DocChat AI

GitHub Repository: https://github.com/topgunprogrammer/DocChat-AI-

DocChat: Local Document Processing & AI Chat Platform

DocChat is a local document processing platform that enables users to upload documents, extract text, and interact with a local AI chatbot powered by Ollama LLM. It preserves document formatting and provides a seamless workflow for both document handling and AI-assisted queries.

Key Features:

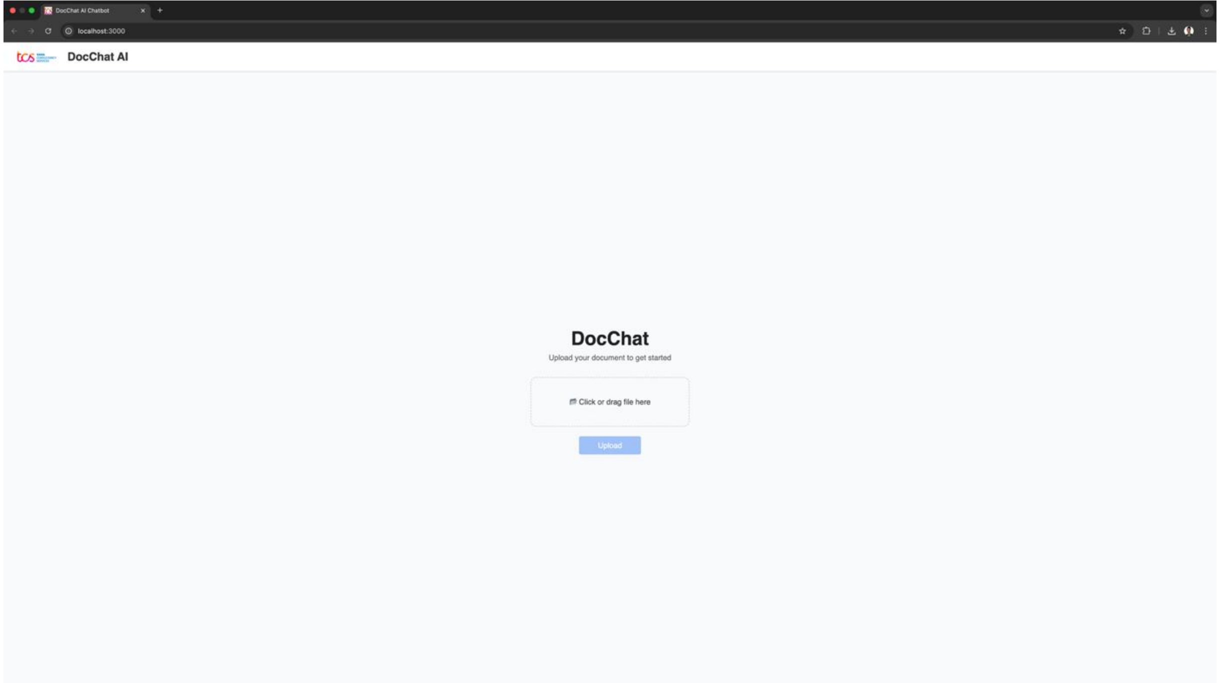
- Upload documents in .txt, .pdf, .docx, and .xlsx formats to local S3 storage.
- Extract text while maintaining original formatting.
- Chat with a local AI model (Ollama LLM) to summarize content or answer document-related questions.
- Fully local AI processing; no cloud AI calls are required.

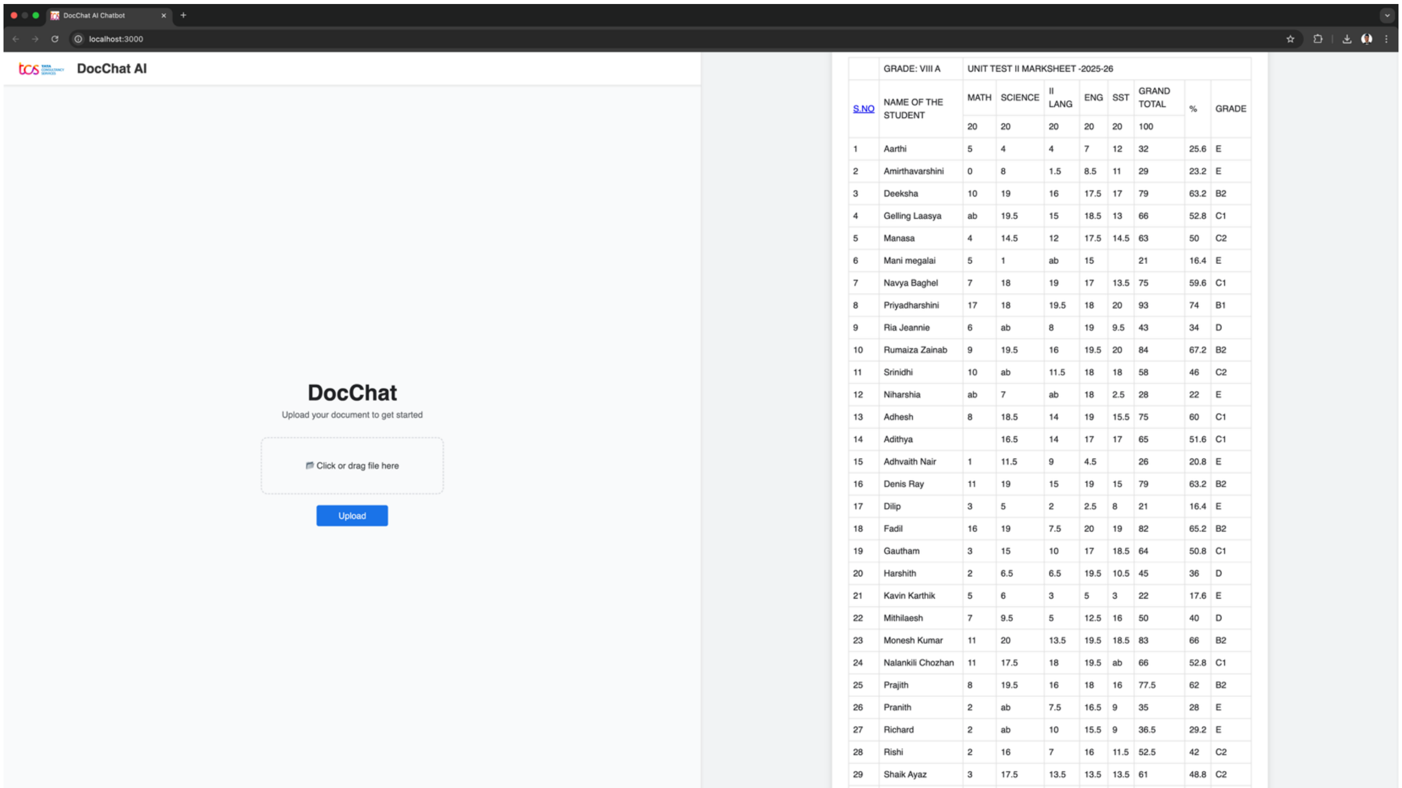
Tech Stack:

- **Frontend:** React with file uploads, chat interface, and document previews. Axios handles API requests.
- **Backend:** Node.js + Express, providing REST APIs for document management and AI interactions. Integrates with local S3 (AWS SDK) and Ollama LLM.
- **Document Parsing:** pdf-parse-fixed, mammoth (DOCX), xlsx (Excel).
- **Local AI Model:** Ollama LLM (llama2) running at http://localhost:11434.

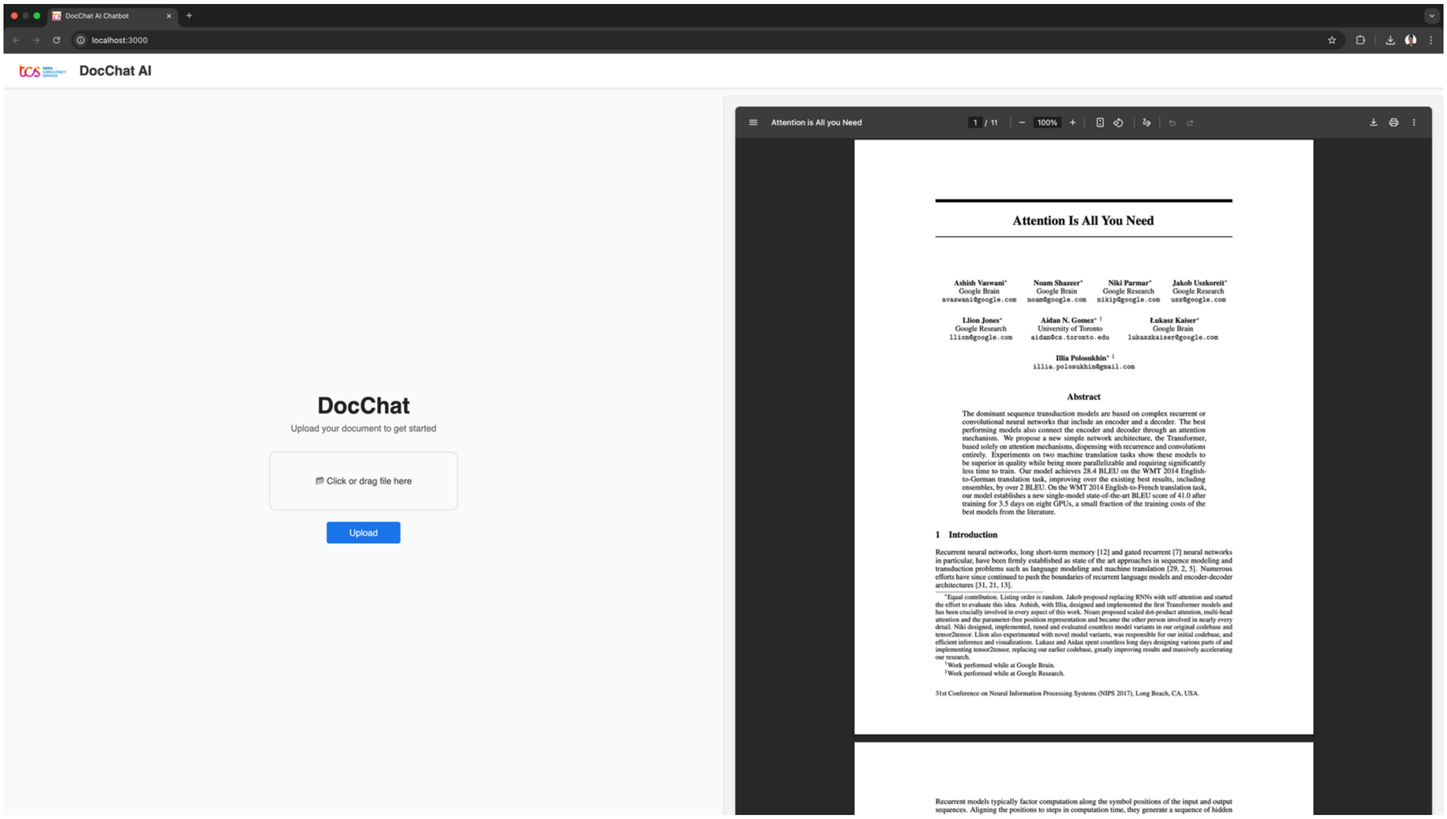
Step 1: Upload Document

- Users can upload any document: `.txt`, `.pdf`, `.docx`, `.xlsx`.
- After selecting a file, a preview is displayed on the right panel.
- This ensures the user can verify the content before processing.





	GRADE: VIII A	UNIT TEST II MARKSHEET -2025-26							
S.NO	NAME OF THE STUDENT	MATH	SCIENCE	II LANG	ENG	SST	GRAND TOTAL	%	GRADE
		20	20	20	20	20	100		
1	Aarthi	5	4	4	7	12	32	25.6	E
2	Amirthavarshini	0	8	1.5	8.5	11	29	23.2	E
3	Deeksha	10	19	16	17.5	17	79	63.2	B2
4	Gelling Laasya	ab	19.5	15	18.5	13	66	52.8	C1
5	Manasa	4	14.5	12	17.5	14.5	63	50	C2
6	Mani megalai	5	1	ab	15		21	16.4	E
7	Navya Baghel	7	18	19	17	13.5	75	59.6	C1
8	Priyadharshini	17	18	19.5	18	20	93	74	B1
9	Ria Jeannie	6	ab	8	19	9.5	43	34	D
10	Rumaiza Zainab	9	19.5	16	19.5	20	84	67.2	B2
11	Srinidhi	10	ab	11.5	18	18	58	46	C2
12	Niharshia	ab	7	ab	18	2.5	28	22	E
13	Adhesh	8	18.5	14	19	15.5	75	60	C1
14	Adithya		16.5	14	17	17	65	51.6	C1
15	Adhvaith Nair	1	11.5	9	4.5		26	20.8	E
16	Denis Ray	11	19	15	19	15	79	63.2	B2
17	Dilip	3	5	2	2.5	8	21	16.4	E
18	Fadil	16	19	7.5	20	19	82	65.2	B2
19	Gautham	3	15	10	17	18.5	64	50.8	C1
20	Harshith	2	6.5	6.5	19.5	10.5	45	36	D
21	Kavin Karthik	5	6	3	5	3	22	17.6	E
22	Mithilaesh	7	9.5	5	12.5	16	50	40	D
23	Monesh Kumar	11	20	13.5	19.5	18.5	83	66	B2
24	Nalankili Chozhan	11	17.5	18	19.5	ab	66	52.8	C1
25	Prajith	8	19.5	16	18	16	77.5	62	B2
26	Pranith	2	ab	7.5	16.5	9	35	28	E
27	Richard	2	ab	10	15.5	9	36.5	29.2	E
28	Rishi	2	16	7	16	11.5	52.5	42	C2
29	Shaik Ayaz	3	17.5	13.5	13.5	13.5	61	48.8	C2



Attention Is All You Need

Ashish Vaswani[*]
Google Brain
avaswani@google.com

Noam Shazeer[*]
Google Brain
noam@google.com

Niki Parmar[*]
Google Research
nikip@google.com

Jakob Uszkoreit[*]
Google Research
usz@google.com

Llion Jones[*]
Google Research
llion@google.com

Aidan N. Gomez[*] [†]
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser[*]
Google Brain
lukaszkaiser@google.com

Illia Polosukhin[*] [‡]
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1 Introduction

Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [29, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [31, 21, 13].

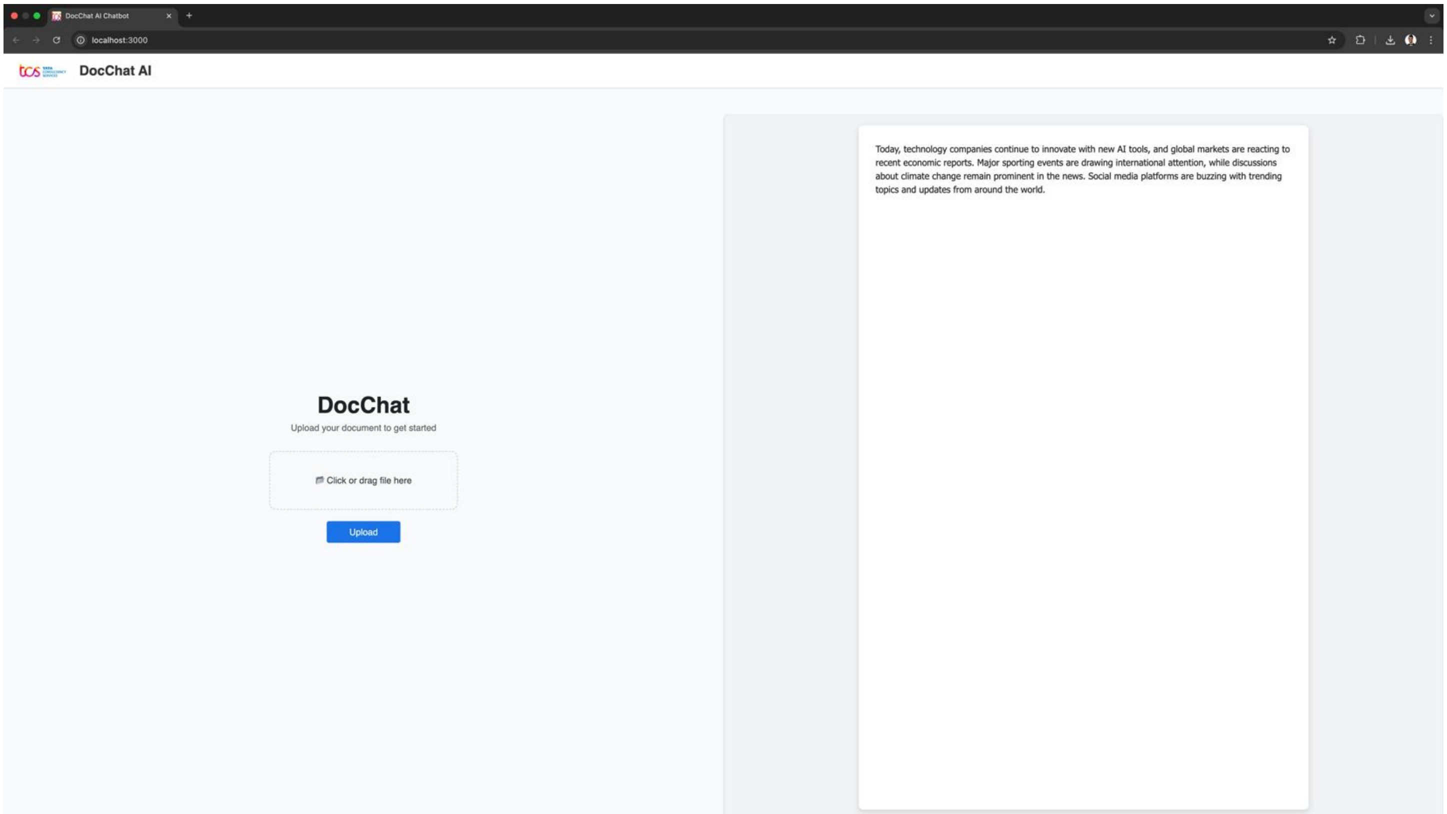
[*] Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

[†] Work performed while at Google Brain.

[‡] Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

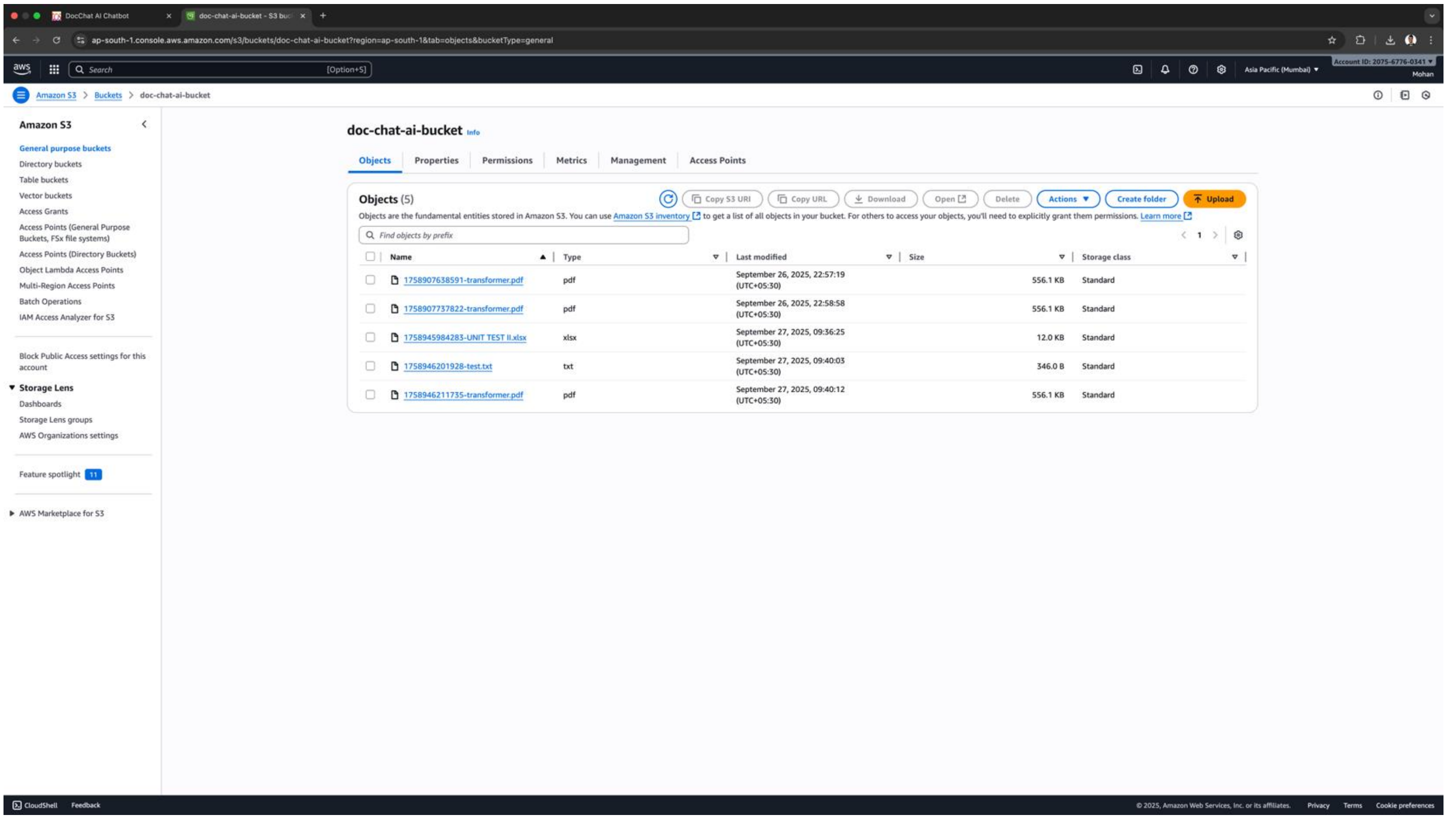
Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden



Today, technology companies continue to innovate with new AI tools, and global markets are reacting to recent economic reports. Major sporting events are drawing international attention, while discussions about climate change remain prominent in the news. Social media platforms are buzzing with trending topics and updates from around the world.

Step 2: Upload to S3

- Once uploaded, the document is stored in the configured S3 bucket (or localstack for testing).
- The uploaded document can be verified in the S3 storage.
- This ensures secure storage and easy retrieval.


DocChat AI Chatbot
doc-chat-ai-bucket - S3 buc
ap-south-1.console.aws.amazon.com/s3/buckets/doc-chat-ai-bucket?region=ap-south-1&tab=objects&bucketType=general
Search [Option+S]
Asia Pacific (Mumbai)
Amazon S3 > Buckets > doc-chat-ai-bucket
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Vector buckets
Access Grants
Access Points (General Purpose Buckets, FSx file systems)
Access Points (Directory Buckets)
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

doc-chat-ai-bucket

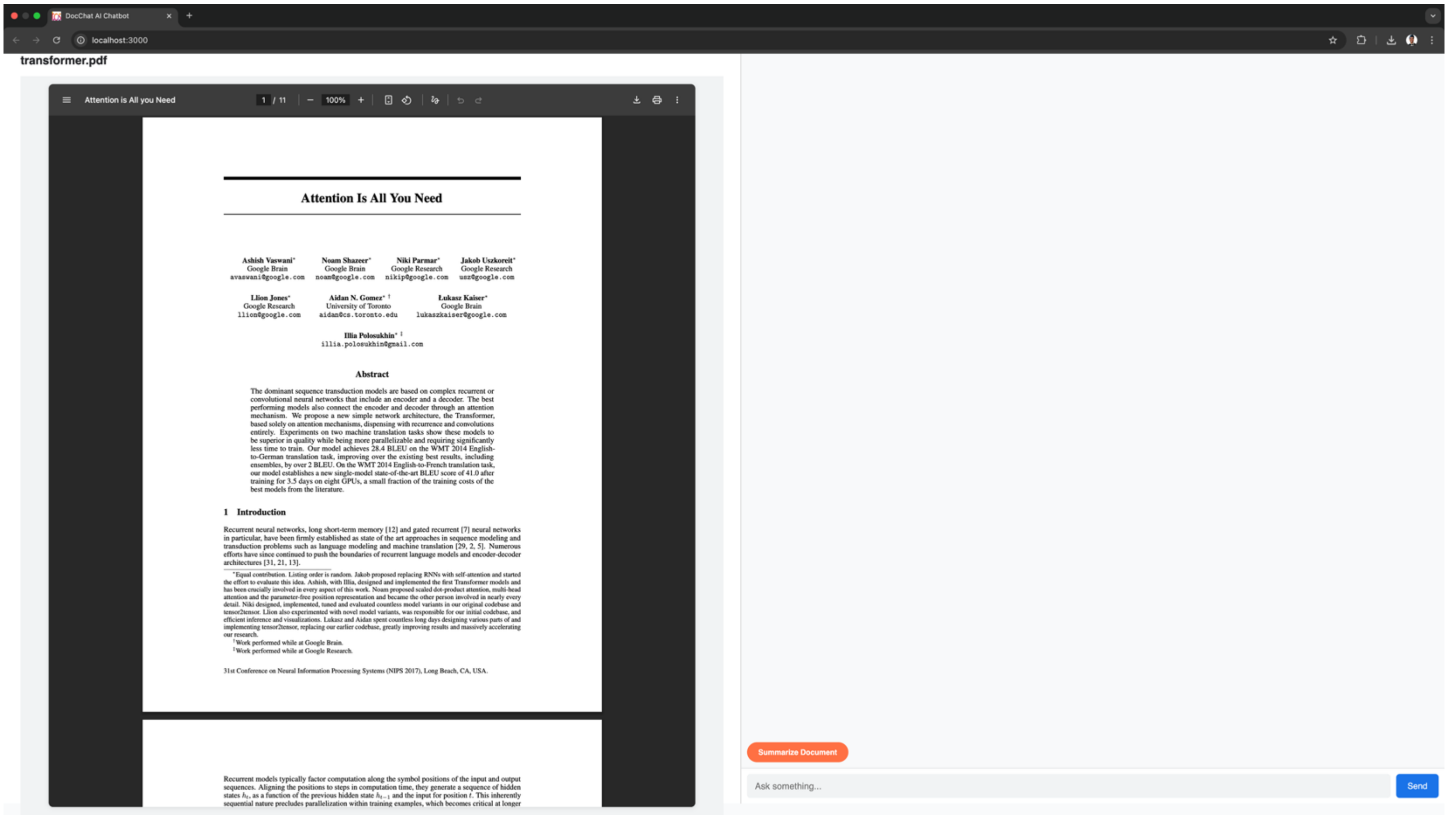
Objects (5)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
1758907638591-transformer.pdf	pdf	September 26, 2025, 22:57:19 (UTC+05:30)	556.1 KB	Standard
1758907737822-transformer.pdf	pdf	September 26, 2025, 22:58:58 (UTC+05:30)	556.1 KB	Standard
1758945984283-UNIT TEST II.xlsx	xlsx	September 27, 2025, 09:36:25 (UTC+05:30)	12.0 KB	Standard
1758946201928-test.txt	txt	September 27, 2025, 09:40:03 (UTC+05:30)	346.0 B	Standard
1758946211735-transformer.pdf	pdf	September 27, 2025, 09:40:12 (UTC+05:30)	556.1 KB	Standard

Step 3: View & Chat Page

- The document is displayed on the left panel for easy reference.
- The right panel hosts the AI chat interface.



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1 Introduction

Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [29, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [31, 21, 13].

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

‡Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states h_t, as a function of the previous hidden state h_{t-1} and the input for position t. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer

Step 4: Summarize Document

- The **Summarize** button generates a concise summary of the document.
- This provides a quick overview without reading the entire content.

Step 5: Chat with Local AI

- Users can interact with the AI to ask questions or get detailed explanations.
- The AI is powered by **Ollama LLM**, running entirely **locally**, ensuring **full data privacy**.
- No cloud AI calls are required, keeping your documents secure.

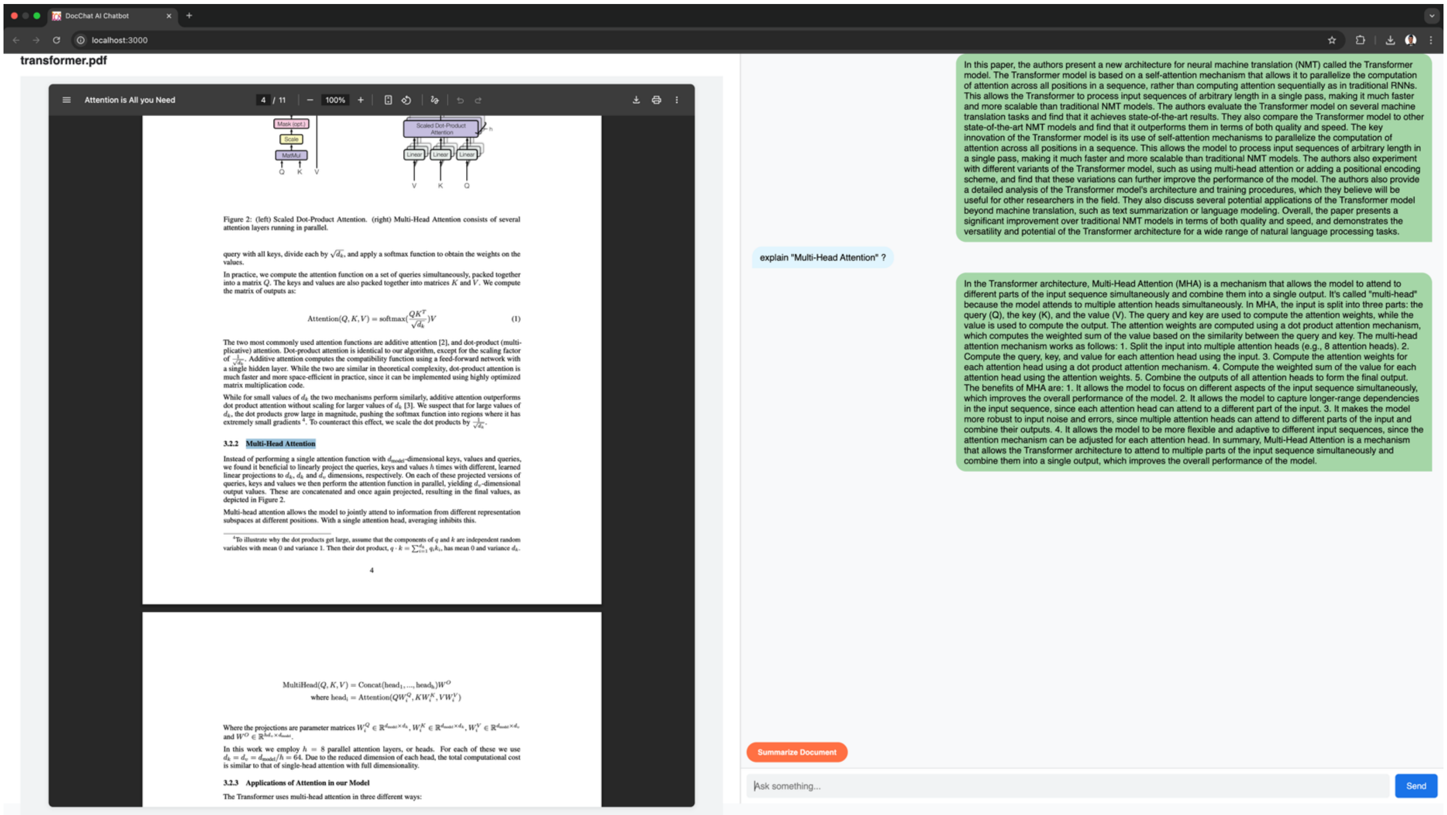


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values.

In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V. We compute the matrix of outputs as:

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

The two most commonly used attention functions are additive attention [2], and dot-product (multiplicative) attention. Dot-product attention is identical to our algorithm, except for the scaling factor of $\frac{1}{\sqrt{d_k}}$. Additive attention computes the compatibility function using a feed-forward network with a single hidden layer. While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code.

While for small values of d_k the two mechanisms perform similarly, additive attention outperforms dot product attention without scaling for larger values of d_k [3]. We suspect that for large values of d_k, the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients [4]. To counteract this effect, we scale the dot products by $\frac{1}{\sqrt{d_k}}$.

3.2.2 Multi-Head Attention

Instead of performing a single attention function with d_{model}-dimensional keys, values and queries, we found it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding d_v-dimensional output values. These are concatenated and once again projected, resulting in the final values, as depicted in Figure 2.

Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this.

[4] To illustrate why the dot products get large, assume that the components of q and k are independent random variables with mean 0 and variance 1. Then their dot product, $q \cdot k = \sum_{i=1}^{d_k} q_i k_i$, has mean 0 and variance d_k.

4

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\mathrm{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\mathrm{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\mathrm{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\mathrm{model}}}$.

In this work we employ $h = 8$ parallel attention layers, or heads. For each of these we use $d_k = d_v = d_{\mathrm{model}}/h = 64$. Due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality.

3.2.3 Applications of Attention in our Model

The Transformer uses multi-head attention in three different ways:

In this paper, the authors present a new architecture for neural machine translation (NMT) called the Transformer model. The Transformer model is based on a self-attention mechanism that allows it to parallelize the computation of attention across all positions in a sequence, rather than computing attention sequentially as in traditional RNNs. This allows the Transformer to process input sequences of arbitrary length in a single pass, making it much faster and more scalable than traditional NMT models. The authors evaluate the Transformer model on several machine translation tasks and find that it achieves state-of-the-art results. They also compare the Transformer model to other state-of-the-art NMT models and find that it outperforms them in terms of both quality and speed. The key innovation of the Transformer model is its use of self-attention mechanisms to parallelize the computation of attention across all positions in a sequence. This allows the model to process input sequences of arbitrary length in a single pass, making it much faster and more scalable than traditional NMT models. The authors also experiment with different variants of the Transformer model, such as using multi-head attention or adding a positional encoding scheme, and find that these variations can further improve the performance of the model. The authors also provide a detailed analysis of the Transformer model's architecture and training procedures, which they believe will be useful for other researchers in the field. They also discuss several potential applications of the Transformer model beyond machine translation, such as text summarization or language modeling. Overall, the paper presents a significant improvement over traditional NMT models in terms of both quality and speed, and demonstrates the versatility and potential of the Transformer architecture for a wide range of natural language processing tasks.

explain "Multi-Head Attention" ?

In the Transformer architecture, Multi-Head Attention (MHA) is a mechanism that allows the model to attend to different parts of the input sequence simultaneously and combine them into a single output. It's called "multi-head" because the model attends to multiple attention heads simultaneously. In MHA, the input is split into three parts: the query (Q), the key (K), and the value (V). The query and key are used to compute the attention weights, while the value is used to compute the output. The attention weights are computed using a dot product attention mechanism, which computes the weighted sum of the value based on the similarity between the query and key. The multi-head attention mechanism works as follows: 1. Split the input into multiple attention heads (e.g., 8 attention heads). 2. Compute the query, key, and value for each attention head using the input. 3. Compute the attention weights for each attention head using a dot product attention mechanism. 4. Compute the weighted sum of the value for each attention head using the attention weights. 5. Combine the outputs of all attention heads to form the final output. The benefits of MHA are: 1. It allows the model to focus on different aspects of the input sequence simultaneously, which improves the overall performance of the model. 2. It allows the model to capture longer-range dependencies in the input sequence, since each attention head can attend to a different part of the input. 3. It makes the model more robust to input noise and errors, since multiple attention heads can attend to different parts of the input and combine their outputs. 4. It allows the model to be more flexible and adaptive to different input sequences, since the attention mechanism can be adjusted for each attention head. In summary, Multi-Head Attention is a mechanism that allows the Transformer architecture to attend to multiple parts of the input sequence simultaneously and combine them into a single output, which improves the overall performance of the model.

Benefits to the Enterprise

- **Increased Productivity:** Instant summaries and direct answers reduce manual scanning.
- **Faster Decision-Making:** Quick insights from complex reports or spreadsheets.
- **Enhanced Compliance:** Local AI ensures adherence to privacy and regulatory mandates.
- **Scalability:** Modular design handles thousands of documents and queries.
- **Competitive Advantage:** Differentiates the organization with secure, AI-driven document intelligence.

Future Scope

- Additional document types (PowerPoint, emails, OCR for scanned documents).
- Multilingual document summarization and querying.
- Domain-specific LLM fine-tuning for finance, healthcare, or legal content.
- Voice interface integration.
- Advanced analytics for tabular data and trend forecasting.
- Knowledge graph integration to link insights across multiple documents.

Conclusion

- DocChat AI represents a next-generation enterprise document assistant, integrating intelligence directly into document workflows. By leveraging locally hosted AI, secure parsing pipelines, and interactive conversational interfaces, it provides enhanced productivity, faster decision-making, and full compliance with enterprise security requirements.
- The solution is scalable, reusable, and adaptable, enabling **clients such as TCS** to adopt secure AI-driven document services across multiple industries and business domains.