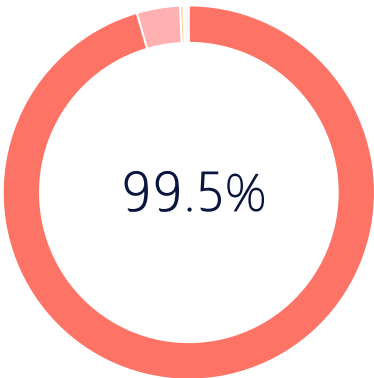


Analysis Report

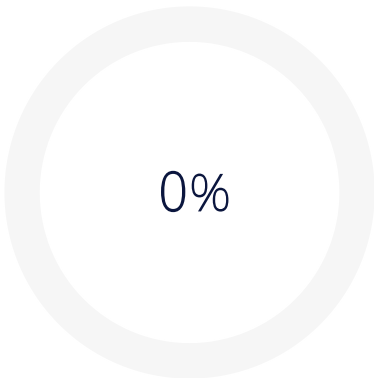
Plagiarism Detection and AI Detection Report
1706.03762v7.pdf

Plagiarism Detection



Plagiarism types	Text coverage	Words
Identical	95.5%	5,826
Minor Changes	3.8%	229
Paraphrased	0.3%	19
Excluded		
Omitted Words		0

AI Detection



	Text coverage	Words
AI Text	0%	0
Human Text	100%	6,102
Excluded		
Omitted Words		0

Plagiarism



Results (83)

*Results may not appear because the feature has been disabled.

 Private Cloud Hub 0	 Shared Data Hub 0	 Filtered / Excluded 0
 Internet Sources 76	 AI Source Match 7	 Current Batch 0

Plagiarism types	Text coverage	Words
 Identical	95.5%	5,826
 Minor Changes	3.8%	229
 Paraphrased	0.3%	19
Excluded		
 Omitted Words		0

About Plagiarism Detection

Our AI-powered plagiarism scans offer three layers of text similarity detection: Identical, Minor Changes, and Paraphrased. Based on your scan settings we also provide insight on how much of the text you are not scanning for plagiarism (Omitted words).

Identical

One to one exact word matches. [Learn more](#)

Minor Changes

Words that hold nearly the same meaning but have a change to their form (e.g. “large” becomes “largely”). [Learn more](#)

Paraphrased

Different words that hold the same meaning that replace the original content (e.g. 'large' becomes 'big') [Learn more](#)

Omitted Words

The portion of text that is not being scanned for plagiarism based on the scan settings. (e.g. the 'Ignore quotations' setting is enabled and the document is 20% quotations making the omitted words percentage 20%) [Learn more](#)

Copyleaks Shared Data Hub

Our Shared Data Hub is a collection of millions of user-submitted documents that you can utilize as a scan resource and choose whether or not you would like to submit the file you are scanning into the Shared Data Hub. [Learn more](#)

Filtered and Excluded Results

The report will generate a complete list of results. There is always the option to exclude specific results that are not relevant. Note, by unchecking certain results, the similarity percentage may change. [Learn more](#)

Current Batch Results

These are the results displayed from the collection, or batch, of files uploaded for a scan at the same time. [Learn more](#)

Plagiarism Detection Results: (83)

 pdf-summarizer.ipynb	92.6%
https://raw.githubusercontent.com/alessandrodorazio/python-cookbook/main/pdf-summarizer/pdf-summariz...	
{ "cells": [{ "cell_type": "markdown", "id": "d2bf60e9", "metadata": {}, "source": ["# PDF Summarizer", "\n", "\n", "Autore: [Alessandro D", "...] }, ...] }	
 Attention Is All You Need	88.3%
https://arxiv.org/html/1706.03762v7	
arXiv is now an independent nonprofit! Learn more License: arXiv.org perpetual non-exclusive licens...	
 [1706.03762] Attention Is All You Need	88.3%
https://ar5iv.labs.arxiv.org/html/1706.03762	
Provided proper attribution is provided, Google hereby grants permission to reproduce the tables and figures in this paper solely fo...	
 ISO 19005 - Electronic document file format for long-term preservation (PDF/A)	87.1%
https://www.overleaf.com/latex/templates/humboldt-university-berlin-hu-thesis-dissertation-template/jgxyvx...	
Your Dissertation Title Optional Subtitle DISSERTATION zur Erlangung des akademischen Grades doctor rerum naturalium (Dr. rer. nat.) im...	
 Attention Is All You Need - DOKUMEN.PUB	87.1%
https://dokumen.pub/attention-is-all-you-need-l-4648890.html	
Attention Is All You Need The dominant sequence transduction models...	
 1706.03762	87%
https://arxiv.org/pdf/1706.03762	
Provided proper attribution is provided, Google hereby grants permission to reproduce the tables and figures in this paper solely for use...	
 example_1-paper_summarizer.ipynb	86.3%
https://raw.githubusercontent.com/shawhint/ai-builders-bootcamp-4/main/session-2/example_1-paper_summ...	
AI Source Match	
{ "cells": [{ "cell_type": "markdown", "id": "5a4ef775-778d-4c0f-b901-65df2ba5caf5", "metadata": {}, "source": ["# Summarizing Research...", ...] }, ...] }	
 Research Paper Summarizer.ipynb	85.5%
https://raw.githubusercontent.com/abhishekkavin/generative-ai/main/research%20paper%20summarizer.ipynb	
{ "cells": [{ "cell_type": "markdown", "id": "7f80ff8b-1d30-4700-9116-09b6779d3714", "metadata": {}, "source": ["# Research Paper Sum...", ...] }, ...] }	

 Medium

84.6%

<https://medium.com/aimpact-all-things-ai/data-chunking-strategies-for-rag-in-2025-acfec4707eaf>

SitemapOpen in appSign inMedium LogoWriteSearchSign inAlmpact - All Things AI · Stories of AI soluti...

 Attention Is All You Need

82.9%

<https://arxiv.org/html/1706.03762v3>

arXiv is now an independent nonprofit! Learn moreLicense: arXiv.org perpetual non-exclusive licens...

 Attention Is All You Need – Vaswani et al., 2017

82.7%

<https://attentionisallyouneed.uk/>

justdoit.workJust Do It – Claude Fable 5 use case · 2026-07-19AbstractThe dominant sequence transduction models are bas...

 example.txt

79.7%

<https://raw.githubusercontent.com/ruslanmv/watsonx-with-langchain/master/example.txt>

Provided proper attribution is provided, Google hereby grants permission to reproduce the tables and figures in this paper solely for use...

 pdf

76.6%

<https://arxiv.org/pdf/1706.03762v1/pdf>

arXiv:1706.03762v1 [cs.CL] 12 Jun 2017Attention Is All You NeedAshish Vaswani* Google Brain avaswani@​google.comLlion Jones* Google R...

 Attention is All you Need

73.1%

https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-paper.pdf

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukh

Attention Is All You NeedAshish Vaswani* Google Brain avaswani@​google.comLlion Jones* Google Research Ilion@​google.comNoam Shazeer* ...

 Attention is All you Need

73.1%

https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-paper.pdf

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukh

Attention Is All You NeedAshish Vaswani* Google Brain avaswani@​google.comLlion Jones* Google Research Ilion@​google.comNoam Shazeer* ...

 Attention is All you Need

73.1%

https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-paper.pdf?ut...

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukh

Attention Is All You NeedAshish Vaswani* Google Brain avaswani@​google.comLlion Jones* Google Research Ilion@​google.comNoam Shazeer* ...

 [Transformer] Attention Is All You Need :: 밤에 쓰는 편지 58.9%
<https://letter-night.tistory.com/278>
...

 paper.json 57.3%
<https://raw.githubusercontent.com/neulab/reviewadvisor/main/extractor/paper.json>
{"name": "NIPS_2017_575.pdf", "metadata": {"source": "CRF", "title": "Attention Is All You Need", "authors": ["Ashish Vaswani", "Noam Sha...

 Attention Is All You Need (Explain Paper) 57.1%
<https://miikahuttunen.com/october-2022/attention-is-all-you-need-explain-paper>
Attention Is All You Need (Explain Paper) Abstract The dominant sequence transduction models are based on complex recurrent or convo...

 Medium 46.1%
<https://medium.com/@shinoavn/attention-near-2c55eed99379>
SitemapOpen in app Sign in Medium Logo Write Search Sign in attention \$near Shinoavn 13 min read· Apr...

 Attention Is All You Need - NotesByLex.com 44.6%
<https://notesbylex.com/attention-is-all-you-need>
Home /permanent Attention Is All You Need Dec 4, 2023 ...

 Attention is All You Need|fendoap 40.8%
https://note.com/ysuie_o/n/na6f2e6583f2e?hl=en
SYSTEM NOTICE Auto translation by AI. Be sure, accuracy, nuances and authorial inten...

 [译][论文] Transformer paper | Attention Is All You Need(Google,2017) 31.4%
<https://arthurchiao.art/blog/attention-is-all-you-need-zh/>
ArthurChiao's Blog [译][论文] Transformer paper | Attention Is All You...

 The Annotated Transformer 30.5%
<http://rush-nlp.com/2018/04/01/attention.html>
The recent Transformer architecture from "Attention is All You Need" @​ NIPS 2017 has been ...

 The Annotated Transformer http://nlp.seas.harvard.edu/2018/04/01/attention.html The recent Transformer architecture from "Attention is All You Need" @Google; NIPS 2017 has been ...	30.5%
 output.md https://raw.githubusercontent.com/yachty66/gpt_pdf_md/main/output.md # Attention Is All You Need Ashish Vaswani* Google Brain avaswani@google.com Noam Shazeer* Google Brain noam@google.com Niki Parmar* G...	25.8%
 Medium https://sh-tsang.medium.com/review-attention-is-all-you-need-transformer-96c787ecdec1 SitemapOpen in app Sign in Medium Logo Write Search Sign in Sik-Ho Tsang PhD, Researcher. I share what I learn. :)...	23.3%
 attention_pages_9_10.pdf https://raw.githubusercontent.com/datastax/ragstack-ai/main/examples/notebooks/resources/attention_page... Table 3: Variations on the Transformer architecture. Unlisted values are identical to those of the base model. All metrics are on the Eng...	13.3%
 LLAMA 3.1. 70B - RAG - Doc QA - ChromaDB.ipynb https://raw.githubusercontent.com/shashankshekharpandey/rag/main/llama%203.1.%2070b%20-%20rag%20... { "cells": [{ "cell_type": "code", "execution_count": 2, "id": "892671f6-elfb-4305-bb7e-14b4376fe238", "metadata": {}, "outputs": [], "source": "...	11.8%
 Medium https://medium.com/dair-ai/papers-explained-01-transformer-474bb60a33f7 SitemapOpen in app Sign in Medium Logo Write Search Sign in DAIR.AI · Democratizing Artificial Intelligenc...	9.8%
 [Transformer] paper, code review https://velog.io/@d4r6j/transformer d4r6j.log d4r6j.log [Transformer] paper, code review d4r6j:2023년 8월 27일 nlp-aipaper-review nlp-paper 목록 보기 2/11 Atten...	9.3%
 llama-parse-attention.ipynb https://raw.githubusercontent.com/datastax/genai-cookbook/main/llama-parse-attention.ipynb { "cells": [{ "cell_type": "markdown", "metadata": {}, "source": "# Using llama-parse with AstraDB" }, { "cell_type": "markdown", "source": "...	9%

 4 Why Self-Attention · GitBook https://windmissing.github.io/nlp-important-papers/aiayn/whyselfattention.html ...	7.8%
 BM25(Illamaindex) - AI Engineering Academy https://aiengineering.academy/rag/01_bm25_rag/notebook/ Skip to content BM25(Illamaindex) BM25 RAG AI Engineering.academy ...	7.5%
 Medium https://medium.com/@childdenaya/transformer-attention-is-all-you-need-a-paper-summary-d5fa82ff65de SitemapOpen in app Sign in Medium Logo Write Search Sign in Denaya chill and learn Software engineer L...	7.3%
 quickstart_example.ipynb https://raw.githubusercontent.com/madrylab/context-cite/main/notebooks/quickstart_example.ipynb <pre>{ "cells": [{ "cell_type": "markdown", "id": "7275acd5", "metadata": {}, "source": ["You can also follow along on Google Colab!"] }] }</pre>	6.9%
 Why Language? · Ruminations https://jessearmand.com/why-language/ Why Language? Posted on April 16, 2023 Now, everyone in the AI – if not the w...	6.3%
 Attention Is All You Need: A New Approach to Sequence Course Hero https://www.coursehero.com/file/210889720/170603762pdf/ 1706.03762.pdf - Provided proper attribution is provided ... Pages 15 Total views 14 University of Texas N N 5710 AmbassadorAn...	5.5%
 References Modern Approaches in Natural Language Processing https://slds-lmu.github.io/seminar_nlp_ss20/references.html Modern Approaches in Natural Language Processing ...	5.2%
 attention is all you need? DSMI Lab's website https://dsmlab.github.io/dsmi-lab-website/2021/01/25/attention%20is%20all%20you%20need/ 2021-01-25 ...	4.4%

 crazy_functions/test_project/latex/attention/introduction.tex · qingxu98/gpt-... https://huggingface.co/spaces/qingxu98/gpt-academic/blob/refs/pr/415/crazy_functions/test_project/latex/a... Fetching metadata from the HF Docker repository... 3v324v23 增加读latex文章的功能.添加测试样例 f238a34 ...	4.3%
 README.md · com3dian/Bart-large-paper2slides-summarizer at 42edfc0f4f77c71194... https://huggingface.co/com3dian/bart-large-paper2slides-summarizer/blame/42edfc0f4f77c71194f26dc7d2636... File size: 7,276 Bytes 01d1ba5 502b69a b3e8ca7 502b69a b3e8ca7 1134b4d cd7a246 502b69a b3e8ca7 01d1...	4.1%
 Transformer's Positional Encoding - Naoki Shibuya https://naokishibuya.github.io/blog/2021-10-31-transformers-positional-encoding/ In 2017, Vaswani et al. published a paper titled Attention Is All You Need for the NeurIPS conference. They in...	3.5%
 GitHub - soumik12345/transformer.pytorch: Pytorch implementation of Transform... https://github.com/soumik12345/transformer.pytorch Skip to content ...	3.3%
 README.md https://raw.githubusercontent.com/soumik12345/transformer.pytorch/master/readme.md # Transformer.Pytorch Pytorch implementation of the **Transformer** Model as proposed by the paper [Attention Is All You Need](https://...	3.3%
 Transformers https://www.shedge.com/05-generative-ai/llm/tools/in-detail/transformers/ ...	3.1%
 Transformer Architecture Knowledgebase https://kb.segersian.com/ai/transformer-architecture/ Transformer Architecture # The transformer architecture is the current mainstream architecture used for these new foundational models, o...	3.1%
 Attention is All You Need https://research.google/pubs/attention-is-all-you-need/ Skip to main content ...	3%

 Attention is all you need EMIL https://ghasemzadeh.com/event/attention-is-all-you-need/ Attention is all you need Reza Rahimi Azghan PDF Abstract The dominant sequence transduction models are based on complex recurrent o...	2.9%
 Attention is All You Need https://research.google.com/pubs/pub46201.html?source=post_page----- ...	2.9%
 AI-Powered Paper Summarization about the arXiv paper 1706.03762v5 https://www.summarizepaper.com/en/arxiv-id/1706.03762v5/ AI Source Match Attention Is All You Need AI-generated keywords: Transformer Attent...	2.9%
 Science Cast https://sciencecast.org/casts/j23kxg6qlanz AI Source Match Attention Is All You Need paulclintonFebruary 26, 2024 11:35am Views (769) Comments (2) Export Citation ShareShare Tweet Ema...	2.8%
 Llion Jones's research works Google Inc. and other places https://www.researchgate.net/scientific-contributions/lion-jones-2163046838 What is this page? This page lists works of an author who doesn't have a ResearchGate profile or hasn't added the works to their prof...	2.8%
 Niki Parmar alphaXiv https://www.alphaxiv.org/@niki-parmar Niki Parmar Member of Technical Staff @8203; Anthropic, Previously Co-Founder @8203; Essential AI About By co-authoring Attention is all yo...	2.8%
 Attention Is All You Need https://bibbase.org/network/publication/vaswani-shazeer-parmar-uszkoreit-jones-gomez-kaiser-polosukhin-a... Attention Is All You Need. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I...	2.7%
 CS7015 (Deep Learning) : Lecture 1 - (Partial/Brief) History of Deep Learning https://www.cse.iitm.ac.in/~miteshk/cs7015/slides/teaching/src/lecture1/modules/module7/lecture1_7.pdf Mitesh M. Khapra References Beating humans at their own game (literally) 1/14 Module 7 References Playing Atari Games Human-level control through deep...	2.6%

 Attention Is All You Need Papers We Love https://paperswelope.org/papers/attention-is-all-you-need-b04a08f7/ AI Source Match paper Attention Is All You Need Authors: Aidan N. Gomez ...	2.4%
 Old NLP Paper Reviews Sungwon Kim https://sungwon-kim.com/blog/2025/old-paper-reviews/ Premise This is part of an ongoing series where I revisit influential papers in NLP. Each post will briefly summarize the core id...	2.2%
 15 Transformer - A look at Attention - aPaperADay https://www.apaperaday.com/11/10/2020/15attention-3attention/ Home Archive ...	1.9%
 3.3 Position-wise Feed-Forward Networks · GitBook https://windmissing.github.io/nlp-important-papers/aiayn/positionwisefeedforwardnetworks.html ...	1.6%
 Medium https://medium.com/@sivaneni1992/inside-transformers-116248802f8e SitemapOpen in app Sign in Medium Logo Write Search Sign in Sivaneni Prasanna Inside Transformers ...	1.6%
 Medium https://medium.com/data-science/attending-to-attention-eba798f0e940 SitemapOpen in app Sign in Medium Logo Write Search Sign in TDS Archive · An archive of data science, data...	1.4%
 Attending to Attention Towards Data Science https://towardsdatascience.com/attending-to-attention-eba798f0e940/ A summary of a revolutionary paper "Attention is All You Need" and Implementing the Transformer using PyTorch Vincent van Gogh ****_...	1.4%
 Attention Is All You Need · Gist Share https://gist.is/papers.baulab.info/en/w3d6-nbph1w2 Eight Transcript Gist ...	1.3%

 Transformer Encoder-Decoder Structure https://apxml.com/courses/how-to-build-a-large-language-model/chapter-4-transformer-architecture/encode... Masterclass How To Build A Large Language Model Chapter 1: Introduction to Large-Scale Language Modeling Defining Large Language M...	1.2%
 Understanding Positional Encoding in Transformers - Blog by Kemal Erdem https://erdem.pl/2021/05/understanding-positional-encoding-in-transformers/ All ArticlesUnderstanding Positional Encoding in Transformers Published 10 May 2021 · 17 min read What is positional encoding? As I've ex...	1.1%
 Medium https://medium.com/data-science/understanding-positional-encoding-in-transformers-dc6bafc021ab SitemapOpen in app Sign in Medium Logo Write Search Sign in TDS Archive · An archive of data science, data...	1.1%
 Medium https://medium.com/data-science/understanding-positional-encoding-in-transformers-dc6bafc021ab?response... SitemapOpen in app Sign in Medium Logo Write Search Sign in TDS Archive · An archive of data science, data...	1.1%
 README.md https://raw.githubusercontent.com/fuzihaofzh/most-cited-papers/master/readme.md # most-cited-papers Statistics on most cited papers in recent years of each conferences title authors venue year citation ...	1%
 Google Research Transformer AI Ecosystem - AI Product Engineer https://aiproduct.engineer/ai-ecosystem/google-research-transformer-d95645b8c4ce Back to AI EcosystemGoogle Research Transformer Advancing Natural Language Processing through pioneering Transformer-based models for su...	1%
 09화 10. 멀티헤드 어텐션 https://brunch.co.kr/@aideveloper/167 10. 멀티헤드 어텐션 2장 GPT의 기반 기술, Transformer란 무엇인가? by에이전트김Nov 10. 2025 멀티헤드 어텐션(Multi-Head Attention)은 Transfo...	1%
 File:Attention Is All You Need - Isolated attentions from just the word 'its'... https://commons.wikimedia.org/wiki/file:attention_is_all_you_need_-_isolated_attentions_from_just_the_word_... Jump to content ...	1%

 Fast Adaptation to New Environments via Policy-Dynamics Value Functions http://proceedings.mlr.press/v119/raileanu20a/raileanu20a-supp.pdf Roberta Raileanu, Max Goldstein, Arthur Szlam, Rob Fergus Fast Adaptation to New Environments via Policy-Dynamics Value Functions A. Network Architectures A.1. Autoencoders The policy and dynami...	0.9%
 Demystifying Large Language Models: A Technical Deep Dive https://iaeme.com/masteradmin/journal_uploads/ijcet/volume_16_issue_1/ijcet_16_01_180.pdf Debu Sinha AI Source Match International Journal of Computer Engineering and Technology (IJCET) Volume 16, Issue 1, Jan-Feb 2025, pp. 2529-2555, Article ID: IJCET_1...	0.9%
 Modeling Structured Data in Attention-based Models https://publications.idiap.ch/attachments/papers/2023/mohammadshahi_thesis_2023.pdf Mohammadshahi, Alireza Thèse n° 9804 Modeling Structured Data in Attention-based Models Présentée le 13 octobre 2023 Faculté informatique et communications La...	0.9%
 File:Attention Is All You Need - Full attentions for head 5.png - Wikimedia C... https://commons.wikimedia.org/wiki/file:attention_is_all_you_need_-_full_attentions_for_head_5.png Jump to content ...	0.8%
 The Random Transformer - hackerllama https://osanseviero.github.io/hackerllama/blog/posts/random_transformer/?utm_source=www.hungryminds.d... AI Source Match In this blog post, we'll do an end-to-end example of the math within a transformer model. The goal is to g...	0.7%
 2502.0023v1.pdf https://vixra.org/pdf/2502.0023v1.pdf AI Source Match Mathematical Foundations of Deep Learning Sourangshu Ghosh* *sourangshug@iisc.ac.in, Department of Civil Engineering, Indian Institute of...	0.5%
 Transformer: Attention Is All You Need Learning-Deep-Learning https://patrick-llgc.github.io/learning-deep-learning/paper_notes/transformer.html Learning-Deep-Learning Transformer: Attention Is All You Need June 2020 tl;dr: Transformer architectur...	0.4%
 Medium https://medium.com/@wasowski.jarek/lions-eat-hyenas-positional-encoding-bb335b61b6fd SitemapOpen in app Sign in Medium Logo Write Search Sign in Jaroslaw Wasowski AI Architect. I build enterprise AI ...	0.4%

 claims/multi-head-attention-attends-multiple-subspaces.md · attention-wiki/kn... 0.4%
<https://huggingface.co/datasets/attention-wiki/knowledge-base/blob/refs/pr/30/claims/multi-head-attentio...>
Ivwerra HF Staff Process arXiv:1706.03762 – Attention Is All You Need (#1) 1953b29 2 months ago preview ...

 Layer Norm & Residuals | Transformer Architecture 0.3%
<https://dekut-dsail.github.io/tutorials/transformer-architecture/6.%20layer%20normalisation%20&%20residua...>
Course Content ...

 What is the difference between position embedding and position encoding?-zhihu 0.3%
<https://www.zhihu.com/en/answer/1336546000>
What is the difference between position embedding and position encoding? Visit original page View all 7 answers zhouxiangyu 1 pe...

Provided proper attribution is provided, Google hereby grants permission to reproduce the tables and figures in this paper solely for use in journalistic or scholarly works.

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

Lukasz 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 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

* 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.

1 Introduction

Recurrent neural networks, long short-term memory [13] 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 [35, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [38, 24, 15].

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 sequence lengths, as memory constraints limit batching across examples. Recent work has achieved significant improvements in computational efficiency through factorization tricks [21] and conditional computation [32], while also improving model performance in case of the latter. The fundamental constraint of sequential computation, however, remains.

Attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences [2, 19]. In all but a few cases [27], however, such attention mechanisms are used in conjunction with a recurrent network.

In this work we propose the Transformer, a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output. The Transformer allows for significantly more parallelization and can reach a new state of the art in translation quality after being trained for as little as twelve hours on eight P100 GPUs.

2 Background

The goal of reducing sequential computation also forms the foundation of the Extended Neural GPU [16], ByteNet [18] and ConvS2S [9], all of which use convolutional neural networks as basic building block, computing hidden representations in parallel for all input and output positions. In these models, the number of operations required to relate signals from two arbitrary input or output positions grows in the distance between positions, linearly for ConvS2S and logarithmically for ByteNet. This makes it more difficult to learn dependencies between distant positions [12]. In the Transformer this is reduced to a constant number of operations, albeit at the cost of reduced effective resolution due to averaging attention-weighted positions, an effect we counteract with Multi-Head Attention as described in section 3.2.

Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. Self-attention has been used successfully in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations [4, 27, 28, 22].

End-to-end memory networks are based on a recurrent attention mechanism instead of sequence-aligned recurrence and have been shown to perform well on simple-language question answering and language modeling tasks [34].

To the best of our knowledge, however, the Transformer is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution. In the following sections, we will describe the Transformer, motivate self-attention and discuss its advantages over models such as [17, 18] and [9].

3 Model Architecture

Most competitive neural sequence transduction models have an encoder-decoder structure [5, 2, 35]. Here, the encoder maps an input sequence of symbol representations $(x_1, \dots, x_n)$ to a sequence of continuous representations $\mathbf{z} = (z_1, \dots, z_n)$. Given $\mathbf{z}$, the decoder then generates an output sequence $(y_1, \dots, y_m)$ of symbols one element at a time. At each step the model is auto-regressive [10], consuming the previously generated symbols as additional input when generating the next.

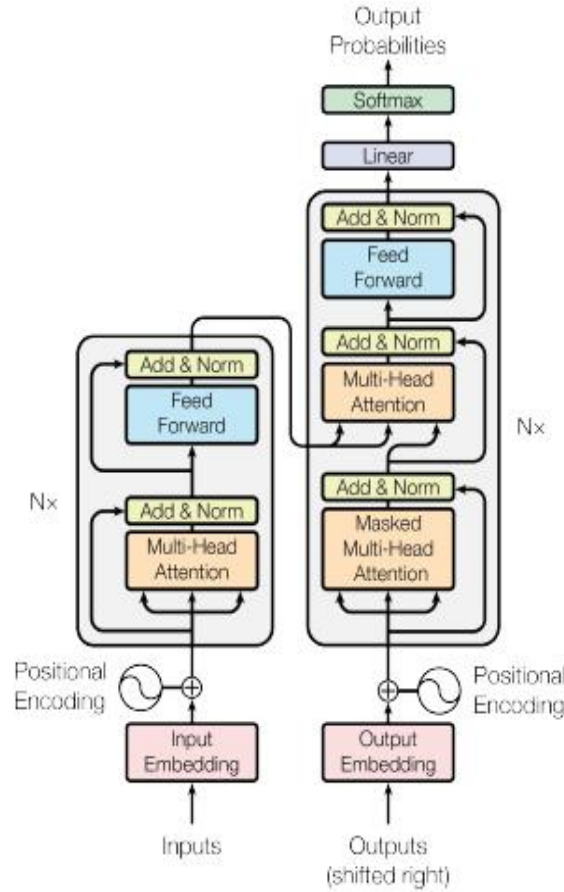


Figure 1: The Transformer - model architecture.

The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, shown in the left and right halves of Figure 1, respectively.

3.1 Encoder and Decoder Stacks

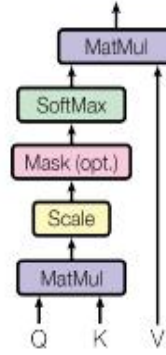
Encoder: The encoder is composed of a stack of $N = 6$ identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. We employ a residual connection [1] around each of the two sub-layers, followed by layer normalization [1]. That is, the output of each sub-layer is $\text{LayerNorm}(x + \text{Sublayer}(x))$, where $\text{Sublayer}(x)$ is the function implemented by the sub-layer itself. To facilitate these residual connections, all sub-layers in the model, as well as the embedding layers, produce outputs of dimension $d_{\text{model}} = 512$.

Decoder: The decoder is also composed of a stack of $N = 6$ identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, we employ residual connections around each of the sub-layers, followed by layer normalization. We also modify the self-attention sub-layer in the decoder stack to prevent positions from attending to subsequent positions. This masking, combined with fact that the output embeddings are offset by one position, ensures that the predictions for position i can depend only on the known outputs at positions less than i .

3.2 Attention

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum

Scaled Dot-Product Attention



Multi-Head Attention

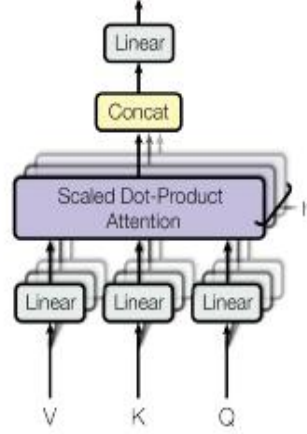


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

3.2.1 Scaled Dot-Product Attention

We call our particular attention "Scaled Dot-Product Attention" (Figure 2). The input consists of queries and keys of dimension d_k , and values of dimension d_v . We compute the dot products of the 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:

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

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 gradients⁴. 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

⁴To 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 .

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.

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

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

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

In this work we employ $h = 8$ parallel attention layers, or heads. For each of these we use $d_k = d_v = d_{\text{model}}/h = 64$. 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 "encoder-decoder attention" layers, the queries come from the previous decoder layer, and the memory keys and values come from the output of the encoder. This allows every position in the decoder to attend over all positions in the input sequence. This mimics the typical encoder-decoder attention mechanisms in sequence-to-sequence models such as [38, 2, 9].
- The encoder contains self-attention layers. In a self-attention layer all of the keys, values and queries come from the same place, in this case, the output of the previous layer in the encoder. Each position in the encoder can attend to all positions in the previous layer of the encoder.
- Similarly, self-attention layers in the decoder allow each position in the decoder to attend to all positions in the decoder up to and including that position. We need to prevent leftward information flow in the decoder to preserve the auto-regressive property. We implement this inside of scaled dot-product attention by masking out (setting to $-\infty$) all values in the input of the softmax which correspond to illegal connections. See Figure 2.

3.3 Position-wise Feed-Forward Networks

In addition to attention sub-layers, each of the layers in our encoder and decoder contains a fully connected feed-forward network, which is applied to each position separately and identically. This consists of two linear transformations with a ReLU activation in between.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2)$$

While the linear transformations are the same across different positions, they use different parameters from layer to layer. Another way of describing this is as two convolutions with kernel size 1. The dimensionality of input and output is $d_{\text{model}} = 512$, and the inner-layer has dimensionality $d_{\text{FF}} = 2048$.

3.4 Embeddings and Softmax

Similarly to other sequence transduction models, we use learned embeddings to convert the input tokens and output tokens to vectors of dimension d_{model} . We also use the usual learned linear transformation and softmax function to convert the decoder output to predicted next-token probabilities. In our model, we share the same weight matrix between the two embedding layers and the pre-softmax linear transformation, similar to [30]. In the embedding layers, we multiply those weights by $\sqrt{d_{\text{model}}}$.

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

3.5 Positional Encoding

Since our model contains no recurrence and no convolution, in order for the model to make use of the order of the sequence, we must inject some information about the relative or absolute position of the tokens in the sequence. To this end, we add "positional encodings" to the input embeddings at the bottoms of the encoder and decoder stacks. The positional encodings have the same dimension d_{model} as the embeddings, so that the two can be summed. There are many choices of positional encodings, learned and fixed [9].

In this work, we use sine and cosine functions of different frequencies:

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

where pos is the position and i is the dimension. That is, each dimension of the positional encoding corresponds to a sinusoid. The wavelengths form a geometric progression from 2π to $10000 \cdot 2\pi$. We chose this function because we hypothesized it would allow the model to easily learn to attend by relative positions, since for any fixed offset k , PE_{pos+k} can be represented as a linear function of PE_{pos} .

We also experimented with using learned positional embeddings [9] instead, and found that the two versions produced nearly identical results (see Table 3 row (E)). We chose the sinusoidal version because it may allow the model to extrapolate to sequence lengths longer than the ones encountered during training.

4 Why Self-Attention

In this section we compare various aspects of self-attention layers to the recurrent and convolutional layers commonly used for mapping one variable-length sequence of symbol representations $(x_1, \dots, x_n)$ to another sequence of equal length $(z_1, \dots, z_n)$, with $x_i, z_i \in \mathbb{R}^d$, such as a hidden layer in a typical sequence transduction encoder or decoder. Motivating our use of self-attention we consider three desiderata.

One is the total computational complexity per layer. Another is the amount of computation that can be parallelized, as measured by the minimum number of sequential operations required.

The third is the path length between long-range dependencies in the network. Learning long-range dependencies is a key challenge in many sequence transduction tasks. One key factor affecting the ability to learn such dependencies is the length of the paths forward and backward signals have to traverse in the network. The shorter these paths between any combination of positions in the input and output sequences, the easier it is to learn long-range dependencies [12]. Hence we also compare the maximum path length between any two input and output positions in networks composed of the different layer types.

As noted in Table 1 a self-attention layer connects all positions with a constant number of sequentially executed operations, whereas a recurrent layer requires $O(n)$ sequential operations. In terms of computational complexity, self-attention layers are faster than recurrent layers when the sequence

length n is smaller than the representation dimensionality d , which is most often the case with sentence representations used by state-of-the-art models in machine translations, such as word-piece [38] and byte-pair [31] representations. To improve computational performance for tasks involving very long sequences, self-attention could be restricted to considering only a neighborhood of size r in the input sequence centered around the respective output position. This would increase the maximum path length to $O(n/r)$. We plan to investigate this approach further in future work.

A single convolutional layer with kernel width $k \leq n$ does not connect all pairs of input and output positions. Doing so requires a stack of $O(n/k)$ convolutional layers in the case of contiguous kernels, or $O(\log_k(n))$ in the case of dilated convolutions [18], increasing the length of the longest paths between any two positions in the network. Convolutional layers are generally more expensive than recurrent layers, by a factor of k . Separable convolutions [6], however, decrease the complexity considerably, to $O(k \cdot n \cdot d + n \cdot d^2)$. Even with $k = n$, however, the complexity of a separable convolution is equal to the combination of a self-attention layer and a point-wise feed-forward layer, the approach we take in our model.

As side benefit, self-attention could yield more interpretable models. We inspect attention distributions from our models and present and discuss examples in the appendix. Not only do individual attention heads clearly learn to perform different tasks, many appear to exhibit behavior related to the syntactic and semantic structure of the sentences.

5 Training

This section describes the training regime for our models.

5.1 Training Data and Batching

We trained on the standard WMT 2014 English-German dataset consisting of about 4.5 million sentence pairs. Sentences were encoded using byte-pair encoding [3], which has a shared source-target vocabulary of about 37000 tokens. For English-French, we used the significantly larger WMT 2014 English-French dataset consisting of 36M sentences and split tokens into a 32000 word-piece vocabulary [38]. Sentence pairs were batched together by approximate sequence length. Each training batch contained a set of sentence pairs containing approximately 25000 source tokens and 25000 target tokens.

5.2 Hardware and Schedule

We trained our models on one machine with 8 NVIDIA P100 GPUs. For our base models using the hyperparameters described throughout the paper, each training step took about 0.4 seconds. We trained the base models for a total of 100,000 steps or 12 hours. For our big models, (described on the bottom line of table 3), step time was 1.0 seconds. The big models were trained for 300,000 steps (3.5 days).

5.3 Optimizer

We used the Adam optimizer [20] with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. We varied the learning rate over the course of training, according to the formula:

$$lr_{rate} = d_{model}^{-0.5} \cdot \min(step_num^{-0.5}, step_num \cdot warmup_steps^{-1.5}) \quad (3)$$

This corresponds to increasing the learning rate linearly for the first $warmup_steps$ training steps, and decreasing it thereafter proportionally to the inverse square root of the step number. We used $warmup_steps = 4000$.

5.4 Regularization

We employ three types of regularization during training:

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	41.29	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$3.3 \cdot 10^{18}$	
Transformer (big)	28.4	41.8	$2.3 \cdot 10^{19}$	

Residual Dropout We apply dropout [33] to the output of each sub-layer, before it is added to the sub-layer input and normalized. In addition, we apply dropout to the sums of the embeddings and the positional encodings in both the encoder and decoder stacks. For the base model, we use a rate of $P_{drop} = 0.1$.

Label Smoothing During training, we employed label smoothing of value $\epsilon_{ls} = 0.1$ [36]. This hurts perplexity, as the model learns to be more unsure, but improves accuracy and BLEU score.

6 Results

6.1 Machine Translation

On the WMT 2014 English-to-German translation task, the big transformer model (Transformer (big) in Table 2) outperforms the best previously reported models (including ensembles) by more than 2.0 BLEU, establishing a new state-of-the-art BLEU score of 28.4. The configuration of this model is listed in the bottom line of Table 3. Training took 3.5 days on 8 P100 GPUs. Even our base model surpasses all previously published models and ensembles, at a fraction of the training cost of any of the competitive models.

On the WMT 2014 English-to-French translation task, our big model achieves a BLEU score of 41.0, outperforming all of the previously published single models, at less than 1/4 the training cost of the previous state-of-the-art model. The Transformer (big) model trained for English-to-French used dropout rate $P_{drop} = 0.1$, instead of 0.3.

For the base models, we used a single model obtained by averaging the last 5 checkpoints, which were written at 10-minute intervals. For the big models, we averaged the last 20 checkpoints. We used beam search with a beam size of 4 and length penalty $\alpha = 0.6$ [38]. These hyperparameters were chosen after experimentation on the development set. We set the maximum output length during inference to input length + 50, but terminate early when possible [38].

Table 2 summarizes our results and compares our translation quality and training costs to other model architectures from the literature. We estimate the number of floating point operations used to train a model by multiplying the training time, the number of GPUs used, and an estimate of the sustained single-precision floating-point capacity of each GPU⁵.

6.2 Model Variations

To evaluate the importance of different components of the Transformer, we varied our base model in different ways, measuring the change in performance on English-to-German translation on the

⁵We used values of 2.8, 3.7, 6.0 and 9.5 TFLOPS for K80, K40, M40 and P100, respectively.

Table 3: Variations on the Transformer architecture. Unlisted values are identical to those of the base model. All metrics are on the English-to-German translation development set, newstest2013. Listed perplexities are per-wordpiece, according to our byte-pair encoding, and should not be compared to per-word perplexities.

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$	
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65	
(A)					1	512	512				5.29	24.9	
					4	128	128				5.00	25.5	
					16	32	32				4.91	25.8	
					32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58	
					32					5.01	25.4	60	
(C)	2									6.11	23.7	36	
	4									5.19	25.3	50	
	8									4.88	25.5	80	
		256			32	32				5.75	24.5	28	
		1024			128	128				4.66	26.0	168	
			1024								5.12	25.4	53
			4096								4.75	26.2	90
(D)							0.0			5.77	24.6		
							0.2			4.95	25.5		
								0.0		4.67	25.3		
								0.2		5.47	25.7		
(E)	positional embedding instead of sinusoids									4.92	25.7		
big	6	1024	4096	16				0.3	300K	4.33	26.4	213	

development set, newstest2013. We used beam search as described in the previous section, but no checkpoint averaging. We present these results in Table 3.

In Table 3 rows (A), we vary the number of attention heads and the attention key and value dimensions, keeping the amount of computation constant, as described in Section 3.2.2. While single-head attention is 0.9 BLEU worse than the best setting, quality also drops off with too many heads.

In Table 3 rows (B), we observe that reducing the attention key size d_k hurts model quality. This suggests that determining compatibility is not easy and that a more sophisticated compatibility function than dot product may be beneficial. We further observe in rows (C) and (D) that, as expected, bigger models are better, and dropout is very helpful in avoiding over-fitting. In row (E) we replace our sinusoidal positional encoding with learned positional embeddings [9], and observe nearly identical results to the base model.

6.3 English Constituency Parsing

To evaluate if the Transformer can generalize to other tasks we performed experiments on English constituency parsing. This task presents specific challenges: the output is subject to strong structural constraints and is significantly longer than the input. Furthermore, RNN sequence-to-sequence models have not been able to attain state-of-the-art results in small-data regimes [37].

We trained a 4-layer transformer with $d_{\text{model}} = 1024$ on the Wall Street Journal (WSJ) portion of the Penn Treebank [25], about 40K training sentences. We also trained it in a semi-supervised setting, using the larger high-confidence and BerkleyParser corpora from with approximately 17M sentences [37]. We used a vocabulary of 16K tokens for the WSJ only setting and a vocabulary of 32K tokens for the semi-supervised setting.

We performed only a small number of experiments to select the dropout, both attention and residual (section 5.4), learning rates and beam size on the Section 22 development set, all other parameters remained unchanged from the English-to-German base translation model. During inference, we

Table 4: The Transformer generalizes well to English constituency parsing (Results are on Section 23 of WSJ)

Parser	Training	WSJ 23 F1
Vinyals & Kaiser et al. (2014) [37]	WSJ only, discriminative	88.3
Petrov et al. (2006) [29]	WSJ only, discriminative	90.4
Zhu et al. (2013) [40]	WSJ only, discriminative	90.4
Dyer et al. (2016) [8]	WSJ only, discriminative	91.7
Transformer (4 layers)	WSJ only, discriminative	91.3
Zhu et al. (2013) [40]	semi-supervised	91.3
Huang & Harper (2009) [14]	semi-supervised	91.3
McClosky et al. (2006) [26]	semi-supervised	92.1
Vinyals & Kaiser et al. (2014) [37]	semi-supervised	92.1
Transformer (4 layers)	semi-supervised	92.7
Luong et al. (2015) [23]	multi-task	93.0
Dyer et al. (2016) [8]	generative	93.3

increased the maximum output length to input length + 300. We used a beam size of 21 and $\alpha = 0.3$ for both WSJ only and the semi-supervised setting.

Our results in Table 4 show that despite the lack of task-specific tuning our model performs surprisingly well, yielding better results than all previously reported models with the exception of the Recurrent Neural Network Grammar [8].

In contrast to RNN sequence-to-sequence models [37], the Transformer outperforms the Berkeley-Parser [29] even when training only on the WSJ training set of 40K sentences.

7 Conclusion

In this work, we presented the Transformer, the first sequence transduction model based entirely on attention, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention.

For translation tasks, the Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers. On both WMT 2014 English-to-German and WMT 2014 English-to-French translation tasks, we achieve a new state of the art. In the former task our best model outperforms even all previously reported ensembles.

We are excited about the future of attention-based models and plan to apply them to other tasks. We plan to extend the Transformer to problems involving input and output modalities other than text and to investigate local, restricted attention mechanisms to efficiently handle large inputs and outputs such as images, audio and video. Making generation less sequential is another research goal of ours.

The code we used to train and evaluate our models is available at <https://github.com/tensorflow/tensor2tensor>.

Acknowledgements We are grateful to Nal Kalchbrenner and Stephan Gouws for their fruitful comments, corrections and inspiration.

References

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.
- [3] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc V. Le. Massive exploration of neural machine translation architectures. *CoRR*, abs/1703.03906, 2017.
- [4] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

- [5] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.
- [6] François Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint [arXiv:1610.02357](#)*, 2016.
- [7] Junyoung Chung, Çağlar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.
- [8] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. In *Proc. of NAACL*, 2016.
- [9] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *arXiv preprint [arXiv:1705.03122](#)*, v2, 2017.
- [10] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint [arXiv:1308.0850](#)*, 2013.
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [12] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.
- [13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [14] Zhongqiang Huang and Mary Harper. Self-training PCFG grammars with latent annotations across languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 832–841. ACL, August 2009.
- [15] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint [arXiv:1602.02410](#)*, 2016.
- [16] Łukasz Kaiser and Samy Bengio. Can active memory replace attention? In *Advances in Neural Information Processing Systems (NIPS)*, 2016.
- [17] Łukasz Kaiser and Ilya Sutskever. Neural GPUs learn algorithms. In *International Conference on Learning Representations (ICLR)*, 2016.
- [18] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *arXiv preprint [arXiv:1610.10099](#)*, v2, 2017.
- [19] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks. In *International Conference on Learning Representations*, 2017.
- [20] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [21] Oleksii Kuchaiev and Boris Ginsburg. Factorization tricks for LSTM networks. *arXiv preprint [arXiv:1703.10722](#)*, 2017.
- [22] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint [arXiv:1703.03130](#)*, 2017.
- [23] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Łukasz Kaiser. Multi-task sequence to sequence learning. *arXiv preprint [arXiv:1511.06114](#)*, 2015.
- [24] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint [arXiv:1508.04025](#)*, 2015.

- [25] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.
- [26] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159. ACL, June 2006.
- [27] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model. In *Empirical Methods in Natural Language Processing*, 2016.
- [28] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint [arXiv:1705.04304](https://arxiv.org/abs/1705.04304)*, 2017.
- [29] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 433–440. ACL, July 2006.
- [30] Ofir Press and Lior Wolf. Using the output embedding to improve language models. *arXiv preprint [arXiv:1608.05859](https://arxiv.org/abs/1608.05859)*, 2016.
- [31] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint [arXiv:1508.07909](https://arxiv.org/abs/1508.07909)*, 2015.
- [32] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint [arXiv:1701.06538](https://arxiv.org/abs/1701.06538)*, 2017.
- [33] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [34] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems* 28, pages 2440–2448. Curran Associates, Inc., 2015.
- [35] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.
- [36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.
- [37] Vinyals & Kaiser, Koo, Petrov, Sutskever, and Hinton. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, 2015.
- [38] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint [arXiv:1609.08144](https://arxiv.org/abs/1609.08144)*, 2016.
- [39] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. Deep recurrent models with fast-forward connections for neural machine translation. *CoRR*, abs/1606.04199, 2016.
- [40] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. Fast and accurate shift-reduce constituent parsing. In *Proceedings of the 51st Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 434–443. ACL, August 2013.

Attention Visualizations

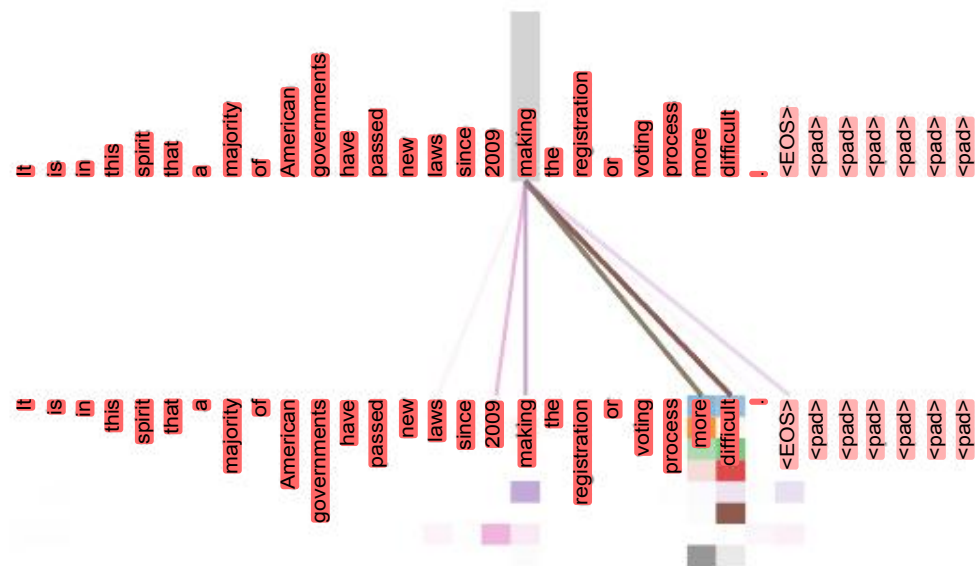


Figure 3: An example of the attention mechanism following long-distance dependencies in the encoder self-attention in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb 'making', completing the phrase 'making...more difficult'. Attentions here shown only for the word 'making'. Different colors represent different heads. Best viewed in color.

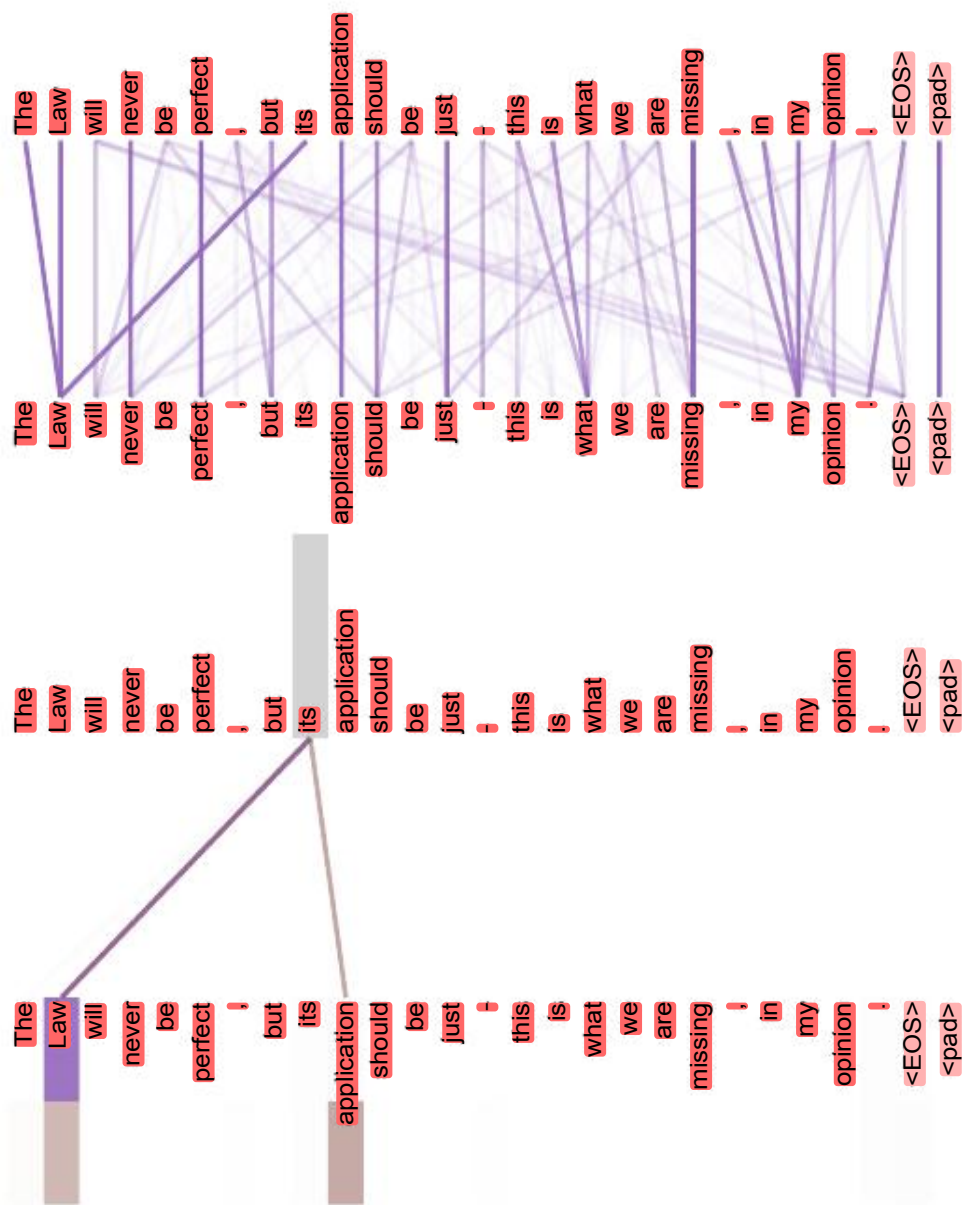


Figure 4: Two attention heads, also in layer 5 of 6, apparently involved in anaphora resolution. Top: Full attentions for head 5. Bottom: Isolated attentions from just the word 'its' for attention heads 5 and 6. Note that the attentions are very sharp for this word.

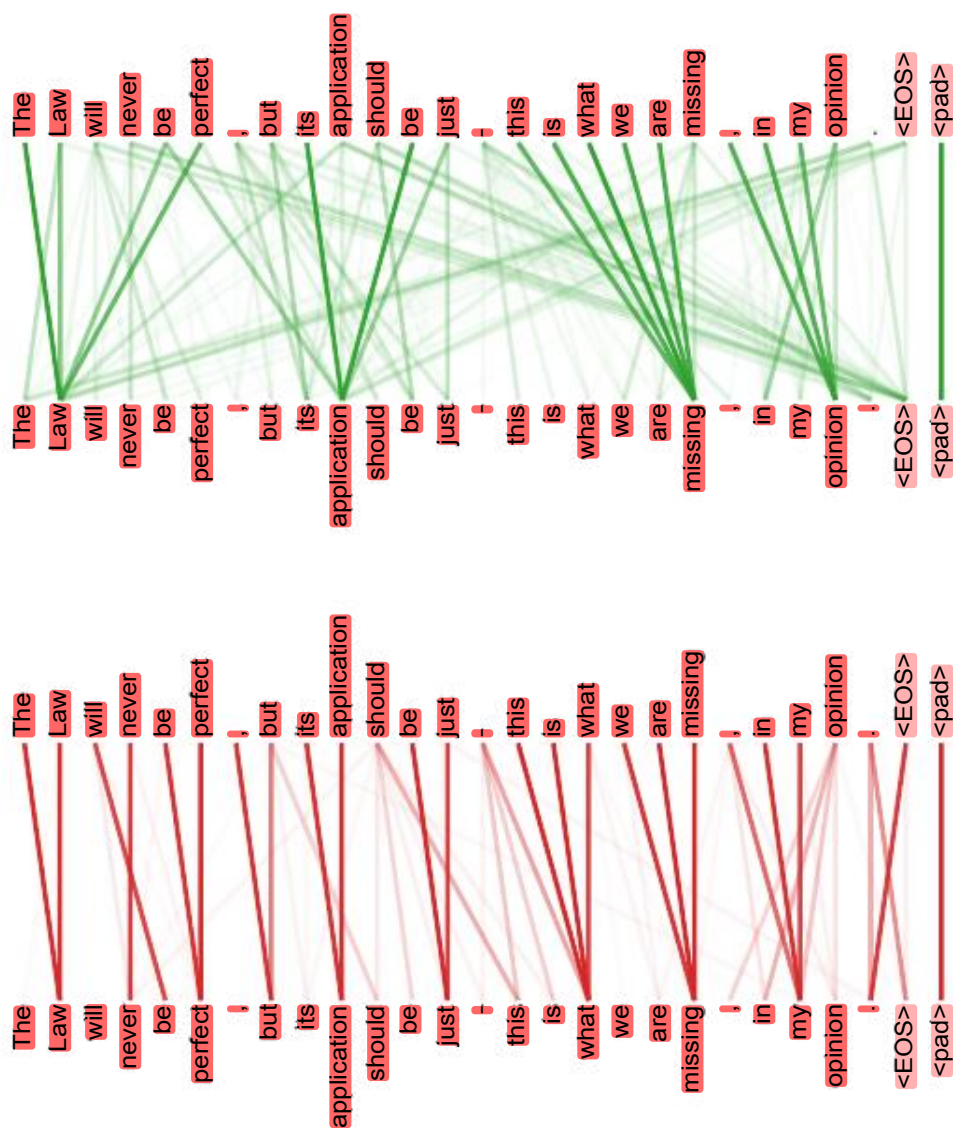
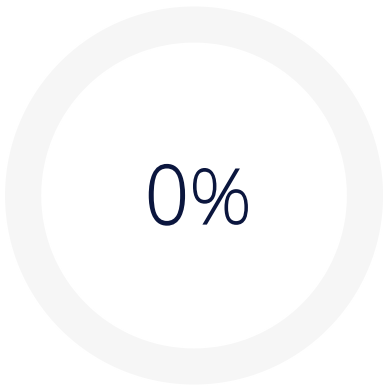


Figure 5: Many of the attention heads exhibit behaviour that seems related to the structure of the sentence. We give two such examples above, from two different heads from the encoder self-attention at layer 5 of 6. The heads clearly learned to perform different tasks.

AI Content



	Text coverage	Words
☐ AI Text	0%	0
☐ Human Text	100%	6,102
Excluded		
☐ Omitted Words		0

About AI Detection

Our AI Detector is the only enterprise-level solution that can verify if the content was written by a human or generated by AI, including source code and text that has been plagiarized or modified. [Learn more](#)

☐ AI Text

A body of text that has been generated or altered by AI technology. [Learn more](#)

☐ Human Text

Any text that has been fully written by a human and has not been altered or generated by AI. [Learn more](#)

Copyleaks AI Detector Effectiveness

Credible data at scale, coupled with machine learning and widespread adoption, allows us to continually refine and improve our ability to understand complex text patterns, resulting in over 99% accuracy—far higher than any other AI detector—and improving daily. [Learn more](#)

Ideal Text Length

The higher the character count, the easier for our technology to determine irregular patterns, which results in a higher confidence rating for AI detection. [Learn more](#)

Reasons It Might Be AI When You Think It's Not

The AI Detector can detect a variety of AI-generated text, including tools that use AI technology to paraphrase content, auto-complete sentences, and more. [Learn more](#)

User AI Alert History

Historical data of how many times a user has been flagged for potentially having AI text within their content. [Learn more](#)

AI Logic

The number of times a phrase was found more frequently in AI vs human text is shown according to low, medium, and high frequency. [Learn more](#)