentence that includes the phrase 'fragment'. The sentence must be between 4 and 12 words.

(c) Sentence correction prompt template

You are a grammar correction assistant

Input: 'sentence'

Output only one corrected sentence, no extra text no explanations:

Chapter 6

Experimental Results

In this chapter, the numerical results of the proposed sentence retrieval system are presented and analyzed. The goal is to determine how effective the system can be in reconstructing complete and coherent sentences from handwritten texts, given a single word image as input. For this purpose, as mentioned in Section 5.2.1, three strategies are compared. The first one is a fixed and symmetric context window, while the other two are dynamic context window techniques, in which the number of left and right neighbors is estimated using LLMs.

In addition to accuracy, the evaluation also takes into account the practical aspects of the system. Specifically, the trade-off between performance, latency, and the number of LLM calls is examined. Finally, common failure cases and limitations of the system are discussed in order to provide a realistic understanding of the challenges involved in sentence retrieval from handwritten texts.

6.1 Numerical Results

When using the Seq2Seq model, which gives more accurate transcriptions (see Table 5.5), more accurate results are shown. The dynamic windowing method, and more specifically the segment-based strategy, achieves the highest scores in all metrics. As shown in Table 6.1, the segment-based strategy consistently achieved higher BLEU and BERTScore values than both the few-shot and fixed window approaches, confirming the benefit of contextual prompting and adaptive boundary estimation. For example, with LLaMA, its BLEU metric score is 77.7% (Table 6.1) compared to 73.7% (Table 6.2) for the few-shot prompting approach and only 40.5% (Table 6.3) for the fixed window approach. This roughly corresponds to a 5% BLEU improvement and a 3% BERTScore gain over the few-shot method, indicating a tangible increase in contextual coherence and sentence completeness. A similar pattern is observed in semantic similarity, where the BERTScore reached 85.0%, while for fixed windowing it reached 58.2%. The 27% difference in BERTScore suggests that the sentences generated by the segment-based method preserve substantially more meaning from the reference sentences. These differences show that dynamic windowing methods combined with the Seq2Seq model not only give more accurate results in token retrieval but also produce sentences that are much closer semantically to the ground-truth sentences.

This advantage is consistent across all three QbE cases and both LLMs. Although Mistral showed slightly lower scores than LLaMA, the relative ranking between the strategies remains the same: segment-based > few-shot prompting > fixed-size windowing. As summarized in Tables 6.1 - 6.3 these trends demonstrate that adaptively estimating the context window yields consistently higher textual and semantic similarity that fixed method. This confirms that dynamically adjusting the context length provides more robust and reliable sentence reconstruction than using a fixed-size window.

When the TrOCR model was used to transcribe the word-images, the overall scores decreased due to the high noise generated during transcription. However, the ranking of the strategies remains unchanged. For example, in Table 6.4, the segment-based strategy using LLaMA achieved a BLEU score of 48.6%, while in Table 6.5, the few-shot prompting strategy reached 38.7%, and in Table 6.6, the fixed window lagged again at 25.3%. Even in more difficult cases, the segment-based strategy achieved better results than the other methods. This shows that this strategy is not only more accurate but also more robust to transcription errors.

During the experiments in the fixed window scenario, a single value of m was applied to every retrieved instance in the ranked list produced by KWS. Although

Table 6.1: Quantitative results for the segment-based strategy using LLaMA and Mistral. Token-level similarity and semantic-level similarity indicate the quality of reconstructed sentences generated from the Seq2Seq transcriptions.

Token-level Similarity				
#	model	**BLEU±SD**	**ROUGE-N±SD**	**METEOR±SD**
1	LLaMA	77.7 ± 27.0	80.4 ± 23.9	81.8 ± 21.5
2		74.8 ± 20.2	75.9 ± 15.7	78.6 ± 14.8
3		76.4 ± 13.6	75.4 ± 21.4	78.5 ± 17.6
1	Mistral	68.7 ± 30.0	77.0 ± 27.8	77.4 ± 24.5
2		50.5 ± 29.2	57.7 ± 21.5	61.0 ± 22.0
3		59.4 ± 24.8	61.5 ± 27.5	65.2 ± 23.8
Semantic-level Similarity				
#	model	**BERT$_{\text{score}}$**	**Sen. BERT** **BERT$_{\text{large}}$ ± SD**	
1	LLaMA	85.0 ± 11.0	0.856 ± 0.083	
2		81.9 ± 5.2	0.892 ± 0.061	
3		77.1 ± 4.5	0.841 ± 0.06	
1	Mistral	79.7 ± 12.7	0.824 ± 0.073	
2		73.0 ± 9.4	0.84 ± 0.093	
3		69.3 ± 8.2	0.779 ± 0.075	

this makes the approach simple and easy to implement, it also makes it essentially inflexible. In practice, sentence lengths vary considerably, and a single window size cannot accommodate this variability. As a result, the selected window was often too small, leading to incomplete or incorrect sentence reconstructions. In such cases, the LLM used for sentence completion did not receive enough context to produce a correct or meaningful sentence that aligns with the ground truth. Consequently, the fixed window method is unable to adapt to the structure of each sentence, which explains its consistently lower performance.

In contrast, the dynamic windowing approach performs consistently better than the fixed window strategy, confirming that sentence boundaries should be estimated adaptively rather than predetermined. Within the dynamic methods, the segment-based strategy achieves the highest performance. The presence of even a small local

Table 6.2: Quantitative results for the few-shot prompting strategy using LLaMA and Mistral. Token-level similarity and semantic-level similarity indicate the quality of reconstructed sentences generated from the Seq2Seq transcriptions.

Token-level Similarity				
#	model	**BLEU±SD**	**ROUGE-N±SD**	**METEOR±SD**
1		73.7 ± 32.3	77.5 ± 27.5	78.9 ± 25.0
2	LLaMA	69.0 ± 12.4	70.3 ± 8.7	73.6 ± 9.1
3		47.8 ± 22.6	53.7 ± 25.9	57.4 ± 21.6
1		56.7 ± 44.5	68.0 ± 39.2	67.0 ± 37.1
2	Mistral	40.0 ± 19.8	49.1 ± 12.8	53.1 ± 13.7
3		51.3 ± 30.1	54.3 ± 32.8	57.2 ± 30.8
Semantic-level Similarity				
			Sen. BERT	
#	model	**BERT$_{\text{score}}$**	**BERT$_{\text{large}} \pm$ SD**	
1		82.9 ± 14.3	0.846 ± 0.09	
2	LLaMA	79.0 ± 4.7	0.868 ± 0.063	
3		65.3 ± 8.8	0.79 ± 0.048	
1		75.8 ± 17.1	0.807 ± 0.091	
2	Mistral	68.5 ± 5.9	0.83 ± 0.08	
3		65.1 ± 14.4	0.769 ± 0.089	

context (up to two neighboring words on each side of the hit) allows the LLM to better capture sentence boundaries and syntactic dependencies, leading to $4-6\%$ improvement in BLEU and 3% in BERTScore. This indicates that even a minimal context helps the LLM to better understand the sentence structure and produce more accurate predictions for L_n and R_n.

In comparison, the few-shot prompting strategy relies solely on the target word, without access to its surrounding context. As a result, it often struggles to infer the correct boundaries, especially when the word can appear in different syntactic roles. Furthermore, this strategy is inherently biased by the examples included in the prompt (see Table 5.5), since the model's predictions depend not only on the target word but also on the prior patterns shown in the examples. This limitation further explains why the segment-based method provides more reliable and contextually

Table 6.3: Quantitative results for the fixed symmetric window configuration using LLaMA and Mistral. Token-level similarity and semantic-level similarity indicate the quality of reconstructed sentences generated from the Seq2Seq transcriptions.

Token-level Similarity				
#	model	**BLEU±SD**	**ROUGE-N±SD**	**METEOR±SD**
1		40.5 ± 27.0	43.9 ± 34.3	45.7 ± 30.5
2	LLaMA	37.3 ± 22.8	28.7 ± 18.9	33.6 ± 17.6
3		33.2 ± 18.6	30.0 ± 21.3	32.5 ± 20.8
1		35.7 ± 23.2	42.7 ± 33.8	45.5 ± 32.6
2	Mistral	17.1 ± 14.3	26.6 ± 14.5	30.3 ± 15.5
3		26.5 ± 21.7	29.5 ± 21.7	34.3 ± 22.2

Semantic-level Similarity			
			Sen. BERT
#	model	$\mathbf{BERT_{score}}$	$\mathbf{BERT_{large} \pm SD}$
1		58.2 ± 14.7	0.731 ± 0.125
2	LLaMA	54.2 ± 8.9	0.753 ± 0.067
3		56.2 ± 10.1	0.765 ± 0.077
1		60.3 ± 18.2	0.731 ± 0.125
2	Mistral	54.2 ± 8.9	0.659 ± 0.131
3		54.2 ± 9.4	0.662 ± 0.042

appropriate boundary estimates.

Overall, the results show that sentence retrieval depends significantly on adapting the context length to each specific result. The fixed window strategy is inflexible to handle the natural variability of sentences, leading to incomplete results. Dynamic windowing addresses this limitation by predicting the context window in each case, resulting in more accurate, complete, and coherent sentences. Among dynamic windowing methods, the segment-based strategy achieves the best performance because it exploits minimal but capable local context. These findings highlight the importance of context adaptability in sentence reconstruction from handwritten texts.

Table 6.4: Quantitative results for the segment-based strategy using LLaMA and Mistral. Token-level similarity and semantic-level similarity indicate the quality of reconstructed sentences generated from the TrOCR transcriptions.

Token-level Similarity				
#	model	**BLEU±SD**	**ROUGE-N±SD**	**METEOR±SD**
1		48.6 ± 6.3	49.5 ± 18.7	49.6 ± 13.8
2	LLaMA	48.3 ± 26.6	51.6 ± 17.3	53.3 ± 18.5
3		33.3 ± 18.4	35.8 ± 23.2	33.4 ± 14.3
1		35.4 ± 16.5	39.2 ± 20.5	39.6 ± 18.7
2	Mistral	31.2 ± 20.4	35.0 ± 19.3	36.0 ± 19.1
3		37.4 ± 19.2	39.0 ± 22.1	39.8 ± 17.6

Semantic-level Similarity			
#	model	$\mathbf{BERT_{score}}$	**Sen. BERT** $\mathbf{BERT_{large} \pm SD}$
1		48.5 ± 5.9	0.70 ± 0.04
2	LLaMA	50.6 ± 11.4	0.663 ± 0.10
3		43.7 ± 5.0	0.626 ± 0.038
1		45.7 ± 7.6	0.684 ± 0.042
2	Mistral	51.2 ± 10.3	0.668 ± 0.09
3		47.1 ± 8.6	0.661 ± 0.034

6.2 Cost-Performance

In addition to accuracy, it is important to consider the computational cost of each strategy. The fixed-window approach is the simplest and most computationally efficient, as it does not require prediction for each retrieved instance. As a result, it has the lowest latency and minimal LLM usage. However, this efficiency affects the quality of the final sentences.

The few-shot prompting strategy introduces a moderate increase in computational cost, as one LLM call is required to estimate L_n and R_n for each hit. The cost remains manageable, as the LLM is called once for the estimation and once for the correction of the final sentence. This improves accuracy but adds a small latency compared to the fixed-window strategy.

Consequently, the segment-based method offers the highest accuracy, but increases

Table 6.5: Quantitative results for the few-shot prompting strategy using LLaMA and Mistral. Token-level similarity and semantic-level similarity indicate the quality of reconstructed sentences generated from the TrOCR transcriptions.

Token-level Similarity				
#	**model**	**BLEU±SD**	**ROUGE-N±SD**	**METEOR±SD**
1		38.7 ± 7.2	46.2 ± 22.5	45.6 ± 17.3
2	LLaMA	38.6 ± 25.6	46.2 ± 22.6	44.4 ± 21.0
3		37.6 ± 10.4	49.1 ± 26.4	47.1 ± 19.1
1		42.4 ± 21.7	51.8 ± 29.5	49.1 ± 22.3
2	Mistral	30.2 ± 16.3	32.6 ± 20.9	34.3 ± 18.8
3		25.1 ± 16.7	33.2 ± 24.1	33.5 ± 18.2

Semantic-level Similarity			
#	**model**	**$\mathbf{BERT_{score}}$**	**Sen. BERT $\mathbf{BERT_{large}} \pm$ SD**
1		47.2 ± 5.6	0.706 ± 0.034
2	LLaMA	52.2 ± 10.9	0.681 ± 0.137
3		45.4 ± 5.7	0.65 ± 0.052
1		50.4 ± 6.0	0.66 ± 0.056
2	Mistral	47.7 ± 7.4	0.611 ± 0.069
3		46.3 ± 7.7	0.658 ± 0.033

computational cost. Since it requires the transcription of an initial segment, the LLM calls to assess the bounds, which depend on the number of sentences to be generated to estimate L_n and R_n, performing additional transcriptions based on the estimate, and finally employing the LLM again to correct the sentences. This results in the longest latency and more LLM calls. On average, the total processing time for the segment-based strategy increased by approximately 10% compared to the few-shot approach, primarily due to additional LLM inference steps. However, this overhead yields a $5 - 6\%$ BLEU improvement and around 3% gain in BERTScore, reflecting a favorable accuracy-cost trade-off.

Nevertheless, the significant improvement in completeness, coherence, and semantic accuracy makes it the most efficient method overall. In practical terms, the additional computational cost corresponds to roughly 1 minute per processed hit, which

Table 6.6: Quantitative results for the fixed symmetric window configuration using LLaMA and Mistral. Token-level similarity and semantic-level similarity indicate the quality of reconstructed sentences generated from the TrOCR transcriptions.

Token-level Similarity				
#	model	**BLEU±SD**	**ROUGE-N±SD**	**METEOR±SD**
1	LLaMA	25.3 ± 23.2	31.2 ± 30.3	32.5 ± 28.7
2	LLaMA	12.4 ± 10.7	13.5 ± 7.4	16.7 ± 10.1
3	LLaMA	6.8 ± 4.6	13.5 ± 5.9	16.6 ± 7.1
1	Mistral	25.3 ± 23.2	31.3 ± 29.3	32.5 ± 28.7
2	Mistral	19.2 ± 14.7	13.5 ± 7.8	16.7 ± 10.17
3	Mistral	6.8 ± 4.6	13.5 ± 5.9	16.6 ± 7.1

Semantic-level Similarity			
#	model	**BERT$_{\text{score}}$**	**Sen. BERT BERT$_{\text{large}}$ ± SD**
1	LLaMA	44.4 ± 10.6	0.58 ± 0.08
2	LLaMA	41.0 ± 3.4	0.565 ± 0.094
3	LLaMA	42.5 ± 3.4	0.588 ± 0.083
1	Mistral	44.4 ± 10.6	0.58 ± 0.081
2	Mistral	41.0 ± 3.4	0.566 ± 0.094
3	Mistral	42.5 ± 3.4	0.588 ± 0.086

is acceptable for research-scale document collections.

In summary, there is a clear trade-off between cost and quality. The fixed window is efficient but imprecise, the few-shot approach balances cost and efficiency, and finally, the dynamic segment-based method achieves the best results at the highest computational cost. Therefore, when high textual fidelity and contextual reconstruction accuracy are prioritized, the segment-based method provides the optimal balance despite its increased runtime demands.

Chapter 7

Concluding Remarks and Future Directions

7.1 Concluding Remarks

This thesis presented a novel framework that extends the classical Handwritten Keyword Spotting (KWS) paradigm toward the retrieval of semantically coherent sentences from historical handwritten document images. The proposed approach bridges the gap between purely visual retrieval and linguistic understanding by integrating four complementary components: visual matching through a Seq2Seq KWS baseline, selective word-level transcription via Handwritten Text Recognition (HTR), dynamic context window estimation driven by Large Language Models (LLMs), and sentence correction through language-based reasoning.

The experimental results obtained on the IAM handwriting dataset demonstrated that the proposed dynamic windowing mechanism substantially improves the linguistic completeness and contextual fidelity of retrieved sentences compared to fixed-size approaches. Among the examined variants, the *segment-based prompting* strategy achieved the highest overall performance, with BLEU and BERTScore values of 77.7% and 85.0%, respectively, outperforming the few-shot alternative. The system proved robust even under the presence of transcription noise introduced by the HTR module, highlighting its potential for large-scale deployment in historical document collections.

Beyond quantitative metrics, the framework introduced here establishes a generalizable methodology for combining visual perception and language reasoning in a unified retrieval process. By treating LLMs as adaptive sentence boundary estimators and contextual refiners, the system demonstrates how linguistic priors can be leveraged to reconstruct coherent textual fragments directly from image-based queries. This contribution is particularly relevant to the fields of document analysis, digital humanities, and cultural heritage, where the recovery of textual meaning from handwritten artifacts remains a major challenge.

Overall, this research contributes both technically and conceptually to the advancement of sentence-level retrieval from handwritten sources. Technically, it introduces a scalable architecture for integrating visual and language models through selective transcription and prompt-based contextual reasoning. Conceptually, it redefines the notion of retrieval in document image analysis, moving from isolated word detection toward context-aware textual reconstruction.

7.2 Future Research Directions

While the proposed system achieved promising results, several directions for further research remain open. These directions aim to enhance generalization, scalability, and interpretability while broadening applicability to more diverse document types and languages.

(i) **Multilingual and historical adaptation.** Extending the current approach to multilingual and multi-script datasets would require retraining or adapting both the HTR and LLM components to handle varying alphabets, writing conventions, and historical orthography. Low-resource adaptation through transfer learning and few-shot fine-tuning could enable generalization to underrepresented languages and scripts.

(ii) **Layout and structure-aware retrieval.** Integrating document layout analysis (e.g., region segmentation, reading order estimation) could allow retrieval of entire paragraphs or article-level structures rather than isolated sentences. Combining sentence retrieval with layout reasoning would facilitate article reconstruction and page-level summarization.

(iii) Retrieval-augmented summarization. A natural extension of this work involves using the top-k retrieved sentence hypotheses as inputs to an LLM-based summarizer. This would enable the generation of concise, query-driven summaries of entire documents while preserving historical language characteristics.

(iv) End-to-end optimization. Although the current pipeline is modular by design, future implementations could explore end-to-end training strategies that jointly optimize the retrieval, transcription, and language reasoning stages. Such integration could minimize cascading errors and improve the overall semantic consistency of the reconstructed text.

(v) Human-centered evaluation and applications. Beyond quantitative metrics such as BLEU or BERTScore, human-centered evaluation could assess the interpretability and usefulness of the retrieved sentences for historians, archivists, and scholars. Developing interactive interfaces for semantic exploration of digitized archives would transform the system into a practical tool for digital humanities research.

In summary, this thesis demonstrates that bridging vision and language models can fundamentally transform information retrieval from handwritten documents. Future research building on these findings may lead to fully multimodal systems capable of query-guided summarization, semantic exploration, and contextual reconstruction of handwritten heritage archives.

Bibliography

[1] A. P. Giotis, G. Sfikas, B. Gatos, and C. Nikou, "A survey of document image word spotting techniques," *Pattern Recognition*, vol. 68, pp. 310–332, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317300870

[2] S. Sudholt and G. A. Fink, " PHOCNet: A Deep Convolutional Neural Network for Word Spotting in Handwritten Documents ," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2016, pp. 277–282. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICFHR.2016.0060

[3] Z. Zhong, W. Pan, L. Jin, H. Mouchère, and C. Viard-Gaudin, "Spottingnet: Learning the similarity of word images with convolutional neural network for word spotting in handwritten historical documents," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Oct 2016, pp. 295–300.

[4] G. Sfikas, G. Retsinas, and B. Gatos, "Zoning aggregated hypercolumns for keyword spotting," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Oct 2016, pp. 283–288.

[5] S. Sudholt and G. A. Fink, "Evaluating word string embeddings and loss functions for cnn-based word spotting," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, Nov 2017, pp. 493–498.

[6] N. Gurjar, S. Sudholt, and G. A. Fink, "Learning deep representations for word spotting under weak supervision," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, April 2018, pp. 7–12.

[7] P. Krishnan, K. Dutta, and C. Jawahar, "Deep feature embedding for accurate recognition and retrieval of handwritten text," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Oct 2016, pp. 289–294.

[8] P. Krishnan and C. V. Jawahar, "Matching handwritten document images," in *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 766–782.

[9] M. Al-Rawi, E. Valveny, and D. Karatzas, "Can one deep learning model learn script-independent multilingual word-spotting?" in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sep. 2019, pp. 260–267.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Tech. Rep., 2015. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[11] F. Wolf, K. Brandenbusch, and G. A. Fink, "Improving handwritten word synthesis for annotation-free word spotting," in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Sep. 2020, pp. 61–66.

[12] F. Wolf and G. A. Fink, "Annotation-free learning of deep representations for word spotting using synthetic data and self labeling," in *Document Analysis Systems*. Springer International Publishing, 2020, pp. 293–308.

[13] L. Gómez, M. Rusiñol, and D. Karatzas, "Lsde: Levenshtein space deep embedding for query-by-string word spotting," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, Nov 2017, pp. 499–504.

[14] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," Tech. Rep., 2013.

[15] G. Retsinas, G. Sfikas, G. Louloudis, N. Stamatopoulos, and B. Gatos, "Compact deep descriptors for keyword spotting," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Aug 2018, pp. 315–320.

[16] P. Krishnan, K. Dutta, and C. Jawahar, "Word spotting and recognition using deep embedding," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, April 2018, pp. 1–6.

[17] P. Krishnan and □. C. V. Jawahar, "Hwnet v2: an efficient word image representation for handwritten documents," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 22, pp. 387–405, 2019. [Online]. Available: https://doi.org/10.1007/s10032-019-00336-x

[18] F. Daraee, S. Mozaffari, and S. M. Razavi, "Handwritten keyword spotting using deep neural networks and certainty prediction," *Computers & Electrical Engineering*, vol. 92, p. 107111, 6 2021.

[19] S. Majumder, S. Ghosh, S. Malakar, R. Sarkar, and M. Nasipuri, "A voting-based technique for word spotting in handwritten document images," 2021. [Online]. Available: https://doi.org/10.1007/s11042-020-10363-0

[20] A. P. Giotis, G. Sfikas, and C. Nikou, "Adversarial deep features for weakly supervised document image keyword spotting," 2022. [Online]. Available: https://fki.tic.heia-fr.ch/databases/washington-database

[21] P. Krishnan, □. K. Dutta, and □. C. V. Jawahar, "Hwnet v3: a joint embedding framework for recognition and retrieval of handwritten text," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 26, pp. 401–417, 2023. [Online]. Available: https://doi.org/10.1007/s10032-022-00423-6

[22] F. Wolf and G. A. Fink, "Self-training for handwritten word recognition and retrieval," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 27, pp. 225–244, 2024. [Online]. Available: https://doi.org/10.1007/s10032-024-00484-9

[23] A. Matos, P. Almeida, P. L. Correia, and O. Pacheco, "iforal: Automated handwritten text transcription for historical medieval manuscripts," 2025. [Online]. Available: https://creativecommons.org/licenses/by/4.0/

[24] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, "Trocr: Transformer-based optical character recognition with pre-trained models," Tech. Rep., 2023. [Online]. Available: www.aaai.org

[25] G. Retsinas, G. Sfikas, C. Nikou, and P. Maragos, "From seq2seq recognition to handwritten word embeddings." British Machine Vision Conference (BMVC), 11 2021. [Online]. Available: https://www.bmvc2021-virtualconference.com/conference/papers/paper_1481.html

[26] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation."

[27] T. Liu, Y. E. Jiang, N. Monath, R. Cotterell, M. Sachan, and E. Zürich, "Autoregressive structured prediction with language models." [Online]. Available: https://github.com/lyutyuh/ASP

[28] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models." [Online]. Available: https://github.com/facebookresearch/xformers

[29] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. Michael, S. Ranjan, S. Xiaoqing, E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open foundation and fine-tuned chat models," 2023.

[30] L. Team and A. . Meta, "The llama 3 herd of models," 2024.

[31] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b."

[32] W. Fei, X. Niu, P. Zhou, L. Hou, B. Bai, L. Deng, and W. Han, "Extending context window of large lan-guage models via semantic compression," Tech. Rep., 2023.

[33] J. Ainslie, J. Lee-Thorp, M. D. Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, "Gqa: Training generalized multi-query transformer models from multi-head checkpoints," pp. 4895–4901. [Online]. Available: https://github.com/google/flaxformer

[34] C. Zhu, W. Ping, C. Xiao, M. Shoeybi, T. Goldstein, A. Anandkumar, and B. Catanzaro, "Long-short transformer: Efficient transformers for language and vision." [Online]. Available: https://github.com/NVIDIA/transformer-ls.

[35] Z. Fu, W. Song, Y. Wang, X. Wu, Y. Zheng, Y. Zhang, D. Xu, X. Wei, T. Xu, and X. Zhao, "Sliding window attention training for efficient large language models," 2025.

[36] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," Tech. Rep. [Online]. Available: http://www.statmt.org/moses/

[37] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," Tech. Rep.

[38] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," Tech. Rep., 2017.

[39] A. Vaswani, G. Brain, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[40] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," 2023. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/roformer.

[41] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2552–2566, 2014.

[42] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Association for Computational Linguistics, 2002, p. 311–318. [Online]. Available: https://doi.org/10.3115/1073083.1073135

[43] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," 01 2004, p. 10.

[44] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Association for Computational Linguistics, 06 2005, p. 65–72. [Online]. Available: https://aclanthology.org/W05-0909/

[45] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," Tech. Rep., 2020. [Online]. Available: https://github.com/Tiiiger/bert_score.

[46] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "Bert: Pre-training of deep bidirectional transformers for language understanding," Tech. Rep., 2019. [Online]. Available: https://github.com/tensorflow/tensor2tensor

[47] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," Tech. Rep., 2019. [Online]. Available: https://github.com/UKPLab/

[48] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, and P. G. Allen, "Roberta: A robustly optimized bert pretraining approach," Tech. Rep., 2019. [Online]. Available: https://github.com/pytorch/fairseq

[49] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in bertology: What we know about how bert works," 2020. [Online]. Available: https://doi.org/10.1162/tacl

[50] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019. [Online]. Available: http://dx.doi.org/10.1561/2200000056

[51] E. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large lan-guage models," Tech. Rep., 2021. [Online]. Available: https://github.com/microsoft/LoRA.

[52] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, and F. L. Wang, "Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment," Tech. Rep., 2023. [Online]. Available: https://huggingface.co/blog/falcon-180b#hardware-requirements

[53] Z. Han, C. Gao, J. Liu, J. Zhang, and S. Q. Zhang, "Parameter-efficient fine-tuning for large models: A comprehensive survey," Tech. Rep., 2024.

Short Biography

Georgios Voudiotis was born in Samos in 1995. He received his B.Sc. in Mathematics from the University of Ioannina in 2022. During his undergraduate studies, he developed a strong interest in computer science, particularly in the field of data science. Motivated by this interest, he pursued a Master's degree in the Department of Computer Science and Engineering at the same university, specializing in Data Science and Engineering, which he completed in 2025.

His main academic interests include Machine Learning, Computer Vision, and Optimization. He is proficient in Python and has extensive experience with scientific computing and machine learning libraries such as NumPy, Pandas, Matplotlib, Scikit-learn, PyTorch, and TensorFlow.

He is passionate about bridging theoretical foundations with practical applications in data science and aims to contribute to innovative research in the broader fields of intelligent systems and applied machine learning.