Algerian Democratic and Popular Republic
Ministry of Higher Education and Scientific Research
Higher School of Computer Science
- 08 May 1945 - Sidi Bel Abbès

THESIS

To obtain the **Master's Degree**

Field: **Computer Science**
Specialty: **Artificial Intelligence and Data Science**

Theme:

Intelligent Processing of Medical Documents Using Deep Learning.

Presented by:

SENKADI Khawla

BELMILOUD Maroua

Submission Date: **June 2025**

In front of the jury composed of:

Dr. BOUSMAHA Rabab	President
Dr. MEZRAR Samiha	Examiner
Dr. BENSLIMANE Sidi Mohamed	Supervisor
Dr. DIF Nassima	Supervisor

Academic Year: **2024/2025**

Abstract

Intelligent medical document processing has become essential for improving clinical workflows and supporting timely, evidence-based medical decisions. The increasing volume and complexity of clinical data present significant challenges in the healthcare domain, particularly in organizing, interpreting, and leveraging this information effectively. The initial part of our discussion focuses on the use of **Optical Character Recognition (OCR)** for digitizing unstructured clinical texts, often derived from scanned records, and **Natural Language Processing (NLP)** for extracting meaningful insights from this textual content. The second part of this dissertation is dedicated to the latest research in analyzing complex medical documents using **multi-label classification** techniques, which allow the identification of multiple relevant clinical labels per document. Powered by advanced **deep learning** models, this dissertation focuses on interpreting, extracting, and classifying medical information with enhanced efficiency, accuracy, and scalability.

Keywords: Intelligent Medical Document Processing, OCR, NLP, Deep Learning, Multi-Label Classification, decision-making.

Résumé

Le traitement intelligent des documents médicaux est devenu essentiel pour améliorer les flux de travail cliniques et soutenir des décisions médicales éclairées et prises en temps opportun. L'augmentation du volume et de la complexité des données cliniques représente un défi majeur pour le secteur de la santé, en particulier lorsqu'il s'agit d'organiser, d'interpréter et d'exploiter efficacement ces informations. La première partie de notre étude se concentre sur l'utilisation de l'**Optical Character Recognition (OCR)** pour numériser les textes cliniques non structurés, souvent issus de documents numérisés, ainsi que sur le **Natural Language Processing (NLP)** pour extraire des informations pertinentes de ce contenu textuel. La seconde partie de cette dissertation est consacrée aux travaux de recherche les plus récents portant sur l'analyse de documents médicaux complexes à l'aide de techniques de **classification multi-étiquette**, qui permettent d'identifier plusieurs étiquettes cliniques pertinentes par document. S'appuyant sur des modèles avancés d'**apprentissage profond**, cette dissertation se focalise sur l'interprétation, l'extraction et la classification d'informations médicales avec une efficacité, une précision et une scalabilité accrues.

Mots-clés : Traitement Intelligent des Documents Médicaux, OCR, NLP, Deep Learning, Classification Multi-Étiquette, prise de décision.

الملخص

معالجة المستندات الطبية الذكية أصبحت ضرورية لتحسين سير العمل السريري ودعم اتخاذ قرارات طبية مبنية على الأدلة وفي الوقت المناسب. إن التزايد المستمر في حجم وتعقيد البيانات السريرية يمثل تحديًا كبيرًا في مجال الرعاية الصحية، خاصة فيما يتعلق بتنظيم هذه المعلومات وتفسيرها واستغلالها بشكل فعّال. تركز الجزء الأول من هذا العمل على استخدام تقنية **التعرف الضوئي على الحروف** لرقمنة النصوص السريرية غير المهيكلة، والتي غالبًا ما تكون مستخرجة من سجلات ممسوحة ضوئيًا، بالإضافة إلى استخدام تقنيات **معالجة اللغة الطبيعية** لاستخلاص معلومات ذات مغزى من هذا المحتوى النصي.

أما الجزء الثاني من هذه الأطروحة فهو مكرّس لأحدث الأبحاث في تحليل الوثائق الطبية المعقدة باستخدام تقنيات **التصنيف متعدد التسميات**، والتي تتيح تحديد عدة تسميات سريرية مرتبطة بكل مستند. وبالاعتماد على نماذج متقدمة من **التعلم العميق**، تركز هذه الأطروحة على تفسير واستخلاص وتصنيف المعلومات الطبية بكفاءة ودقة وقابلية توسع محسّنة.

الكلمات المفتاحية: المعالجة الذكية للوثائق الطبية، التعرف البصري على الحروف، معالجة اللغة الطبيعية، التعلم العميق، التصنيف المتعدد، اتخاذ القرار.

Acknowledgements

We begin by thanking Allah for the strength and perseverance to complete this work.

We would like to express our sincere gratitude to our supervisors, **Dr. Sidi Mohamed Benslimane** and **Dr. Nassima Dif**, for their guidance, valuable feedback, and continuous support throughout this project. We also thank the jury president and members for their time and constructive remarks, which contributed to improving the quality of this thesis.

We are deeply thankful to the CASNOS institution for welcoming us and providing an excellent environment and the necessary resources for our internship. Our appreciation goes especially to **Mr. Chems Eddine Boulassel**, General Manager, **Mr. Mohand Alibey**, Head of the DMSI Department, and **Mr. Amine Kellaci**, for their trust and support throughout our time there. We are equally grateful to all the staff, doctors, and engineers at CASNOS, and in particular to **Kadi Amir Khalid** and **Bourahla Amayas**, whose assistance and technical guidance were instrumental to the success of our work.

This thesis is the result of collective support, effort, and belief. We are truly thankful to have been surrounded by people who made it possible.

Dedication

All praise is due to **Allah**. I am grateful for the strength to think, write, and believe in my path, and for the patience to persevere. Above all, I thank Allah for the joy of raising my hands in prayer and saying, **Alhamdulillah**.

This work is lovingly dedicated to my parents, **Fatima** and **Miloud**. To my mother, the source of unconditional love and strength your sacrifices, prayers, and unwavering support have guided me in every step. To my father, whose wisdom, values, and belief in me have shaped who I am thank you for being my strength and inspiration. I owe so much of who I am to both of you and will always strive to make you proud.

To my brother, **Omar Sid Ahmed**, and my sister, **Safa**, thank you for your strength, support, and comforting presence. Your encouragement has lifted me in difficult times and made every step more meaningful. I am truly blessed to walk through life with you by my side.

To my grandfather **Abdellah** and grandmother **Ouda**, your love and values continue to guide me. Thanks also to my uncles and aunts especially **Rachid** and **Fatiha**, **Saad** and **Sead**, **Samira** and my uncle **Mhammed** for your constant support.

To **Khawla**, my teammate and sister at heart your dedication and friendship were key to this accomplishment. These five years together have been truly special, and I am deeply thankful for the incredible bond we have built.

To **Yousra**, **Dina**, and **Melissa**, your friendship has been a source of joy and strength. The countless moments of laughter, support, and encouragement will always hold a special place in my heart.

To the people of **Gaza**, your courage and faith are a source of deep inspiration. I dedicate this work to the hope of peace and freedom for you.

To all of you, thank you for being part of this journey.

Maroua.

Dedication

All praise and thanks are due to Allah, the Most Merciful, who granted me the strength and patience to complete this work. His guidance and blessings have supported me throughout this journey and beyond, in ways I can never fully express.

I dedicate this work to my dear parents, **Abdelkader** and **Malika**, whose unwavering support, endless sacrifices, and constant belief in me have shaped who I am today. I am deeply grateful for everything you've done and for always standing by my side.

To my siblings **Meryem, Amina,** and **Mohamed Elfatih**, thank you for the joy you bring into my life and for the support you've shown along the way. I'm grateful to have you, and I look forward to seeing each of you grow and succeed on your own paths.

To **my grandmother**, whose prayers and quiet presence have always been a comfort.

To my dear **Hafsa**, thank you for simply being there. Whether through laughter, happy moments, or quiet ones, your friendship was a comfort, and I'll always be grateful for it.

To **Maroua**, my project partner and companion from the very beginning — thank you for being there through every step of this long journey. Your commitment and constant support in work and beyond meant a lot. I'm truly thankful we shared it all together.

To all the beautiful souls I met at the Musalla and the university residence, each one by her name — thank you for the peaceful company, the kindness, and the precious moments I'll always remember. I'm glad that you were part of this journey.

To the souls of martyrs in Gaza, who taught us that dignity comes at a price,
To Gaza, to its pulse of resistance, to its voices that became a hymn of heroism,
To its women and men, to their steadfastness in the face of every storm.
To all of them, I dedicate this success to say:
We follow your path, and we will never forget.

Khawla.

Contents

List of Figures

List of Tables

Acronyms

Acc Accuracy

AI Artificial Intelligence

ANN Artificial Neural Networks

Arr Arrhythmia

CNN Convolution Neural Networks

CVDS Cardiovascular Diseases

DL Deep Learning

ECG Electrocardiogram

EKG Electrocardiography

FN False Negative

FP False Positive

GRU Gated Recurrent Unit

HDF5 Hierarchical Data Format version 5

IoT Internet of Things

LSTM Long Short-Term Memory

ML Machine Learning

NN Neural Network

RNN Recurrent Neural Networks

SVM Support Vector Machines

TN True Negatives

TP True Positives

ReLU The Rectified Linear Unit

Chapter 1

Introduction

1.1 Introduction

Artificial Intelligence (AI) is revolutionizing the medical field by enabling more accurate, faster, and cost-effective solutions across a wide range of clinical tasks. From diagnostic imaging and predictive analytics to personalized treatment recommendations and administrative automation, AI enhances both patient outcomes and healthcare efficiency. In particular, deep learning a subfield of AI has shown remarkable success in interpreting complex medical data, including images, signals, and free-text clinical records. Its integration into healthcare systems has supported tasks such as disease detection, clinical decision support, and patient monitoring. A particularly promising application lies in processing unstructured medical documents radiology reports, lab test results, and discharge summaries which contain essential information for diagnosis and treatment but are challenging to analyze automatically due to their unstructured nature. By combining deep learning with natural language processing (NLP), optical character recognition (OCR), it becomes possible to transform raw textual and image-based medical data into structured, interpretable insights. Such systems hold the potential to significantly reduce the manual workload of healthcare professionals and provide faster, more accurate access to critical clinical information.

1.2 Challenges

A primary challenge in processing medical documents is their **unstructured and varied nature**. These documents appear in many different formats, such as digital records, scanned files, printed forms, or handwritten notes, making it difficult to apply one standard method of analysis. The **quality** of the documents also varies; some may have low resolution, unclear handwriting, or incomplete information, which can reduce the accuracy of systems like OCR (Optical Character Recognition). In addition, there is a high level of **heterogeneity** across hospitals and departments, where each may use different structures, medical terms, or languages in their documentation. Another common issue is the **multi-label nature** of clinical texts, as one document can be linked to multiple diagnoses or findings. This requires models that can handle overlapping and complex labels. Finally, **data privacy concerns** and the **limited access to large, annotated medical datasets** make it difficult to train and evaluate reliable AI models. These challenges must be addressed to develop effective systems for understanding and using medical documents in healthcare.

1.3 Motivation

The motivation for this study arises from the increasing need for **intelligent systems** capable of assisting healthcare professionals in managing the growing **complexity** and **volume of clinical data**. Automating the **extraction** and **classification** of medical information from **unstructured documents** can significantly reduce clinicians' **administrative workload**, enhance the **accuracy** and **accessibility** of patient records, and contribute to **early disease detection** and **clinical decision support**. Furthermore, such systems facilitate **large-scale clinical data analysis**, which is essential for advancing **medical research** and **public health initiatives**. Integrating **deep learning** with **Optical Character Recognition (OCR)** and **Natural Language Processing (NLP)** offers a promising solution for converting unstructured medical documents into **structured data**, thereby advancing **digital healthcare** through more **efficient**, **personalized**, and **data-driven** care delivery.

1.4 Organisation and Structure

This document represents the first phase of our comprehensive project, focused on the academic analysis and evaluation of various approaches for medical document processing and intelligent diagnostic systems. The report is structured to provide both the necessary technical foundations and a detailed overview of state-of-the-art methods in the field.

- **Chapter 2: INTELLIGENT DOCUMENT PROCESSING (IDP) IN HEALTHCARE** – This chapter introduces IDP and its role in automating the processing of medical documents using AI technologies like OCR, NLP, and machine learning. It highlights key benefits and use cases in healthcare, to improve efficiency and data accuracy.
- **Chapter 3: Deep Learning** – Introduces the fundamentals of deep learning, covering core architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Transformers. It also outlines commonly used evaluation metrics relevant to medical applications.
- **Chapter 4: Natural Language Processing (NLP)** – Presents key NLP concepts and techniques, including word embeddings, keyword extraction, and clinical language models. This chapter provides the foundation for understanding how unstructured text from medical records can be processed effectively.
- **Chapter 5: Optical Character Recognition (OCR)** – Describes the components and processing pipeline of OCR technology, emphasizing its application to the digitization and interpretation of medical documents.
- **Chapter 6: State of the Art** – Reviews the latest research related to medical document text extraction and multi-label classification, with a particular focus on AI-based diagnostic algorithms. It also discusses the outcomes and key findings from the literature.

The document concludes with a summary of the main contributions and findings of this initial research phase, setting the stage for future development and implementation work.

Part I

Background

Chapter 2

INTELLIGENT DOCUMENT PROCESSING (IDP) IN HEALTHCARE

1 Introduction

Intelligent Document Processing (IDP) is an AI-powered approach designed to handle large volumes of unstructured and semi-structured healthcare documents. It combines technologies like OCR, natural language processing (NLP), and machine learning to automatically extract, classify, and interpret medical data from sources such as clinical notes, prescriptions, insurance forms, and lab reports.

In healthcare, IDP enhances data accuracy, reduces manual workload, speeds up administrative tasks such as claims processing and electronic health record (EHR) updates, and ensures better regulatory compliance. By transforming raw documents into structured, actionable information, IDP supports faster decision-making, improves patient care delivery, and contributes to more efficient healthcare operations.

2 Understanding intelligent document processing (IDP)

Intelligent Document Processing (IDP) streamlines document management by integrating artificial intelligence technologies such as machine learning (ML), optical character recognition (OCR), and natural language processing (NLP). This combination enables the automatic extraction, processing, and management of information with greater speed and accuracy than traditional manual approaches, making IDP particularly valuable in data-intensive sectors like healthcare [IDP, 2025].

The IDP process follows a structured workflow that includes several key phases: **capture, separation, classification, extraction,** and **validation**.

- **Capture:** Physical or digital documents are collected and ingested into the system.
- **Separation:** The input is divided into individual documents for targeted processing.
- **Classification:** Documents are organized based on their type, content, or layout.
- **Extraction:** Relevant data is identified and retrieved using AI-driven techniques.
- **Validation:** The extracted information is checked for accuracy and compliance with predefined rules.

Finally, the processed data is routed to the appropriate systems or end users for further action.

3 The Benefits of IDP in Healthcare

Healthcare organizations that adopt Intelligent Document Processing (IDP) can gain several important benefits. These include:

- **Reduced operational costs:** Automation reduces the need for manual data entry and document handling, leading to significant cost savings.
- **Prevention of dangerous human mistakes:** AI-driven extraction and validation minimize the risk of errors that could affect patient safety or compliance.
- **Improved workflow efficiency:** IDP streamlines document-centric processes, reducing delays and administrative bottlenecks.

- **Easy data accessibility:** Digitized and structured data becomes easily searchable and available across systems, supporting better decision-making.
- **Improved data security:** Automated systems help enforce access controls and audit trails, enhancing the protection of sensitive health information.

The diagram in Figure 2.1 illustrates the workflow of Intelligent Document Processing (IDP) in healthcare. It begins with input files or documents—such as medical records or insurance forms—being fed into the IDP software. The system then extracts relevant data using AI-based techniques, followed by verification and cross-check validation to ensure the accuracy and reliability of the extracted information. After that, the data undergoes automated entry and anonymization to protect patient privacy. Finally, the processed data is converted into structured formats like JSON or CSV and sent to an Electronic Health Record (EHR) system or database, enabling easy access and integration across healthcare platforms.

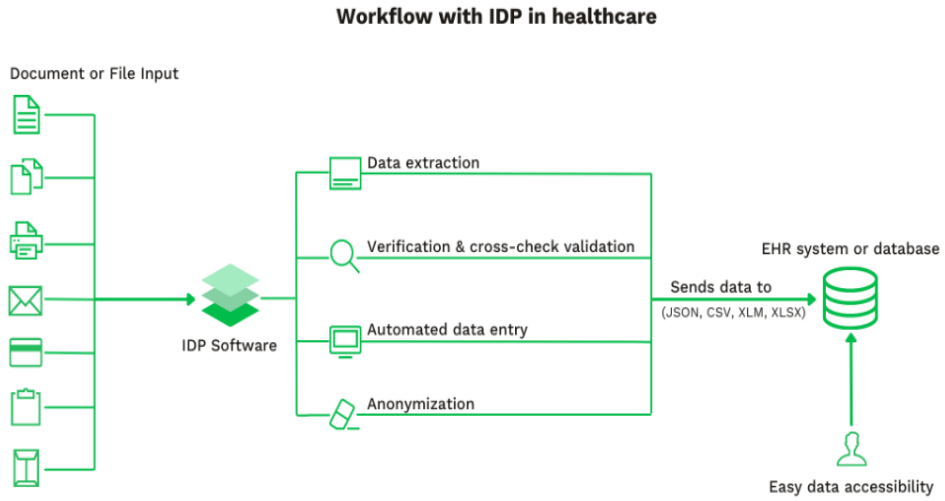

Figure 2.1: Overview of the Intelligent Document Processing (IDP) workflow. [klippa, 2025]

4 Use Cases of IDP in the Healthcare Industry

Intelligent Document Processing (IDP) has numerous applications in the healthcare sector. Among the most common and impactful use cases are:

- **Automated patient onboarding**
- **Data extraction from patient records**
- **Medical records classification**
- **Prior authorization**
- **Clinical research and documentation**

The following section explores these use cases in greater detail.

4.1 Automated Patient Onboarding

Automated patient onboarding refers to the use of digital tools to streamline the process of registering new patients and integrating their information into healthcare systems. Intelligent Document Processing (IDP) plays a key role in this by automatically extracting and structuring data from essential medical documents, many of which are still paper-based or unstructured. This reduces manual entry, speeds up administrative workflows, and ensures greater accuracy during the onboarding process.

Insurance Card Scanning with IDP Intelligent Document Processing (IDP) enables automatic extraction of key details from scanned insurance cards using OCR. The structured data is then directly integrated into the Electronic Health Record (EHR) system, streamlining patient onboarding.

1. Insurance Card Scanning Insurance card scanning can be performed using a mobile device to capture an image of the document for digital processing, as illustrated in Figure 2.2.

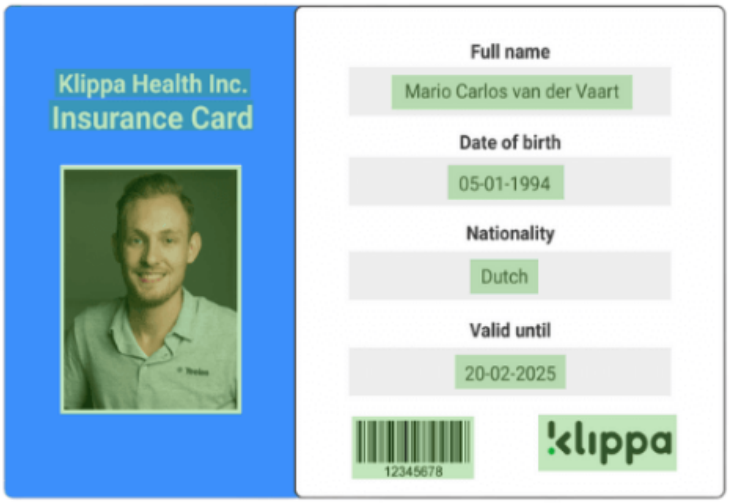

Figure 2.2: Insurance Card Scanning Example. [klippa, 2025]

2. Data Extraction After scanning, relevant data fields from the insurance card are automatically extracted and converted into structured text, as shown in Figure 2.3.

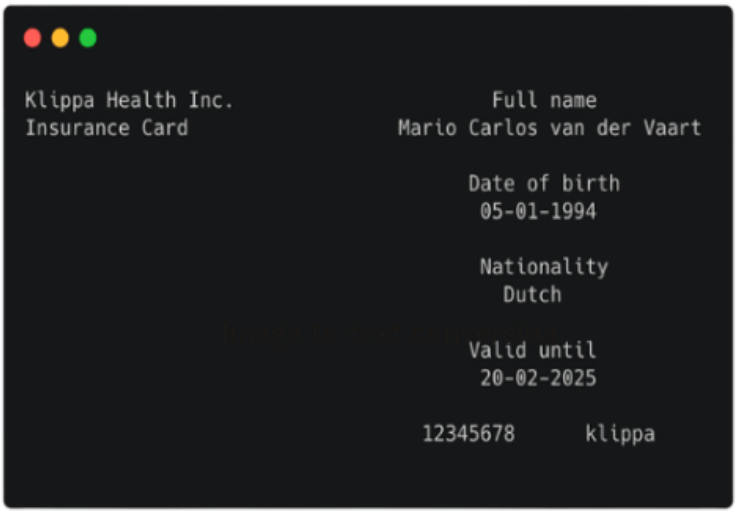

Figure 2.3: Data Extraction Example. [klippa, 2025]

3. Data Conversion Once extracted, the text is converted into a machine-readable format (e.g., JSON, CSV, XML, XLSX), making it suitable for integration with downstream systems and processes, as depicted in Figure 2.4.

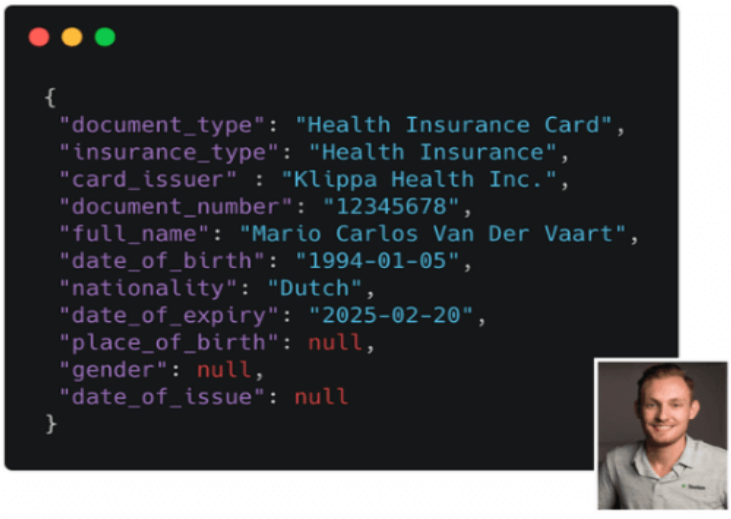

Figure 2.4: Data Conversion Example. [klippa, 2025]

4.2 Data Extraction from Patient Records

Manually processing patient records in an EHR system is both time-consuming and prone to errors, potentially leading to legal issues.

It is essential to find a solution to automatically sort, identify, extract and match the data from documents, faxes, PDFs, and scans to a patient's EHR.

The extracted information from medical documents typically includes details such as the hospital name, patient's name, age, date of birth, medical history, blood type, doctor's name, diagnosis, and relevant dates, as illustrated in Figure 2.5.

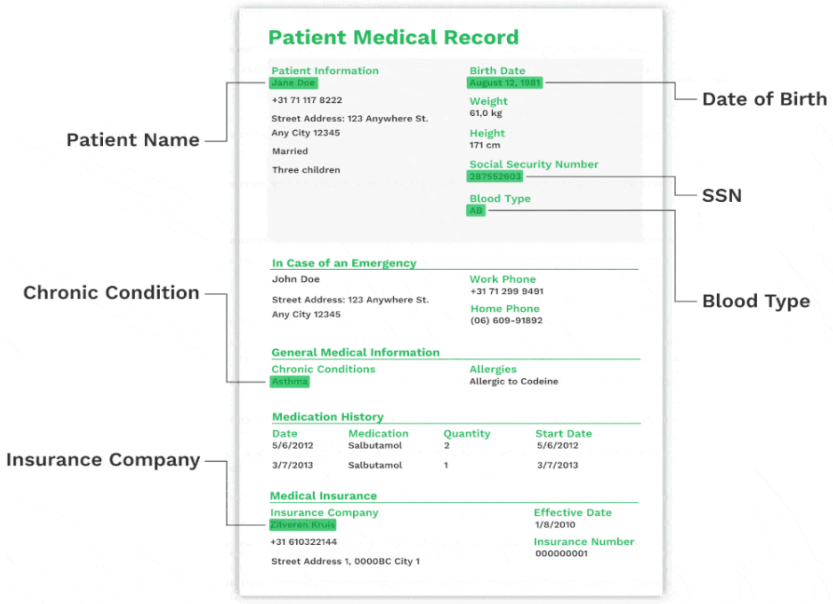

Figure 2.5: Data Extraction from Patient Record. [klippa, 2025]

4.3 Medical records classification

Intelligent Document Processing (IDP) enhances the management of medical records by applying automated classification techniques. These techniques use AI to analyze document content and assign each file to a specific category—such as physician notes, lab reports, or discharge summaries—based on its structure, language, and metadata. This classification process replaces traditional manual indexing, saving time and reducing errors [IDP, 2025].

Once classified, documents are automatically sorted and organized, making it much easier to retrieve patient files and locate relevant information. When integrated with Electronic Health Record (EHR) systems, IDP enables faster access to clinical data, streamlining workflows and allowing healthcare professionals to focus more on patient care and make quicker, data-driven decisions.

4.4 Prior authorization

IDP helps simplify the prior authorization process for payers by automatically capturing and extracting the necessary information from provider requests. This significantly reduces processing

time, making it easier to stay compliant with deadlines.

By streamlining these approvals, IDP not only accelerates decision-making for insurers but also helps healthcare providers deliver timely care leading to smoother operations and a better overall experience for patients.

4.5 Clinical research and documentation

IDP enhances clinical research by automating the management of research documents, ensuring secure access, proper routing, and regulatory compliance. It streamlines the digitization and analysis of data, enabling faster, more accurate information sharing across research teams. This not only improves collaboration but also supports more efficient, evidence-based decision-making throughout the research process.

5 Conclusion

Intelligent Document Processing (IDP) is transforming the way healthcare organizations handle unstructured and semi-structured data. By leveraging technologies such as OCR, NLP, and machine learning, IDP automates the extraction, classification, and integration of information from various medical documents. This not only enhances data accuracy and operational efficiency but also reduces the risks associated with manual processing. Through use cases like automated patient onboarding and data extraction from patient records, IDP proves to be a powerful tool for improving administrative workflows, supporting better clinical decisions, and ensuring regulatory compliance. Ultimately, IDP contributes to more reliable, secure, and efficient healthcare delivery.

Chapter 3

Deep Learning and Natural Language Processing

1 Introduction

Artificial Intelligence (AI) has experienced a major leap forward with the advent of deep learning, a subset of machine learning that enables systems to automatically learn complex patterns from large volumes of data. Fueled by advances in computing power and the availability of massive datasets, deep learning has significantly outperformed traditional algorithms in areas such as image recognition, speech processing, and, increasingly, Natural Language Processing (NLP).

NLP is the field of AI focused on enabling machines to understand, interpret, and generate human language, whether it be written, spoken, or handwritten. As AI-driven tools become more embedded in daily life, the role of NLP in powering chatbots, voice assistants, document summarization, sentiment analysis, and medical text classification has grown substantially. These breakthroughs are largely enabled by deep learning techniques, which have transformed how machines handle the intricacies of human language.

At the heart of deep learning lie neural networks—models composed of layers of interconnected nodes that progressively extract and refine features from raw input. Through processes such as

forward propagation and backpropagation, these networks learn to make increasingly accurate predictions or classifications. In NLP, these models are used to perform tasks like tokenization, keyword extraction, named entity recognition, and semantic understanding.

This chapter explores how deep learning serves as a powerful foundation for modern NLP, illustrating its importance in developing intelligent systems capable of understanding and generating human language across diverse applications.

2 A brief history of Natural Language Processing

Computers gain capacity to process human language through Natural Language Processing (NLP), which constitutes a sub-discipline of Artificial Intelligence (AI). NLP research developed through history from its initial stage in the late 1940s before important breakthroughs occurred in the 1950s through to the 1960s. The first computer-based NLP application started with machine translation programs motivated by Weaver's 1949 memorandum. Research teams during this period built language rule systems by hand, although these methods proved ineffective when dealing with ambiguity in language use. The establishment of generative grammar took place during 1957.

The excessively optimistic tone of that era produced unrealistic goals for linguistic-assisted machine translation during the late 1960s slowdown period. Theoretical along with practical breakthroughs persisted throughout the 1970s because of new introductions such as conceptual ontologies and symbolic approaches. During the 1980s, researchers changed their approach to use statistical models instead of hand-written rules and began placing greater value on machine learning algorithms. The statistical models operated through soft probabilistic decisions while also overcoming hand-written rule complexity limitations. Deep learning and neural networks emerged during the late 1980s and early 1990s and transformed NLP into a modern discipline that remains actively developed through these technologies [Foote et al, 2023].

A summarized timeline illustrating these major milestones is presented in Figure 3.3

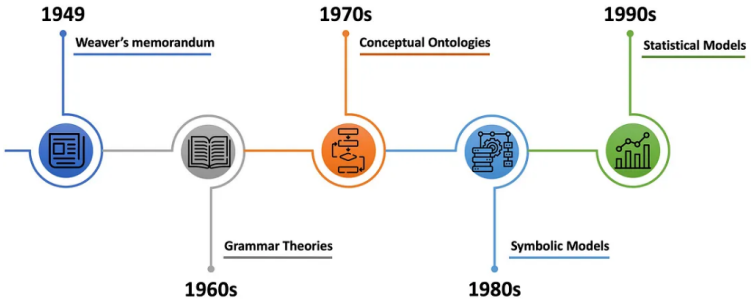

Figure 3.1: NLP History. [Louis et al, 2023].

3 Natural Language Processing

Natural language processing functions as a specialized domain which belongs to artificial intelligence and computer science along with linguistics since its main mission involves making computers understand human dialogues through speech and written content. Various daily products and services employ this technology including voice-activated digital assistants which appear on smartphone platforms [Gillis et al, 2023].

4 NLP componenets

For any communication to take place, these two things are necessary. The first understands, and the other generates (known as a response in a more common language).

A high-level overview of these components is illustrated in Figure 3.2

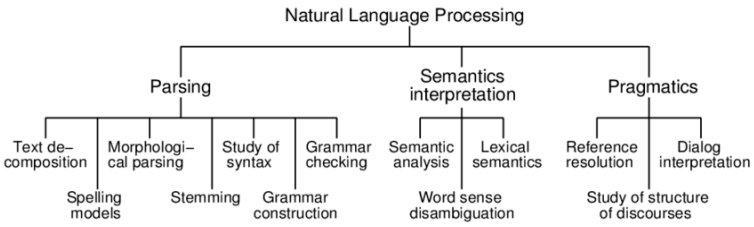

Figure 3.2: NLP componenets. [Taxonomy, 2024].

4.1 Natural Language Understanding (NLU)

NLU according to Gartner stands for a computer system's capability to decode the construction and semantic content of spoken languages thus permitting users to converse with technology through normal verbal communication. The subset of Artificial Intelligence known as NLU enables computer software to convert unstructured data and texts into machine language which results in straightforward human interpretable outputs. [Mcshane et al, 2017].

4.2 Natural Language Generation (NLG)

The creation of natural human language or speech from machine-generated data forms the basis of NLG. The system aims to converge machines with human understanding through generation of text content that people understand. The technology of NLG serves various operational fields including chatbots together with automated report creation and text summarization systems [Semaan et al, 2012].

5 NLP techniques

The NLP techniques used in natural language processing apps include :

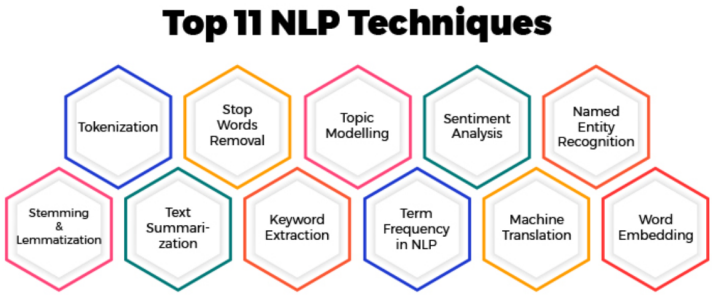

Figure 3.3: NLP Techniques. [Hrithik, 2022].

5.1 Tokenization

Tokenization is a fundamental and essential technique in Natural Language Processing (NLP) that involves breaking down a continuous text string into smaller units called tokens. These tokens can represent words, symbols, numbers, or other meaningful elements.

5.2 Stop words removal

In the preprocessing pipeline, the next step is stop words removal. Stop words are common words in a language that serve as connectors or relationship indicators but do not contribute significant meaning to the text.

5.3 Stemming and Lemmatization

After tokenization there exists a crucial NLP technique in the preprocessing pipeline which involves either stemming or lemmatization processing of text. These techniques are essential for normalizing words to their root forms, especially in scenarios like search engines where users expect results for various forms of a word.

5.4 Keyword extraction

The text analysis approach in Natural Language Processing (NLP) called keyword extraction automatically finds significant words and phrases in documents to provide quick knowledge about topic subject matter. Similar to text reading comprehension humans naturally notice vital words in context while dismissing nonessential words to comprehend meaning. Keyword extraction processes extensive text documents to extract key terms which in turn saves time and effort.

5.5 TF-IDF and CountVectorizer

While not technically word embeddings, Term Frequency-Inverse Document Frequency (TF-IDF) and CountVectorizer are classical vectorization techniques. TF-IDF reflects how important a word is to a document in a collection, while CountVectorizer simply counts the frequency of each word.

These methods produce sparse and high-dimensional vectors but remain useful for traditional machine learning models.

5.6 Word Embeddings

Machine learning and deep learning algorithms require numerical representations of textual data. Word embeddings convert words into continuous vector representations in an n-dimensional space, capturing syntactic and semantic relationships between them. Words with similar meanings are placed closer in the vector space. These embeddings are usually learned from large text corpora and are essential for various NLP tasks.

The most widely used techniques are detailed below [Swimm, 2023].

Word2Vec

Word2Vec is one of the most influential word embedding models, introduced by Mikolov et al. It is based on a shallow neural network that learns word associations from a large corpus of text. Word2Vec offers two architectures: Continuous Bag of Words (CBOW), which predicts the target word from surrounding context words, and Skip-Gram, which does the inverse—predicting context words given a target word. The resulting word vectors reflect syntactic and semantic similarity.

GloVe

GloVe (Global Vectors for Word Representation), developed by Stanford, combines the advantages of global matrix factorization and local context-based learning. Unlike Word2Vec, which relies on local context windows, GloVe uses word co-occurrence statistics from the entire corpus to learn embeddings. This enables it to capture broader relationships between words based on their overall frequency patterns in documents.

FastText

FastText, developed by Facebook AI Research, extends Word2Vec by representing words as bags of character n-grams. This allows FastText to create embeddings for out-of-vocabulary words (OOV)

by composing them from subword units. It is particularly useful for morphologically rich languages or datasets with rare or misspelled words.

ELMo

ELMo (Embeddings from Language Models) is a deep contextualized word representation model developed by AllenNLP. Unlike static embeddings like Word2Vec and GloVe, ELMo generates context-sensitive embeddings for words depending on their usage in a sentence. It is based on bidirectional LSTM language models trained on large corpora, capturing both syntax and semantics.

BERT

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model developed by Google. It produces dynamic contextual embeddings for each word, allowing for understanding of complex language phenomena such as polysemy. BERT is pre-trained on a large corpus with masked language modeling and next sentence prediction tasks and fine-tuned on specific downstream tasks.

5.7 Named Entity Recognition

NER is a subfield of Information Extraction (IE) that focuses on identifying and classifying predefined named entities, such as person names, organizations, and quantitative values, from unstructured text. NER is similar to keyword extraction, but it goes a step further by categorizing the extracted keywords into predefined classes.

6 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs), also known as ConvNets, are a type of deep neural network used primarily in computer vision and image classification applications. They can extract features and identify patterns from images and videos through convolutional layers, which apply filters to detect visual patterns. CNNs also include pooling layers, which reduce the dimensionality of feature

maps via down-sampling operations. Finally, fully connected layers assign the final classification based on the extracted features from the convolutional stages [Keita, 2023].

The overall architecture of a CNN, as applied to digit recognition, is illustrated in Figure 3.4

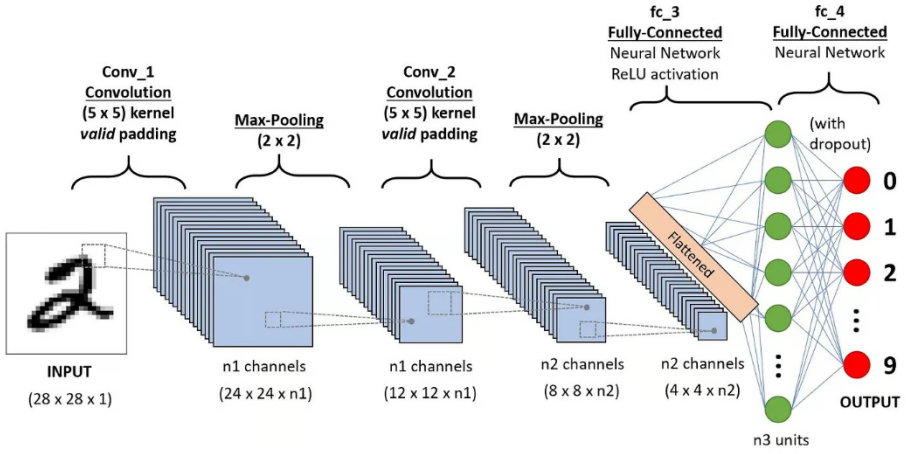

Figure 3.4: Architecture of the CNNs applied to digit recognition. [Keita, 2023].

6.1 Components of CNN

- **Convolutional Layer:** The input receives convolutional filter processing which extracts local features from the input through the process such as edges or textures.
- **Pooling Layer:** The network performs spatial data reduction by executing down-sampling operations like max pooling or average pooling.
- **Fully Connected Layer:** The model applies full interconnectivity between every neuron from one layer with every neuron from another layer which activates at the end of the network for classification or regression tasks.

6.2 Types of CNN

- **LeNet:** LeNet stands as one of the original CNN architectures dedicated to performing character recognition tasks. like digit classification.

- **AlexNet:** This development added deeper network architectures while implementing ReLU activation functions. It won the ImageNet competition in 2012.
- **VGG:** is a convolutional neural network developed by the Visual Geometry Group at Oxford. It uses a homogeneous architecture where convolutional layers are stacked to increase depth, enabling the extraction of increasingly abstract features [VGG, 2020].
- **ResNet:** To resolve the vanishing gradient problem resilient connections were added. for very deep networks. ResNet-50 and ResNet-101 are popular variants [resnet, 2020].
- **MobileNet:** This model has a specific target market of embedded vision applications to reduce both the number of parameters and computational cost [IBM, 2025].

7 Reccurent Neural Networks (RNN)

The deep neural network structure named Recurrent Neural Network (RNN) processes sequential or time-series data to create predictions by considering previous input context. RNNs serve applications including natural language processing, language translation and sentiment analysis, speech recognition and time-series forecasting because their architecture supports processing data that depends on sequence order.

Working Mechanism The principal function of Recurrent Neural Networks (RNNs) involves keeping an internal hidden state to remember previous sequence inputs. RNNs combine the current input alongside their hidden state at each time step, which provides memory functionality for preceding context. The feedback design enables RNNs to process sequential relationships. RNNs accomplish their reduction of complexity by utilizing identical weight parameters throughout time steps as opposed to feedforward networks. The training method uses Backpropagation Through Time (BPTT) to modify shared parameters by accumulating propagated errors during training [O'Shea, 2015]. The structure of RNNs enables them to work effectively on tasks that require processing sequences with time-dependent elements, as illustrated in Figure 3.5.

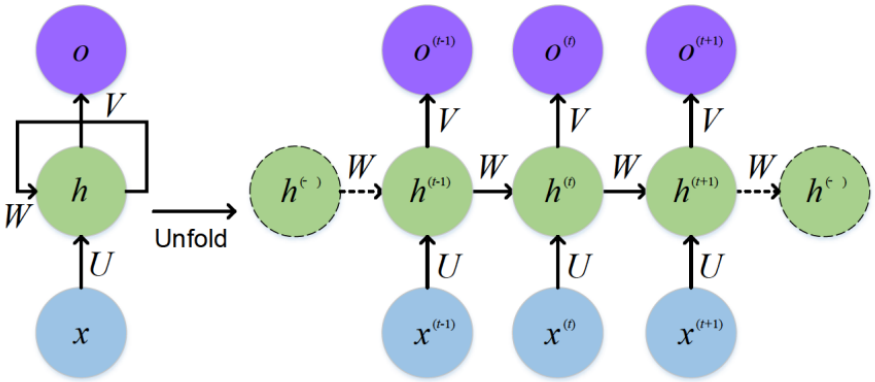

Figure 3.5: Recurrent Neural Networks. [Feng, 2017].

8 Long Short-term Memory Networks (LSTM)

As an advanced type of recurrent neural network, Long Short-Term Memory networks (LSTM) are capable of retaining more information about past values, making them particularly effective in tasks that require remembering long-term dependencies [Praveenkumar, 2024]. This architecture addresses the gradient vanishing problem present in basic RNNs, enabling better performance on complex sequential data tasks, as illustrated in Figure 3.6.

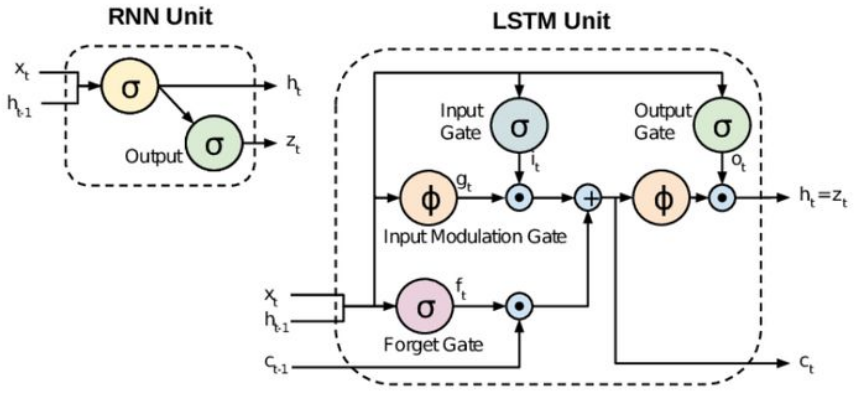

Figure 3.6: Comparison of RNN and LSTM. [Tripathi, 2021].

9 Bidirectional LSTM (BiLSTM)

Bidirectional LSTM (BiLSTM) is a kind of neural network that is commonly used in natural language processing tasks. It operates on sequences using both forward and backward directions simultaneously, enabling it to capture contextual information from both ends [Alkhawaldeh, 2023]. This dual processing makes BiLSTM particularly effective in understanding the relationships between words and phrases in a sentence or text, as shown in Figure 3.7.

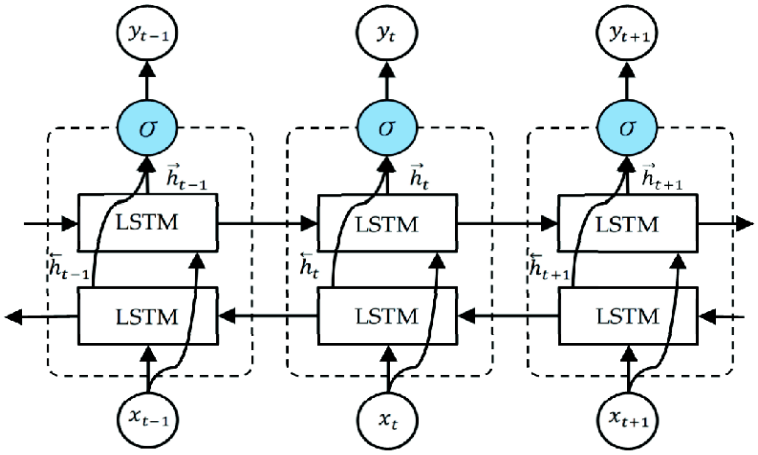

Figure 3.7: Structure of BiLSTM. [Jatavallabha, 2024].

10 Transformers

A group of Google researchers introduced the Transformer architecture in their paper "Attention is All You Need" [Vaswani, 2017], which marked a major breakthrough in sequential computation. Traditional sequence-to-sequence (seq2seq) models had two primary limitations—sequential processing and inefficient performance.

The Transformer model revolutionized these architectures by introducing attention mechanisms that focus processing on the most relevant parts of the input. This approach enables parallel computation over the entire input sequence and leads to significant performance improvements. The general structure of the Transformer architecture is illustrated in Figure 3.8.

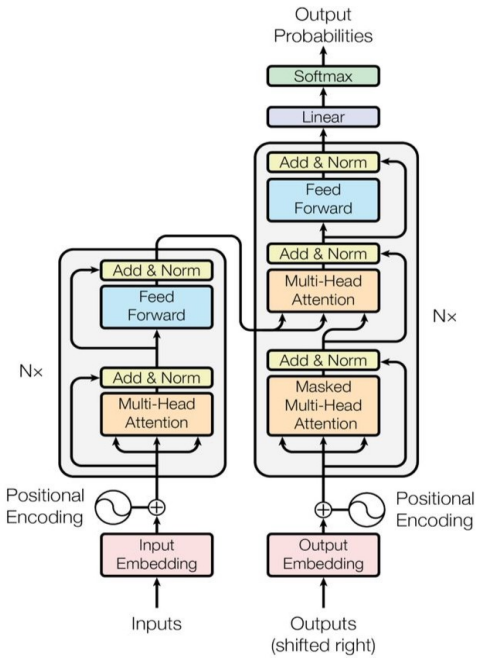

Figure 3.8: Transformer architecture. [Vaswani, 2017].

10.1 Components of Transformer

- **Encoder:** The encoder is composed of a stack identical layers. Each layer has two sub-layers:
  - The first is a multi-head self-attention mechanism.
  - The second is a simple, position-wise fully connected feed-forward network.

  Aresidual connection around each of the two sub-layers is used, followed by layer normalization. That is, the output of each sub-layer is: **LayerNorm(x +Sublayer(x))** where Sublayer(x) is the function implemented by the sub-layer itself. To facilitate these residual connections, all sub-layers in the model, as well as the embedding layers, produce outputs of dimension dmodel = 512.

- **Decoder:** The decoder is also composed of N identical layers (where N = 6 in the original "Attention is all you need" paper). In addition to the two sub-layers in each encoder layer,

the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, residual connections are used around each of the sub-layers, followed by layer normalization. Additionally, the self-attention sub-layer in the decoder stack was modified from the encoder self-attention to prevent positions from attending to subsequent positions. This masking, combined with the fact that the output embeddings are offset by one position, ensures that the predictions for position i can depend only on the known outputs at positions less than i.

- **Scaled Dot Product:** A method used in the calculation of attention scores in the Transformer model. It involves taking the dot product of query and key vectors and scaling the result by the square root of the dimensionality of the key vectors. This helps to prevent the gradient from vanishing or exploding during training. The process is illustrated in Figure 3.9.

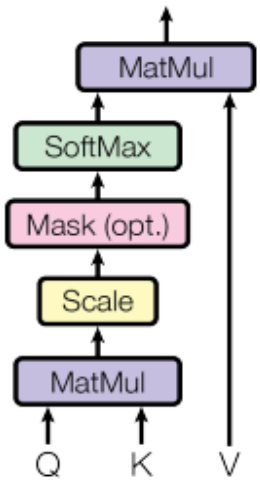

Figure 3.9: Scaled Dot Product. [Vaswani, 2017].

- **Embeddings :** The word embeddings are dense, fixed-length word representations that are based on the distributional hypothesis, that is, the hypothesis that words that occur in the same context have the same meaning. Originally, word embeddings were developed on the basis of the Vector Space Model fromInformation Retrieval, later they have become the combination of statistical language models and neural networks. These embeddings are the

structures that have the complex syntactic and semantic information and are the ones that are now the most important in various NLP tasks. These embeddings are the ones that are really useful in such tasks. They are made by casting the raw word vectors on a coefficient layer, hence the word co-occurrence statistics are used [almeida, 2019].

- **Attention Mechanism :** A mechanism used in neural networks to weigh the importance of different input elements when making predictions. It allows the model to focus on relevant parts of the input, improving its performance on tasks such as machine translation and image captioning.
- **Multi-head Attention:** An extension of the attention mechanism in the Transformer model that allows the model to focus on different parts of the input simultaneously. It achieves this by computing multiple attention scores in parallel using different sets of learnable parameters, then combining the results to obtain a richer representation of the input. This mechanism is illustrated in Figure 3.10.

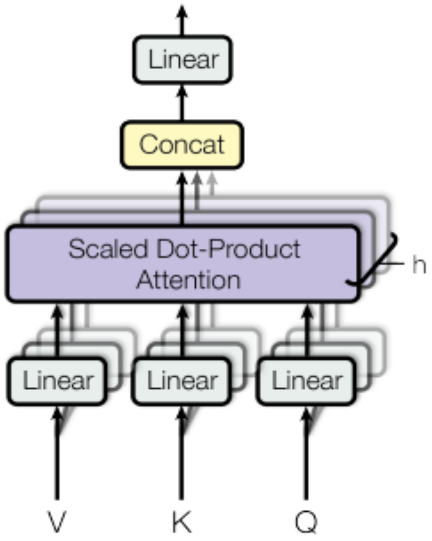

Figure 3.10: Multi-head Attention. [Vaswani, 2017].

- **Positional Encodings :** Given that the Transformer architecture operates without inherent

sequential constraints, a method is required to incorporate positional information within the model. To address this, positional encodings are introduced. These encodings utilize a cosine function to represent the positions of tokens within the sequence. They are then added to the embeddings, thereby preserving the order of the sequence. This approach ensures that the Transformer effectively captures the sequential relationships between tokens, despite its non-sequential nature.

- **Masked Attention :** Masked attention is a technique used in the Transformer model to ensure that during training, the model only focuses on information that it should realistically have access to at any given time. It does this by hiding certain parts of the input sequence from the model's attention mechanism, preventing it from "peeking ahead" into future information. This is crucial for tasks like language modeling, where the model needs to predict the next word based only on the words that came before it. By masking out future information, the model learns to make predictions based solely on past context, leading to more accurate and realistic results.

10.2 Key Transformer-Based Language Models

- **BERT :** which is a product of the research group of Google AI and abbreviated as Bidirectional Encoder Representations from Transformers. It is an advanced Transformer-based model, which notably focuses on the capturing context in which a word appears within a text, both out of words to the left and right of it. This two-way approach is what makes BERT to come up with richer and more context-aware representations, which would have been difficult for previous unidirectional text processing models. BERT is pre-trained on large corpora using two objectives: one of the techniques is masked language modeling , wherein certain words are masked and the model need to predict them and the next sentence prediction , which may present the model with how to relate the sentence pairs. Because of its strength in robust pre-training, it becomes sufficient to fine-tune BERT for a complex of NLP tasks such as question answering, sentiment analysis, and named entity recognition and these achieve state-of-the-art results in many benchmarks[Delvin, 2019].

- **RoBERTa :** is a model for computer vision produced by Facebook AI. It is the next iterative model of BERT and builds on the foundation that BERT established by implementing several critical improvements that improve performance. In contrast to the BERT, however, RoBERTa does not use the next sentence prediction objective, focusing solely on the masked language modeling (MLM) task. The masking working is dynamic, where the masking pattern varies with each training, making the model sturdier. In addition RoBERTa is trained over considerably larger dataset with one of the many features being longer training periods, bigger batch sizes and higher learning rates. Being an improved version of BERT, RoBERTa stands out in all NLP comparison benchmarks showing to be a powerful tool for all text classification, sequence labeling plus other language understanding functions [Liu, 2019].

- **GPT :** GPT (Generative Pre-trained Transformer) functions as a pre-trained model which creates human-oriented texts. The system demonstrates top performance across multiple NLP operations while needing no task-specific training. This model finds use in three main areas: text creation, language translation and writing innovative material [Gillioz, 2020].

- **GPT-2 :** In 2019, OpenAI introduced GPT-2 as an enhanced version of GPT-1. By utilizing a larger dataset and increasing the number of parameters, GPT-2 builds upon its predecessor. GPT-2 achieves impressive performance in various language tasks through its 1.5 billion parameters which surpass the 117 million parameters of GPT-1 by a factor of ten. GPT-2 excels in various language tasks—such as translation and summarization—using raw text input and minimal or no specific training examples. Its evaluation across downstream tasks demonstrates improved accuracy in handling long-range dependencies and predicting sentences

- **GPT-3 :** GPT-3 the Generative Pre-trained Transformer which is its third version released from OpenAI in 2020. The latest version of GPT-3 features 175 billion parameters which exceeding GPT-2 by ten times. The developers spent time building high-quality training data from multiple sources including Common Crawl, WebText, books and Wikipedia which contained roughly 500 billion tokens. The wide range of uses for GPT-3 spans more than three hundred different fields which include productivity applications alongside education use

and creative possibilities and gaming adaptations thus offering developers new potential uses [Zong, 2022].

- **GPT-4 :** The current version of OpenAI language models known as GPT-4 demonstrates advanced capabilities in executing advanced tasks. GPT-4 operates as a major model which utilizes text and image data to generate text results that mimic human writing. Contextual understanding runs through GPT-4 because it uses deep learning principles to follow sequential information patterns. The latest version provides superior performance to preceding models for reasoning tasks along with knowledge retention and coding abilities and intellectual functions. [Waisberg, 2023].

11 Large Language Models

11.1 Definition

Large language models are a type of artificial intelligence algorithm that use deep learning techniques on extensive datasets to understand, summarize, generate and predict new content. They are based on transformer models, which are neural networks that can process data by tokenizing the input and conducting mathematical equations to discover relationships between tokens [Dhaduk et al, 2023].

11.2 How Large Language Models work

Pre-training

Large Language Models (LLMs) leverage advanced deep learning techniques to process massive amounts of textual data, enabling them to comprehend, generate, and manipulate human language across a wide variety of natural language processing (NLP) tasks. In the pre-training phase, large language models (LLMs) are exposed to massive volumes of text collected from diverse internet-based sources, including books, publications, and websites. This stage allows the models to learn natural language patterns, covering grammar, syntax, and semantic relationships. Depending on the model architecture, different training strategies are applied, such as predicting the next word

in a sequence. Throughout the process, the model continuously updates its internal parameters (weights) to improve accuracy and generate contextually meaningful text. Pre-training is highly resource-intensive; for example, the training of GPT-3 alone is estimated to have cost over $4 million.

Fine-tuning

The next stage following pre-training involves LLMs receiving fine-tuning operations with a specific task-ready dataset. This phase applies supervised learning techniques for the model to absorb labeled examples demonstrating the wanted outputs. The adjustment through fine-tuning enables models to use their pre-trained knowledge for dealing with specific tasks which include translation and summarization as well as sentiment analysis. The model's parameters receive updates through methods like gradient descent and backpropagation for boosting its performance level.

In-context learning

In-context learning is a powerful capability of modern language models, allowing them to perform new tasks simply by being given a few examples during the input prompt. Without the need for explicit training or adjusting the model's parameters, the system can generalize from the context provided. For instance, if a model is given a few example sentences labeled as positive or negative in sentiment, it can accurately predict the sentiment of new, unseen sentences. This approach enables the model to behave as if it has learned the task, even though it was never trained specifically for it. In-context learning plays a key role in the flexibility and adaptability of large language models across a wide range of natural language processing tasks.

11.3 Natural Language Processing (NLP) in HealthCare

Modern medicine relies heavily on Natural Language Processing (NLP) because it allows machines to analyze vast amounts of unorganized clinical text within electronic health records along with radiology reports and discharge summaries. The application of NLP enables researchers to extract valuable insights from clinical data which supports activities including risk assessment for patients and adverse event recognition and decision support systems and customized healthcare services. As

standard healthcare infrastructure adopts this technology it helps to create faster and more precise diagnostic procedures together with treatment methods [Wang et al, 2021].

11.4 Challenges and limitations in NLP

The core function of NLU or Natural Language Understanding serves numerous NLP operations especially the implementation of nlp rules. However, Current NLU algorithms battle to achieve complete understanding of natural language because of its multifaceted complexity patterns. The rules of natural language usage expand beyond standard conventions because people successfully communicate by interpreting hidden meanings from incomplete or faulty statements. The flexible nature of language presents machines with a substantial computational obstacle because only expertise within natural language understanding together with environmental knowledge leads to understanding words. The training and replication of NLP models prove difficult mainly because deep learning systems need highly customized adjustments for long training periods on modern GPUs. Custom programming creates obstacles for researchers to easily reproduce or use their work. Builder techniques typically fail to provide exact contextual details so analysts need to understand class classifications while sometimes requiring manual data adjustment at increased expenses. Model selection is also labor-intensive. Model development can be improved through cognitive science and neuroscience principles that enable the integration of human language processing methods [Ray et al, 2023] And there are other challenges and limitations including:

- **Lack of sufficient training data :** Effective deep learning models for natural language processing and natural language understanding tasks need access to superior quality labeled datasets. The process of obtaining suitable datasets remains challenging because languages which have poor available resources create additional difficulties. Building datasets which represent multiple languages with different texts across various domains requires lengthy time commitment and lots of effort [Deva, 2023].

- **Low resource languages :** The creation of successful NLP solutions for languages which do not have adequate training data presents a major challenge. The need for NLP models to embrace diverse languages remains essential because it leads to global accessibility alongside

inclusivity standards [Magueresse et al, 2020].

- **Ambiguity and comprehension of context :** Human language is inherently ambiguous, with words often possessing multiple meanings that vary based on the context in which they are employed. To comprehend context effectively, it is essential to not only consider the words directly preceding and following a specific term but also to analyze the broader context of the discourse [Deva, 2023].

- **Privacy and Ethical Concerns:** Because NLP and NLU systems handle users' sensitive information their ethical applications and privacy concerns become necessary to address. Some personal information requires discussion through virtual assistants or customer service bots as well as chatbots. It remains essential to maintain data security and adhere to ethical standards and implement safe authorization systems. The development of NLP and NLU systems faces a constant challenge between providing assistance and securing user privacy through proper protection [Deva, 2023].

- **Bias and Fairness :** The significant concern about bias exists in NLP and NLU technologies. The models tend to learn and spread biases from their training datasets producing invalid yet biased results. The elimination of bias together with the achievement of non-biased results must become a critical priority for NLP and NLU models that handle applications such as sentiment analysis and automated content moderation and hiring procedures. Scientists continue their work to find detection and reduction methods for prejudice [Deva, 2023].

12 Evaluation metrics

Evaluation metrics play a vital role in evaluating the effectiveness of machine learning models by providing quantitative measures that aid in model selection and hyperparameter tuning [Pretnar, 2022]. The choice of metric depends on the specific task at hand, and selecting the appropriate metric is essential for accurately interpreting the performance of models.

12.1 Multi-label Classification metrics

In classification tasks, where the output is a discrete label, common evaluation metrics include:

1. Confusion matrix A table used to evaluate the performance of a classification model, displaying true positives, true negatives, false positives, and false negatives [GeeksforGeeks, 2025]. In both dimensions, the class instances are categorized as follows:

- **True positive (TP):**When a class is predicted as true and is indeed true in reality.
- **True negative (TN):** When a class is predicted as false and is indeed false in reality.
- **False positive (FP):** When a class is predicted as true but is actually false.
- **False negative (FN):** When a class is predicted as false but is actually true.

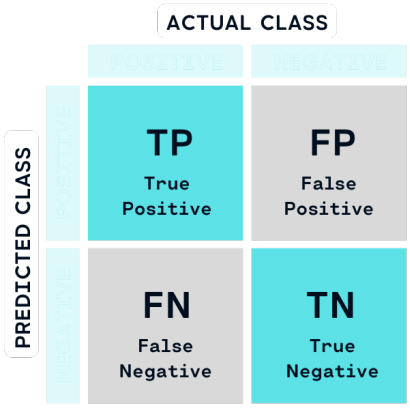

Figure 3.11: The basic structure of a confusion matrix. [AiDevelopper, 2024].

2. Area Under Cueve (AUC) The AUC (Area Under the ROC Curve) represents the area under the ROC Curve, always ranging between 0 and 1, similar to TPR and FPR. The goal is to maximize this area to achieve the highest TPR and lowest FPR for a specific threshold. AUC possesses characteristics like threshold invariance and scale invariance, ensuring that the metric is

not influenced by the chosen threshold or the scale of probabilities. These properties make AUC a valuable metric for assessing binary classifiers, allowing for comparison without the need to consider the classification threshold.

3. Accuracy The ratio of correct predictions to the total number of input samples. [Hossin, 2015].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

4. Precision The ratio of true positives to the sum of true positives and false positives [Hossin, 2015].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.2}$$

5. Recall, or sensitivity The ratio of true positives to the sum of true positives and false negatives [Hossin, 2015].

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.3}$$

6. F1 Score The harmonic mean of precision and recall, providing a balanced measure of both [Hossin, 2015].

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

7. Exact Match Ratio Exact Match Ratio, also known as subset accuracy, measures the percentage of samples that have all their labels predicted correctly.

$$\text{Exact Match Ratio} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[Y_i = \hat{Y}_i] \tag{3.5}$$

where Y_i and $\hat{Y}_i$ are the sets of true and predicted labels for instance i, and $\mathbf{1}$ is the indicator function.

12.2 Loss Functions

1. Categorical Cross-Entropy Loss The measurement of distance between predicted probability distribution and true distribution serves as the method to calculate the Categorical Cross-Entropy. It is calculated as:

$$\text{CE} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{3.6}$$

where:

- C is the number of classes,
- y_i is the true probability distribution (one-hot encoded vector for the true class),
- $\hat{y}_i$ is the predicted probability for class i.

2. Binary Cross-Entropy (BCE) Loss for Multi-Label Classification When performing multi-label classification tasks the BCE Loss functions by treating the outcome categories as two-part binary datasets. It is calculated as:

$$\text{BCE Loss} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C}[y_{ic}\log(\hat{y}_{ic}) + (1-y_{ic})\log(1-\hat{y}_{ic})] \tag{3.7}$$

where:

- n is the number of samples,
- C is the number of classes,
- y_{ic} is the true label for sample i and class c (0 or 1),
- $\hat{y}_{ic}$ is the predicted probability for sample i and class c.

3. Focal Loss Focal Loss is a modified cross-entropy loss function designed to address class imbalance by down-weighting easy examples and focusing training on hard negatives [Lin, 2017].

$$\text{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{3.8}$$

where p_t is the model's estimated probability for the true class, α_t is a weighting factor for class balance, and γ is the focusing parameter that adjusts the rate at which easy examples are down-weighted.

4. Hamming Loss Hamming Loss evaluates how many labels are incorrectly predicted. It is especially useful in multi-label classification tasks.

$$\text{Hamming Loss} = \frac{1}{n \cdot L} \sum_{i=1}^{n} \sum_{j=1}^{L} \mathbf{1}[y_{ij} \neq \hat{y}_{ij}] \tag{3.9}$$

where n is the number of samples, L is the number of labels, y_{ij} is the ground truth, and $\hat{y}_{ij}$ is the predicted label[Sckit, 2022].

13 Conclusion

Deep Learning has transformed AI, enabling breakthroughs in image recognition, natural language processing, and time-series forecasting. Key architectures like RNNs, LSTMs, BiLSTMs, and Transformers have advanced our ability to process sequential data efficiently. Transformer-based models such as BERT, RoBERTa, and GPT have set new performance standards in language tasks. These innovations highlight the dynamic and evolving nature of deep learning, equipping us to tackle complex real-world problems and drive future AI advancements.

Chapter 4

OPTICAL CHARACTER RECOGNITION (OCR)

1 Introduction

OCR stands as a fundamental technology used for converting printed documents into digital forms. it provides the functionality which turns scanned images together with documents into an electronic file that enables searching and editing. OCR is widely used across various industries to automate data entry processes, reduce human errors, and enhance overall efficiency. Additionally, it is often integrated with artificial intelligence along with machine learning techniques to extract data from images for use by intelligent models. Digital transformation between many fields depends on Optical Character Recognition as an essential technology which continues to be a key technology driving progress in many fields.

2 Definition

OCR represents the collection of methods that allow computers to identify and process text found in scanned documents or images. The system reads character shapes to turn them into computer-

readable text by analyzing the shapes and structures of characters through its operation. The first appearance of OCR during the 1950s restricted its usefulness because of its narrow character recognition capabilities and unknown accuracy standards. However, with advancements in computer vision and the integration of deep learning algorithms, OCR technology received major improvements through deep learning algorithms including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) models which boosted its accuracy along with its versatility. The current version of OCR technology supports various fonts and languages to reach recognition accuracy levels exceeding 99% [Wang et al, 2023].

3 The Brief History of OCR Technology

Since the 1920s OCR technology started with its initial use for mail sorting systems and bank check decoding processes. With GISMO technology from the 1950s and subsequently ICR and MICR technologies developed in the 1960s and 1970s organizations gained the ability to convert printed text into machine code and recognize handwritten and banking text. During the 1980s through the 1990s digital imaging and machine learning programs brought OCR technology into widespread commercial availability. The research community achieved improved accuracy levels through Tesseract OCR technology together with deep learning methods during the 2010s. Today, OCR is widely used in industries like logistics, insurance, banking, and real estate for automating document processing [Tripathi, 2025].

4 How does OCR work?

The OCR engine or OCR software works by using the following steps [OCR, 2025]:

4.1 Image acquisition

A scanner reads documents and converts them to binary data. The OCR software examines the captured image through analysis to identify light areas as background and dark areas as text content.

4.2 Preprocessing

The OCR software commences by cleansing the image before error removal allows the document to be ready for interpretation. These are some of its cleaning techniques:

- Deskewing or tilting the scanned document slightly to fix alignment issues during the scan.
- The process includes despeckling digital image spots while also removing text image spots and text edge smoothing.
- Cleaning up boxes and lines in the image.
- Script recognition for multi-language OCR technology.

4.3 Text recognition

Two main software processes for text recognition within OCR software operate as pattern matching and feature extraction.

- **Pattern matching :** The OCR pattern matching process achieves its best results by comparing glyphs with pre-stored templates under conditions of matched font types and sizes for scanned standardized documents.
- **Feature extraction**The analysis process divides glyph elements into distinct features which include lines together with closed loops followed by line direction and line intersection evaluation. It then uses these features to find the best match or the nearest neighbor among its various stored glyphs.

4.4 Postprocessing

The system transforms analyzed text data into digital file format after conclusion of analysis. Some OCR systems can create annotated PDF files that include both the before and after versions of the scanned document.

Text extraction using Optical Character Recognition (OCR) plays a crucial role in automating the processing of scanned documents by converting unstructured visual data into machine-readable

text. This process is essential for downstream tasks such as document classification and information retrieval. The workflow of OCR-based text extraction is illustrated in Figure 4.1.

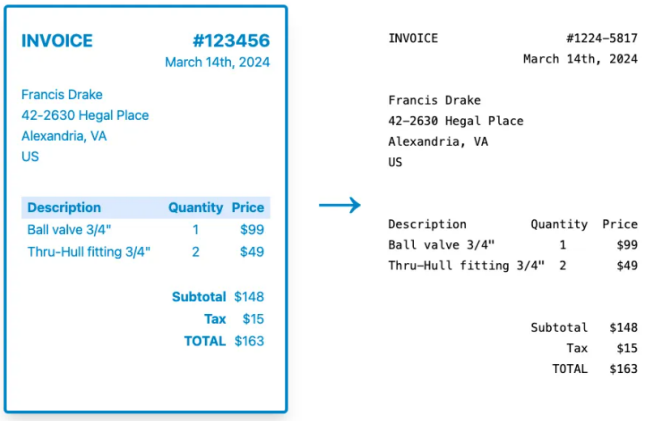

Figure 4.1: Information Extraction from documents. [Parseur, 2025].

5 OCR challenges

- **Fonts and handwriting :** The combination of various types of fonts with unclear handwriting pose challenges to OCR accuracy.
- **Image quality :** OCR shows sensitivity to the quality of images used for text recognition operations. so, Poor scan can affect the accuracy of text recognition.
- **Multilingualism :** The process of recognizing text from different language sources represents a challenge because languages contain unique characteristics and variations of fonts.
- **Accuracy :** Complex documents along with poor quality material tends to reduce the accuracy of text recognition systems.
- **Formatting and structuring :** The recognition of text-based formatting elements including tables, columns, font sizes, or text alignments can be a challenge [Horn et al, 2023].

6 Conclusion

OCR has become a key tool for turning printed or handwritten text into digital data. It has evolved from simple character recognition to advanced systems powered by deep learning. Despite ongoing challenges such as handwriting variability and poor image quality, modern OCR systems have become significantly faster, more accurate, and widely applied across a range of practical domains.

Part II

State of the art

Chapter 5

State of the art

1 Introduction

Predictive medicine focuses on anticipating and mitigating the onset of diseases by leveraging artificial intelligence to analyze large volumes of medical data, uncover patterns, and enable proactive healthcare interventions. In this context, clinical documents such as patient reports, diagnostic summaries, and scientific abstracts represent a valuable source of unstructured information requiring intelligent processing techniques.

This chapter comprehensively explores the current advancements and methodologies in intelligent document analysis and multi-label classification for clinical texts. Emphasis is placed on recent state-of-the-art approaches that integrate deep learning architectures for effective text extraction, semantic understanding, and accurate label prediction in complex, domain-specific settings. These techniques are essential for transforming unstructured clinical narratives into structured, meaningful information, thereby enhancing clinical decision-making and supporting more effective and data-driven medical practices.

2 Related Work

2.1 Advanced Techniques in Document Text Extraction: From Data to Implementation

2.1.1 Used datasets for Text extraction

The success of modern Optical Character Recognition (OCR) systems is largely influenced by the quality, diversity, and scale of the datasets used during training and evaluation. These datasets vary by content type (handwritten, printed, scene text), complexity, language, and annotation level. This section outlines key datasets employed in recent OCR research, each playing a vital role in shaping the capabilities of state-of-the-art models. In this section, we will highlight the most commonly utilized datasets for tasks related to Text extraction.

1. Large-Scale Pretraining Dataset : A large-scale pretraining dataset was assembled by extracting **684 million printed text lines** from **2 million digital PDFs**. To improve performance on handwritten content, **17.9 million synthetic handwritten text lines** were created using **5,427 fonts** and a text generation tool. Additionally, **3.3 million printed receipt text lines** and **16 million scene text images** from synthetic datasets were included. For evaluation, popular datasets such as a receipt dataset, a handwriting benchmark, and several scene text datasets were used [Li et al, 2022].

2. Mixed-Mode Dataset for Printed, Handwritten, and Scene Text : A balanced mix of datasets was used to train a model capable of handling various text modalities. For handwritten text (IAM), the well-known English handwriting dataset was included with **747 training** and **336 testing** samples. For printed receipts (SROIE), **626 training** and **361 testing** examples were used. Scene text benchmarks included multiple datasets such as IC13, IC15, IIIT5K, SVT, SVTP, and CUTE, covering thousands of cropped words or lines. This diversity supported robust learning across handwriting, printed forms, and real-world images [Chang et al, 2025].

3. Custom Medical Handwriting Dataset : A custom dataset was constructed from **1,050 medical forms** manually filled out in English and German. The scanned forms yielded **3,850 English words** and **1,434 German words**, totaling approximately **14,600 characters**. The dataset includes three main types of content: **handwritten text**, **printed checkboxes**, and **page numbers**. All elements were manually annotated using a Python-based labeling tool, ensuring accurate ground truth for OCR tasks in healthcare settings [Zaryab et al, 2023].

4. Unstructured Medical Document Images : A collection of real-world medical reports was assembled using **PDF scans and mobile photographs**. To maintain uniformity in the preprocessing pipeline, all images were converted to `.jpg` format. Although this dataset lacks fine-grained annotations, it represents a realistic and challenging setting for text extraction systems working with noisy, heterogeneous input [Malashin et al, 2024].

5. Lightweight OCR Dataset with Scene and Synthetic Images: This dataset includes a total of **97,000 images** for text detection, comprising **68,000 real-world scene images** and **29,000 synthetic images**, sourced from public datasets such as LSVT, RCTW-17, and MTWI 2018. For text recognition, the dataset contains **17.9 million images**, with a validation set of **18,700 images**. It includes both **1.9 million real images** and **16 million synthetic images**, featuring diverse conditions like rotated text, noisy backgrounds, and vertical layouts. Additionally, **300 real application images**—including contracts, licenses, and tickets—are included to represent practical document scenarios [Du et al, 2021].

6. IIT-CDIP, FUNSD, and SROIE: The **IIT-CDIP dataset** contains over **11 million scanned documents**, including both images and extracted text, along with basic metadata. The **FUNSD dataset** includes just under **200 annotated forms**, where thousands of words are labeled and organized into categories such as headers, questions, answers, and other form content. The **SROIE dataset** is composed of nearly **1,000 scanned receipts**, each annotated with key fields like company name, date, and total amount [Xu et al, 2020].

8. Transfusion Reaction Reports Dataset : A curated dataset of **387 validated transfusion reaction reports** was collected from a **German university hospital** over an **eight-year period (2017–2024)**. Each report is a scanned form containing **checkbox selections**, **handwritten annotations** (in 10% of cases), and **physical stickers**. These reports cover **488 transfused blood products**, averaging **1.26 products** and **3.9 findings** per case. The documents reflect a **balanced gender distribution** and evolving form structures over time. Gold-standard annual summaries, manually compiled for regulatory reporting, were used to validate extraction methods [Schaffer et al, 2025].

Table 5.1 gives an overview of the main datasets used for training and testing OCR systems. It shows a wide variety of text types from large synthetic collections of printed and handwritten text to real scanned documents and handwritten medical forms. Some datasets, like the OCR Pretraining sets, are huge and synthetic, helping models learn the basics. Others, such as the STR benchmarks and medical reports, offer real-world challenges that test how well models work on actual text images. Altogether, these datasets cover a broad range of scenarios, helping researchers build OCR tools that can handle everything from neat printed pages to messy handwriting and complex scene text.

Dataset Name	Type	Size	Content
OCR Pretraining (Printed)	Printed (Synthetic)	684M (train)	Extracted PDF text
OCR Pretraining (Handwritten)	Handwritten (Synthetic)	17.9M (train)	Synthetic handwriting using 5,427 fonts
Scene Text (Synthetic)	Scene Text	16M (train)	MJSynth, SynthText
Handwriting Dataset	Handwritten	747 / 336	English handwriting
Receipt Dataset	Printed	626 / 361	Printed receipts
IC13, IC15, IIIT5K, SVT, SVTP, CUTE	Scene Text	Various	Scene Text
Medical Forms (Handwritten)	Mixed	1,050 forms	English/German handwritten + checkboxes
Medical Report Images	Scanned/Photographed	/	Real scans and photos in .jpg format
Lightweight OCR Detection	Mixed	97K / 500	Scene and synthetic images
Lightweight OCR Recognition	Mixed	17.9M / 18.7K	Real + synthetic for recognition
SROIE, FUNSD, CORD, RVL-CDIP	Mixed Scanned/Photographed	412K (train)	Forms, receipts, reports
Clinical reports Dataset	Scanned Medical Forms	3,615 reports	Transfusion reaction reports with checkboxes (2017–2024), scanned from a German university hospital

Table 5.1: Overview of commonly used datasets for OCR training and evaluation.

2.1.2 Data Preprocessig techniques for Text Extraction

Text extraction from complex documents, whether scanned, photographed, or synthetic, requires careful pre-processing to ensure clean, structured input for downstream OCR tasks. Various studies have proposed tailored techniques depending on the context and data type. [Li et al, 2022], split the image into small patches, which the encoder linearly embeds and uses to produce many visual features. Another study [Chang et al, 2025] takes a unified route across different text types handwritten, printed, and scene text by resizing all images, slicing them into equal patches, and embedding positional information to match the needs of a Vision Transformer. In a medical form

setting [Zaryab et al, 2023], preprocessing begins with identifying text regions using YOLOv5 [1] before cropping and resizing them for OCR input, using consistent dimensions and character length limits to fit models like Gated-CNN-BLSTM and TrOCR. A separate effort [Malashin et al, 2024] focuses on improving image quality by converting all inputs to .jpg and evaluating sharpness with filters like Laplacian[2] and Sobel[3], while manually tagging issues like blur, lighting, and handwriting presence to support tools like PyTesseract and EasyOCR. another contribution [Du et al, 2021] combines real-world images with synthetic ones that simulate complex layouts and noisy conditions, applying contrast adjustments and format normalization to enhance the model's generalization in both detection and recognition tasks. Finally, Another study [Xu et al, 2020] Preprocessing starts by cleaning up the document images—getting rid of noise, fixing any tilted pages, and boosting the contrast so the text really stands out. After that, the images are resized or normalized so everything looks consistent. Next, we identify the areas where the text sits and note their exact positions, adjusting these coordinates to fit a common scale. This way, we keep the layout details clear, which helps a lot when it's time to pull out the actual text later on. Together, these works show just how important and varied preprocessing strategies can be when building robust OCR systems.

[1] YOLOv5 is a real-time object detection model based on a convolutional neural network (CNN), commonly used for text region detection in natural scenes and scanned documents. It achieves high accuracy and speed using techniques like mosaic data augmentation and anchor-free detection.

[2] Laplacian filter is a second-order derivative operator used to highlight regions of rapid intensity change in images, making it effective for detecting edges and assessing sharpness.

[3] Sobel filter is a gradient-based edge detection operator that computes image derivatives in horizontal and vertical directions, helping to identify and enhance edges for clarity evaluation.

Table 5.2 summarizes key preprocessing strategies adopted in recent research, highlighting their context, methodology, and intended use.

Study	**Technique**	**Description**
[Li et al, 2022]	Splitting images	split the image into small patches, which the encoder linearly embeds and uses to produce many visual features.
[Chang et al, 2025]	Resizing and Positionnal	Resize all images, slice into patches. Add positional embeddings for Vision Transformers.
[Zaryab et al, 2023]	Text Region Detection	Use YOLOv5 to detect and crop text zones. Normalize image size and character limits for OCR models.
[Malashin et al, 2024]	Image Quality Enhancement	Convert to .jpg, apply Laplacian and Sobel filters. Tag issues: blur, lighting, handwriting presence.
[Du et al, 2021]	Contrast Enhancement & Format Normalization	Applying contrast adjustments and format normalization.
[Xu et al, 2020]	Document Cleanup & Layout Preservation	Cleaned images by removing noise, de-skewing, and enhancing contrast. Resize and align text regions while preserving layout.

Table 5.2: Preprocessing Techniques for Text Extraction

2.1.3 Text Extraction Techniques and Approaches

Recent advancements in text extraction have led to the development of increasingly accurate and robust techniques, combining deep learning with traditional OCR methods. This section highlights key approaches that represent the current state of the art in the field.

1. [Li et al, 2022] Li et al present a novel approach for optical character recognition (OCR) called TrOCR, which uses a transformer encoder-decoder architecture and combines a Vision Transformer (ViT) for image encoding and a pre-trained RoBERTa model for text decoding. Therefore, this process begins by splitting the image into small patches, which the encoder linearly embeds and uses to produce many visual features. When the neural network receives the results from the encoder, which generates the corresponding text sequence. First, the encoder is trained using large datasets of paired images and text (from the M4C OCR dataset) to learn visual-linguistic representations; then, the model is fine-tuned on real OCR datasets such as IAM for handwritten text and SROIE/SST for printed documents. TrOCR eliminates the need for complex OCR pipelines by using end-to-end learning that generalizes well across printed and handwritten texts. It achieves state-of-the-art results, including a CER of 2.61% and WER of 6.75% on the IAM dataset. Training includes Adam optimizer with weight decay, a linear scheduler with warm-up, and data augmentation for robustness. Both base and large variants of ViT and RoBERTa are tested, showing TrOCR's ability to capture visual and linguistic features effectively. Consequently, recognition performance is high and this method can manage a wide variety of documents and identify them successfully.

2. [Chang et al, 2025] Chang et al. present DLoRA-TrOCR, a parameter-efficient OCR framework tailored for complex mixed-text scenarios by integrating DoRA into the encoder and LoRA into the decoder of a pre-trained TrOCR model. Addressing challenges such as font diversity, variable layouts, and natural scene backgrounds, the model leverages a custom dataset combining printed, handwritten, and street-view texts. Built on TrOCR's ViT-based image encoder and RoBERTa-based text decoder, the method incorporates PEFT techniques to drastically reduce trainable parameters from 333.9M to just 2M (0.6%) and requires 30% less GPU memory, without degrading performance. Running on several tough benchmarks, DLoRA-TrOCR performs better

than several leading OCR models and scores nearly 85% on a dataset with both plain and mixed types of text. The optimal configuration, using DoRA in the encoder and LoRA in the decoder, achieved the best balance between efficiency and recognition quality with a CER of 5.42 and an F1-score of 85.07, highlighting the importance of efficient image encoding in robust text extraction.

3. [Zaryab et al, 2023] Zaryab et Chuen. focuses on developing an OCR system tailored for medical documents using deep learning techniques. For the text extraction process, two architectures were evaluated: a lightweight Gated-CNN-BLSTM and a more advanced Transformer-based model, TrOCR. the Gated-CNN-BLSTM architecture uses 11 convolutional layers, including gated layers and two bidirectional LSTM layers, allowing it to model a sequence with just 820,000 parameters. It uses CTC loss for decoding sequences and supports a 99-character multilingual charset. On the other hand, the leveraging of TrOCR means using a ViT in its encoder and a BERT-like decoder that has been fine-tuned with medical records. The experiments showed that while the Gated-CNN-BLSTM achieved a respectable Character Error Rate (CER) of 9%, the TrOCR model delivered better performance with a CER of just 6%, making it the preferred choice for high-accuracy text recognition in complex medical forms.

4. [Malashin et al, 2024] Ivan et al. presented an approach that combines manual annotation and automated tools to extract structured data from a set of 2041 real-world Russian medical reports, which were acquired via scans or smartphone photos. The authors began by categorizing each document according to visual characteristics such as blur, brightness, skew, and the presence of handwriting to analyze their impact on extraction performance. Experienced researchers reviewed and annotated the sample, identifying tax identification numbers, license numbers, dates of treatment, payments made, the money involved and organization names. The extracted annotations showed a high level of agreement (F1-score = 0.93), validating the dataset's reliability. Optical Character Recognition (OCR) was applied using both PyTesseract[4] and EasyOCR[5] engines to

[4]PyTesseract is a Python wrapper for Google's Tesseract-OCR Engine, commonly used for text extraction from images.

[5]EasyOCR is an open-source OCR library that supports multiple languages and scripts, based on deep learning models.

convert visual documents into text. For entity recognition, they employed Russian-language NLP libraries like Natasha and PullEnti to extract structured fields. To quantify document quality and its correlation with extraction performance, the authors calculated 14 image-level and text-level features, such as brightness, entropy, number of OCR-detected characters, and skew angle. A Pearson correlation analysis revealed strong links between certain features like brightness and entropy (r = 0.75), and skew angle and text density (r = 0.52) highlighting the importance of visual quality in accurate data extraction. Finally, they introduced a genetic algorithm to dynamically optimize OCR parameters in order to maximize F1-scores for entity extraction tasks, showing that adaptive tuning significantly improved downstream information retrieval performance compared to static configurations.

5. [Du et al, 2021] Du et al. focus on making text extraction in OCR systems both smarter and lighter. For detecting text, they use a clever teamwork approach called Collaborative Mutual Learning (CML)[6], where two compact models based on MobileNetV3[7] learn side by side, helping each other improve, while also taking guidance from a stronger, fixed teacher model (ResNet18)[8]. This setup helps the lightweight models become more accurate without adding extra complexity. When it comes to recognizing the text, they switch to a faster and more efficient backbone called PP-LCNet[9]. On top of that, they use something called Unified Deep Mutual Learning (UDML)[10], where several recognition modules (decoders) share a single encoder and learn together by exchanging knowledge throughout the training. To make the model even more robust, they apply a simple but effective technique called CopyPaste[11] this means mixing text from different images to create new training samples, helping the model deal with more variety. In real tests, these methods pay

[6] A training method where two models learn together and help improve each other by sharing knowledge during training.

[7] A lightweight convolutional neural network designed for efficient performance on mobile and embedded devices.

[8] A more powerful, pre-trained model used to guide smaller models during training by providing better predictions or features.

[9] A lightweight and fast CNN backbone optimized for OCR tasks, developed by Baidu's PaddleOCR team.

[10] A strategy where multiple modules (like decoders) learn together using shared features from one encoder, exchanging knowledge throughout training.

[11] A data augmentation method where text regions from different images are combined to create diverse training samples.

off: the detection system reaches a solid 76.5% F-score on the ICDAR2015 dataset while staying fast and light, and the recognition setup hits 74.8% accuracy with noticeably quicker predictions. Altogether, their work offers a practical and efficient way to improve OCR performance without overloading the system.

6. [Xu et al, 2020] Xu et al. propose a text extraction method that begins with Optical Character Recognition (OCR) using Tesseract, which identifies the words in a document and captures their exact positions using bounding boxes. The model then brings together three kinds of information: the actual text from the OCR, the position of each word on the page (using 2D coordinates), and visual details from the document images processed by Faster R-CNN [12]. By combining these, LayoutLM understands not just what the words say, but also how they're arranged on the page, which is especially useful for reading structured documents like forms and receipts. The model is built on a Transformer architecture similar to BERT, with a base version having 12 layers and about 113 million parameters, and a larger one with 24 layers and 343 million parameters. It's pre-trained using a masked token prediction task with the Adam optimizer, a learning rate of 5e-5, and a batch size of 80, taking roughly 80 hours per epoch on powerful GPUs. When fine-tuned for text extraction, it's trained over 100 epochs with a batch size of 16. Thanks to this approach, LayoutLM achieves an F1 score of 79.27% on the FUNSD dataset, clearly outperforming earlier models like BERT that score around 60-65%. This shows how adding layout and visual information to text can make a big difference in understanding and extracting data from documents.

7. [Schaffer et al, 2025] Henning Schäfer et al. presented an approach that combines a YOLOv8-based detection model[13] with two distinct text extraction strategies to process scanned transfusion reaction reports. The YOLOv8 model, trained on over 10,000 synthetically generated images using Copy-Paste augmentation[14] and layout-aware design, is used to accurately detect checkbox

[12] Faster R-CNN is a deep model for detecting and extracting text regions in documents by treating text blocks as visual objects.

[13] YOLOv8: "You Only Look Once" version 8, a state-of-the-art object detection architecture known for real-time performance and high accuracy.

[14] Copy-Paste augmentation: a data augmentation technique where objects from one image are copied and pasted into another to enrich training data diversity.

regions in the documents. For extracting the associated text labels, the authors implemented two complementary methods: the first relies on PaddleOCR[15] to read text adjacent to each detected checkbox, followed by Levenshtein distance[16] matching to align the outputs with a predefined list of medical categories; the second leverages the vision-language model Pixtral-Large-Instruct-2411[17], which interprets entire checkbox sections using carefully crafted prompts that embed contextual instructions, enabling more robust identification of checked categories, especially in the presence of scan noise or handwritten marks. The VLM-based approach achieved an average accuracy of 92.04%, outperforming the OCR+Levenshtein method, which reached 85.17%..

Table 5.3 provides a comparative overview of recent OCR-based text extraction approaches, focusing on the models, training strategies, evaluation metrics, and datasets used. All studies leverage deep learning architectures—most notably transformer-based models like TrOCR—to address varied challenges in recognizing handwritten, printed, and scene text. Techniques such as parameter-efficient fine-tuning (DLoRA), genetic optimization, and collaborative mutual learning highlight different strategies to improve accuracy and efficiency. Across these works, metrics like CER, F1-score, and recognition accuracy show the effectiveness of these models, with transfer learning commonly applied to enhance performance on specific domains such as medical documents or natural scene text. This comparison emphasizes the diversity and innovation in OCR systems, supporting robust and adaptable text extraction pipelines.

[15] PaddleOCR: an open-source optical character recognition (OCR) system developed by Baidu, designed for robust text extraction from images.

[16] Levenshtein distance: a string metric for measuring the number of edits needed to change one word into another, useful in fuzzy text matching.

[17] Vision-language model: a model that jointly processes visual and textual information, enabling tasks like image captioning, document understanding, and multimodal reasoning.

Paper	Model	Optimization / Tuning	Metrics	Dataset(s)
[Li et al, 2022]	ViT + RoBERTa	Adam, weight decay, scheduler, data aug.	CER: 2.61%, WER: 6.75%	IAM, SROIE, SST
[Chang et al, 2025]	TrOCR (DoRA/LoRA)	Param.-efficient (333M → 2M), low GPU	CER: 5.42, F1: 85.07	Mixed (print, hand., street)
[Zaryab et al, 2023]	Gated-CNN-BLSTM, TrOCR	CTC loss on med. records	CER: 9% / 6%	Medical docs
[Ivan et al, 2024]	EasyOCR, NLP tools (Natasha, PullEnti)	Genetic tuning of OCR params	F1: 0.93 (NER)	2041 Russian med. reports
[Du et al, 2021]	PP-LCNet, U-DML	CML, CopyPaste data augmentation	F1: 76.5% (detection), 74.8% (recognition)	ICDAR2015
[Xu et al, 2020]	LayoutLM + Tesseract	Pretrain (11M), fine-tune (30–100 epochs)	F1: 79.27%	FUNSD, SROIE
[Schaffer et al, 2025]	YOLOv8 + VLM (Pixtral)	CopyPaste aug., prompt-based inference	Accuracy: 92.04%	Transfusion reaction scans (2017–2024)

Table 5.3: Comparison of recent OCR-based text extraction methods, including model type, optimization strategy, evaluation metrics, and datasets.

2.1.4 Discussion

Recent advances in OCR and text extraction have significantly improved the ability to handle complex documents, especially in challenging domains like medical records. Notably, Li et al [Li et al, 2022] introduced TrOCR, a transformer-based model combining a Vision Transformer with a RoBERTa decoder, which achieves impressive accuracy on both handwritten and printed texts by leveraging end-to-end training. Building on this, Chang et al [Chang et al, 2025] proposed a more efficient variant, DLoRA-TrOCR, which reduces model size and resource use while maintaining strong performance across mixed-text scenarios. Similarly, Zaryab et Chuen [Zaryab et al, 2023] demonstrated that transformer architectures fine-tuned on medical data outperform traditional CNN-LSTM approaches in recognizing medical forms. On the practical side, Ivan et al [Ivan et al, 2024] showed the importance of document quality and parameter tuning using a genetic algorithm to optimize OCR settings for real-world Russian medical reports. Du and his team [Du et al, 2021] improved OCR by combining lightweight models through collaborative learning and innovative data augmentation, balancing speed and accuracy. Finally, LayoutLM by Xu et al [Xu et al, 2020] highlights the power of integrating text, layout, and visual cues to better understand structured documents, achieving superior results on form-like data. in a complementary study, Schäfer et al [Schaffer et al, 2025] demonstrated that a VLM-based approach outperforms OCR+Levenshtein in scanned medical forms, especially for faint marks, large category sets, and poor-quality scans. Their model adds robustness to clinical workflows and offers potential for validation and continuity in low-infrastructure contexts. Collectively, these studies, each employing distinct strategies, underscore the critical role of combining advanced model architectures, efficient training techniques, and contextual understanding to significantly enhance the accuracy and robustness of text extraction systems in complex and specialized document domains. Each reviewed study introduced innovative techniques to address the challenges of messy layouts, diverse text types, and domain-specific language, pushing the boundaries of what OCR technology can achieve.

2.2 Advances in Multi-Label Classification for Clinical Texts

2.2.1 Benchmark Datasets and Pre-processing Pipelines

1. Liu et al worked with three medical datasets: the widely used **MIMIC-III** in English, and two real-world hospital datasets in Dutch and French. MIMIC-III includes over **52,000 discharge summaries** and nearly **9,000 ICD-9 diagnosis codes**, making it a standard benchmark in medical AI. The Dutch and French datasets, though not publicly available, were used to test the model on non-English clinical texts. Unlike MIMIC-III, they include all patient documents, resulting in longer and more complex records especially in Dutch, with some cases exceeding **5,000 words**. For preprocessing, the authors applied common NLP techniques such as **text cleaning**, **tokenization**, and **truncation** of long documents (up to 3,500 tokens). They also preserved small numerical values that can carry clinical meaning, like dosage or lab results. To adapt to the language and style of each dataset, they trained custom **word embeddings** using the **word2vec** model. This preprocessing pipeline ensured consistent and meaningful input across all datasets, enabling fair comparisons between English and non-English data [Liu et al, 2021].

2. **The DEFT 2021** dataset includes **275 French clinical cases** annotated with various medical entities like symptoms and diseases, along with attributes such as negation, hypothesis, and MeSH-C labels. Each document contains a set of labels describing the patient's condition, with **multiple labels possible per case** (e.g., "tumor," "immune," "hemic"). Due to the small and imbalanced size of the dataset, the authors enriched it using external terminological resources from the UMLS metathesaurus,[18] extracting French and English synonyms of MeSH concepts.[19] They used CUI mappings,[20] PymedTermino2,[21] and opus-mt-en-fr[22] to build French-only, bilingual, and translated term sets, improving generalization for multilabel classification. The preprocessing pipeline also included cleaning, entity annotation alignment and label aggregation per document to support robust training. [Gerardin et al, 2021].

[18] A big medical database that links terms from many sources to help computers understand medical language.
[19] Medical Subject Headings—standard medical terms used in research databases.
[20] Unique IDs that connect different words for the same medical idea in UMLS.
[21] A Python tool to access and work with medical term databases like UMLS.
[22] An AI model that translates medical terms from English to French accurately.

3. A dataset comprising 79,912 bilingual drug allergy records from Songklanagarind Hospital's EHR, combining Thai and English text. Each entry included free-text descriptions and multiple symptom terms validated by pharmacists. After filtering, 36 common symptoms were retained, with an average of 1.8 labels per record. Preprocessing involved cleaning, punctuation removal, and Thai word segmentation using a bidirectional maximum matching algorithm. The dataset was split into training (80%), validation (10%), and test (10%) sets using stratified sampling to maintain symptom label balance [Chaichulee et al, 2022].

4. This study [Rouabhi et al, 2023] leverages the Ohsumed dataset, a widely used medical corpus composed of over 34,000 abstracts related to cardiovascular diseases, categorized into 23 classes based on the MeSH taxonomy. The dataset originates from the 1991 MEDLINE medical abstracts and serves as a benchmark for multi-label classification tasks in the biomedical domain. For the classification process, a comprehensive preprocessing pipeline was implemented. This included data cleaning and augmentation using the NLPAug library, which enables contextual word-level augmentation by replacing terms using pre-trained BERT embeddings. Two augmentation strategies were tested: a uniform strategy producing two augmented samples per input, and a variable approach generating between one to three augmented texts per instance.

5. The study [Kumawat et al, 2023] draws on two datasets made up of PubMed article abstracts, each labeled with multiple relevant categories. In the case of the Hallmarks of Cancer dataset, only ten top-level classes were used, following conventions found in related studies. The Chemical Exposure Information dataset was also considered. Since both datasets often included structural headings like OBJECTIVE, BACKGROUND, CONCLUSIONS, STUDY DESIGN, and RESULTS, these were removed to ensure cleaner input. Additional text cleaning steps involved converting all characters to lowercase, removing line breaks, joining sentences within each abstract, stripping away punctuation, and tidying up extra spaces. Interestingly, when using transformer-based models, only the section headers were removed no other preprocessing was done—so that these models could take full advantage of the original context present in the raw text. Raw text must be converted into numerical form for deep learning. Traditional methods like BoW and TF-IDF ignore context. Neural models such as Word2Vec, GloVe, and FastText introduced semantic word embeddings but

remain static. BERT and similar models offer dynamic embeddings that adapt word meaning based on context, improving text representation.

6. As described in [Chen et al, 2022], two biomedical multi-label datasets were used: LitCovid BioCreative and Hallmarks of Cancer (HoC). The LitCovid BioCreative dataset includes a total of 33,699 PubMed[23] articles distributed across training (24,960), development (6,239), and testing (2,500) sets, each annotated with seven topic labels such as Case Report, Diagnosis, and Treatment. The HoC dataset contains 1,580 PubMed abstracts annotated with ten hallmark cancer labels and is split into 1,108 training, 157 development, and 315 test articles. All articles in both datasets include titles and abstracts. During preprocessing, section headers were removed, and the text was tokenized with a maximum sequence length of 512 tokens for transformer-based models. For traditional deep learning baselines such as ML-Net[24], character-based inputs were used with a sequence limit of 2,000 characters. The label distributions were preserved across splits to maintain representative learning conditions, and the datasets were publicly sourced to ensure reproducibility.

7. As described in [Yanis et al, 2023], NACHOS and NBDW are two French medical datasets used for language model pre-training. NACHOS is a public corpus built from 24 high-quality French medical websites, containing up to 7.4 GB of diverse health-related texts such as disease descriptions, drug leaflets, scientific reports, and clinical case summaries. A smaller 4 GB version, NACHOSsmall, was created for comparison purposes. NBDW is a private dataset composed of 1.7 million de-identified hospital stay reports from the Nantes University Hospital, totaling 4 GB of clinical text covering a wide range of medical specialties. For both datasets, the text was preprocessed using SentencePiece, a subword segmentation algorithm that avoids the need for language-specific tokenization. A vocabulary of 32,000 subword tokens was trained from the full set of sentences in each dataset, allowing consistent and efficient tokenization for models pre-trained from scratch.

[23] PubMed is a free search engine accessing primarily the MEDLINE database of references and abstracts on life sciences and biomedical topics, maintained by the U.S. National Library of Medicine.

[24] ML-Net is a deep learning framework designed for multi-label text classification, utilizing hierarchical attention mechanisms to improve label prediction performance.

As part of the dataset analysis for clinical document classification tasks, Table 5.4 summarizes the main characteristics of commonly used corpora, including their availability, language, size, document type, and publication year. In parallel, Table 5.5 details the preprocessing steps applied to each dataset, such as text cleaning, tokenization, truncation, and entity annotation, providing a clearer view of the methodological pipeline adopted across studies.

Dataset	Publicly available	Language	Size	Document type	Year
MIMIC-III [Liu et al, 2021]	✓	English	Large	Discharge summaries	2016
Dutch Dataset [Liu et al, 2021]	✗	Dutch	Medium	Full records	2023
French records [Liu et al, 2021]	✗	French	Medium	Full records	2023
DEFT [Gerardin et al, 2021]	✓	French	Small	Clinical cases	2021
Songklanagarind Allergy Dataset [Chaichulee et al, 2022]	✗	Thai + English	Large	Drug allergy reports	2022
Ohsumed [Rouabhi et al, 2023]	✓	English	34,000 abstracts	Medical abstracts	1991
HoC [Kumawat et al, 2022], [Chen et al, 2022]	✓	English	1,580 abstracts	PubMed abstracts	—
LitCovid BioCreative [Chen et al, 2022]	✓	English	33,699 articles	PubMed articles	—
NACHOS [Yanis et al, 2023]	✓	French	7.4 GB	Medical documents	2023
NBDW [Yanis et al, 2023]	✗	French	4 GB	Medical abstracts	2023

Table 5.4: Comparison of key features across benchmark datasets used for clinical document classification.

Dataset	Text cleaning	Tokenization	Truncation	Preserve numeric values	Custom embeddings	Entity annotation
MIMIC-III[Liu et al, 2021]	✓	✓	✓	✓	✓	✗
Dutch records[Liu et al, 2021]	✓	✓	✓	✓	✓	✗
French records[Liu et al, 2021]	✓	✓	✓	✓	✓	✗
DEFT[Gerardin et al, 2021]	✓	✓	✗	✗	✓	✓
Drug Allergy Dataset[Chaichulee et al, 2022]	✓	✓	✗	✗	✓	✗
Ohsumed [Rouabhi et al, 2023]	✓	✓	✗	✗	✓	✗
HoC [Kumawat et al, 2022][Chen et al, 2022]	✓	✓	✓	✗	✓	✗
LitCovid BioCreative [Chen et al, 2022]	✓	✓	✓	✗	✓	✗
NACHOS[Yanis et al, 2023]	✓	✓	✓	✗	✓	✗
NBDW[Yanis et al, 2023]	✓	✓	✓	✗	✓	✗

Table 5.5: Comparison of preprocessing steps applied across clinical datasets.

2.2.2 Multi-Label Classification for Clinical Documents Methods

1. The architecture presented in this study [Liu et al, 2021] is a deep convolutional model specifically designed for multi-label classification of clinical documents. It consists of four main parts: **an input layer** that converts raw text into pretrained word embeddings, **a convolutional encoder** built from multiple residual squeeze-and-excitation (Res-SE) blocks that capture patterns and enhance important **features**, **an attention mechanism** that focuses on the most relevant parts of the text for each label, and an output layer that makes the final label predictions. The model uses **word embeddings** trained using the word2vec CBOW method, with embedding sizes of 100 for English texts and 200 for non-English ones. To ensure efficiency and scalability, the convolutional encoder is preferred over self-attention for handling long documents. The architecture includes four Res-SE blocks, and hyperparameters such as the convolutional filter sizes, the number of output channels, dropout rate, and focal loss parameters were optimized using the Ray Tune library. Training was done using the **Adam optimizer** with a learning rate of 0.00015.

2. This study [Gerardin et al, 2021] proposes a four-step pipeline to classify French clinical documents, starting with a **NER** model that extracts key medical terms by unifying "disease" and "sign or symptom" categories to better handle overlaps and filter out negated, hypothetical, or non-patient references.

Next, a **gender classifier** predicts patient gender using weighted linguistic cues like gendered nouns, adjectives, pronouns, and first names, focusing more on early sentences where patient details usually appear.

The third component is a **multi-label classification model**. It takes the validated terms from the NER step and predicts one or more MeSH-C labels for each term. This model is built on top of pretrained language models such as **CamemBERT (for French)** or **multilingual BERT**. A simple linear layer is added on top for classification. Terms indicating normal or negative exams are filtered out using rules to focus on medical issues.

The final step aggregates all term-level predictions to produce document-level MeSH-C classifications.

Training used different learning rates for transformer and classification layers, optimized with Adam

and learning rate scheduling. The NER model combined **BERT**, **bidirectional LSTM**, and **highway layers**, while the gender classifier used AdaBoost on features.

3. Chaichulee et al. [Chaichulee et al, 2022] formulated the task as a multi-label classification problem to identify symptom terms from bilingual drug allergy descriptions. They explored three main approaches: NB-SVM[25], ULMFiT[26], and BERT-based models. For NB-SVM, they created 36 binary classifiers using a one-vs-rest[27] strategy. Texts were vectorized into document term matrices with unigrams and bigrams, and Naive Bayes log-count ratios[28] were computed before passing the data to linear SVMs, with class imbalance addressed through weighted penalties. For ULMFiT, they used the AWD-LSTM architecture with 400-dimensional embeddings and 1,152 hidden units per layer. They first pre-trained the model on their domain-specific corpus using SentencePiece tokenization, then fine-tuned it with two fully connected layers (100 and 36 units) and applied slanted triangular learning rates and weighted focal loss over 60 epochs, gradually unfreezing layers to stabilize learning. They also fine-tuned several transformer-based models: mBERT, XLM-RoBERTa[29], WangchanBERTa[30], and a custom AllergyRoBERTa[31] trained from scratch on their dataset. These models included two linear layers (768 hidden and 36 output units) and were trained using focal loss with a learning rate of 1e-5 for 30 epochs.

4. According to [Rouabhi et al, 2023], the multi-label classification task was addressed using the BioBERT model, a variant of BERT pretrained on biomedical texts. The model architecture

[25] NB-SVM is a hybrid text classification method that combines Naive Bayes features with a linear Support Vector Machine (SVM) to leverage the strengths of both probabilistic and discriminative models.

[26] ULMFiT (Universal Language Model Fine-tuning for Text Classification) is a transfer learning method that fine-tunes a pre-trained language model, typically based on LSTMs, for downstream NLP tasks.

[27] The one-vs-rest (OvR) strategy involves training one classifier per class, where each classifier distinguishes between samples of one class and all other classes.

[28] Naive Bayes log-count ratios are calculated by taking the logarithm of the ratio of word frequencies in the positive and negative classes, and are used as features in hybrid classifiers like NB-SVM.

[29] XLM-RoBERTa is a multilingual transformer model based on RoBERTa, trained on 100 languages using CommonCrawl data, and designed for cross-lingual NLP tasks.

[30] WangchanBERTa is a RoBERTa-based transformer model trained specifically on Thai language corpora, developed to support Thai NLP tasks.

[31] AllergyRoBERTa is a domain-specific RoBERTa-based transformer model pre-trained from scratch on allergy-related medical text to better capture specialized vocabulary and context.

consists of two main components: the BioBERT encoder, which transforms input text into rich contextual embeddings, and a classification head that predicts multiple labels for each input. To handle class imbalance, the training phase applied the Focal Loss function, which reduces the influence of well-classified samples and emphasizes harder, underrepresented classes. Additionally, class weights were used to further reinforce learning on minority labels. The classification layer was fine-tuned specifically for multi-label output using Hugging Face's BertForSequenceClassification class, adapted for this task. Training was carried out with the Adam optimizer and standard BioBERT parameters 20 epochs, a batch size of 64, and a learning rate of 6e-6 ensuring stable and effective learning throughout the experiments.

5. The study [Kumawat et al, 2023] implemented various deep learning architectures for multi-label classification, including MLP for TF-IDF, Bi-LSTM for word embeddings (GloVe, Word2Vec, FastText, BioWordVec), and fine-tuned transformer models (BERT, SciBERT, BioBERT, PubMedBERT). The annotated datasets were split using a train-validation-test technique: 80% of the data was used for training and 20% for testing, with 10% of the training set reserved for validation. Each model employed a sigmoid-activated output layer to support multi-label outputs. Inputs were limited to 256 tokens using padding and truncation for uniformity.
For TF-IDF and embedding-based models, training was conducted over 50 epochs with early stopping based on validation loss monitored for three consecutive steps, using a batch size of 32. A learning rate of 0.01 with exponential decay scheduling was applied. Multiple optimizers were explored, including RMSprop, Adam, and AdamW. Transformer-based models were trained under the same batch and epoch conditions but used the AdamW optimizer with a learning rate of 6×10^{-6} and a weight decay of 0.01. Training was performed on an AMD Ryzen 7 4800H CPU for non-transformer models and on Kaggle's T4 GPU environment for transformer models.

6. In [Chen et al, 2022], the authors proposed **LITMC-BERT**, a transformer-based multi-label classification model tailored for biomedical literature. The study was conducted on two benchmark datasets: *LitCovid BioCreative* and *HoC*. Several approaches were compared: **ML-Net** (a deep learning model using ELMo embeddings and a label prediction/counting module), **Binary BERT** (one BioBERT model per label with sigmoid output), **Linear BERT** (a single BioBERT model out-

putting probabilities for all labels via sigmoid), and the proposed **LITMC-BERT**, which extends Linear BERT with additional modules. LITMC-BERT uses a shared *BioBERT* backbone, a *Label Module* that combines CLS tokens and label-specific representations via multi-head self-attention and MLP layers, and a *Label Pair Module* that models label co-occurrence through co-attention mechanisms. The model is trained using multi-task learning to optimize both label prediction and co-occurrence detection. All BERT-based models used the same training parameters: maximum sequence length of 512 tokens, batch size of 16, learning rate of 5e-2, and early stopping after 2 epochs. LITMC-BERT additionally used a label pair threshold of 0.4, an auxiliary task weight of 0.25, and MLP layers with 512, 256, and 128 units, respectively, for final classification.

7. [Yanis et al, 2023] Yanis et al introduced DrBERT, a French biomedical language model pre-trained entirely from scratch using the CamemBERT base configuration, which follows the RoBERTa-base architecture with 12 transformer layers, 768 hidden dimensions, 12 attention heads, and approximately 110 million parameters. The model was trained on a Masked Language Modeling (MLM) objective using a custom SentencePiece tokenizer with a vocabulary of 32k subword units. Pre-training was conducted for 80,000 steps, with a batch size of 4,096 sequences of 512 tokens each (2.1M tokens per step), and a learning rate linearly warmed up to 5×10 over 10,000 steps. Training used mixed-precision (FP16) on 128 Nvidia V100 GPUs for 20 hours. Among the evaluated downstream tasks, DrBERT was applied to a multi-label classification task using the MUSCA-DET dataset, where it identified multiple social determinants of health within French clinical sentences.

Table 5.6 summarizes recent approaches to multi-label classification in the medical domain. Transformer-based models like BioBERT, PubMedBERT, and CamemBERT consistently achieved high F1 scores. The results highlight how model choice and training settings directly impact classification performance.

Study	**Model**	**Experiment Parameters & Details**	**Results**
[Liu et al, 2021]	EffectiveCAN (Res-SE + Attention)	Word2vec embeddings (100/200 dim), Adam (lr=0.00015), Ray Tune for hyperparams	F1 Score: 0.669
[Gerardin et al, 2021]	CamemBERT	Differentiated LR for transformers + Adam optimizer	F1 Score: 0.809
[Chaichulee et al, 2022]	NB-SVM, ULMFiT, BERT-based	SentencePiece tokenization, focal loss, LR=1e-5, 30–60 epochs.	F1 Score: 0.988, Hamming Loss: 0.17
[Rouabhi et al, 2023]	BioBERT	Focal loss + class weights, Adam (lr=6e-6), 20 epochs, batch size=64	F1: 0.96
[Kumawat et al, 2023]	Bi-LSTM, Transformers (PubMedBERT)	50 epochs, batch size=32, early stopping, LR decay, trained on CPU/GPU depending on model	PubMedBERT F1: 0.91, Hamming Loss: 0.03
[Chen et al, 2022]	LITMC-BERT (BioBERT + Label & Co-attention Modules)	Batch size=16, max len=512, early stopping (2 epochs), LR=5e-2, MLP: 512→256→128	F1: 0.9384
[Yanis et al, 2023]	DrBERT (CamemBERT-based 12-layer Transformer)	SentencePiece tokenizer (32k), batch=4,096×512, LR warm-up (5e-5), 80k steps on 128 V100 GPUs	F1: 0.95

Table 5.6: Summary of Models, Experimental Settings, and Results

2.2.3 Evaluation Metrics

Micro-F1 and Macro-F1 Scores: As used in [Liu et al, 2021], Micro-F1 serves as the main metric for multi-label classification by balancing precision and recall across all labels, making it suitable for imbalanced datasets. Macro-F1, on the other hand, evaluates performance equally across all classes, highlighting model behavior on both frequent and rare labels.

AUC (Macro and Micro): Used in [Liu et al, 2021] to assess a model's ability to rank relevant labels higher than irrelevant ones. Macro-AUC treats all labels equally, while Micro-AUC is weighted by label frequency.

General Precision, Recall, F1 Score: Reported in [Gerardin et al, 2021], [Liu et al, 2021], [Chaichulee et al, 2022], [Rouabhi et al, 2023], [Kumawat et al, 2023], [Chen et al, 2022], [Yanis et al, 2023] for evaluating classification models (EffectiveCAN, CamemBERT and multilingual BERT...), with and without fine-tuning, to assess correctness, coverage, and prediction balance.

Exact Match Ratio and Hamming Loss: Following [Chaichulee et al, 2022],[Kumawat et al, 2023] Exact Match Ratio was used as a strict criterion to measure fully correct predictions across all labels. Hamming Loss quantified the fraction of incorrect label assignments, providing insight into partial prediction errors.

Hamming Loss: Reported in [Rouabhi et al, 2023], [Kumawat et al, 2023] as a key evaluation metric for multi-label classification tasks. Hamming Loss measures the fraction of incorrect labels to the total number of labels. Lower values indicate better predictive performance and fewer misclassifications across all labels.

2.2.4 Discussion

The extensive review of techniques and methodologies in multi-label classification of medical documents reveals the remarkable evolution and growing sophistication of models designed to process complex clinical narratives. The exploration of various datasets, including HoC and LitCovid, underscores the importance of high-quality, domain-specific corpora for training robust models capable of capturing nuanced biomedical concepts. Despite growing access to such resources, data limitations remain a significant challenge, particularly in terms of label imbalance and linguistic variability across languages and clinical contexts.

Crucially, the preprocessing pipelines such as entity extraction, token normalization, and input truncation play a foundational role in ensuring consistency and relevance in model training. Techniques like padding, sequence length management, and label filtering help adapt raw clinical text into model-friendly formats. Moreover, feature extraction using pretrained embeddings, whether static (e.g., Word2Vec, FastText) or contextual (e.g., BERT, BioBERT, CamemBERT), has proven critical in capturing semantic detail necessary for accurate multi-label prediction.

In terms of architectural evolution, early pipelines integrated traditional classifiers like SVMs and NB-SVMs for binary decisions across label sets [Chaichulee et al, 2022], while more recent approaches favor deep learning and transformer-based frameworks. Architectures such as those proposed in [Liu et al, 2021] introduced Res-SE convolutional blocks and attention mechanisms for more efficient and scalable document encoding. Others have built multi-step pipelines that combine named entity recognition (NER), gender classification, and document-level label aggregation [Gerardin et al, 2021], showcasing a shift toward modular, task-aware designs. Transformer models, including BioBERT, SciBERT, and PubMedBERT [Kumawat et al, 2023], [Rouabhi et al, 2023] have further advanced performance by leveraging biomedical pretraining, often incorporating focal loss and class weighting to handle label imbalance.

Notably, hybrid architectures such as LITMC-BERT [Chen et al, 2022], which combine multi-head self-attention with label-pair modeling, have improved the capture of inter-label dependencies crucial for handling correlated medical labels. However, these sophisticated models often demand significant computational resources, including GPUs and meticulous hyperparameter tuning, to achieve optimal performance. This complexity extends to training strategies, where techniques

such as learning rate scheduling, layer freezing, and optimizer selection (e.g., AdamW or RMSprop) play a vital role in model effectiveness. Complementing these architectural innovations, DrBERT [Yanis et al, 2023] represents a domain-specific effort to enhance French biomedical NLP. Pretrained from scratch on the NACHOS and NBDW datasets spanning both public and private medical texts DrBERT demonstrates the value of tailoring models to language and domain. Its strong performance on multi-label classification tasks across both datasets (F1-score: 0.95) underscores the importance of aligning model architecture, training strategy, and pretraining corpus with the linguistic and semantic characteristics of the target domain.

In conclusion, while current models showcase increasing capabilities in handling clinical document classification, challenges remain. The dependence on high-quality annotations, sensitivity to input length, and model scalability continue to require attention. Future directions should emphasize the creation of larger, multilingual, and ethically curated datasets, as well as the development of lightweight, interpretable models suited for real-world clinical deployment. Techniques like transfer learning, weak supervision, and synthetic data generation may serve as practical avenues to address existing limitations and move closer to AI-driven decision support in healthcare settings.

3 Conclusion

This chapter has highlighted how recent progress in intelligent document processing and multi-label classification is transforming the landscape of clinical document analysis. By addressing challenges such as heterogeneous layouts, domain-specific terminology, and complex annotation schemes, modern approaches now offer significantly improved performance, adaptability, and scalability. These advances not only streamline the processing of diverse healthcare documents but also enhance the accuracy and depth of multi-label categorization, which is essential for medical decision-making and data-driven research. Ultimately, the convergence of intelligent extraction methods and classification strategies marks a pivotal step toward more automated, reliable, and intelligent management of clinical textual data.

Conclusion

In this comprehensive study, we have delved into the transformative role of Artificial Intelligence (AI) in the field of medical document analysis, specifically focusing on intelligent document processing techniques. Our exploration has highlighted the vital importance of various AI-driven approaches in enabling machines to interpret, extract, and classify complex medical data with greater efficiency, accuracy, and scalability.

An initial focus is placed on deep learning, highlighting its essential role in intelligent document processing by enabling AI systems to effectively handle sequential and contextual data. This is followed by an examination of Natural Language Processing (NLP), which facilitates the semantic understanding and classification of clinical texts, thereby supporting informed medical decision-making. The discussion then turns to Optical Character Recognition (OCR), a key tool for converting unstructured clinical text into structured, machine-readable data.

Finally, our review of recent developments in intelligent document processing and multi-label classification illustrated how advanced AI techniques address key structural and semantic challenges in clinical documents. These methods enhance automation, data reliability, and support early diagnosis and informed medical decision-making. Building on these insights, the literature review also provided a solid theoretical basis for the development of the Automated Prediction of Medical Acts for Cardiovascular Disease from Patient Records System. Key insights from recent advances in deep learning, natural language processing, optical character recognition, and multi-label classification guided the methodological choices. This alignment between theory and application enabled the design of an effective system for predicting medical acts from unstructured clinical data.

Bibliography

AI Developer. (2024, February 14). *Confusion matrix in machine learning: A hands-on explanation* [Accessed: 2025-06-12]. https://www.deeplearningnerds.com/confusion-matrix-in-machine-learni

Alkhawaldeh, R. S., Al-Ahmad, B., Ksibi, A., et al. (2023). Convolution neural network bidirectional long short-term memory for heartbeat arrhythmia classification. *International Journal of Computational Intelligence Systems*, *16*, 197. https://doi.org/10.1007/s44196-023-00374-8

Almeida, F., & Xexéo, G. (2019). Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*. https://arxiv.org/abs/1901.09069

Amazon Web Services. (2025). What is ocr? - optical character recognition explained.

Ateeque, Z. M., & Ng, C. R. (2023). Optical character recognition for medical records digitization with deep learning. *2023 IEEE International Conference on Image Processing (ICIP)*, 3260–3263. https://doi.org/10.1109/ICIP49359.2023.10222038

Chaichulee, S., Promchai, C., Kaewkomon, T., Kongkamol, C., Ingviya, T., & Sangsupawanich, P. (2022). Multi-label classification of symptom terms from free-text bilingual adverse drug reaction reports using natural language processing. *PLOS ONE*, *17*(8), e0270595. https://doi.org/10.1371/journal.pone.0270595

Chang, D., & Li, Y. (2025). DLoRA-TrOCR: Mixed text mode optical character recognition based on transformer. https://arxiv.org/abs/2404.12734

Chen, Q., Du, J., Allot, A., & Lu, Z. (2022). LitMC-BERT: Transformer-based multi-label classification of biomedical literature with an application on COVID-19 literature curation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *19*(5), 2586–2599. https://doi.org/10.1109/TCBB.2022.3167991

Chowdhary, K., & Chowdhary, K. (2020). Natural language processing. In *Fundamentals of artificial intelligence* (pp. 603–649). Springer.

DataScientest. (n.d.). Qu'est-ce que le modèle vgg ? [Accessed: 2025-06-17].

Developers, S.-l. (n.d.). Sklearn.metrics.hamming_loss [Accessed: 2025-06-17].

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186. https://aclanthology.org/N19-1423

Dhaduk, H. (2023). How do large language models work? *Simform*. https://www.simform.com/blog/how-do-llm-work/

Feng, W., Guan, N., Li, Y., Luo, Z., & Chen, B. (2017). Audio visual speech recognition with multimodal recurrent neural networks. *2017 International Joint Conference on Neural Networks (IJCNN)*, 681–688. https://doi.org/10.1109/IJCNN.2017.7965918

Foote, K. D. (2023, July). *A brief history of natural language processing*. DATAVERSITY. https://www.dataversity.net/a-brief-history-of-natural-language-processing-nlp/

GeeksforGeeks. (2025). Metrics for machine learning model. https://www.geeksforgeeks.org/metrics-for-machine-learning-model/

Gérardin, C., Wajsbürt, P., Vaillant, P., Bellamine, A., Carrat, F., & Tannier, X. (2021). Multilabel classification of medical concepts for patient clinical profile identification. *Proceedings of DEFT 2021 (Défi Fouille de Textes)*.

Gillioz, A., Casas, J., Mugellini, E., & Abou Khaled, O. (2020). Overview of the transformer-based models for nlp tasks. In *Proceedings of the international conference on artificial intelligence and data science* (pp. 179–183). Springer.

Gillis, A. S., Lutkevich, B., & Burns, E. (2023). *What is natural language processing (nlp)?* TechTarget. https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP

Gneiting, T., & Vogel, P. (2022). Receiver operating characteristic (roc) curves: Equivalences, beta model, and minimum distance estimation. *Machine Learning*, *111*, 2147–2159. https://doi.org/10.1007/s10994-021-06115-2

Horn Janina, J. S. (2023, June). Ocr technology: Fundamentals, applications, and challenges.

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 1–11.

Hyland. (2025). The power of intelligent document processing (idp) in healthcare [Accessed: 2025-06-23]. https : / / www . hyland . com / en / resources / articles / idp - healthcare % 5C # keybenefitsofidpinhealthcare

Ivan, M., Masich, I. S., Tynchenko, V., Gantimurov, A., Nelyub, V., & Borodulin, A. (2024). Image text extraction and natural language processing of unstructured data from medical reports. *Machine Learning and Knowledge Extraction*, *6*(2), 1361–1377. https://doi.org/10.3390/make6020064

Jatavallabha, A., Gerlach, J., & Naresh, A. (2024). Deciphering air travel disruptions: A machine learning approach [License: CC BY 4.0. Accessed: 2025-06-12]. https://doi.org/10.48550/arXiv.2408.02802

Keita, Z. (2023, November). *An introduction to convolutional neural networks (cnns)*. https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns

Klippa. (2024, May). How intelligent document processing improves healthcare [Accessed: 2025-06-23]. https://www.klippa.com/en/blog/information/idp-in-healthcare/

Kumawat, H., Sharan, A., & Verma, S. (2025). Impact analysis of text representation on biomedical multi-label text classification with deep learning [International Conference on Machine Learning and Data Engineering]. *Procedia Computer Science*, *258*, 3294–3304. https://doi.org/10.1016/j.procs.2025.04.587

Labrak, Y., Bazoge, A., Dufour, R., Rouvier, M., Morin, E., Daille, B., & Gourraud, P.-A. (2023). Drbert: A robust pre-trained model in french for biomedical and clinical domains [HAL Id: hal-04056658]. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*. https://arxiv.org/abs/2304.00958

Li, M., Zhang, Y., Xu, Y., Cui, L., Huang, S., Wei, F., & Zhou, M. (2021). TrOCR: Transformer-based optical character recognition with pre-trained models. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 981–993. https://arxiv.org/abs/2109.10282

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2980–2988.

Liu, Y., Cheng, H., Klopfer, R., Schaaf, T., & Gormley, M. R. (2021). Effective convolutional attention network for multi-label clinical document classification. *Proceedings of the 20th Workshop on Biomedical Language Processing*, 161–170. https://aclanthology.org/2021.bionlp-1.17

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*. https://arxiv.org/abs/1907.11692

Louis, A. (2023). *A brief history of natural language processing (part 1)* [Accessed: 2025-04-20]. Medium. https://medium.com/@antoine.louis/a-brief-history-of-natural-language-processing-part1-ffbcb937ebce

Magueresse, A., Carles, V., & Heetderks, E. (2020). Low-resource languages: A review of past work and future challenges. *arXiv preprint arXiv:2006.07264*. https://arxiv.org/abs/2006.07264

McShane, M. (2017). Natural language understanding (nlu, not nlp) in cognitive systems. *AI Magazine*, *38*(4), 43–56.

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Pankaj, T. (2025). The brief history of ocr technology. https://www.docsumo.com/blog/optical-character-recognition-history

Parseur. (2025). Ocr zonal - extraction de données à partir de documents structurés. https://parseur.com/fr/ocr-zonal

Praveenkumar, A., Jha, G., Madival, S., et al. (2024). Deep learning approaches for potato price forecasting: Comparative analysis of lstm, bi-lstm, and am-lstm models. *Potato Research*. https://doi.org/10.1007/s11540-024-09823-z

Pretnar Žagar, A., & Demšar, J. (2022). Model evaluation. In R. Egger (Ed.), *Applied data science in tourism*. Springer, Cham.

Priya, D. (2023). 5 major challenges in nlp and nlu. *Analytics Insight*. https://www.analyticsinsight.net/latest-news/5-major-challenges-in-nlp-and-nlu

Ray, P. P. (2023). Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, *3*, 121–154.

Rouabhi, R., & Azizi, N. (2023). Bert and biobert models for improved multi-label medical text augmentation and classification [Received June 19, 2023; Revised August 27, 2023; Accepted August 30, 2023]. *VOLUME 2 Issue 1*.

SainiJul, H. (07, 2022). *Natural language processing: 11 key nlp techniques*. Analytics Steps. https://www.analyticssteps.com/blogs/natural-language-processing-11-key-nlp-techniques

Schaffer, H., Schmidt, C. S., Wutzkowsky, J., Lorek, K., Reinartz, L., Rückert, J., Temme, C., Böckmann, B., Horn, P. A., & Friedrich, C. M. (2025). A multimodal pipeline for clinical data extraction: Applying vision-language models to scans of transfusion reaction reports [Submitted on 28 Apr 2025]. https://doi.org/10.48550/arXiv.2504.20220

Semaan, P. (2012). Natural language generation: An overview. *Journal of Computer Science Research*, *1*(3), 50–57.

The taxonomy of natural language processing. (n.d.). ResearchGate. https://www.researchgate.net/figure/The-taxonomy-of-natural-language-processing

Team, S. (2023). *5 types of word embeddings and example nlp applications* [Accessed: 2025-06-15]. https://swimm.io/learn/large-language-models/5-types-of-word-embeddings-and-example-nlp-applications

Tripathi, A. (2021, July 2). *What is the main difference between rnn and lstm | nlp | rnn vs lstm* [Accessed: 2025-06-12]. https://ashutoshtripathi.com/2021/07/02/what-is-the-main-difference-between-rnn-and-lstm-nlp-rnn-vs-lstm/

Ultralytics. (n.d.). Residual networks (resnet) [Accessed: 2025-06-17].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*. https://arxiv.org/abs/1706.03762

Waisberg, E., Ong, J., Masalkhi, M., et al. (2023). Gpt-4: A new era of artificial intelligence in medicine. *Irish Journal of Medical Science*, *192*, 3197–3200. https://doi.org/10.1007/s11845-023-03377-8

Wang, J. (2023). A study of the ocr development history and directions of development. *Highlights in Science, Engineering and Technology*, *72*, 409–415. https://doi.org/10.54097/bm665j77

Wang, Y., Wang, L., Rastegar-Mojarad, M., Liu, S., Shen, F., Liu, H., & Afzal, N. (2021). Clinical information extraction applications: A literature review. *Journal of Biomedical Informatics*, *116*, 103671. https://doi.org/10.1016/j.jbi.2021.103671

What are convolutional neural networks? (n.d.). https://www.ibm.com/think/topics/convolutional-neural-networks

Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., & Zhou, M. (2020). Layoutlm: Pre-training of text and layout for document image understanding. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining (KDD)*, 1192–1200. https://doi.org/10.1145/3394486.3403172

Yuning, D., Chenxia, L., Ruoyu, G., Cheng, C., Weiwei, L., Jun, Z., Bin, L., Yang, Y., Liu, Q., Hu, X., Yu, D., & Ma, Y. (2021). PP-OCRv2: Bag of tricks for ultra lightweight ocr system. https://doi.org/10.48550/arXiv.2109.03144

Zong, M., & Krishnamachari, B. (2022). A survey on gpt-3. *arXiv preprint arXiv:2212.00857*.