EDITOR
Doç. Dr. Fatih YUCALAR

Research and
Evaluations in the Field of

COMPUTER SCIENCE ENGINEERING

June 2025

Privilege Holder • Yaşar Hız
Editor-in-chief • Eda Altunel
Prepared for Publication • Gece Kitaplığı
Editor • Doç. Dr. Fatih YUCALAR

First Edition • June 2025 / ANKARA

ISBN • 978-625-388-440-6

Gece Kitaplığı
Adress: Kızılay Mah. Fevzi Çakmak 1. Sokak Ümit Apt
No: 22/A Çankaya/ANKARA Tel: 0312 384 80 40

www.gecekitapligi.com
gecekitapligi@gmail.com

Printing and Binding
Bizim Buro
Certificate No: 42488

Research And Evaluations In The Field Of Computer Science Engineering

June 2025

Editor:
Doç. Dr. Fatih YUCALAR

CONTENTS

CHAPTER 1

DEEP LEARNING FOR DENTISTRY: YOLO VARIANTS IN TOOTH ABNORMALITY DETECTION

Davut ARI, Mehmet BURUKANLI ***1***

CHAPTER 2

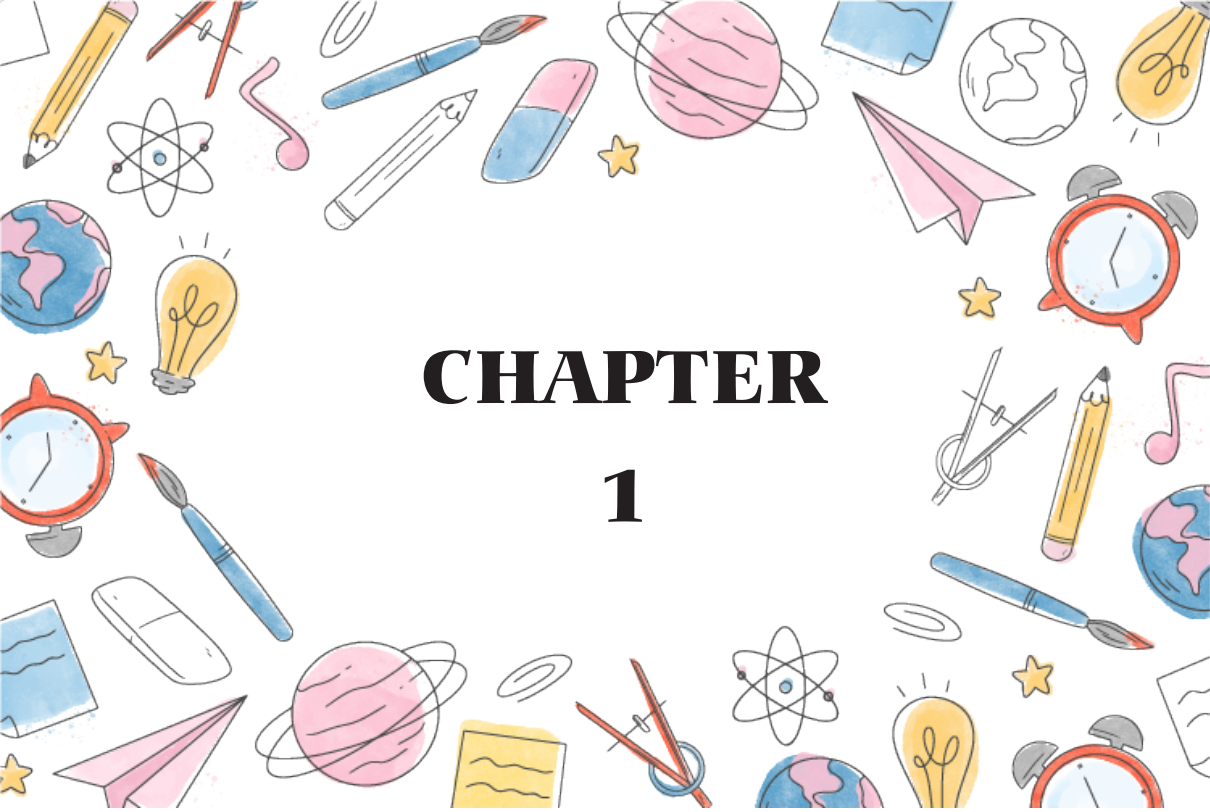
FROM N-GRAMS TO TRANSFORMERS: NLP TECHNİQUES AND APPLİCATİONS

İlhami SEL ***17***

CHAPTER 3

GENERATIVITY IN LANGUAGE AND VISION: THE EVOLUTION, ARCHITECTURES, AND EMERGING PARADIGMS OF LARGE MODELS

İlhami SEL ***35***



CHAPTER 1

DEEP LEARNING FOR DENTISTRY: YOLO VARIANTS IN TOOTH ABNORMALITY DETECTION

Davut ARI[1]

Mehmet BURUKANLI[2]

1 Research Assistant Dr., Bitlis Eren University, Faculty of Engineering and Architecture, Computer Engineering, dari@beu.edu.tr, ORCID: 0000-0001-6439-7957.

2 Lecturer Dr., Bitlis Eren University, Rectorate, Department of Common Courses, mburukanli@beu.edu.tr, ORCID: 0000-0003-4459-0455.

1. INTRODUCTION

Artificial intelligence (AI), and particularly deep learning (DL) techniques, have brought about major paradigm shifts in the healthcare domain in recent years, offering revolutionary innovations in clinical processes such as diagnosis, treatment planning, and patient monitoring. One of the most prominent areas of this transformation is dentistry and dental radiology, where imaging-based disciplines are experiencing rapid advancements driven by AI integration. Limitations in radiographic evaluations—such as subjectivity, the need for expert interpretation, time consumption, and anatomical overlaps—have strongly highlighted the need for robust artificial intelligence solutions to support diagnostic automation. The ability to assess common clinical conditions like dental caries, periodontal diseases, periapical lesions, and maxillofacial tumors in a more reliable and reproducible manner significantly enhances the potential of AI in this domain. Recent studies in the literature have shown that AI-powered systems not only enhance diagnostic accuracy but are also effectively utilized in subdomains such as image quality enhancement, automatic anatomical landmarking, and implant identification. However, several methodological and technical challenges still persist, preventing the seamless integration of these applications into routine clinical practice. Data scarcity, lack of standardization, generalization issues, and inconsistencies in performance metrics significantly hinder the safe, ethical, and effective deployment of these technologies. This is particularly critical in fields like dental radiology, which involve high variability in visual data. The need for explainable, clinically validated models developed through interdisciplinary collaboration is becoming increasingly essential(Sivari et al. 2023)(Singh and Raza 2022)(Putra et al. 2022)(Yazdanian et al. 2022).

Chen et al. made a significant contribution to the integration of artificial intelligence-based diagnostic systems into dentistry by proposing an effective and comprehensive deep learning–based approach for the simultaneous detection of periodontitis and dental caries in periapical radiographic images. In the study, a total of 2,850 single-tooth images were automatically cropped from 1,525 periapical radiographs using the YOLOv7 algorithm. For image enhancement, CLAHE and bilateral filtering techniques were integrated. EfficientNet-B0 architecture was selected for the classification task, achieving high accuracy (over 95%) and an AUC close to 98%. The proposed model demonstrated significant advantages in terms of both diagnostic accuracy and clinical time efficiency, offering notable superiority over single-disease-focused approaches and manual workflows commonly found in the literature. However, the study's exclusive focus on radiographic data without incorporating additional clinical factors has been identified as a potential

limitation that may restrict the overall generalizability of the proposed method(Chen et al. 2023). Lian et al. conducted an innovative study focusing on the automatic detection of carious lesions in panoramic dental radiographs using deep learning techniques, as well as their classification according to radiographic depth. The study utilized region-of-interest images extracted from 1,160 panoramic radiographs, performing segmentation with nnU-Net and four-stage (D0–D3) caries depth classification using DenseNet121. During training, techniques such as data augmentation, dropout, and label smoothing were employed, and transfer learning was utilized to enhance model performance. The model demonstrated high accuracy in segmentation metrics including IoU and Dice, as well as strong classification performance with accuracies of 95.7%, 83.2%, and 86.3% for stages D1, D2, and D3, respectively. Comparisons with dental professionals revealed that the model was particularly more sensitive in detecting early-stage lesions, although no statistically significant difference was observed. This study addresses significant gaps in the literature by focusing on the rarely explored tasks of caries depth classification, the combined application of segmentation and classification, performance comparison with dental practitioners, and automated screening on panoramic radiographs. It offers a highly promising approach that could be integrated into clinical decision support systems. However, the study has limitations, including reliance on data from a single center, the absence of a gold standard reference, and lack of validation in clinical practice(Lian et al. 2021). Chen et al. developed a deep learning model based on Faster R-CNN for detecting lesions such as caries, periapical periodontitis, and periodontitis in periapical radiographs. Trained on 2,900 images, the model classified lesions according to both disease type and severity level and was evaluated using various training strategies. While the model achieved higher accuracy in detecting severe lesions, its performance was lower for early-stage lesions. The study highlights the applicability of deep CNNs for automated diagnosis while also drawing attention to challenges such as data imbalance and detection of small lesions (Chen et al. 2021). Lee et al. developed a U-Net-based deep learning model to detect early dental caries in bitewing radiographs and evaluated its impact on clinician performance. The model achieved an overall F1-score of approximately 64% in detecting all types of carious lesions and significantly improved clinicians' sensitivity, particularly for early-stage lesions. However, the increased sensitivity with model assistance was accompanied by a decrease in positive predictive value and an elevated false positive rate. Despite limitations such as a limited dataset, low-quality images, and the absence of a gold standard, the study demonstrated the supportive role of deep learning in clinical diagnosis. (Lee et al. 2021). Khan et al. proposed Ded-Net, a novel adaptive enhancement method designed to address issues such as contrast

deficiency and uneven brightness in low-quality dental images, capable of working effectively with small datasets while preserving structural integrity. Trained on only 216 images, this network separates dental images into reflection and illumination components, applying distinct enhancement processes and offering user-specific contrast adjustments. With its improved image quality, Ded-Net contributes to early diagnosis and enhances the performance of intelligent systems. Supported by low computational cost, texture and edge preservation, and clinical validation, this method represents a significant innovation in dental imaging(Khan et al. 2023). In their proposed approach, Almalki et al. focused on the automatic classification of dental diseases using orthopantomogram (OPG) X-ray images and deep learning techniques. To address the limitations of manual diagnosis—such as inefficiency, subjectivity, and the risk of human error—they developed a custom dataset consisting of 800 high-resolution panoramic dental X-rays. The authors applied a YOLOv3-based object detection model, fine-tuned through extensive data augmentation and annotated under expert supervision. Their system was trained and evaluated using Google Colab with GPU acceleration, achieving notable results, including an mAP of 99.33%, F1-score of 0.99, and an IOU of 84.56%. Unlike earlier methods that struggled with low generalizability and lacked automation, this work demonstrates high accuracy and robustness in detecting four common dental conditions: cavities, root canals, crowns, and broken-down root canals. However, its reliance on a relatively small, localized dataset limits broader applicability. Future directions suggested include real-time detection capabilities and the inclusion of a more diverse and larger dataset encompassing additional dental disorders(Almalki et al. 2022).

In this study, 1075 training data, 121 validation data and 73 testing data were used to detect dental anomalies using YOLO models. YOLO models were compared in terms of precision, recall, mAP50 and mAP50-95 values. The results showed that YOLO-based models were quite successful in detecting dental anomalies.

2. DENTAL RADIOGRAFHY DATASET AND YOLO MODELS

In this section, the dental radiography dataset and YOLO deep learning models used in the detection of dental abnormalities are explained.

2.1. Dental radiography dataset

In this study, the dental radiography dataset consists of 1269 X-ray images (Momeni 2023)(Roy 2024). The dental radiography dataset consists of 4

classes in total: ' Fillings ', ' Implant ', ' Cavity ', 'Impacted Tooth '. Some image examples found in the dental radiography dataset used in this study are shown in Figure 1.

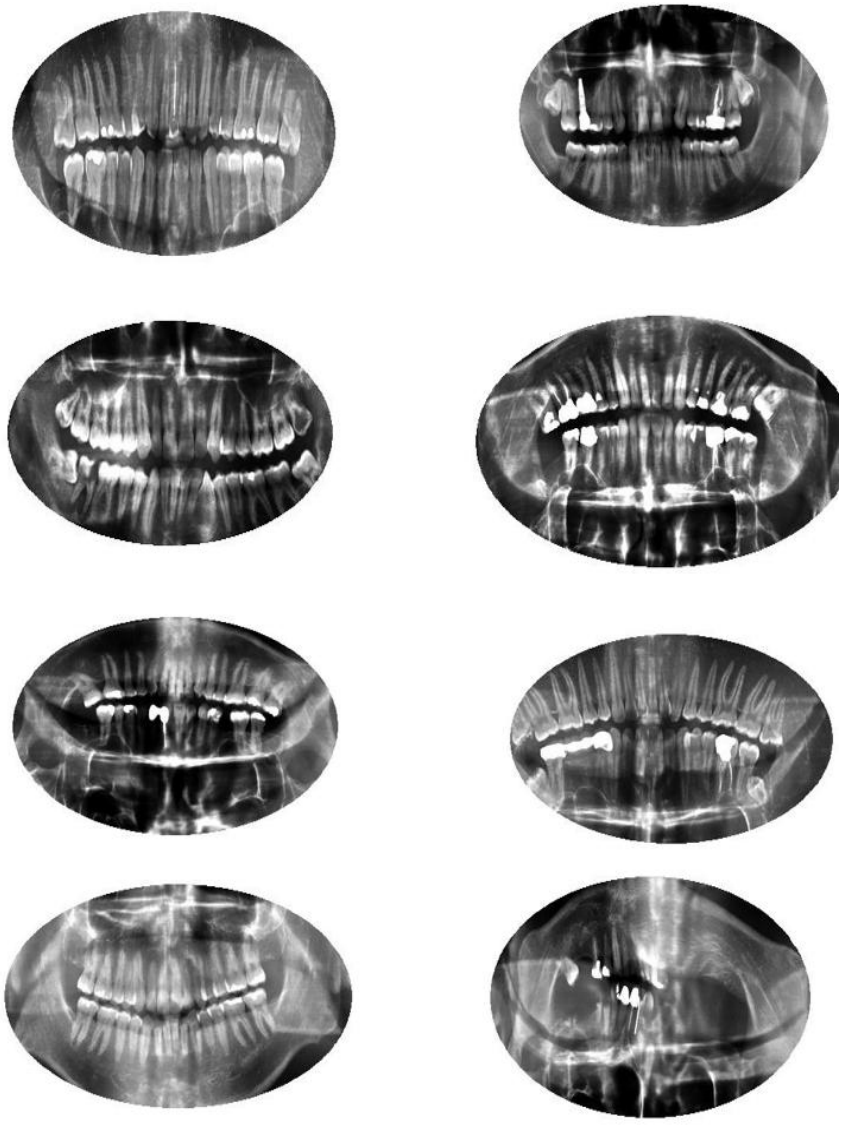


Figure 1. Some image examples found in the dental radiography dataset (Momeni 2023)

The total amount of data in the training, validation and test datasets for YOLO models in the dental radiography dataset is shown in Table 1.

Table 1. The total amount of training, validation and testing datasets in the dental radiography dataset

Dataset	Amount of dataset
Training	1075
Validation	121
Testing	73

As seen in Table 1, for each YOLO model, 1075 image samples were used for training, 121 image samples for validation, and 73 image samples for testing.

2.2. YOLO models

In this section, YOLOv9, YOLOv10, YOLO11 and YOLOv12 models for the dental radiography dataset are explained below.

2.2.1. YOLOv9 model

YOLOv9 model (version 9) is a deep learning based model that is frequently used for object detection and segmentation. YOLOv9 model is trained on COCO dataset having 80 classes. In this study, pre-trained yolov9t.pt weights were used for dental radiography anomaly detection (Wang, Yeh, and Mark Liao 2025)

2.2.2. YOLOv10 model

YOLOv10 model (version 10) is a deep learning based model designed in real-time object detection. YOLOv10 model is trained on COCO dataset which has 80 classes similar to YOLOv9 model. In this study, pre-trained yolov10n.pt weights were used for dental radiography anomaly detection (Wang et al. 2024).

2.2.3. YOLO11 model

YOLO11 model (version 11) is a deep learning based model designed in real-time object detection similar to YOLO11 model. YOLO11 model is also trained on COCO dataset with 80 classes. In this study, pre-trained yolo11n.pt weights were used for dental radiography anomaly detection. (Jocher and Qiu 2024).

2.2.4. YOLOv12 model

YOLOv12 model (version 12) is a real-time attention-based deep learning model. YOLOv12 model is also trained on COCO dataset with 80 classes. In this study, pre-trained yolo12n.pt weights were used for dental radiography anomaly detection (Tian, Ye, and Doermann 2025).

3. RESULTS

To measure the performance of each YOLO-based model, precision, recall and f1 score, average precision (AP), mean average precision (mAP) values were used. The formulas of these values are shown in equations (1), (2), (3) and (4), respectively (Ari and Burukanli 2025). In addition, during the training of YOLO models, optimizer = AdamW, epochs = 70, batch size = 16, learning rate = 0.00125, momentum = 0.937 and weight_decay: 0.0005 were set. S in equations (4) is the number of classes.

$$\text{Precision} = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad (1)$$

$$Recall = \frac{TP}{TP + False\ Negative\ (FN)} \quad (2)$$

$$F1 - Score = 2 * \frac{\text{Precision} * Recall}{\text{Precision} + Recall} \quad (3)$$

$$Mean\ average\ precision\ (mAP) = \frac{\sum_{f=1}^{S}(AP_f)}{S} \quad (4)$$

3.1. Experimental results

In this study, experimental results obtained using YOLOv9, YOLOv10, YOLO11 and YOLOv12 models for the detection of real-time dental radiography abnormalities are discussed. The performance comparison of YOLOv9, YOLOv10, YOLO11 and YOLOv12 models on the dental radiography dataset is shown in Table 2.

Table 2. The performance values of YOLOv9, YOLOv10, YOLO11 and YOLOv12 models on the dental radiography dataset

Model	Precision (%)	Recall (%)	mAP50 (%)	mAP50-95 (%)	GFLOPS
YOLOv9	73.3	75.2	**77.7**	51.2	7.9
YOLOv10	**73.9**	71.1	76.1	49.9	8.4
YOLO11	67.4	76.1	76.2	51.1	**6.4**
YOLOv12	70.3	**79.9**	76.7	**51.3**	6.5

When the results are examined in detail in Table 2, among the YOLO based models on the dental radiography dataset, the best score in terms of precision value was obtained by the YOLOv10 model with 73.9%, while the worst result was obtained by the YOLO11 model with 67.4%. Similarly, the best score in terms of recall value was obtained by the YOLOv12 model with 79.9% on the dental radiography data set, while the worst result was obtained by the YOLOv10 model with 71.1%. In addition, the best score in terms of mAP50 value on the dental radiography dataset was obtained by the YOLOv9 model with 77.7%, while the worst result was obtained by the YOLOv10 model with 76.1%. Similarly, the best score in terms of mAP50-95 value on the dental radiography dataset was obtained by the YOLOv12 model with 51.3%, while the worst result was obtained by the YOLOv10 model with 49.9%. The normalized confusion matrix for the YOLOv12 model on the dental radiography dataset is shown in Figure 2.

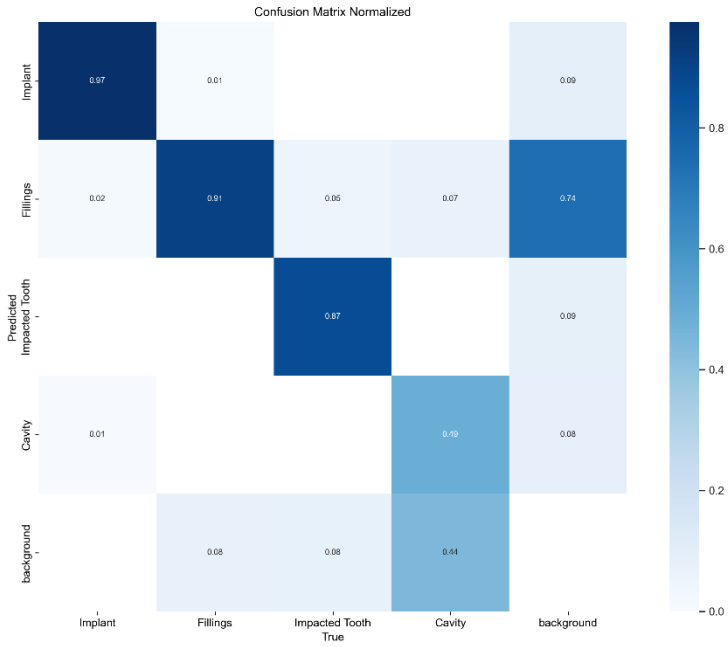


Figure 1. The normalized confusion matrix for the YOLOv12 model on the dental radiography dataset

The precision-confidence curve for the YOLOv12 model on the dental radiography dataset is shown in Figure 2. The recall-confidence curve for the YOLOv12 model on the dental radiography dataset is shown in Figure 3. The precision-recall curve for the YOLOv12 model on the dental radiography dataset is shown in Figure 4. The f1-confidence curve for the YOLOv12 model on the dental radiography dataset is shown in Figure5.

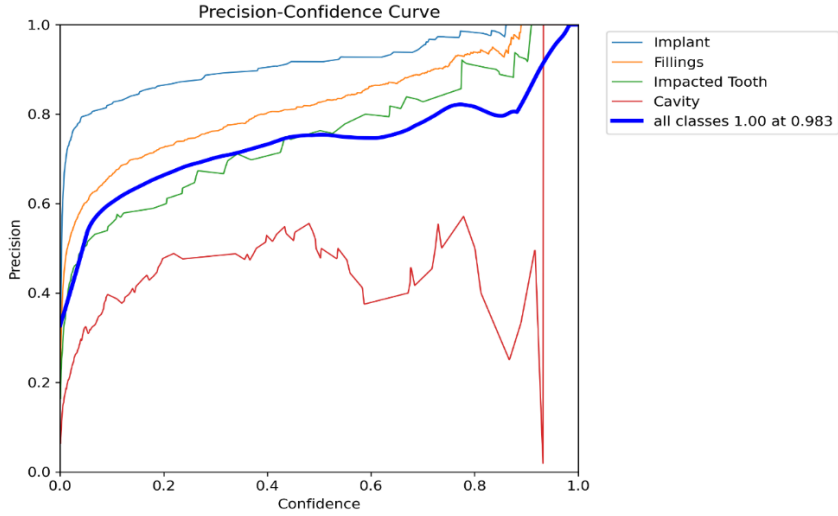


Figure 2. The precision-confidence curve for the YOLOv12 model on the dental radiography dataset

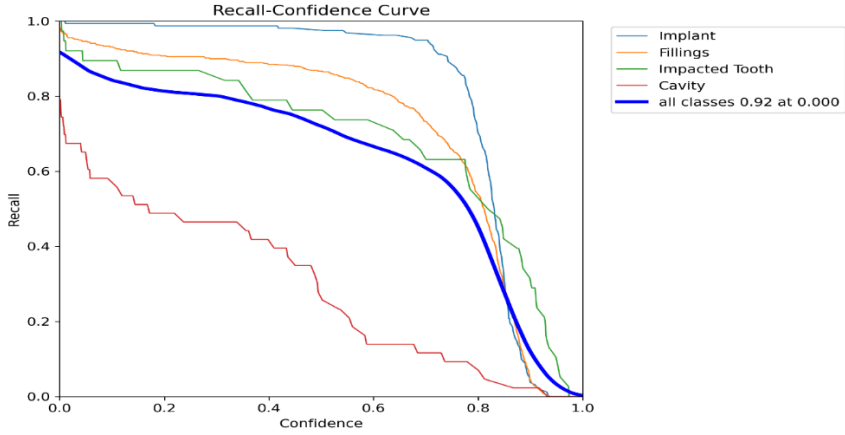


Figure 3. The precision-confidence curve for the YOLOv12 model on the dental radiography dataset

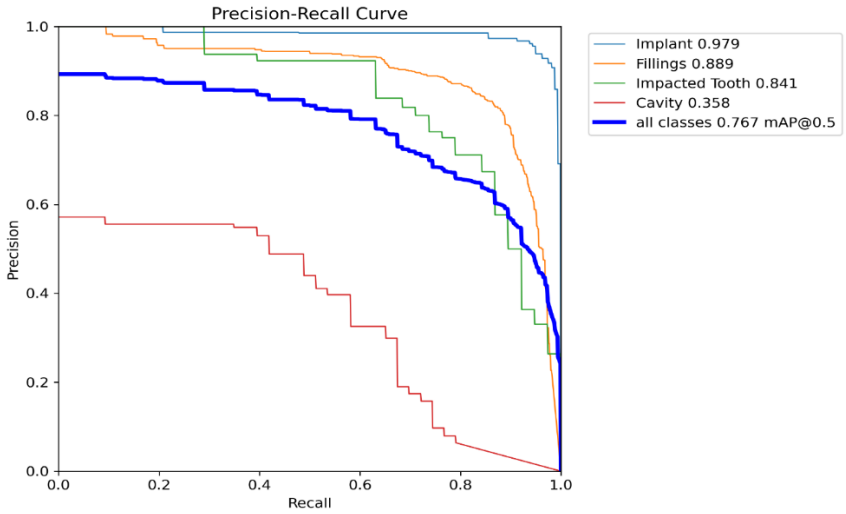


Figure 4. The precision-confidence curve for the YOLOv12 model on the dental radiography dataset

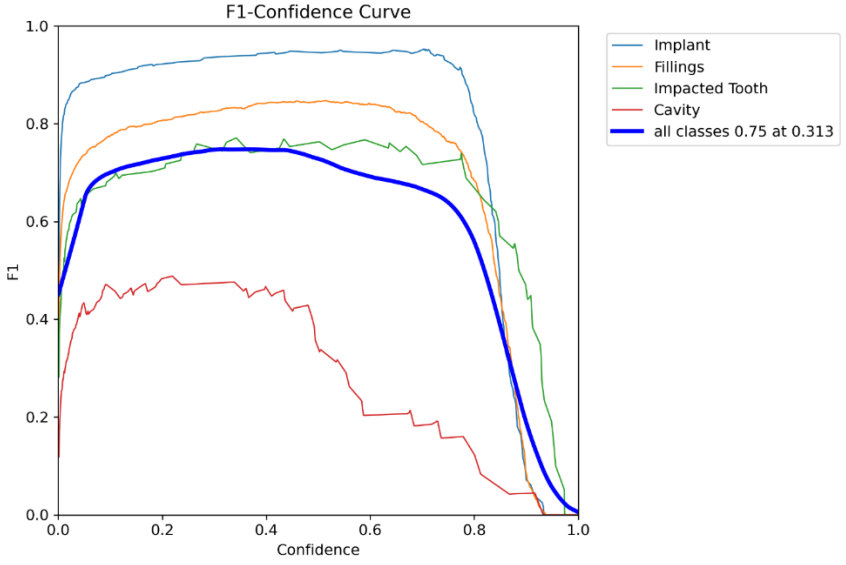


Figure 5. The precision-confidence curve for the YOLOv12 model on the dental radiography dataset

In this study, the performance results of the YOLOv12 model on training and validation data sets for the detection of dental abnormalities are given in Figure 6.

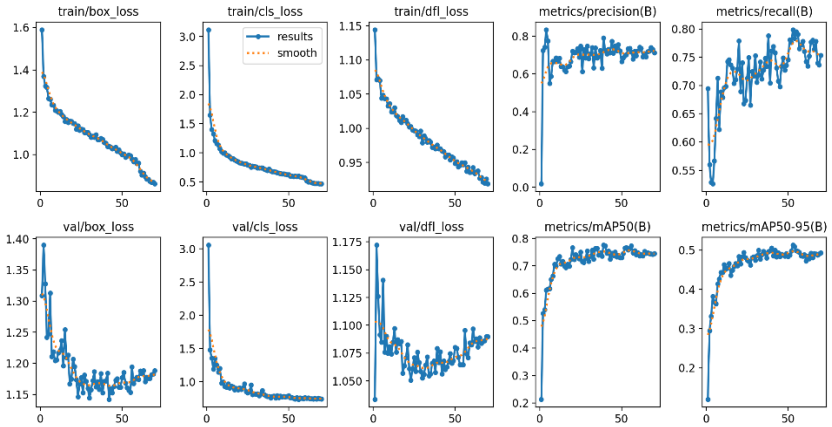


Figure 6. Training and validation results of the YOLOv12 model on the dental radiograph dataset

As seen in Figure 6, the graphical results obtained during the training and validation phase of the YOLOv12 model on the dental radiography dataset show how robust the YOLOv12 model is. These results are promising for future studies. Some prediction results obtained using YOLOv12 model on dental radiography testing dataset are shown in Figure 7.

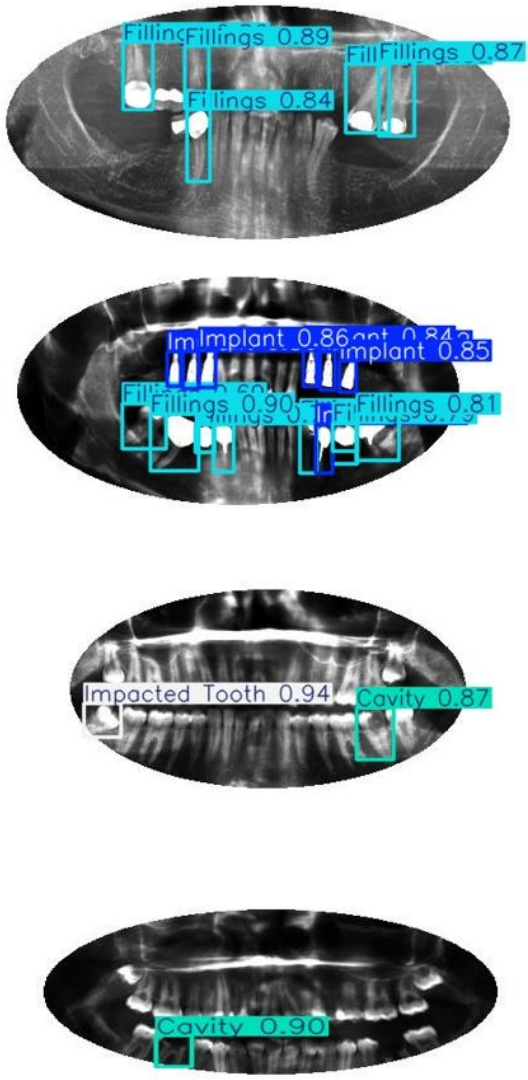


Figure 7. Some prediction results obtained using YOLOv12 model on dental radiography testing dataset

As seen in Figure 7, the YOLOv12 model was very successful in detecting images in the dental radiography test dataset that it had never seen before in real time. For example, the YOLOv12 model successfully detected a tooth image in the 'cavity' class in the test dataset, which it had never seen before, with a 90% success rate, as seen in Figure 7. Similarly, an image in the 'impacted tooth' class was successfully detected with a 94% success

rate. These successful prediction results show that the YOLOv12 model is also promising in real-world tasks.

4. CONCLUSION

In this study, dental anomalies were detected using YOLOv9, YOLOv10, YOLO11 and YOLOv12 models on a dental radiography dataset consisting of 1269 images. In addition, YOLO models were compared in terms of precision, recall, mAP50 and mAP50-95 values. Experimental results were examined in detail and as a result, it was observed that YOLO-based models, especially the YOLOv12 model, were quite successful in detecting dental anomalies. In the next study, we plan to improve their overall performance (especially in terms of mAP50 and mAP50-95 values) on the dental radiography image dataset.

REFERENCES

Almalki, Yassir Edrees, Amsa Imam Din, Muhammad Ramzan, Muhammad Irfan, Khalid Mahmood Aamir, Abdullah Almalki, Saud Alotaibi, Ghada Alaglan, Hassan A. Alshamrani, and Saifur Rahman. 2022. "Deep Learning Models for Classification of Dental Diseases Using Orthopantomography X-Ray OPG Images." *Sensors* 22(19). doi: 10.3390/s22197370.

Ari, Davut, and Mehmet Burukanli. 2025. "Real-Time Human Bone Fracture Detection Using Yolo Models." Pp. 172–83 in *Ases IX. International Scientific Research Congress*. Adıyaman, Turkey.

Chen, Hu, Hong Li, Yijiao Zhao, Jianjiang Zhao, and Yong Wang. 2021. "Dental Disease Detection on Periapical Radiographs Based on Deep Convolutional Neural Networks." *International Journal of Computer Assisted Radiology and Surgery* 16(4):649–61. doi: 10.1007/s11548-021-02319-y.

Chen, Ivane Delos Santos, Chieh Ming Yang, Mei Juan Chen, Ming Chin Chen, Ro Min Weng, and Chia Hung Yeh. 2023. "Deep Learning-Based Recognition of Periodontitis and Dental Caries in Dental X-Ray Images." *Bioengineering* 10(8):1–13. doi: 10.3390/bioengineering10080911.

Jocher, Glenn, and Jing Qiu. 2024. "Ultralytics YOLO11." Retrieved (https://github.com/ultralytics/ultralytics).

Khan, Rizwan, Saeed Akbar, Ali Khan, Muhammad Marwan, Zahid Hussain Qaisar, Atif Mehmood, Farah Shahid, Khushboo Munir, and Zhonglong Zheng. 2023. "Dental Image Enhancement Network for Early Diagnosis of Oral Dental Disease." *Scientific Reports* 13(1):1–14. doi: 10.1038/s41598-023-30548-5.

Lee, Shinae, Sang il Oh, Junik Jo, Sumi Kang, Yooseok Shin, and Jeong won Park. 2021. "Deep Learning for Early Dental Caries Detection in Bitewing Radiographs." *Scientific Reports* 11(1):1–8. doi: 10.1038/s41598-021-96368-7.

Lian, Luya, Tianer Zhu, Fudong Zhu, and Haihua Zhu. 2021. "Deep Learning for Caries Detection and Classification." *Diagnostics* 11(9). doi: 10.3390/DIAGNOSTICS11091672.

Momeni, Mohamadreza. 2023. "Dental Radiography Analysis and Diagnosis Dataset." *Kaggle*. Retrieved (https://www.kaggle.com/datasets/imtkaggleteam/dental-radiography/data).

Putra, Ramadhan Hardani, Chiaki Doi, Nobuhiro Yoda, Eha Renwi Astuti, and Keiichi Sasaki. 2022. "Current Applications and Development of

Artificial Intelligence for Digital Dental Radiography." *Dentomaxillofacial Radiology* 51(1). doi: 10.1259/DMFR.20210197.

Roy, Sumit Kumar. 2024. "Dental Radiography Analysis and Diagnosis Dataset (Penaromic X-Ray)." *Kaggle*. Retrieved (https://www.kaggle.com/datasets/lesumitkumarroy/yolo-penaromic-xray/data).

Singh, Nripendra Kumar, and Khalid Raza. 2022. "Progress in Deep Learning-Based Dental and Maxillofacial Image Analysis: A Systematic Review." *Expert Systems with Applications* 199(February):116968. doi: 10.1016/j.eswa.2022.116968.

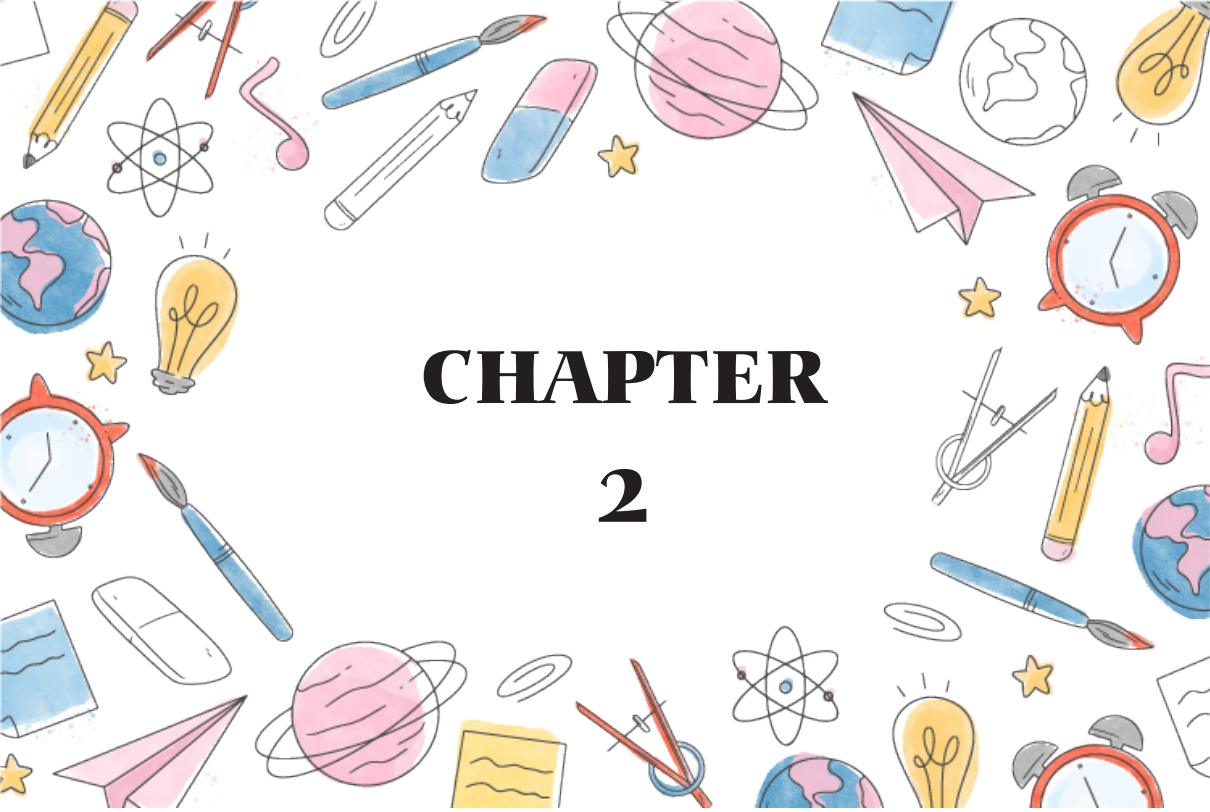
Sivari, Esra, Guler Burcu Senirkentli, Erkan Bostanci, Mehmet Serdar Guzel, Koray Acici, and Tunc Asuroglu. 2023. "Deep Learning in Diagnosis of Dental Anomalies and Diseases: A Systematic Review." *Diagnostics* 13(15):1–28. doi: 10.3390/diagnostics13152512.

Tian, Yunjie, Qixiang Ye, and David Doermann. 2025. "YOLOv12: Attention-Centric Real-Time Object Detectors."

Wang, Ao, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. 2024. "YOLOv10: Real-Time End-to-End Object Detection." *Advances in Neural Information Processing Systems* 37(NeurIPS):1–28.

Wang, Chien-Yao, I. Hau Yeh, and Hong-Yuan Mark Liao. 2025. "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information." Pp. 1–21 in *arXiv:2402.13616v2*.

Yazdanian, Mohsen, Shahryar Karami, Elahe Tahmasebi, Mostafa Alam, Kamyar Abbasi, Mahdi Rahbar, Hamid Tebyaniyan, Reza Ranjbar, Alexander Seifalian, and Alireza Yazdanian. 2022. "Dental Radiographic/Digital Radiography Technology along with Biological Agents in Human Identification." *Scanning* 2022. doi: 10.1155/2022/5265912.



CHAPTER 2

FROM N-GRAMS TO TRANSFORMERS: NLP TECHNIQUES AND APPLICATIONS

İlhami SEL[1]

1 Engineer, Ministry of National Education, ilhamisel23@gmail.com,
ORCID: 0000-0003-0222-7017

Natural Language Processing (NLP) is a subfield of artificial intelligence that aims to convert human language into formats understandable and processable by computer systems. Human language, as the fundamental medium of interpersonal communication, poses significant challenges for computational understanding due to its complex structural properties and rich semantic layers. Therefore, NLP systems must not only analyze the surface-level structures of words and sentences but also consider deeper linguistic elements such as context, meaning, user intent, and general world knowledge (Khurana et al., 2023). The development of NLP has occurred at the intersection of several disciplines, including computer science, linguistics, artificial intelligence, and statistics. Advances in hardware technology and the substantial increase in available data have enabled NLP applications to become widespread today, encompassing a diverse range of tasks from search engines and chatbots to machine translation and automatic summarization. The foundations of NLP were laid in the 1950s. Alan Turing's provocative question, "Can machines think?", introduced via the Turing Test, provided an early benchmark for assessing the human-like communicative abilities of computer systems. Additionally, the concept of "artificial intelligence," proposed by John McCarthy at the 1956 Dartmouth Conference, marked the beginning of a new research field that encompassed NLP as well (Chowdhary, 2020).

Early NLP implementations were predominantly rule-based systems, relying on manually crafted linguistic rules to analyze text. However, the inherent complexity and numerous exceptions within natural languages quickly exposed the limitations of these systems. The 1990s saw the rise of statistical approaches, introducing probabilistic perspectives to NLP through techniques such as n-gram models and hidden Markov models. This era notably advanced fields such as machine translation and speech recognition (Lauriola et al., 2022). In the past decade, deep learning-based methods have led to revolutionary advancements in NLP. Sequential neural network architectures, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks (Mikolov et al., 2010; Schuster & Paliwal, 1997), along with the Transformer architecture introduced in 2017 (Vaswani et al., 2017), have enabled faster and more effective modeling approaches. Derived from this architecture, large-scale language models such as BERT and GPT have achieved human-level performance across numerous NLP tasks (Lauriola et al., 2022).

Natural Language Processing is fundamentally categorized into two main branches: Natural Language Understanding (NLU) and Natural Language Generation (NLG).

- NLU focuses on extracting meaning from textual or spoken input, including tasks such as named entity recognition (NER), semantic parsing, sentiment analysis, and co-reference resolution.

- NLG involves generating human-like natural language outputs from data, applied in tasks like automatic report creation, summarization, and storytelling (Khurana et al., 2023).

These tasks constitute the core components of contemporary AI-driven communication systems.

Currently, key research areas in NLP center around large-scale language models, transfer learning approaches, multilingualism, and various challenges posed by deep learning techniques. Particularly, pretrained large language models such as BERT, GPT, and T5 have significantly advanced the field by demonstrating outstanding performance on various NLP tasks. Transfer learning approaches have further enhanced the effectiveness of these models, allowing high performance even with limited data. Moreover, developing multilingual systems capable of supporting numerous languages has become an important research direction, considerably improving global accessibility. Nevertheless, critical challenges remain, including the high computational costs associated with deep learning-based NLP systems, difficulties in reproducibility of experiments, and insufficient interpretability of model decisions (Lauriola et al., 2022).

In conclusion, NLP forms the cornerstone of artificial intelligence systems designed to interact with human language. Historically evolving from rule-based approaches through statistical models to deep learning architectures, NLP has reached a stage where systems are capable of understanding, interpreting contextually, and generating text dynamically. Despite these advancements, extensive research continues to enhance the reliability, fairness, and efficiency of NLP systems.

1. Language Modeling

In natural language processing (NLP) applications, it is essential to first represent textual data in a numerical form in order to process it effectively. Two of the most fundamental techniques developed for this purpose are n-gram-based language models and the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization method. Both approaches allow for the representation of texts within a vector space, which enables their widespread use in tasks such as classification, similarity analysis, and information extraction across various NLP applications.

1.1 n-gram Based Language Modeling

An n-gram refers to a group of n consecutive items within a sequence. In the context of language modeling, n-gram models are probabilistic methods used to represent sequences of words or characters. The principal aim of these models is to predict the likelihood of a word

based on the preceding n-1 items. In an n-gram model, the probability of a sequence of words is computed as follows:

$$P(w_1, w_2, \dots, w_n) \approx \prod_{i=1}^{n} P(w_i \mid w_{i-n+1}, \dots, w_{i-1}) \quad (1)$$

For example, for the bigram model (n=2), this formula becomes:

$$P(w_1, w_2, \dots, w_n) \approx \prod_{i=1}^{n} P(w_i \mid w_{i-1}) \quad (2)$$

This approach serves as a fundamental technique in applications such as language modeling, auto-completion, and text generation (Jurafsky & Martin, 2025). Unlike the "bag-of-words" (BoW) method, n-gram models take word order into account, thus encoding richer contextual information. As the value of n increases, the model captures broader contextual dependencies; however, this also results in a higher number of parameters and exacerbates the sparsity problem. For example:
"I am reading a book"

- Unigram: ["I", "am", "reading", "a", "book"]
- Bigram: [("I", "am"), ("am", "reading"), ("reading", "a"), ("a", "book")]
- Trigram: [("I", "am", "reading"), ("am", "reading", "a"), ("reading", "a", "book")]

One of the key advantages of n-gram-based language models is their simplicity. These models are computationally efficient and provide fast, practical solutions, especially at the early stages of language modeling. Their sensitivity to word order allows them to better capture structural properties of language, such as word combinations and contextual relations, compared to unigram models. Higher-order n-grams like bigrams and trigrams are particularly effective in modeling inter-word dependencies within sentences.

However, n-gram models also have certain limitations. As n increases, the model becomes more complex, and the number of unseen n-gram combinations in the training data rises significantly. This phenomenon, known as the "data sparsity problem," leads to the model being unable to estimate probabilities for previously unseen word sequences, resulting in the so-called "zero-frequency problem." To address this, smoothing techniques are commonly applied to increase the flexibility and reliability of probability estimates. Popular methods such as Laplace (add-one) smoothing and Good-Turing smoothing assign small, non-zero probabilities to unseen n-grams, thereby improving the model's generalization capacity and yielding more realistic predictions (Manning et al., 2008).

1.2 TF-IDF (Term Frequency – Inverse Document Frequency)

TF-IDF is a statistical measure used to assess the importance of words within a document. The underlying idea is that if a word appears frequently in a document, it is likely to be significant for that document; however, if it is common across many documents, its discriminative power decreases. The components TF and IDF are defined as follows:

Term Frequency (TF):

$$\mathrm{TF}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{3}$$

Here f$f_{t,d}$ is the frequency of term t in document d.

Inverse Document Frequency (IDF):

$$\mathrm{IDF}(t,D) = \log\left(\frac{N}{1+|\{d \in D : t \in d\}|}\right) \tag{4}$$

Here N is the total number of documents, and adding 1 to the denominator prevents the log(0) error.

TF-IDF Score:

$$\text{TF-IDF}(t,d,D) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t,D) \tag{5}$$

This approach ensures that terms which are both frequent within a document and rare across the corpus are highlighted as important. For example, in a dataset with three documents:

D1: "This book is beautiful"
D2: "I enjoy reading books"
D3: "I liked this movie"

The IDF values for the words:

"book": appears in 2 documents → log(3/2) ≈ 0.176
"movie": appears in 1 document → log(3/1) ≈ 0.477

As shown, the word "movie" is considered more distinctive in this collection.

TF-IDF is widely applied in tasks such as text classification, document clustering, and information retrieval. In fact, many modern search engines rely fundamentally on this algorithm for ranking results (Salton & Buckley, 1988). Jurafsky and Martin (2025) emphasize that n-gram modeling forms the backbone of various applications, including

language classification, sentiment analysis, and spam detection. Moreover, combining n-gram representations with naive Bayes classifiers has shown to yield effective results for text classification tasks (Jurafsky & Martin, 2025).

In the literature, TF-IDF is recognized as a standard technique for document classification and information extraction. Some recent studies report that combining TF-IDF scores with word embedding methods leads to hybrid vector representations with enhanced expressive power.

1.3 Word Embedding: Word2Vec, GloVe, FastText

Representing the meaning of words numerically is a fundamental step in language modeling processes for NLP applications. While traditional approaches such as n-grams or TF-IDF represent words as indices or sparse vectors, these representations fail to capture semantic relationships among words. To overcome this limitation, word embedding techniques have been developed; these methods map words into continuous, dense vector spaces. In such spaces, semantic and contextual similarities between words can be quantitatively assessed using measures like Euclidean distance or cosine similarity (Bengio et al., 2003).

Word2Vec

Word2Vec, developed by Mikolov et al. in 2013, is a predictive word embedding approach (Mikolov et al., 2013). It is based on two primary architectures:

- Continuous Bag-of-Words (CBOW): Predicts the target word given its surrounding context.
- Skip-Gram: Predicts the surrounding words given the target word.

For the Skip-Gram model, the mathematical formulation is as follows:

Given a target word wt , the probability of the context words w_{t-j}, …, w_{t+j} within a window of size c centered around wt, is defined as follows:

$$\prod_{-c\le j\le c, j\neq 0} P\left(w_{t+j} \mid w_t\right) \tag{6}$$

Each conditional probability is usually calculated with softmax:

$$P(w_o \mid w_i) = \frac{\exp\left(v_{w_o} \cdot v_{w_i}\right)}{\sum_{w\in V} \exp\left(v_w \cdot v_{w_i}\right)} \tag{7}$$

Here, v_{wi} is the vector of the input word, v_{wo} is the vector of the output word, and V is the entire vocabulary. This structure allows words to be grouped into similar vectors based on context relationships. For example, inferences such as "king" – "man" + "woman" ≈ "queen" can be made.

1.4 GloVe (Global Vectors for Word Representation)

GloVe is a statistically based counting method developed by Stanford University (Pennington et al., 2014), unlike Word2Vec. GloVe uses a co-occurrence matrix to infer meaning based on the ratios between word pairs. The basic assumption of GloVe is that the ratio of two words occurring together can be modeled by their vector difference.

$$w_i^T \widetilde{w_j} + b_i + \widetilde{b_j} = \log(X_{ij}) \tag{8}$$

w_i: vector of target word
$\widetilde{w_j}$: vector of context word
b_i, $\widetilde{b_j}$: bias terms
X_{ij}: frequency of co-occurrence of word i with context word j

The following loss function is minimized for the optimization of the model:

$$J = \sum_{i,j=1}^{|V|} f(X_{ij}) \cdot \left(w_i^T \widetilde{w_j} + b_i + \widetilde{b_j} - \log X_{ij}\right)^2 \tag{9}$$

As a result, GloVe provides statistical information on large text collections. learns a more global structure of meaning based on the context (Pennington et al., 2014).

1.5 FastText

FastText is an embedding method developed by Facebook AI. While it is based on the architecture of Word2Vec, it considers words not only as a whole but also at the subword level (Bojanowski et al., 2017). Each word is represented by dividing it into character n-grams.

Thanks to this structure, it is possible to assign close vectors to words that are similar in meaning but different in form, and to create vectors from their subunits even for words that are never seen (OOV).

The mathematical FastText model is:

$$v_w = \sum_{g \in G_w} z_g \tag{10}$$

Here:

v_w: vector representation of w
$g \in G_w$: set of all character n-grams belonging to w
z_g: character vector

This structure enables the creation of vector representations even for previously unseen words (out-of-vocabulary, OOV).

The primary contribution of word embedding techniques is that they do not merely represent words as indices, but rather as continuous, multi-dimensional vectors, thereby making it possible to mathematically compare semantic similarities between words. As a result, word embeddings allow for the modeling of similarity measures, grammatical contexts, and semantic relations within a vector space—capabilities that are not possible with traditional frequency-based methods.

The Word2Vec model learns the meanings of words from their surrounding context, leveraging its local, context-predictive architecture. In this way, it effectively captures both syntactic and semantic properties of language at the word level. In contrast, the GloVe model incorporates global statistical information by leveraging co-occurrence matrices, which provide distributional cues derived from the overall usage patterns in a language corpus, thereby enabling the modeling of broader semantic patterns. FastText, on the other hand, represents words not only as whole tokens but also incorporates their subword (character-level n-gram) components, allowing for the creation of meaningful vector representations even for morphologically complex or previously unseen words. This characteristic provides a particular advantage for agglutinative languages like Turkish, which are rich in inflectional and derivational morphology.

Each of these three approaches has distinct strengths and limitations. While methods such as Word2Vec and GloVe yield strong results when trained on large-scale datasets, they may be inadequate for representing OOV words. FastText largely overcomes this limitation by virtue of its subword modeling, although it comes at the cost of increased computational requirements and longer training times. Nevertheless, all three embedding techniques are still widely used in many applications today, especially in scenarios where pretrained models are not readily available.

With advances in NLP, these traditional embedding techniques have formed the foundation for new contextual embedding methods (such as ELMo, BERT, and GPT), which learn word meanings not only from the word itself but also from the context of the entire sentence or paragraph. These contextual models provide much more flexible and powerful representations; however, their training can be computationally intensive. Therefore, many systems still employ Word2Vec, GloVe, and FastText

either directly or as part of their preprocessing pipelines. In this regard, classic embedding methods continue to serve as both the precursors and complements to today's advanced NLP architectures (Devlin et al., 2018; Radford et al., 2018).

2. Transformer-Based NLP Models

One of the most revolutionary advancements in the field of natural language processing (NLP) in recent years has been the introduction of the Transformer architecture by Vaswani et al. in 2017 (Vaswani et al., 2017). Their seminal work, "Attention is All You Need," eliminated the need for sequential processing and paved the way for more efficient systems in terms of both computation time and model capacity. The Transformer was designed as an alternative to previously dominant recurrent neural network (RNN) architectures such as RNN, LSTM, and GRU, which were widely used in NLP tasks. Today, Transformer serves as the foundational architecture for state-of-the-art models including GPT (Devlin et al., 2018), BERT (Radford et al., 2018), and T5 (Raffel et al., 2019).

The Transformer architecture is primarily composed of two main components:

- Encoder: Processes the input data and generates contextual representations.
- Decoder: Utilizes the encoder's outputs to generate target sequences.

Each encoder or decoder block consists of the following sub-components:

- Multi-head self-attention
- Feed-forward neural network (fully connected layers)
- Layer normalization
- Residual connections

The general structure of the Transformer architecture is illustrated in Figure 1.

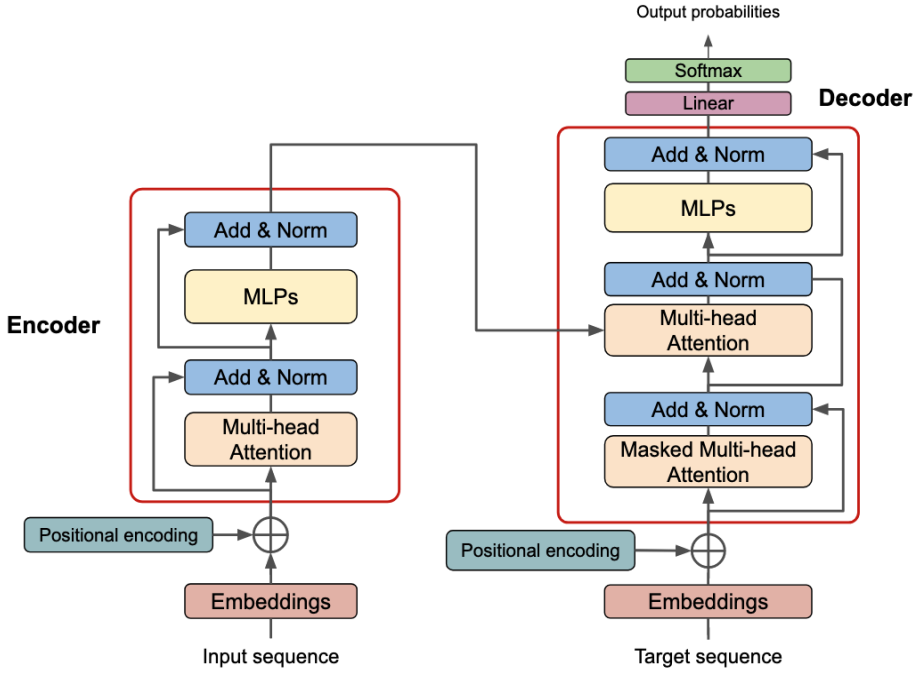


Figure 1: Transformer Architecture (Vaswani et al., 2017)

The Transformer architecture uses positional encoding to model the sequential nature of input sequences. This mechanism incorporates information about the position of each word into its continuous vector representation, allowing the model to capture word order and positional relationships within the sequence.

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{11}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{12}$$

pos:position of the word in the sentence
i: dimension index
d_{model}: embedding dimension

2.1 Self-Attention Mechanism

The most distinctive component of the Transformer architecture is the self-attention mechanism. This mechanism enables each word in a sequence to interact with all other words, effectively capturing contextual dependencies regardless of their distance in the sequence.

Given an input sequence of vectors $X \in R^{(nxd)}$, the model transforms these vectors into three distinct representations:

- Query: $Q = XW^Q$
- Key: $K = XW^K$
- Value: $V = XW^V$

where W^Q, W^K, W^V are learnable parameter matrices.

This setup forms the foundation for computing the self-attention scores, allowing each word to attend to every other word in the sequence and thus aggregate contextual information dynamically.

2.2 Multi-Head Attention

When the self-attention mechanism operates through a single head, it can only capture limited contextual information. To address this, the Transformer architecture performs the self-attention operation in parallel across multiple heads:

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{14}$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{15}$$

This structure enables the model to learn different semantic relationships through distinct attention heads.

The Transformer architecture offers several critical advantages over traditional recurrent neural networks (RNNs, LSTMs, GRUs), establishing a new standard in the field of NLP. The first major advantage is its ability to process data in parallel. Whereas RNN-based models must process input sequences sequentially—making each word's computation dependent on the previous one and thus incurring significant computational time—the Transformer can process all words simultaneously via the self-attention mechanism. This results in dramatically reduced training times and better utilization of parallel hardware architectures such as GPUs.

Secondly, the Transformer excels at modeling long-range dependencies. Although RNNs can, in theory, model long-distance relationships within sequences, in practice, issues such as vanishing and exploding gradients undermine their effectiveness. In contrast, the Transformer's self-attention mechanism allows every word in the sequence to directly attend to every other word, thereby simultaneously capturing both short- and long-range dependencies and producing highly context-sensitive representations.

A third major advantage is computational efficiency. The attention mechanisms used in both the encoder and decoder blocks are based on

intensive matrix operations, which can be efficiently scaled on modern parallel hardware such as GPUs and TPUs. This architecture is particularly well-suited for training very large language models (LLMs), as it utilizes computational resources more efficiently than previous approaches.

Finally, the generality of the Transformer architecture is noteworthy. Although it was initially designed for NLP tasks, it has since been successfully adapted to other domains, including computer vision (e.g., Vision Transformer—ViT), speech processing, biomedical signal analysis, and even protein structure prediction (Lin et al., 2022). This adaptability demonstrates that the Transformer is not merely a language model but a general-purpose deep learning architecture.

3. Language Generation, Summarization, Question Answering, and Sentiment Analysis

Some of the most critical application-oriented tasks in natural language processing (NLP) include language generation, text summarization, question answering, and sentiment analysis. These tasks go beyond mere statistical or structural analysis of language, requiring the modeling of contextual meaning and the production of semantically coherent responses. In recent years, the development of Transformer-based models has led to remarkable performance improvements across each of these tasks (Devlin et al., 2018; Lewis et al., 2019; Raffel et al., 2019).

3.1 Natural Language Generation (NLG)

Natural language generation refers to the process of producing meaningful, fluent, and human-like text from data or semantic representations. While traditional rule-based and statistical language models have limited generative capabilities, the emergence of models such as GPT (Generative Pretrained Transformer) (Radford et al., 2018), BART (Bidirectional and Auto-Regressive Transformers) (Lewis et al., 2019), and T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2019) has revolutionized language generation. These models share the characteristic of being pretrained on large-scale text corpora and can be fine-tuned for specific downstream tasks. As a result, they are capable of producing text that is both contextually coherent and semantically consistent. Today, NLG systems are widely applied in automatic news writing, social media content creation, dialog systems, and data-driven reporting.

3.2 Text Summarization

Text summarization is the task of generating concise and meaningful summaries from lengthy documents. Summarization techniques are generally categorized as either extractive, which selects important sentences directly from the source text, or abstractive, which generates new sentences to express the main ideas. For abstractive

summarization, Transformer-based encoder-decoder models such as T5, BART, and PEGASUS have achieved state-of-the-art results (Lin et al., 2022). These models encode the input text to learn its essential content and then generate a summary via the decoder. Notably, BART demonstrates high performance in summarization by leveraging masked language modeling and learning from noisy inputs. Such systems are extensively used for news summarization, scientific paper condensation, and the shortening of medical records (Khurana et al., 2023).

3.3 Question Answering (QA)

Question answering is an NLP subfield that aims to generate meaningful and accurate answers to questions posed in natural language, based on textual context. Unlike traditional information retrieval systems that rely on simple keyword matching, modern QA systems are able to match contextual meaning. The Transformer-based BERT (Bidirectional Encoder Representations from Transformers) model (Devlin et al., 2018) has achieved groundbreaking results on context-based QA datasets such as SQuAD. In these models, both the question and the context are encoded together, and the attention mechanism enables the model to pinpoint the start and end positions of the answer span within the text. QA systems are widely used in educational technologies, digital assistants, customer service bots, and domain-specific applications in law and healthcare.

3.4 Sentiment Analysis

Sentiment analysis is a classification-based task that seeks to determine whether a text expresses a positive, negative, or neutral sentiment. Traditional approaches relied on techniques such as TF-IDF, Naive Bayes, and SVM; however, with the rise of deep learning, word embedding-based LSTM models and, most recently, Transformer-based models have become prevalent (Lauriola et al., 2022; Sel et al., 2019).

Pretrained models like BERT, RoBERTa, and XLNet can achieve high accuracy in sentiment classification, even with only a few labeled examples (few-shot learning) (Lauriola et al., 2022; Lin et al., 2022). Their key advantage lies in the ability to generate contextual representations, allowing the correct interpretation of word meaning across different contexts. Sentiment analysis plays a crucial role in areas such as social media monitoring, customer feedback analysis, financial news evaluation, and public opinion polling, providing valuable strategic decision support.

Applications such as language generation, summarization, question answering, and sentiment analysis constitute some of the most concrete and practical outputs of NLP for end users. Transformer architectures not only improve accuracy in these tasks but also enable important strategies such as transfer learning, multi-task learning, and few-shot/fine-tuning, broadening their applicability. Consequently, models like

GPT, BERT, and T5 are now extensively used in both research and production environments. The development of Transformer-based architectures has thus redefined the performance limits of NLP on these key tasks.

4. Multimodal NLP Applications

One of the most striking recent developments in artificial intelligence has been the emergence of multimodal systems, which extend natural language processing beyond text to integrate information from images, video, audio, and other data modalities. Multimodal NLP seeks to build models that project the meaning of diverse data types into a common representation space, thereby enabling the discovery of meaningful relationships and inferences across these modalities. This approach emulates the way humans interact with their environment, substantially enhancing the capacity of machines to extract deep, context-rich meaning.

Multimodal NLP systems are particularly prominent in tasks that require visual-textual interaction. For example, Visual Question Answering (VQA) (Antol et al., 2015) is designed to answer natural language questions about images. These systems typically process visual features extracted via convolutional neural networks (CNNs) or Transformer-based architectures, while simultaneously encoding the textual content of the question using language models. Ultimately, the representations from these different modalities are fused through a unified multimodal Transformer network to generate answers. VQA systems are increasingly used in assistive technologies for the visually impaired, automatic image captioning, and digital assistants.

Another major area in multimodal NLP is image captioning, which automates content creation by integrating text and visual data. In these systems, visual features from images are transferred contextually to language models, which then generate fluent textual descriptions. With the advent of multimodal pretrained models such as the Vision Transformer (ViT) (Khan et al., 2022) and CLIP (Contrastive Language-Image Pretraining) (Ramesh et al., 2022), the performance of image captioning systems has approached human-level accuracy.

Multimodal NLP is not limited to text and images. Increasingly, systems are being developed that combine video, audio, and textual modalities. For example, automatic video subtitling, content summarization, podcast analysis, sentiment detection, and speaker identification all make effective use of multimodal NLP techniques. These systems typically convert audio data into text using speech recognition models, which are then analyzed by language models to extract information such as sentiment, topic, or speaker identity.

In summary, multimodal NLP applications significantly enhance the human-like perceptual and reasoning capabilities of AI systems by

meaningfully integrating diverse sensory modalities. Success in this field—particularly with the advent of multimodal Transformers such as CLIP, DALL-E, Flamingo, BLIP, and LLaVA (Alayrac et al., 2022; Ramesh et al., 2022)—has enabled innovative applications that were previously unachievable. As large-scale, multimodal pretrained models become more prevalent, it is expected that both the range and the effectiveness of these applications will continue to grow.

5. Explainable Artificial Intelligence (XAI) and NLP

With the increasing prevalence of deep learning-based models in NLP, understanding and interpreting the decisions made by these models has become ever more critical. The complexity and scale of large Transformer-based language models—such as GPT (Radford et al., 2018) and BERT (Devlin et al., 2018)—which comprise millions of parameters, make their decisions particularly opaque and challenging to explain. This poses significant challenges for ensuring the reliability, fairness, and transparency of NLP applications. Here, explainable artificial intelligence (XAI) methods play a crucial role in making the decision-making processes of NLP models more transparent (Danilevsky et al., 2020).

XAI methods in NLP provide mechanisms for revealing which words, expressions, or contexts models focus on when arriving at specific outcomes. Among these, general-purpose explanation tools such as SHAP (SHapley Additive exPlanations) (Mosca et al., 2022) and LIME (Local Interpretable Model-agnostic Explanations), as well as visualization techniques for attention mechanisms, are particularly notable. For instance, the self-attention mechanisms used in Transformer-based models allow for the visualization of the relative importance assigned to each word, thus enabling users to better understand model decisions. Such visualizations are frequently used in sentiment analysis, text classification, and question answering to provide insight into the model's reasoning.

The importance of explainability in NLP extends beyond technical considerations to encompass ethical and social responsibility. In contexts such as automated job application screening or legal decision-support systems, it is essential to clearly articulate the reasoning behind model decisions. Otherwise, bias, unfair outcomes, or unintended consequences may arise. By increasing transparency, XAI methods help identify and correct such biases, which is crucial for wider societal acceptance and trust in NLP technologies.

In conclusion, the effective use of explainable AI techniques in NLP not only clarifies model decision-making and improves accuracy, but is also vital for fulfilling ethical and social responsibilities. For these reasons, the integration of XAI approaches is increasingly becoming a standard part of NLP system development.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., … Simonyan, K. (2022). *Flamingo: a Visual Language Model for Few-Shot Learning*. http://arxiv.org/abs/2204.14198

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., & Parikh, D. (2015). Vqa: Visual question answering. *Proceedings of the IEEE International Conference on Computer Vision*, 2425–2433.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*(Feb), 1137–1155.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Chowdhary, K. R. (2020). Fundamentals of artificial intelligence. In *Fundamentals of Artificial Intelligence*. Springer India. https://doi.org/10.1007/978-81-322-3972-7

Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., & Sen, P. (2020). A survey of the state of explainable AI for natural language processing. *ArXiv Preprint ArXiv:2010.00711*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. *Mlm*. http://arxiv.org/abs/1810.04805

Jurafsky, D., & Martin, J. H. (2025). *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models Third Edition draft Summary of Contents*.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in Vision: A Survey. *ACM Computing Surveys*, *54*(10s), 1–41. https://doi.org/10.1145/3505244

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, *82*(3), 3713–3744. https://doi.org/10.1007/s11042-022-13428-4

Lauriola, I., Lavelli, A., & Aiolli, F. (2022). An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools. *Neurocomputing*, *470*, 443–456. https://doi.org/10.1016/j.neucom.2021.05.103

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ArXiv Preprint ArXiv:1910.13461*.

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, *3*, 111–132. https://doi.org/10.1016/j.aiopen.2022.10.001

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Boolean retrieval. *Introduction to Information Retrieval*, 1–18.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. http://arxiv.org/abs/1301.3781

Mikolov, T., Karafiát, M., Burget, L., Jan, C., & Khudanpur, S. (2010). Recurrent neural network based language model. *Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010*, *September*, 1045–1048.

Mosca, E., Szigeti, F., Tragianni, S., Gallagher, D., & Groh, G. (2022). SHAP-based explanation methods: a review for NLP interpretability. *Proceedings of the 29th International Conference on Computational Linguistics*, 4593–4603.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2018). *Language Models are Unsupervised Multitask Learners*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. http://arxiv.org/abs/1910.10683

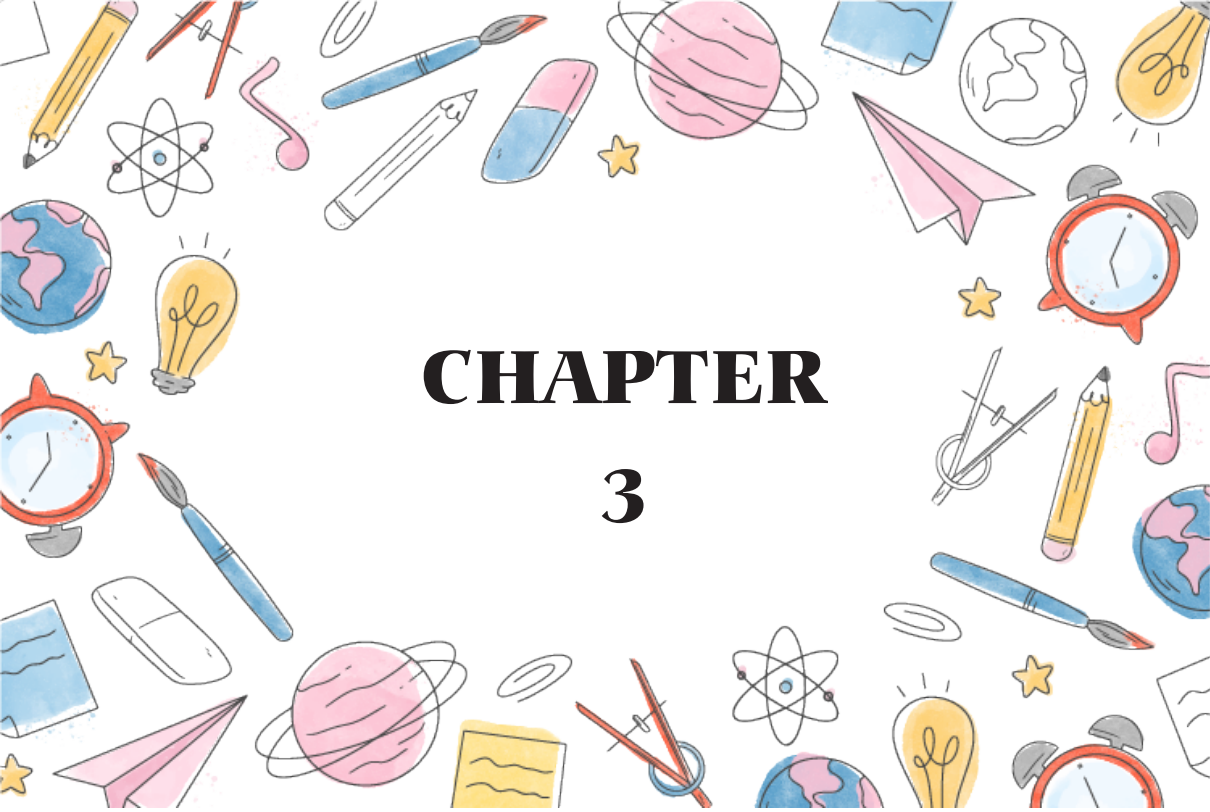
Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). *Hierarchical Text-Conditional Image Generation with CLIP Latents*. http://arxiv.org/abs/2204.06125

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional Recurrent Neural Networks. In *IEEE TRANSACTIONS ON SIGNAL PROCESSING* (Vol. 45, Issue 11).

Sel, İ., Karci, A., & Hanbay, D. (2019). Feature selection for text classification using mutual information. *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 1–4.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*, 5999–6009.



CHAPTER 3

GENERATIVITY IN LANGUAGE AND VISION: THE EVOLUTION, ARCHITECTURES, AND EMERGING PARADIGMS OF LARGE MODELS

İlhami SEL[1]

1 Engineer, Ministry of National Education, ilhamisel23@gmail.com,
ORCID: 0000-0003-0222-7017

One of the most significant breakthroughs in artificial intelligence (AI) in recent years has been the emergence of generative models and their integration with large-scale architectures. While traditional AI systems have primarily focused on tasks such as data classification, clustering, or labeling, generative AI systems have advanced to a stage where they can autonomously produce original and creative content—spanning text, images, audio, and video—at a level reminiscent of human output. This progress has been made possible by combining statistical learning techniques with probability theory, enabling models to learn the underlying distribution of observed data and generate new samples from this learned distribution. Consequently, these systems are no longer limited to reproducing existing data but are capable of synthesizing novel and innovative outputs.

The transformative impact of generative models has been further amplified with the advent of large language models (LLMs) and vision-language models (VLMs). For instance, LLMs such as GPT, T5, and LLaMA, with their billions of parameters, are trained on vast text and multimodal datasets, allowing them to generate contextually coherent, meaningful, and creative texts. Leveraging zero-shot and few-shot learning capabilities, these models can successfully perform complex tasks for which they have not been explicitly trained. Similarly, advanced multimodal models like CLIP, Flamingo, and LLaVA have bridged the gap between textual and visual modalities, significantly enhancing the capabilities of AI systems in understanding and generating multimodal information.

As a result, generative AI has become not merely a technological innovation, but a transformative tool for scientific discovery, knowledge production, creative industries, and digital media. The opportunities presented by these models—especially in creative sectors, scientific research, and knowledge-intensive services—are pushing the boundaries of human–machine collaboration and driving substantial social, economic, and cultural change.

Moreover, the increasing multimodality and scalability of generative AI empower machines to better interpret contextual information and play more effective roles in complex tasks. In sum, generative artificial intelligence has evolved beyond a mere technological trend, inaugurating a paradigmatic transformation in the ways knowledge is generated, disseminated, and reconstructed across a wide array of disciplines.

1. Large Language Models (LLMs)

In recent years, the field of natural language processing (NLP) has undergone a profound transformation, with large language models (LLMs) emerging as the central drivers of this shift. These models, equipped with

billions of parameters and constructed using deep learning architectures, are trained on enormous datasets and have achieved remarkable results on a wide variety of complex language tasks, surpassing the capabilities of previous approaches. Unlike earlier models that primarily captured shallow statistical patterns, LLMs are designed to understand and generate language at multiple levels, including semantics, syntax, pragmatics, and broader contextual nuances.

The remarkable performance of LLMs is rooted in a two-stage training paradigm: pretraining followed by task-specific fine-tuning. During the pretraining phase, the model is exposed to massive collections of text through self-supervised tasks, such as masked language modeling or next-word prediction. This enables the model to acquire fundamental statistical and semantic structures of language. The subsequent fine-tuning stage involves training the model on smaller, labeled datasets tailored to particular NLP tasks such as summarization, question answering, or classification. This approach allows the general language competencies acquired during pretraining to be effectively transferred to specialized applications (Chang et al., 2024).

The effectiveness of LLM training is shaped not only by the sheer volume of data but also by the specific strategies and architectures employed. For instance, autoregressive models (e.g., the GPT series (Radford et al., 2018)) predict each output token based solely on preceding tokens, whereas encoder-decoder models (such as T5 (Raffel et al., 2019)) seek to model the full structural relationships between inputs and outputs. These architectural choices have direct implications for model performance in various tasks like text generation, translation, and question answering.

Furthermore, to ensure stable learning and optimize resource usage during LLM training, a number of advanced optimization techniques have become standard practice. These include:

- Learning rate scheduling: This method adjusts the step size of weight updates dynamically throughout training. Using a constant learning rate can be inefficient or unstable, especially for large models. Scheduling methods such as cosine annealing or inverse square root decay enable rapid early learning and smooth convergence in later stages.
- Warm-up strategies: Particularly vital for very large models, warm-up gradually increases the learning rate at the beginning of training, mitigating instabilities and protecting model weights from abrupt changes.
- Gradient clipping: Large gradient values can destabilize learning by causing excessive parameter updates. By capping gradients at a threshold, this technique effectively prevents the “exploding gradients” problem.

- Layer normalization: To address the accumulation of minor deviations across many layers—which can disrupt stable learning—layer normalization regularizes activations at each layer, promoting more reliable convergence.
- Mixed-precision training: By carrying out certain computations in lower-precision formats (e.g., FP16) while retaining full precision (FP32) for critical calculations, this approach significantly reduces memory usage and training time, enabling larger models to be trained on available hardware without compromising accuracy.

Each of these techniques plays a crucial role in overcoming common challenges in training models with billions of parameters, such as divergence (training instability) or overfitting (excessive adaptation to training data). As a result, successful LLM training demands not only extensive data and powerful hardware but also carefully designed optimization strategies (Touvron et al., 2023).

The performance of large language models also heavily depends on the scale and diversity of the training corpus. Data sources typically include web crawls (such as Common Crawl), Wikipedia, books, news articles, academic papers, and even code repositories. For instance, GPT-3 was trained on roughly 570 GB of compressed text, reflecting the broad informational spectrum of the internet (Raffel et al., 2019). In contrast, LLaMA models are trained exclusively on publicly available, high-quality datasets, thereby enhancing reproducibility and adherence to open science principles (Touvron et al., 2023). Nevertheless, it is vital to consider the ethical and reliability aspects of pretraining data, as biases, inaccuracies, or harmful content within the corpus can directly influence model outputs. Thus, data cleaning, filtering, and quality assurance procedures are essential steps in the model training process.

Training LLMs is computationally intensive, often requiring clusters of GPUs or TPUs with hundreds of gigabytes of memory and thousands of processing cores. For example, GPT-3's training involved thousands of A100 GPUs on a supercomputing infrastructure, and the process took several weeks (Chang et al., 2024). The considerable energy consumption and associated carbon footprint have also fueled ongoing debates. Accordingly, efficiency-focused training strategies are gaining increasing attention. Chinchilla Scaling Laws, for instance, demonstrate that models with fewer parameters can achieve superior results when trained on larger datasets, enhancing both performance and computational efficiency (Hoffmann et al., 2022).

While parameter count is a common way to describe LLMs, increases in parameter size do not always yield proportional improvements in performance. For example, although GPT-3 contains 175 billion parameters, this scale brings not only greater capacity but also higher risks

of overfitting, slower inference, and greater hardware demands. Consequently, the trend in new model development is toward architectures that can match or surpass the performance of larger models with fewer parameters, as seen in the case of LLaMA-13B achieving results on par with GPT-3 (Touvron et al., 2023).

1.1 GPT (Generative Pre-trained Transformer)

The Generative Pre-trained Transformer (GPT), developed by OpenAI, is a groundbreaking autoregressive language model that has significantly reshaped the landscape of natural language processing (NLP). Its successive iterations—GPT-2, GPT-3, and GPT-4—have progressively increased in both scale and performance, with GPT-3 featuring an architecture comprising 175 billion parameters. These models are initially pre-trained on vast text corpora using self-supervised learning and subsequently demonstrate high accuracy in a range of downstream tasks through few-shot or even zero-shot learning, where only a handful of examples (or none) are needed to generalize to novel tasks (Chang et al., 2024).

Mathematically, a GPT model estimates the joint probability of a given input sequence $X = (x_1, x_2, \ldots, x_n)$ using the following autoregressive factorization:

$$P(X) = \prod_{i=1}^{n} P(x_i \mid x_1, x_2, \ldots, x_{i-1}) \quad (1)$$

This formulation enables GPT to generate coherent outputs by predicting each token based on all preceding tokens in the sequence. Architecturally, GPT is built on the Transformer decoder framework and utilizes causal attention mechanisms that operate in a unidirectional (left-to-right) manner, allowing it to excel in generative language modeling tasks.

In real-world applications, GPT-based models have been widely adopted in diverse domains. They power intelligent assistants (e.g., ChatGPT), support automated content creation, enable advanced code generation, and even assist in academic writing. Notably, GPT-3 has shown proficiency in tasks such as summarization, machine translation, question answering, and dialogue generation, often achieving impressive performance without explicit task-specific fine-tuning.

1.2 T5 (Text-to-Text Transfer Transformer)

The Text-to-Text Transfer Transformer (T5), developed by Google, introduces a unified framework for solving diverse natural language processing (NLP) tasks under a single architecture (Raffel et al.,

2019). Unlike models that rely on task-specific structures or objective formulations, T5 recasts all NLP tasks—including classification, translation, summarization, and question answering—as a text-to-text problem: the input is a textual description of the task, and the output is a corresponding textual response.
For instance, a sentiment analysis task is structured as follows:

- Input: "sentiment analysis: I love this movie."
- Output: "positive"

This design enables the model to operate across a wide range of NLP applications using the same input-output interface, making it highly adaptable and modular.
Architecturally, T5 is based on the standard encoder–decoder Transformer model, wherein the encoder processes the input sequence and the decoder generates the output conditioned on the encoder's representations. This allows T5 to benefit from both bidirectional input understanding and autoregressive output generation, a combination particularly effective in tasks that require both comprehension and fluent generation.

The training of T5 can be summarized in three major phases:

- Pretraining: The model is trained on a large-scale masked language modeling task, where spans of text are replaced with sentinel tokens and the model learns to reconstruct them.
- Data: Training is conducted on the Colossal Clean Crawled Corpus (C4), a massive and curated web-scale dataset that emphasizes data quality and cleanliness (Dodge et al., 2021).
- Fine-tuning: After pretraining, the model is adapted to a range of specific downstream tasks via transfer learning, typically using supervised datasets formulated in the same text-to-text paradigm.

T5 is released in multiple model sizes, enabling users to balance performance and resource constraints:

- T5-Small (60 million parameters)
- T5-Base (220M)
- T5-Large (770M)
- T5-3B (3 billion)
- T5-XXL (11 billion parameters)

One of T5's most impactful contributions lies in its support for multi-task learning. By pretraining the model to perform multiple tasks simultaneously under a shared objective, T5 enhances generalization across domains and improves sample efficiency. Moreover, its ability to handle explicit task definitions in natural language makes it particularly

effective for prompt-based applications, aligning it with recent trends in instruction-tuned and generative AI systems.

T5 has been widely adopted in areas such as:

- Text summarization (e.g., news, legal, and medical documents)
- Machine translation (multilingual settings)
- Grammatical error correction
- Paraphrasing and rewriting
- Question answering over structured and unstructured data

Its flexible design also makes it a suitable base model for further enhancements, such as T5.1.1 (an optimized training version), mT5 (multilingual variant), and Flan-T5, which incorporates instruction tuning to improve performance in zero-shot and few-shot scenarios.

Compared to decoder-only models like GPT, T5's bidirectional encoder allows it to better understand full input sequences, which can lead to superior results in comprehension-heavy tasks. Unlike LLaMA, which is primarily optimized for efficient generative inference, T5 emphasizes structured instruction-following and interpretability. Furthermore, T5's open availability and detailed documentation have made it a foundational model in academic research and industry deployments.

1.3 LLaMA (Large Language Model Meta AI)

LLaMA (Large Language Model Meta AI), developed by Meta AI, is a family of open-access large language models designed in accordance with open science principles. Unlike proprietary models trained on undisclosed datasets, LLaMA is trained exclusively on publicly available, high-quality corpora, ensuring transparency and reproducibility in both research and application contexts (Touvron et al., 2023).

The LLaMA model suite includes variants with 7B, 13B, 33B, and 65B parameters. One of its most striking attributes is its ability to achieve competitive or even superior performance with significantly fewer parameters compared to larger models. For instance, LLaMA-13B has outperformed GPT-3 (with 175 billion parameters) on several standard benchmarks. This efficiency is largely attributed to the adoption of optimal scaling strategies, as outlined in the Chinchilla Scaling Laws, which demonstrate that training smaller models for longer with more high-quality data yields better results than simply increasing parameter counts (Hoffmann et al., 2022).

In addition to its efficient training regimen, LLaMA incorporates several architectural improvements within the Transformer framework that contribute to its robustness and efficiency (Touvron et al., 2023):

- RMSNorm (Root Mean Square Layer Normalization): Replaces standard layer normalization, offering improved training stability and computational simplicity.
- SwiGLU Activation Function: An enhanced gated linear unit that improves model expressivity and learning dynamics.
- Rotary Positional Embeddings (RoPE): Used in place of absolute position encodings, enabling better generalization and efficient attention across long sequences.

LLaMA employs Byte-Pair Encoding (BPE) for tokenization (Sennrich et al., 2016), a widely used subword segmentation technique that balances vocabulary size and token granularity. This design contributes to both training and inference efficiency, making LLaMA suitable for large-scale deployment and constrained-resource environments.

LLaMA is explicitly optimized for low-cost inference, which allows it to be deployed more broadly—including in academic research, startups, and smaller-scale production systems. Its open-source licensing and availability for research use have made it one of the most widely adopted models in 2023–2024, particularly for building instruction-tuned variants (e.g., Alpaca, Vicuna, LLaMA-Adapter) and for experimentation with parameter-efficient fine-tuning (PEFT) techniques such as LoRA.

Furthermore, LLaMA models strike a balance between performance, transparency, and hardware efficiency, offering a viable alternative to larger commercial models like GPT-3. Its public availability has democratized access to powerful language models and fostered a wide ecosystem of downstream applications and model extensions.

LLaMA has been successfully used in:

- Instruction-tuned conversational agents (e.g., Vicuna, OpenAssistant)
- Domain-specific adaptations (e.g., biomedical, legal NLP)
- Education and learning tools with on-device language capabilities
- Research in low-resource languages and PEFT methods

LLaMA exemplifies a shift in large language model development: prioritizing open access, data transparency, and computational efficiency. Its scalable architecture, combined with intelligent training and inference optimizations, enables it to deliver state-of-the-art performance in both academic and real-world settings—often rivaling much larger models at a fraction of the computational cost. As such, LLaMA has become a foundational model for researchers seeking to balance performance with accessibility.

Table 1: Use-Case Suggestions

Use Case	Recommended Models
Multilingual NLP	mT5, GPT-4, LLaMA 2
On-device inference (low resource)	Gemma, LLaMA 2 7B, Mistral 7B
Instruction following	Flan-T5, LLaMA 2 Chat, GPT-4
Dialogue and chatbot agents	GPT-4, LLaMA 2 Chat, Vicuna (based on LLaMA)
Open-source R&D	LLaMA 2, Mistral, Flan-T5, Gemma
Multimodal tasks	GPT-4V (GPT-4 with vision), Flamingo, LLaVA

Table 2: Extended Comparison of Major and Emerging Large Language Models

Model	Developer	Architecture	Parameter Sizes	Training Objective	Open Source	Notable Features
GPT-3	OpenAI	Transformer (Decoder-only)	175B	Next-token prediction	No	Powerful few-shot generalization, closed data
GPT-4	OpenAI	Multimodal Transformer	~1T (unconfirmed)	Next-token prediction + multimodal	No	Strong reasoning, vision-text support, top-tier benchmark performance
T5	Google	Transformer (Encoder–Decoder)	60M – 11B	Span corruption (text-to-text)	Yes	Task-unified format, C4 dataset, multitask learning

Flan-T5	Google	Encoder–Decoder Transformer	80M – 11B	Instruction-tuned text-to-text	Yes	Better generalization on instructions, zero-shot prompt following
LLaMA	Meta	Transformer (Decoder-only)	7B, 13B, 33B, 65B	Autoregressive next-token	Yes (research)	Efficient small models, public training data
LLaMA 2	Meta	Transformer (Decoder-only)	7B, 13B, 70B	Autoregressive next-token	Yes	Chat-tuned, safety and instruction improvements, fine-tuned variants (Chat, Code)
Mistral	Mistral AI	Transformer (Decoder-only)	7B, Mixtral 12.9B (MoE)	Autoregressive (dense + MoE)	Yes	Sparse Mixture-of-Experts (MoE), high efficiency, strong open-source performance
Gemma	Google DeepMind	Transformer (Decoder-only)	2B, 7B	Next-token prediction	Yes	Lightweight, fine-tuned for alignment, comparable to LLaMA 2 in small scale

2. Vision–Language Models

Vision–Language Models (VLMs) refer to deep learning architectures that are trained to jointly process and align different sensory modalities, primarily visual inputs (e.g., images, videos) and natural language (e.g., text, questions, captions). These models are inspired by the human cognitive ability to integrate visual and linguistic information to perceive, reason about, and interact with the environment. In artificial intelligence systems, achieving such cross-modal integration is essential for success in complex tasks like image captioning, visual question answering (VQA), and text-to-image retrieval (Zhang et al., 2024). The core objective of VLMs is to learn a shared semantic embedding space where visual and linguistic representations can be meaningfully compared or fused.

Most vision–language models are composed of two primary components:

- Vision Encoder: This module transforms low-level pixel data into high-level abstract representations. Common choices include Vision Transformers (ViT) (Dosovitskiy et al., 2020) and ResNet architectures (He et al., 2015), which capture spatial and semantic patterns from images or video frames.
- Language Decoder or Model: Built typically on top of transformer-based architectures (e.g., Vaswani et al., 2017), this component processes natural language input or generates textual output conditioned on visual features, enabling the model to perform tasks like caption generation, reasoning, or retrieval.

2.1 Flamingo: In-Context Learning with Vision–Language Models

Flamingo is a vision–language model developed by DeepMind that introduces multimodal in-context learning capabilities—enabling models to perform new visual-language tasks using just a few examples at inference time, without the need for gradient-based fine-tuning (Alayrac et al., 2022). Designed to build upon models such as CLIP, Flamingo offers a more dynamic framework for context-aware generation across modalities.

Flamingo follows a modular architecture consisting of three main components:

- Visual Encoder: A frozen image encoder based on CLIP (Radford et al., 2021), which generates high-level visual embeddings from image inputs.

- Language Model: A large, frozen decoder-only transformer (e.g., Chinchilla, Hoffmann et al., 2022), responsible for text generation and reasoning.

Perceiver Resampler: An intermediate module that transforms the CLIP visual embeddings into a compact set of latent tokens. These are optimized to interface with the language model via attention, allowing effective cross-modal fusion.

Similar to how GPT-3 performs few-shot and zero-shot learning using text prompts, Flamingo is capable of prompt-based multimodal learning. During inference, users can provide a context window that includes both visual and textual examples of a task (e.g., image-question-answer triples), followed by a new input for which the model is to generate an output.

This paradigm allows Flamingo to generalize to unseen tasks with minimal supervision. Notably, Flamingo achieves state-of-the-art (SOTA) performance on various multimodal benchmarks in zero-shot or few-shot settings, such as:

- VQA (Visual Question Answering)
- Image Captioning
- Grounded Question Answering
- Multimodal Reasoning Tasks

Unlike models trained with extensive fine-tuning on narrow tasks, Flamingo offers:

- Task flexibility, by accepting open-ended prompts.
- Modality awareness, through explicit attention-based integration of visual and textual streams.
- Parameter efficiency, by keeping large pretrained components frozen and using a lightweight resampler for adaptation.

Its success highlights a broader shift toward generalist AI systems that can adapt to novel tasks and inputs simply by reconfiguring prompts—without retraining.

2.2 CLIP: Contrastive Language–Image Pretraining

CLIP (Contrastive Language–Image Pretraining), developed by OpenAI, is a vision–language model designed to align visual and textual representations within a shared high-dimensional embedding space (Radford et al., 2021). Trained on a massive dataset of image–text pairs collected from the web, CLIP enables semantic understanding across modalities without requiring task-specific supervision.

At the core of CLIP lies a dual encoder architecture, where:

- A vision encoder (e.g., ViT or ResNet) maps the input image into a fixed-length embedding.
- A text encoder (Transformer-based) generates a representation of the paired text.

During training, CLIP optimizes a contrastive loss (Chen et al., 2020) that encourages matched (image, text) pairs to have similar embeddings, while pushing apart the embeddings of mismatched pairs. Specifically, in a mini-batch of size N, the model learns to maximize the cosine similarity of the correct image–text pair among N^2 possible combinations, using multiple negatives from the same batch. This objective enables the model to generalize effectively to unseen tasks via zero-shot learning.

CLIP has demonstrated exceptional performance on a variety of tasks, including:

- Zero-shot image classification: By mapping class labels to textual descriptions (e.g., "a photo of a cat"), CLIP can classify images without fine-tuning.
- Text-to-image retrieval and vice versa: Thanks to the joint embedding space, it can retrieve relevant visuals or captions based on similarity.
- Multimodal understanding: It enables models to reason about and compare cross-modal content without explicit alignment labels.

An additional advantage of CLIP is its ability to leverage web-scale noisy data without requiring manually annotated labels, thereby lowering the data annotation cost during pretraining. This feature has made it highly scalable and suitable for large-scale foundation model pipelines.

2.3 LLaVA: Large Language and Vision Assistant

LLaVA (Large Language and Vision Assistant) is an instruction-following multimodal conversational AI model that enables natural dialogue grounded in visual content (Liu et al., 2023). Developed as an open-source alternative to proprietary vision–language systems, LLaVA integrates a frozen vision encoder (CLIP) with a powerful autoregressive language model (Vicuna), enabling it to generate fluent and semantically rich textual outputs in response to visual inputs.

LLaVA follows a two-stage training pipeline and features:

- A vision encoder (CLIP) that extracts semantic representations from images.

- A language model (Vicuna) fine-tuned to follow instructions and engage in coherent conversation.

A lightweight adapter mechanism to integrate visual embeddings into the language model's context window.

The training process involves:

- Synthetic Instruction Tuning: Using ChatGPT or GPT-4, the developers generated high-quality instruction-following datasets from (image, caption) pairs. These synthetic annotations included questions, reasoning prompts, and explanatory content.
- End-to-End Fine-tuning: The integrated CLIP–Vicuna model was trained on over 158,000 multimodal instruction samples to align visual grounding with language generation.
- Benchmarking: A custom evaluation suite, LLaVA-Bench, was developed to assess model performance on a diverse set of vision–language tasks with varying difficulty levels.

LLaVA has achieved strong performance on multiple benchmarks, including a state-of-the-art accuracy of 92.53% on the ScienceQA dataset when combined with GPT-4. It excels in tasks such as:

- Visual Question Answering (VQA)
- Interactive Visual Dialog
- Multimodal Instruction Following
- Commonsense Reasoning over Images

Unlike models like CLIP that are optimized for retrieval, or Flamingo, which focuses on contextual in-context learning, LLaVA is built for real-time multimodal dialogue—a step toward generalist agents that can perceive and interact using both vision and language in human-aligned ways.

The development of CLIP, Flamingo, and LLaVA collectively illustrates the field's rapid progression toward versatile, human-like AI systems. These models differ in design goals and capabilities:

- CLIP enables semantic alignment through contrastive learning and is effective in zero-shot retrieval.
- Flamingo expands into in-context learning, fusing vision and text for generative tasks without fine-tuning.
- LLaVA adds conversational and instruction-following abilities, enabling interactive multimodal understanding.

Such advancements have been enabled not only by improvements in computational infrastructure but also by strategic architectural modularity, large-scale pretraining, and instructional fine-tuning on high-quality datasets. Future developments are likely to expand these capabilities to incorporate additional modalities such as audio, video, and even sensorimotor data, bringing us closer to foundation models for general-purpose intelligence (Zhang et al., 2024).

Table 3: Comparative Overview of Vision–Language Models

Feature	CLIP	Flamingo	LLaVA
Developer	OpenAI	DeepMind	UC Berkeley, Microsoft, CMU, et al.
Year Introduced	2021	2022	2023
Architecture Type	Dual encoder (image + text)	Frozen CLIP + Transformer Decoder + Perceiver Resampler	CLIP vision encoder + Vicuna language model
Fusion Strategy	Contrastive alignment	Cross-attention via latent resampler	Feature injection via adapter layers
Training Objective	Contrastive loss (image–text alignment)	Multimodal in-context learning	Instruction tuning with synthetic vision–language prompts
Pretraining Data	400M image–text pairs from web	Web-scale data, CLIP embeddings, GPT-generated prompts	158K image–instruction samples using GPT-generated labels
Modality Handling	Independent encoding of modalities	Fusion through shared attention interface	Visual embeddings injected into LM token stream
Core Capabilities	Zero-shot classification, retrieval	Prompt-based generative tasks, few-shot learning	Vision-grounded conversation, VQA, instruction following
Zero-Shot Performance	Strong on image classification & retrieval	SOTA on VQA and captioning	92.53% accuracy on ScienceQA (with GPT-4)

Openness	Partially open (models not fully released)	Closed-source	Open-source
Limitations	No generation capability, lacks interaction	Large size, non-public weights	Sensitive to image–text alignment quality, limited memory

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., … Simonyan, K. (2022). *Flamingo: a Visual Language Model for Few-Shot Learning*. http://arxiv.org/abs/2204.14198

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2024). A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, *15*(3). https://doi.org/10.1145/3641289

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). *A Simple Framework for Contrastive Learning of Visual Representations*. http://arxiv.org/abs/2002.05709

Dodge, J., Sap, M., Marasović, A., Agnew, W., Ilharco, G., Groeneveld, D., Mitchell, M., & Gardner, M. (2021). *Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus*. http://arxiv.org/abs/2104.08758

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. http://arxiv.org/abs/2010.11929

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*. http://arxiv.org/abs/1512.03385

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. de Las, Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. van den, Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., … Sifre, L. (2022). *Training Compute-Optimal Large Language Models*. http://arxiv.org/abs/2203.15556

Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). *Visual Instruction Tuning*. http://arxiv.org/abs/2304.08485

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2023). *GPT-4 Technical Report*. http://arxiv.org/abs/2303.08774

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). *Learning Transferable Visual Models From Natural Language Supervision*. http://arxiv.org/abs/2103.00020

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2018). *Language Models are Unsupervised Multitask Learners*.
Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. http://arxiv.org/abs/1910.10683
Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, *3*, 1715–1725.
Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). *LLaMA: Open and Efficient Foundation Language Models*. http://arxiv.org/abs/2302.13971
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*, 5999–6009.
Zhang, J., Huang, J., Jin, S., & Lu, S. (2024). Vision-Language Models for Vision Tasks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(8), 5625–5644. https://doi.org/10.1109/TPAMI.2024.3369699