In [2]:

```
## Data Ingestion
from langchain_community.document_loaders import TextLoader
loader=TextLoader("speech.txt")
text_documents=loader.load()
text_documents
```

Out[2]:

```
[Document(page_content='The world must be made safe for democracy. Its peace must be plan
ted upon the tested foundations of political liberty. We have no selfish ends to serve. W
e desire no conquest, no dominion. We seek no indemnities for ourselves, no material comp
ensation for the sacrifices we shall freely make. We are but one of the champions of the
rights of mankind. We shall be satisfied when those rights have been made as secure as th
e faith and the freedom of nations can make them.\n\nJust because we fight without rancor
and without selfish object, seeking nothing for ourselves but what we shall wish to share
with all free peoples, we shall, I feel confident, conduct our operations as belligerents
without passion and ourselves observe with proud punctilio the principles of right and of
fair play we profess to be fighting for.\n\n…\n\nIt will be all the easier for us to cond
uct ourselves as belligerents in a high spirit of right and fairness because we act witho
ut animus, not in enmity toward a people or with the desire to bring any injury or disadv
antage upon them, but only in armed opposition to an irresponsible government which has t
hrown aside all considerations of humanity and of right and is running amuck. We are, let
me say again, the sincere friends of the German people, and shall desire nothing so much
as the early reestablishment of intimate relations of mutual advantage between us—however
hard it may be for them, for the time being, to believe that this is spoken from our hear
ts.\n\nWe have borne with their present government through all these bitter months becaus
e of that friendship—exercising a patience and forbearance which would otherwise have bee
n impossible. We shall, happily, still have an opportunity to prove that friendship in ou
r daily attitude and actions toward the millions of men and women of German birth and nat
ive sympathy who live among us and share our life, and we shall be proud to prove it towa
rd all who are in fact loyal to their neighbors and to the government in the hour of test
. They are, most of them, as true and loyal Americans as if they had never known any othe
r fealty or allegiance. They will be prompt to stand with us in rebuking and restraining
the few who may be of a different mind and purpose. If there should be disloyalty, it wil
l be dealt with with a firm hand of stern repression; but, if it lifts its head at all, i
t will lift it only here and there and without countenance except from a lawless and mali
gnant few.\n\nIt is a distressing and oppressive duty, gentlemen of the Congress, which I
have performed in thus addressing you. There are, it may be, many months of fiery trial a
nd sacrifice ahead of us. It is a fearful thing to lead this great peaceful people into w
ar, into the most terrible and disastrous of all wars, civilization itself seeming to be
in the balance. But the right is more precious than peace, and we shall fight for the thi
ngs which we have always carried nearest our hearts—for democracy, for the right of those
who submit to authority to have a voice in their own governments, for the rights and libe
rties of small nations, for a universal dominion of right by such a concert of free peopl
es as shall bring peace and safety to all nations and make the world itself at last free.
\n\nTo such a task we can dedicate our lives and our fortunes, everything that we are and
everything that we have, with the pride of those who know that the day has come when Amer
ica is privileged to spend her blood and her might for the principles that gave her birth
and happiness and the peace which she has treasured. God helping her, she can do no other
.', metadata={'source': 'speech.txt'})]
```

In [3]:

```
import os
from dotenv import load_dotenv
load_dotenv()

os.environ['OPENAI_API_KEY']=os.getenv("OPENAI_API_KEY")
```

In [4]:

```
# web based loader
from langchain_community.document_loaders import WebBaseLoader
import bs4

## load,chunk and index the content of the html page
```

```
loader=WebBaseLoader(web_paths=("https://lilianweng.github.io/posts/2023-06-23-agent/",),
                     bs_kwargs=dict(parse_only=bs4.SoupStrainer(
                         class_=("post-title","post-content","post-header")

                     )))

text_documents=loader.load()
```

```
USER_AGENT environment variable not set, consider setting it to identify your requests.
```

In [5]:

```
## Pdf reader
from langchain_community.document_loaders import PyPDFLoader
loader=PyPDFLoader('attention.pdf')
docs=loader.load()
```

In [6]:

```
docs
```

Out[6]:

```
[Document(page_content='Provided proper attribution is provided, Google hereby grants per
mission to\nreproduce the tables and figures in this paper solely for use in journalistic
or\nscholarly works.\nAttention Is All You Need\nAshish Vaswani*\nGoogle Brain\navaswani@
google.comNoam Shazeer*\nGoogle Brain\nnoam@google.comNiki Parmar*\nGoogle Research\nniki
p@google.comJakob Uszkoreit*\nGoogle Research\nusz@google.com\nLlion Jones*\nGoogle Resea
rch\nllion@google.comAidan N. Gomez* †\nUniversity of Toronto\naidan@cs.toronto.eduŁukasz
Kaiser*\nGoogle Brain\nlukaszkaiser@google.com\nIllia Polosukhin* ‡\nillia.polosukhin@gma
il.com\nAbstract\nThe dominant sequence transduction models are based on complex recurren
t or\nconvolutional neural networks that include an encoder and a decoder. The best\nperf
orming models also connect the encoder and decoder through an attention\nmechanism. We pr
opose a new simple network architecture, the Transformer,\nbased solely on attention mech
anisms, dispensing with recurrence and convolutions\nentirely. Experiments on two machine
translation tasks show these models to\nbe superior in quality while being more paralleli
zable and requiring significantly\nless time to train. Our model achieves 28.4 BLEU on th
e WMT 2014 English-\nto-German translation task, improving over the existing best results
, including\nensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation tas
k,\nour model establishes a new single-model state-of-the-art BLEU score of 41.8 after\nt
raining for 3.5 days on eight GPUs, a small fraction of the training costs of the\nbest m
odels from the literature. We show that the Transformer generalizes well to\nother tasks
by applying it successfully to English constituency parsing both with\nlarge and limited
training data.\n*Equal contribution. Listing order is random. Jakob proposed replacing RN
Ns with self-attention and started\nthe effort to evaluate this idea. Ashish, with Illia,
designed and implemented the first Transformer models and\nhas been crucially involved in
every aspect of this work. Noam proposed scaled dot-product attention, multi-head\nattent
ion and the parameter-free position representation and became the other person involved i
n nearly every\ndetail. Niki designed, implemented, tuned and evaluated countless model v
ariants in our original codebase and\ntensor2tensor. Llion also experimented with novel m
odel variants, was responsible for our initial codebase, and\nefficient inference and vis
ualizations. Lukasz and Aidan spent countless long days designing various parts of and\ni
mplementing tensor2tensor, replacing our earlier codebase, greatly improving results and
massively accelerating\nour research.\n†Work performed while at Google Brain.\n‡Work perf
ormed while at Google Research.\n31st Conference on Neural Information Processing Systems
(NIPS 2017), Long Beach, CA, USA.arXiv:1706.03762v7  [cs.CL]  2 Aug 2023', metadata={'sou
rce': 'attention.pdf', 'page': 0}),
 Document(page_content='1 Introduction\nRecurrent neural networks, long short-term memory
[ 13] and gated recurrent [ 7] neural networks\nin particular, have been firmly establish
ed as state of the art approaches in sequence modeling and\ntransduction problems such as
language modeling and machine translation [ 35,2,5]. Numerous\nefforts have since continu
ed to push the boundaries of recurrent language models and encoder-decoder\narchitectures
[38, 24, 15].\nRecurrent models typically factor computation along the symbol positions o
f the input and output\nsequences. Aligning the positions to steps in computation time, t
hey generate a sequence of hidden\nstates ht, as a function of the previous hidden state
ht−1and the input for position t. This inherently\nsequential nature precludes paralleliz
ation within training examples, which becomes critical at longer\nsequence lengths, as me
mory constraints limit batching across examples. Recent work has achieved\nsignificant im
provements in computational efficiency through factorization tricks [ 21] and conditional
\ncomputation [ 32], while also improving model performance in case of the latter. The fu
ndamental\nconstraint of sequential computation, however, remains.\nAttention mechanisms
have become an integral part of compelling sequence modeling and transduc-\ntion models i
```

```
n various tasks, allowing modeling of dependencies without regard to their distance in\nt
he input or output sequences [ 2,19]. In all but a few cases [ 27], however, such attenti
on mechanisms\nare used in conjunction with a recurrent network.\nIn this work we propose
the Transformer, a model architecture eschewing recurrence and instead\nrelying entirely
on an attention mechanism to draw global dependencies between input and output.\nThe Tran
sformer allows for significantly more parallelization and can reach a new state of the ar
t in\ntranslation quality after being trained for as little as twelve hours on eight P100
GPUs.\n2 Background\nThe goal of reducing sequential computation also forms the foundatio
n of the Extended Neural GPU\n[16], ByteNet [ 18] and ConvS2S [ 9], all of which use conv
olutional neural networks as basic building\nblock, computing hidden representations in p
arallel for all input and output positions. In these models,\nthe number of operations re
quired to relate signals from two arbitrary input or output positions grows\nin the dista
nce between positions, linearly for ConvS2S and logarithmically for ByteNet. This makes\n
it more difficult to learn dependencies between distant positions [ 12]. In the Transform
er this is\nreduced to a constant number of operations, albeit at the cost of reduced eff
ective resolution due\nto averaging attention-weighted positions, an effect we counteract
with Multi-Head Attention as\ndescribed in section 3.2.\nSelf-attention, sometimes called
intra-attention is an attention mechanism relating different positions\nof a single seque
nce in order to compute a representation of the sequence. Self-attention has been\nused s
uccessfully in a variety of tasks including reading comprehension, abstractive summarizat
ion,\ntextual entailment and learning task-independent sentence representations [4, 27, 2
8, 22].\nEnd-to-end memory networks are based on a recurrent attention mechanism instead
of sequence-\naligned recurrence and have been shown to perform well on simple-language q
uestion answering and\nlanguage modeling tasks [34].\nTo the best of our knowledge, howev
er, the Transformer is the first transduction model relying\nentirely on self-attention t
o compute representations of its input and output without using sequence-\naligned RNNs o
r convolution. In the following sections, we will describe the Transformer, motivate\nsel
f-attention and discuss its advantages over models such as [17, 18] and [9].\n3 Model Arc
hitecture\nMost competitive neural sequence transduction models have an encoder-decoder s
tructure [ 5,2,35].\nHere, the encoder maps an input sequence of symbol representations (
x1, ..., x n)to a sequence\nof continuous representations z= (z1, ..., z n). Given z, the
decoder then generates an output\nsequence (y1, ..., y m)of symbols one element at a time
. At each step the model is auto-regressive\n[10], consuming the previously generated sym
bols as additional input when generating the next.\n2', metadata={'source': 'attention.pd
f', 'page': 1}),
 Document(page_content='Figure 1: The Transformer - model architecture.\nThe Transformer
follows this overall architecture using stacked self-attention and point-wise, fully\ncon
nected layers for both the encoder and decoder, shown in the left and right halves of Fig
ure 1,\nrespectively.\n3.1 Encoder and Decoder Stacks\nEncoder: The encoder is composed o
f a stack of N= 6 identical layers. Each layer has two\nsub-layers. The first is a multi-
head self-attention mechanism, and the second is a simple, position-\nwise fully connecte
d feed-forward network. We employ a residual connection [ 11] around each of\nthe two sub
-layers, followed by layer normalization [ 1]. That is, the output of each sub-layer is\n
LayerNorm( x+ Sublayer( x)), where Sublayer( x)is the function implemented by the sub-lay
er\nitself. To facilitate these residual connections, all sub-layers in the model, as wel
l as the embedding\nlayers, produce outputs of dimension dmodel = 512 .\nDecoder: The dec
oder is also composed of a stack of N= 6identical layers. In addition to the two\nsub-lay
ers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-he
ad\nattention over the output of the encoder stack. Similar to the encoder, we employ res
idual connections\naround each of the sub-layers, followed by layer normalization. We als
o modify the self-attention\nsub-layer in the decoder stack to prevent positions from att
ending to subsequent positions. This\nmasking, combined with fact that the output embeddi
ngs are offset by one position, ensures that the\npredictions for position ican depend on
ly on the known outputs at positions less than i.\n3.2 Attention\nAn attention function c
an be described as mapping a query and a set of key-value pairs to an output,\nwhere the
query, keys, values, and output are all vectors. The output is computed as a weighted sum
\n3', metadata={'source': 'attention.pdf', 'page': 2}),
 Document(page_content='Scaled Dot-Product Attention\n Multi-Head Attention\nFigure 2: (l
eft) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several\natte
ntion layers running in parallel.\nof the values, where the weight assigned to each value
is computed by a compatibility function of the\nquery with the corresponding key.\n3.2.1
Scaled Dot-Product Attention\nWe call our particular attention "Scaled Dot-Product Attent
ion" (Figure 2). The input consists of\nqueries and keys of dimension dk, and values of d
imension dv. We compute the dot products of the\nquery with all keys, divide each by√dk,
and apply a softmax function to obtain the weights on the\nvalues.\nIn practice, we compu
te the attention function on a set of queries simultaneously, packed together\ninto a mat
rix Q. The keys and values are also packed together into matrices KandV. We compute\nthe
matrix of outputs as:\nAttention( Q, K, V ) = softmax(QKT\n√dk)V (1)\nThe two most common
ly used attention functions are additive attention [ 2], and dot-product (multi-\nplicati
ve) attention. Dot-product attention is identical to our algorithm, except for the scalin
g factor\nof1√dk. Additive attention computes the compatibility function using a feed-for
ward network with\na single hidden layer. While the two are similar in theoretical comple
```

xity, dot-product attention is\nmuch faster and more space-efficient in practice, since i
t can be implemented using highly optimized\nmatrix multiplication code.\nWhile for small
values of dkthe two mechanisms perform similarly, additive attention outperforms\ndot pro
duct attention without scaling for larger values of dk[3]. We suspect that for large valu
es of\ndk, the dot products grow large in magnitude, pushing the softmax function into re
gions where it has\nextremely small gradients4. To counteract this effect, we scale the d
ot products by1√dk.\n3.2.2 Multi-Head Attention\nInstead of performing a single attention
function with dmodel-dimensional keys, values and queries,\nwe found it beneficial to lin
early project the queries, keys and values htimes with different, learned\nlinear project
ions to dk,dkanddvdimensions, respectively. On each of these projected versions of\nqueri
es, keys and values we then perform the attention function in parallel, yielding dv-dimen
sional\n4To illustrate why the dot products get large, assume that the components of qand
kare independent random\nvariables with mean 0and variance 1. Then their dot product, q·k
=Pdk\ni=1qiki, has mean 0and variance dk.\n4', metadata={'source': 'attention.pdf', 'page
': 3}),
 Document(page_content='output values. These are concatenated and once again projected, r
esulting in the final values, as\ndepicted in Figure 2.\nMulti-head attention allows the
model to jointly attend to information from different representation\nsubspaces at differ
ent positions. With a single attention head, averaging inhibits this.\nMultiHead(Q, K, V
) = Concat(head 1, ...,head h)WO\nwhere head i= Attention(QWQ\ni, KWK\ni, V WV\ni)\nWher
e the projections are parameter matrices WQ\ni∈Rdmodel×dk,WK\ni∈Rdmodel×dk,WV\ni∈Rdmodel×
dv\nandWO∈Rhdv×dmodel.\nIn this work we employ h= 8 parallel attention layers, or heads.
For each of these we use\ndk=dv=dmodel/h= 64 . Due to the reduced dimension of each head,
the total computational cost\nis similar to that of single-head attention with full dimen
sionality.\n3.2.3 Applications of Attention in our Model\nThe Transformer uses multi-head
attention in three different ways:\n•In "encoder-decoder attention" layers, the queries c
ome from the previous decoder layer,\nand the memory keys and values come from the output
of the encoder. This allows every\nposition in the decoder to attend over all positions i
n the input sequence. This mimics the\ntypical encoder-decoder attention mechanisms in se
quence-to-sequence models such as\n[38, 2, 9].\n•The encoder contains self-attention laye
rs. In a self-attention layer all of the keys, values\nand queries come from the same pla
ce, in this case, the output of the previous layer in the\nencoder. Each position in the
encoder can attend to all positions in the previous layer of the\nencoder.\n•Similarly, s
elf-attention layers in the decoder allow each position in the decoder to attend to\nall
positions in the decoder up to and including that position. We need to prevent leftward\n
information flow in the decoder to preserve the auto-regressive property. We implement th
is\ninside of scaled dot-product attention by masking out (setting to −∞) all values in t
he input\nof the softmax which correspond to illegal connections. See Figure 2.\n3.3 Posi
tion-wise Feed-Forward Networks\nIn addition to attention sub-layers, each of the layers
in our encoder and decoder contains a fully\nconnected feed-forward network, which is app
lied to each position separately and identically. This\nconsists of two linear transforma
tions with a ReLU activation in between.\nFFN(x) = max(0 , xW 1+b1)W2+b2 (2)\nWhile the
linear transformations are the same across different positions, they use different parame
ters\nfrom layer to layer. Another way of describing this is as two convolutions with ker
nel size 1.\nThe dimensionality of input and output is dmodel = 512 , and the inner-layer
has dimensionality\ndff= 2048 .\n3.4 Embeddings and Softmax\nSimilarly to other sequence
transduction models, we use learned embeddings to convert the input\ntokens and output to
kens to vectors of dimension dmodel. We also use the usual learned linear transfor-\nmati
on and softmax function to convert the decoder output to predicted next-token probabiliti
es. In\nour model, we share the same weight matrix between the two embedding layers and t
he pre-softmax\nlinear transformation, similar to [30]. In the embedding layers, we mult
iply those weights by√dmodel.\n5', metadata={'source': 'attention.pdf', 'page': 4}),
 Document(page_content='Table 1: Maximum path lengths, per-layer complexity and minimum n
umber of sequential operations\nfor different layer types. nis the sequence length, dis t
he representation dimension, kis the kernel\nsize of convolutions and rthe size of the ne
ighborhood in restricted self-attention.\nLayer Type Complexity per Layer Sequential Maxi
mum Path Length\nOperations\nSelf-Attention O(n2·d) O(1) O(1)\nRecurrent O(n·d2) O(n) O(n
)\nConvolutional O(k·n·d2) O(1) O(logk(n))\nSelf-Attention (restricted) O(r·n·d) O(1) O(n
/r)\n3.5 Positional Encoding\nSince our model contains no recurrence and no convolution,
in order for the model to make use of the\norder of the sequence, we must inject some inf
ormation about the relative or absolute position of the\ntokens in the sequence. To this
end, we add "positional encodings" to the input embeddings at the\nbottoms of the encoder
and decoder stacks. The positional encodings have the same dimension dmodel\nas the embed
dings, so that the two can be summed. There are many choices of positional encodings,\nle
arned and fixed [9].\nIn this work, we use sine and cosine functions of different frequen
cies:\nPE(pos,2i)=sin(pos/100002i/d model)\nPE(pos,2i+1)=cos(pos/100002i/d model)\nwhere
posis the position and iis the dimension. That is, each dimension of the positional encod
ing\ncorresponds to a sinusoid. The wavelengths form a geometric progression from 2πto100
00 ·2π. We\nchose this function because we hypothesized it would allow the model to easil
y learn to attend by\nrelative positions, since for any fixed offset k,PEpos+kcan be repr
esented as a linear function of\nPEpos.\nWe also experimented with using learned position
al embeddings [9] instead, and found that the two\nversions produced nearly identical re

sults (see Table 3 row (E)). We chose the sinusoidal version\nbecause it may allow the mo
del to extrapolate to sequence lengths longer than the ones encountered\nduring training.
\n4 Why Self-Attention\nIn this section we compare various aspects of self-attention laye
rs to the recurrent and convolu-\ntional layers commonly used for mapping one variable-le
ngth sequence of symbol representations\n(x1, ..., x n)to another sequence of equal lengt
h (z1, ..., z n), with xi, zi∈Rd, such as a hidden\nlayer in a typical sequence transduct
ion encoder or decoder. Motivating our use of self-attention we\nconsider three desiderat
a.\nOne is the total computational complexity per layer. Another is the amount of computa
tion that can\nbe parallelized, as measured by the minimum number of sequential operation
s required.\nThe third is the path length between long-range dependencies in the network.
Learning long-range\ndependencies is a key challenge in many sequence transduction tasks.
One key factor affecting the\nability to learn such dependencies is the length of the pat
hs forward and backward signals have to\ntraverse in the network. The shorter these paths
between any combination of positions in the input\nand output sequences, the easier it is
to learn long-range dependencies [12]. Hence we also compare\nthe maximum path length be
tween any two input and output positions in networks composed of the\ndifferent layer typ
es.\nAs noted in Table 1, a self-attention layer connects all positions with a constant n
umber of sequentially\nexecuted operations, whereas a recurrent layer requires O(n)sequen
tial operations. In terms of\ncomputational complexity, self-attention layers are faster
than recurrent layers when the sequence\n6', metadata={'source': 'attention.pdf', 'page':
5}),
Document(page_content='length nis smaller than the representation dimensionality d, whic
h is most often the case with\nsentence representations used by state-of-the-art models i
n machine translations, such as word-piece\n[38] and byte-pair [31] representations. To
improve computational performance for tasks involving\nvery long sequences, self-attentio
n could be restricted to considering only a neighborhood of size rin\nthe input sequence
centered around the respective output position. This would increase the maximum\npath len
gth to O(n/r). We plan to investigate this approach further in future work.\nA single con
volutional layer with kernel width k < n does not connect all pairs of input and output\n
positions. Doing so requires a stack of O(n/k)convolutional layers in the case of contigu
ous kernels,\norO(logk(n))in the case of dilated convolutions [18], increasing the lengt
h of the longest paths\nbetween any two positions in the network. Convolutional layers ar
e generally more expensive than\nrecurrent layers, by a factor of k. Separable convolutio
ns [6], however, decrease the complexity\nconsiderably, to O(k·n·d+n·d2). Even with k=n,
however, the complexity of a separable\nconvolution is equal to the combination of a self
-attention layer and a point-wise feed-forward layer,\nthe approach we take in our model.
\nAs side benefit, self-attention could yield more interpretable models. We inspect atten
tion distributions\nfrom our models and present and discuss examples in the appendix. Not
only do individual attention\nheads clearly learn to perform different tasks, many appear
to exhibit behavior related to the syntactic\nand semantic structure of the sentences.\n5
Training\nThis section describes the training regime for our models.\n5.1 Training Data a
nd Batching\nWe trained on the standard WMT 2014 English-German dataset consisting of abo
ut 4.5 million\nsentence pairs. Sentences were encoded using byte-pair encoding [3], whi
ch has a shared source-\ntarget vocabulary of about 37000 tokens. For English-French, we
used the significantly larger WMT\n2014 English-French dataset consisting of 36M sentence
s and split tokens into a 32000 word-piece\nvocabulary [38]. Sentence pairs were batched
together by approximate sequence length. Each training\nbatch contained a set of sentence
pairs containing approximately 25000 source tokens and 25000\ntarget tokens.\n5.2 Hardwar
e and Schedule\nWe trained our models on one machine with 8 NVIDIA P100 GPUs. For our bas
e models using\nthe hyperparameters described throughout the paper, each training step to
ok about 0.4 seconds. We\ntrained the base models for a total of 100,000 steps or 12 hour
s. For our big models,(described on the\nbottom line of table 3), step time was 1.0 secon
ds. The big models were trained for 300,000 steps\n(3.5 days).\n5.3 Optimizer\nWe used th
e Adam optimizer [20] with β1= 0.9,β2= 0.98andϵ= 10−9. We varied the learning\nrate over
the course of training, according to the formula:\nlrate =d−0.5\nmodel·min(step_num−0.5,
step _num·warmup _steps−1.5) (3)\nThis corresponds to increasing the learning rate linear
ly for the first warmup _steps training steps,\nand decreasing it thereafter proportional
ly to the inverse square root of the step number. We used\nwarmup _steps = 4000 .\n5.4 Re
gularization\nWe employ three types of regularization during training:\n7', metadata={'so
urce': 'attention.pdf', 'page': 6}),
Document(page_content='Table 2: The Transformer achieves better BLEU scores than previou
s state-of-the-art models on the\nEnglish-to-German and English-to-French newstest2014 te
sts at a fraction of the training cost.\nModelBLEU Training Cost (FLOPs)\nEN-DE EN-FR EN-
DE EN-FR\nByteNet [18] 23.75\nDeep-Att + PosUnk [39] 39.2 1.0·1020\nGNMT + RL [38] 24.6 3
9.92 2.3·10191.4·1020\nConvS2S [9] 25.16 40.46 9.6·10181.5·1020\nMoE [32] 26.03 40.56 2.0
·10191.2·1020\nDeep-Att + PosUnk Ensemble [39] 40.4 8.0·1020\nGNMT + RL Ensemble [38] 26.
30 41.16 1.8·10201.1·1021\nConvS2S Ensemble [9] 26.36 41.29 7.7·10191.2·1021\nTransformer
(base model) 27.3 38.1 3.3·1018\nTransformer (big) 28.4 41.8 2.3·1019\nResidual Dropout W
e apply dropout [33] to the output of each sub-layer, before it is added to the\nsub-lay
er input and normalized. In addition, we apply dropout to the sums of the embeddings and
the\npositional encodings in both the encoder and decoder stacks. For the base model, we
use a rate of\nPdrop= 0.1.\nLabel Smoothing During training, we employed label smoothing

of value ϵls= 0.1[36]. This\nhurts perplexity, as the model learns to be more unsure, but improves accuracy and BLEU score.\n6 Results\n6.1 Machine Translation\nOn the WMT 2014 English-to-German translation task, the big transformer model (Transformer (big)\nin Table 2) outperforms the best previously reported models (including ensembles) by more than 2.0\nBLEU, establishing a new state-of-the-art BLEU score of 28.4. The configuration of this model is\nlisted in the bottom line of Table 3. Training took 3.5days on 8P100 GPUs. Even our base model\nsurpasses all previously published models and ensembles, at a fraction of the training cost of any of\nthe competitive models.\nOn the WMT 2014 English-to-French translation task, our big model achieves a BLEU score of 41.0,\noutperforming all of the previously published single models, at less than 1/4the training cost of the\nprevious state-of-the-art model. The Transformer (big) model trained for English-to-French used\ndropout rate Pdrop= 0.1, instead of 0.3.\nFor the base models, we used a single model obtained by averaging the last 5 checkpoints, which\nwere written at 10-minute intervals. For the big models, we averaged the last 20 checkpoints. We\nused beam search with a beam size of 4and length penalty α= 0.6[38]. These hyperparameters\nwere chosen after experimentation on the development set. We set the maximum output length during\ninference to input length + 50, but terminate early when possible [38].\nTable 2 summarizes our results and compares our translation quality and training costs to other model\narchitectures from the literature. We estimate the number of floating point operations used to train a\nmodel by multiplying the training time, the number of GPUs used, and an estimate of the sustained\nsingle-precision floating-point capacity of each GPU5.\n6.2 Model Variations\nTo evaluate the importance of different components of the Transformer, we varied our base model\nin different ways, measuring the change in performance on English-to-German translation on the\n5We used values of 2.8, 3.7, 6.0 and 9.5 TFLOPS for K80, K40, M40 and P100, respectively.\n8', metadata={'source': 'attention.pdf', 'page': 7}),

Document(page_content='Table 3: Variations on the Transformer architecture. Unlisted values are identical to those of the base\nmodel. All metrics are on the English-to-German translation development set, newstest2013. Listed\nperplexities are per-wordpiece, according to our byte-pair encoding, and should not be compared to\nper-word perplexities.\nN d model dff h d k dvPdrop ϵlstrain PPL BLEU params\nsteps (dev) (dev) ×106\nbase 6 512 2048 8 64 64 0.1 0.1 100K 4.92 25.8 65\n(A)1 512 512 5.29 24.9\n4 128 128 5.00 25.5\n16 32 32 4.91 25.8\n32 16 16 5.01 25.4\n(B)16 5.16 25.1 58\n32 5.01 25.4 60\n(C)2 6.11 23.7 36\n4 5.19 25.3 50\n8 4.88 25.5 80\n256 32 32 5.75 24.5 28\n1024 128 128 4.66 26.0 168\n1024 5.12 25.4 53\n4096 4.75 26.2 90\n(D)0.0 5.77 24.6\n0.2 4.95 25.5\n0.0 4.67 25.3\n0.2 5.47 25.7\n(E) positional embedding instead of sinusoids 4.92 25.7\nbig 6 1024 4096 16 0.3 300K 4.33 26.4 213\ndevelopment set, newstest2013. We used beam search as described in the previous section, but no\ncheckpoint averaging. We present these results in Table 3.\nIn Table 3 rows (A), we vary the number of attention heads and the attention key and value dimensions,\nkeeping the amount of computation constant, as described in Section 3.2.2. While single-head\nattention is 0.9 BLEU worse than the best setting, quality also drops off with too many heads.\nIn Table 3 rows (B), we observe that reducing the attention key size dkhurts model quality. This\nsuggests that determining compatibility is not easy and that a more sophisticated compatibility\nfunction than dot product may be beneficial. We further observe in rows (C) and (D) that, as expected,\nbigger models are better, and dropout is very helpful in avoiding over-fitting. In row (E) we replace our\nsinusoidal positional encoding with learned positional embeddings [9], and observe nearly identical\nresults to the base model.\n6.3 English Constituency Parsing\nTo evaluate if the Transformer can generalize to other tasks we performed experiments on English\nconstituency parsing. This task presents specific challenges: the output is subject to strong structural\nconstraints and is significantly longer than the input. Furthermore, RNN sequence-to-sequence\nmodels have not been able to attain state-of-the-art results in small-data regimes [37].\nWe trained a 4-layer transformer with dmodel = 1024 on the Wall Street Journal (WSJ) portion of the\nPenn Treebank [25], about 40K training sentences. We also trained it in a semi-supervised setting,\nusing the larger high-confidence and BerkleyParser corpora from with approximately 17M sentences\n[37]. We used a vocabulary of 16K tokens for the WSJ only setting and a vocabulary of 32K tokens\nfor the semi-supervised setting.\nWe performed only a small number of experiments to select the dropout, both attention and residual\n(section 5.4), learning rates and beam size on the Section 22 development set, all other parameters\nremained unchanged from the English-to-German base translation model. During inference, we\n9', metadata={'source': 'attention.pdf', 'page': 8}),

Document(page_content='Table 4: The Transformer generalizes well to English constituency parsing (Results are on Section 23\nof WSJ)\nParser Training WSJ 23 F1\nVinyals & Kaiser el al. (2014) [37] WSJ only, discriminative 88.3\nPetrov et al. (2006) [29] WSJ only, discriminative 90.4\nZhu et al. (2013) [40] WSJ only, discriminative 90.4\nDyer et al. (2016) [8] WSJ only, discriminative 91.7\nTransformer (4 layers) WSJ only, discriminative 91.3\nZhu et al. (2013) [40] semi-supervised 91.3\nHuang & Harper (2009) [14] semi-supervised 91.3\nMcClosky et al. (2006) [26] semi-supervised 92.1\nVinyals & Kaiser el al. (2014) [37] semi-supervised 92.1\nTransformer (4 layers) semi-supervised 92.7\nLuong et al. (2015) [23] multi-task 93.0\nDyer et al. (2016) [8] generative 93.3\nincreased the maximum output length to input length + 300. We used a beam size of 21andα= 0.3\nfor both WSJ only and the semi-supervised setting.\nOur results in Table 4 show that despite the lack of task-specific tuning our model performs sur-\nprisingly well, yielding better results than all

previously reported models with the exception of the\nRecurrent Neural Network Grammar [8
].\nIn contrast to RNN sequence-to-sequence models [37], the Transformer outperforms the
Berkeley-\nParser [29] even when training only on the WSJ training set of 40K sentences.\
n7 Conclusion\nIn this work, we presented the Transformer, the first sequence transductio
n model based entirely on\nattention, replacing the recurrent layers most commonly used i
n encoder-decoder architectures with\nmulti-headed self-attention.\nFor translation tasks
, the Transformer can be trained significantly faster than architectures based\non recurr
ent or convolutional layers. On both WMT 2014 English-to-German and WMT 2014\nEnglish-to-
French translation tasks, we achieve a new state of the art. In the former task our best\
nmodel outperforms even all previously reported ensembles.\nWe are excited about the futu
re of attention-based models and plan to apply them to other tasks. We\nplan to extend th
e Transformer to problems involving input and output modalities other than text and\nto i
nvestigate local, restricted attention mechanisms to efficiently handle large inputs and
outputs\nsuch as images, audio and video. Making generation less sequential is another re
search goals of ours.\nThe code we used to train and evaluate our models is available at
https://github.com/\ntensorflow/tensor2tensor .\nAcknowledgements We are grateful to Nal
Kalchbrenner and Stephan Gouws for their fruitful\ncomments, corrections and inspiration.
\nReferences\n[1]Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalizati
on. arXiv preprint\narXiv:1607.06450 , 2016.\n[2]Dzmitry Bahdanau, Kyunghyun Cho, and Yos
hua Bengio. Neural machine translation by jointly\nlearning to align and translate. CoRR
, abs/1409.0473, 2014.\n[3]Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc V . Le. M
assive exploration of neural\nmachine translation architectures. CoRR , abs/1703.03906, 2
017.\n[4]Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for
machine\nreading. arXiv preprint arXiv:1601.06733 , 2016.\n10', metadata={'source': 'atte
ntion.pdf', 'page': 9}),
 Document(page_content='[5]Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bo
ugares, Holger Schwenk,\nand Yoshua Bengio. Learning phrase representations using rnn enc
oder-decoder for statistical\nmachine translation. CoRR , abs/1406.1078, 2014.\n[6]Franco
is Chollet. Xception: Deep learning with depthwise separable convolutions. arXiv\npreprin
t arXiv:1610.02357 , 2016.\n[7]Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua
Bengio. Empirical evaluation\nof gated recurrent neural networks on sequence modeling. Co
RR , abs/1412.3555, 2014.\n[8]Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah
A. Smith. Recurrent neural\nnetwork grammars. In Proc. of NAACL , 2016.\n[9]Jonas Gehring
, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolu-\ntional seque
nce to sequence learning. arXiv preprint arXiv:1705.03122v2 , 2017.\n[10] Alex Graves. Ge
nerating sequences with recurrent neural networks. arXiv preprint\narXiv:1308.0850 , 2013
.\n[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for
im-\nage recognition. In Proceedings of the IEEE Conference on Computer Vision and Patter
n\nRecognition , pages 770–778, 2016.\n[12] Sepp Hochreiter, Yoshua Bengio, Paolo Frascon
i, and Jürgen Schmidhuber. Gradient flow in\nrecurrent nets: the difficulty of learning l
ong-term dependencies, 2001.\n[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-ter
m memory. Neural computation ,\n9(8):1735–1780, 1997.\n[14] Zhongqiang Huang and Mary Har
per. Self-training PCFG grammars with latent annotations\nacross languages. In Proceeding
s of the 2009 Conference on Empirical Methods in Natural\nLanguage Processing , pages 832
–841. ACL, August 2009.\n[15] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazee
r, and Yonghui Wu. Exploring\nthe limits of language modeling. arXiv preprint arXiv:1602.
02410 , 2016.\n[16] Łukasz Kaiser and Samy Bengio. Can active memory replace attention? I
n Advances in Neural\nInformation Processing Systems, (NIPS) , 2016.\n[17] Łukasz Kaiser
and Ilya Sutskever. Neural GPUs learn algorithms. In International Conference\non Learnin
g Representations (ICLR) , 2016.\n[18] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan,
Aaron van den Oord, Alex Graves, and Ko-\nray Kavukcuoglu. Neural machine translation in
linear time. arXiv preprint arXiv:1610.10099v2 ,\n2017.\n[19] Yoon Kim, Carl Denton, Luon
g Hoang, and Alexander M. Rush. Structured attention networks.\nInInternational Conferenc
e on Learning Representations , 2017.\n[20] Diederik Kingma and Jimmy Ba. Adam: A method
for stochastic optimization. In ICLR , 2015.\n[21] Oleksii Kuchaiev and Boris Ginsburg. F
actorization tricks for LSTM networks. arXiv preprint\narXiv:1703.10722 , 2017.\n[22] Zho
uhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen\nZhou, and Yo
shua Bengio. A structured self-attentive sentence embedding. arXiv preprint\narXiv:1703.0
3130 , 2017.\n[23] Minh-Thang Luong, Quoc V . Le, Ilya Sutskever, Oriol Vinyals, and Luka
sz Kaiser. Multi-task\nsequence to sequence learning. arXiv preprint arXiv:1511.06114 , 2
015.\n[24] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches t
o attention-\nbased neural machine translation. arXiv preprint arXiv:1508.04025 , 2015.\n
11', metadata={'source': 'attention.pdf', 'page': 10}),
 Document(page_content='[25] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Sant
orini. Building a large annotated\ncorpus of english: The penn treebank. Computational li
nguistics , 19(2):313–330, 1993.\n[26] David McClosky, Eugene Charniak, and Mark Johnson.
Effective self-training for parsing. In\nProceedings of the Human Language Technology Con
ference of the NAACL, Main Conference ,\npages 152–159. ACL, June 2006.\n[27] Ankur Parik
h, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention\nmodel. I
n Empirical Methods in Natural Language Processing , 2016.\n[28] Romain Paulus, Caiming X
iong, and Richard Socher. A deep reinforced model for abstractive\nsummarization. arXiv p
reprint arXiv:1705.04304 , 2017.\n[29] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan

```
Klein. Learning accurate, compact,\nand interpretable tree annotation. In Proceedings of
the 21st International Conference on\nComputational Linguistics and 44th Annual Meeting o
f the ACL , pages 433–440. ACL, July\n2006.\n[30] Ofir Press and Lior Wolf. Using the out
put embedding to improve language models. arXiv\npreprint arXiv:1608.05859 , 2016.\n[31]
Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare word
s\nwith subword units. arXiv preprint arXiv:1508.07909 , 2015.\n[32] Noam Shazeer, Azalia
Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton,\nand Jeff Dean. Outr
ageously large neural networks: The sparsely-gated mixture-of-experts\nlayer. arXiv prepr
int arXiv:1701.06538 , 2017.\n[33] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky,
Ilya Sutskever, and Ruslan Salakhutdi-\nnov. Dropout: a simple way to prevent neural netw
orks from overfitting. Journal of Machine\nLearning Research , 15(1):1929–1958, 2014.\n[3
4] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory\nn
etworks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors,\nA
dvances in Neural Information Processing Systems 28 , pages 2440–2448. Curran Associates,
\nInc., 2015.\n[35] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence l
earning with neural\nnetworks. In Advances in Neural Information Processing Systems , pag
es 3104–3112, 2014.\n[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Sh
lens, and Zbigniew Wojna.\nRethinking the inception architecture for computer vision. CoR
R , abs/1512.00567, 2015.\n[37] Vinyals & Kaiser, Koo, Petrov, Sutskever, and Hinton. Gra
mmar as a foreign language. In\nAdvances in Neural Information Processing Systems , 2015.
\n[38] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang\nMa
cherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine\n
translation system: Bridging the gap between human and machine translation. arXiv preprin
t\narXiv:1609.08144 , 2016.\n[39] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu.
Deep recurrent models with\nfast-forward connections for neural machine translation. CoRR
, abs/1606.04199, 2016.\n[40] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo
Zhu. Fast and accurate\nshift-reduce constituent parsing. In Proceedings of the 51st Annu
al Meeting of the ACL (Volume\n1: Long Papers) , pages 434–443. ACL, August 2013.\n12', m
etadata={'source': 'attention.pdf', 'page': 11}),
 Document(page_content='Attention Visualizations\nInput-Input Layer5\nIt\nis\nin\nthis\ns
pirit\nthat\na\nmajority\nof\nAmerican\ngovernments\nhave\npassed\nnew\nlaws\nsince\n2009
\nmaking\nthe\nregistration\nor\nvoting\nprocess\nmore\ndifficult\n.\n<EOS>\n<pad>\n<pad>
\n<pad>\n<pad>\n<pad>\n<pad>\nIt\nis\nin\nthis\nspirit\nthat\na\nmajority\nof\nAmerican\n
governments\nhave\npassed\nnew\nlaws\nsince\n2009\nmaking\nthe\nregistration\nor\nvoting\
nprocess\nmore\ndifficult\n.\n<EOS>\n<pad>\n<pad>\n<pad>\n<pad>\n<pad>\n<pad>\nFigure 3:
An example of the attention mechanism following long-distance dependencies in the\nencode
r self-attention in layer 5 of 6. Many of the attention heads attend to a distant depende
ncy of\nthe verb 'making', completing the phrase 'making...more difficult'. Attentions he
re shown only for\nthe word 'making'. Different colors represent different heads. Best vi
ewed in color.\n13', metadata={'source': 'attention.pdf', 'page': 12}),
 Document(page_content='Input-Input Layer5\nThe\nLaw\nwill\nnever\nbe\nperfect\n,\nbut\ni
ts\napplication\nshould\nbe\njust\n-\nthis\nis\nwhat\nwe\nare\nmissing\n,\nin\nmy\nopinio
n\n.\n<EOS>\n<pad>\nThe\nLaw\nwill\nnever\nbe\nperfect\n,\nbut\nits\napplication\nshould\
nbe\njust\n-\nthis\nis\nwhat\nwe\nare\nmissing\n,\nin\nmy\nopinion\n.\n<EOS>\n<pad>\nInpu
t-Input Layer5\nThe\nLaw\nwill\nnever\nbe\nperfect\n,\nbut\nits\napplication\nshould\nbe\
njust\n-\nthis\nis\nwhat\nwe\nare\nmissing\n,\nin\nmy\nopinion\n.\n<EOS>\n<pad>\nThe\nLaw
\nwill\nnever\nbe\nperfect\n,\nbut\nits\napplication\nshould\nbe\njust\n-\nthis\nis\nwhat
\nwe\nare\nmissing\n,\nin\nmy\nopinion\n.\n<EOS>\n<pad>Figure 4: Two attention heads, als
o in layer 5 of 6, apparently involved in anaphora resolution. Top:\nFull attentions for
head 5. Bottom: Isolated attentions from just the word 'its' for attention heads 5\nand 6
. Note that the attentions are very sharp for this word.\n14', metadata={'source': 'atten
tion.pdf', 'page': 13}),
 Document(page_content='Input-Input Layer5\nThe\nLaw\nwill\nnever\nbe\nperfect\n,\nbut\ni
ts\napplication\nshould\nbe\njust\n-\nthis\nis\nwhat\nwe\nare\nmissing\n,\nin\nmy\nopinio
n\n.\n<EOS>\n<pad>\nThe\nLaw\nwill\nnever\nbe\nperfect\n,\nbut\nits\napplication\nshould\
nbe\njust\n-\nthis\nis\nwhat\nwe\nare\nmissing\n,\nin\nmy\nopinion\n.\n<EOS>\n<pad>\nInpu
t-Input Layer5\nThe\nLaw\nwill\nnever\nbe\nperfect\n,\nbut\nits\napplication\nshould\nbe\
njust\n-\nthis\nis\nwhat\nwe\nare\nmissing\n,\nin\nmy\nopinion\n.\n<EOS>\n<pad>\nThe\nLaw
\nwill\nnever\nbe\nperfect\n,\nbut\nits\napplication\nshould\nbe\njust\n-\nthis\nis\nwhat
\nwe\nare\nmissing\n,\nin\nmy\nopinion\n.\n<EOS>\n<pad>Figure 5: Many of the attention he
ads exhibit behaviour that seems related to the structure of the\nsentence. We give two s
uch examples above, from two different heads from the encoder self-attention\nat layer 5
of 6. The heads clearly learned to perform different tasks.\n15', metadata={'source': 'at
tention.pdf', 'page': 14})]
```


Transform

In [7]:

```
from langchain.text_splitter import RecursiveCharacterTextSplitter
text_splitter=RecursiveCharacterTextSplitter(chunk_size=1000,chunk_overlap=200)
```

```
documents=text_splitter.split_documents(docs)
documents[:5]
```

Out[7]:

```
[Document(page_content='Provided proper attribution is provided, Google hereby grants per
mission to\nreproduce the tables and figures in this paper solely for use in journalistic
or\nscholarly works.\nAttention Is All You Need\nAshish Vaswani*\nGoogle Brain\navaswani@
google.comNoam Shazeer*\nGoogle Brain\nnoam@google.comNiki Parmar*\nGoogle Research\nniki
p@google.comJakob Uszkoreit*\nGoogle Research\nusz@google.com\nLlion Jones*\nGoogle Resea
rch\nllion@google.comAidan N. Gomez* †\nUniversity of Toronto\naidan@cs.toronto.eduŁukasz
Kaiser*\nGoogle Brain\nlukaszkaiser@google.com\nIllia Polosukhin* ‡\nillia.polosukhin@gma
il.com\nAbstract\nThe dominant sequence transduction models are based on complex recurren
t or\nconvolutional neural networks that include an encoder and a decoder. The best\nperf
orming models also connect the encoder and decoder through an attention\nmechanism. We pr
opose a new simple network architecture, the Transformer,\nbased solely on attention mech
anisms, dispensing with recurrence and convolutions', metadata={'source': 'attention.pdf'
, 'page': 0}),
 Document(page_content='mechanism. We propose a new simple network architecture, the Tran
sformer,\nbased solely on attention mechanisms, dispensing with recurrence and convolutio
ns\nentirely. Experiments on two machine translation tasks show these models to\nbe super
ior in quality while being more parallelizable and requiring significantly\nless time to
train. Our model achieves 28.4 BLEU on the WMT 2014 English-\nto-German translation task,
improving over the existing best results, including\nensembles, by over 2 BLEU. On the WM
T 2014 English-to-French translation task,\nour model establishes a new single-model stat
e-of-the-art BLEU score of 41.8 after\ntraining for 3.5 days on eight GPUs, a small fract
ion of the training costs of the\nbest models from the literature. We show that the Trans
former generalizes well to\nother tasks by applying it successfully to English constituen
cy parsing both with\nlarge and limited training data.', metadata={'source': 'attention.p
df', 'page': 0}),
 Document(page_content='best models from the literature. We show that the Transformer gen
eralizes well to\nother tasks by applying it successfully to English constituency parsing
both with\nlarge and limited training data.\n*Equal contribution. Listing order is random
. Jakob proposed replacing RNNs with self-attention and started\nthe effort to evaluate t
his idea. Ashish, with Illia, designed and implemented the first Transformer models and\n
has been crucially involved in every aspect of this work. Noam proposed scaled dot-produc
t attention, multi-head\nattention and the parameter-free position representation and bec
ame the other person involved in nearly every\ndetail. Niki designed, implemented, tuned
and evaluated countless model variants in our original codebase and\ntensor2tensor. Llion
also experimented with novel model variants, was responsible for our initial codebase, an
d\nefficient inference and visualizations. Lukasz and Aidan spent countless long days des
igning various parts of and', metadata={'source': 'attention.pdf', 'page': 0}),
 Document(page_content='efficient inference and visualizations. Lukasz and Aidan spent co
untless long days designing various parts of and\nimplementing tensor2tensor, replacing o
ur earlier codebase, greatly improving results and massively accelerating\nour research.\
n†Work performed while at Google Brain.\n‡Work performed while at Google Research.\n31st
Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.arXi
v:1706.03762v7  [cs.CL]  2 Aug 2023', metadata={'source': 'attention.pdf', 'page': 0}),
 Document(page_content='1 Introduction\nRecurrent neural networks, long short-term memory
[ 13] and gated recurrent [ 7] neural networks\nin particular, have been firmly establish
ed as state of the art approaches in sequence modeling and\ntransduction problems such as
language modeling and machine translation [ 35,2,5]. Numerous\nefforts have since continu
ed to push the boundaries of recurrent language models and encoder-decoder\narchitectures
[38, 24, 15].\nRecurrent models typically factor computation along the symbol positions o
f the input and output\nsequences. Aligning the positions to steps in computation time, t
hey generate a sequence of hidden\nstates ht, as a function of the previous hidden state
ht−1and the input for position t. This inherently\nsequential nature precludes paralleliz
ation within training examples, which becomes critical at longer\nsequence lengths, as me
mory constraints limit batching across examples. Recent work has achieved', metadata={'so
urce': 'attention.pdf', 'page': 1})]
```

In [8]:

```
## Vector Embedding And Vector Store
from langchain_openai import OpenAIEmbeddings
from langchain_community.vectorstores import Chroma
db = Chroma.from_documents(documents,OpenAIEmbeddings())
```

In [15]:

```
## Vector Database
query = "An attention function can be described as mapping query"
```

```
retireved_results=db.similarity_search(query)
print(retireved_results[0].page_content)
```

```
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
3
```

In [12]:

```
## FAISS Vector Database
from langchain_community.vectorstores import FAISS
db1 = FAISS.from_documents(documents[:15], OpenAIEmbeddings())
```

In [14]:

```
## Vector Database
query = "An attention function can be described as mapping query"
retireved_results=db1.similarity_search(query)
print(retireved_results[0].page_content)
```

```
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
3
```

In [16]:

```
db
```

Out[16]:

```
<langchain_community.vectorstores.faiss.FAISS at 0x7fa808e5e940>
```

In [17]:

```
from langchain_community.llms import Ollama
##Load llama2 model
llm=Ollama(model="llama2")
llm
```

Out[17]:

```
Ollama()
```

In [18]:

```
##Design ChatPrompt Template
from langchain_core.prompts import ChatPromptTemplate
prompt = ChatPromptTemplate.from_template("""
Answer the following question based only on the provided context.
Think step by step before providing a detailed answer. I will tip you $1000 if the user finds the answer helpful.
<context>
{context}
</context>
Question: {input}""")
```

In [ ]: