and

"How much do we know at any time?Much more,or so I believe, than weknow we know:"

Fluent speakers of a language bring an enormous amount of knowledge to bear dur-ing comprehension and production.This knowledge is embodied in many forms,perhaps most obviously in the vocabulary, the rich representations we have of wordsand their meanings ad usage.This makes the vocabulary a useful lens to explorethe acquisition of knowledge from text; by both people and machines:

Estimates ot the Size f adult vocabularies vary widely both Within and acrosslanguages. For example, estimates of the vocabulary size of young adult speakers ofAmerican English range from 30,000 to 100,000 depending on the resources usedto make the estimate and the definition of what it means to knowaword.Whatisagreed upon is that the vast majority of words that mature speakers use in theirday-to-day interactions are acquired early in life through spoken interactions withcare givers and peers, usually well before the start of formal schooling: This activevocabulary is extremely limited compared to the size of the adult vocabulary (usuallyon the order of 2000 words for young speakers) and is quite stable, with very fewadditional words learned via casual conversation beyond this early stage. Obviously,this leaves a very large number of words to beacquired by other means.

Asimple consequence of these facts is that children have to learn about 7to 10wordsaday, every singleto arrive at observedvocabulary levels by the timeare 20 years of age.And indeed empirical estimates of vocabulary growth inlate elementary through high school are consistent with this rate. How do childrenachieve this rate of vocabulary growth? Most of this growth is not happening throughdirect vocabulary instruction in school, which is not deployed at the rate that wouldbe required to result in sufficient vocabulary growth:day;they

The mostlikely explanation is that the bulk of this knowledge acquisitionpens as a by-product ofreading, aS part of the rich processing and reasoning that weperform whenwe read:Research into the average amount of time children spendreading, and the lexical diversity of the textsTead, indicate that it is possibleto achieve the desired rate.But the mechanism behind this rate of learning mustbe remarkable indeed, since at some points during learning the rate of vocabularygrowth exceeds the rate at which new words are appearing to the learner!hap-|they

of these facts have motivated approaches to word learning based on thedistributional hypothesis, introduced in Chapter 6.This is the idea that somethingabout what we're loosely calling word meanings can be learned even without anygrounding in the real world, solely based 0n the content of the textswe encounterover our lives.This knowledge is based on the complex association of words withthe wordsco-occur with (and with the words that those words occur with).Manythey

The crucial insight of the distributional hypothesis is that the knowledge that weacquire through this process can be brought to bear long after its initial acquisition.

Of course, adding grounding from visionor from real-world interactioncanhelpbuild even more powerful models, but even text alone is remarkably useful.

In this chapter we formalize this idea of pretraininglearning knowledge aboutlanguage and the world from vast amounts of text--and call the resulting pretrainedlanguage models large language models Large language models exhibit remark-able performance on all sorts of natural language tasks because of the knowledgethey learn in pretraining, andwill play a role throughout the rest f this bookThey have been especially transformative for tasks where we need to produce text;like summarization, machine translation; question answering, Or chatbotsthey

transformer

The standard architecture for building large language models is the transformer.We thus begin this chapter by introducing this architecture in detail. The transformermakes use of a novel mechanism called self-attention, which developed out of theidea of attention thatwas introduced for RNNs in Chapter 9.Self-attentioncanbe thought of away to build contextual representations of aword smeaning thatintegrate information from surrounding words, helping the model learn how wordsrelate to each other over large spans of text.

We'Il then see how to apply the transformer to language modeling, in a setting of-ten called causal O autoregressive language models, in which we iteratively predictwords left-to-right from earlier wordsThese language models, like the feedforwardand RNN language models we have already seen, are thus self-trained: given a largecOrpus of text; we iteratively teach the model to guess the next word in the text fromthewords.In addition totraining;we Il introduce algorithms for generatingtexts, including important methods like greedy decoding; beam search, and sam-pling:And we'Il talk about the components of popular large language models likethe GPT family:prior

Finally, we'11see the great power of language models:almost any NLP taskcan be modeledas wordprediction, ifwe think about it in the right way:We 11work throughanexample of using large language models to solve one NLP taskof summarization (generating a short text that summarizes some larger document).The use of a large language model to generate text is one of the areas in which theimpact of the last decade of neural algorithms for NLP has been the largest: Indeed,textgeneration,withgeneration and code generation, constitute a newarea of AI that is often called generative AIalongimage

We'Il save three more areas of large language models for the next three chapters;Chapter 11 will introduce the bidirectional transformer encoder and the method ofmasked language modeling, used for the popular BERT family of models. Chapter 12 will introduce the most powerful way to interact with large language modelszprompting them to perform other NLP tasks by simply giving directions O instruc-tions in natural language to a transformer that is pretrained on language modeling:And Chapter 13 will introduce the use of the encoder-decoder architecture for transformers in the context of machine translation.

The Transformer:ASelf-Attention Network

In this section we introduce the architecture of the transformer, the algorithm thatunderlies most modern NLP systems When used for causal language modeling, theinput to a transformer is a sequence of words, and the output is a prediction for whatword comes next;as wellasasequence of contextual embedding that representsthe contextual meaning of each of the input words.Like the LSTMs of Chapter 9

self-attention

10.1.1Transformers: the intuition

transformers are a neural architecture that can handle distant information. But unlikeLSTMs, transformers are not based on recurrent connections (which can be hard toparallelize) , which means that transformers can be more efficient to implement atscale.

Transformers are made up of stacks of transformer blocks, each of which is amultilayer network that maps sequences of input vectors (X1Xn) to sequences ofoutput vectors (21,-Zn) of the same length:These blocks are made by combin-ing simple linear layers, feedforward networks, and self-attention layers, theinnovation of transformers\_Self-attention allowsanetwork to directly extract anduse information from arbitrarily large contexts.We'Il start by describing how self-attention works and then return to how it fits into larger transformer blocks Finally,we'Il describe how to use the transformer block together with some input and outputmechanisms as alanguage model, to predict upcoming words fromwords inthe context;keyprior

The intuition of a transformer is that across a series of layers, we build up richer andricher contextualized representations of the meanings of input words O tokens (wewill refer to the input as a sequence of words for convenience, although technicallythe input is first tokenized by an algorithm like BPE, so it is a series of tokens ratherthan words).At eachof a transformer; to compute the representation of aword1we combine information from the representation ofi at theprevious layerwith information from the representations of the neighboring words. Theis toproduce a contextualized representation for each word at each position. We can thinkof these representations aS & contextualized version of the static vectors we saw inChapter 6, which each represented the meaning ofa word type. By contrast, our goalin transformers is toproduce a contextualized version, something that representswhat this word means in the particular context in which it occurslayergoal

We thus need a mechanism that tells us how to weigh and combine the represen-tations of the different words from the context at thelevel in Order to computeourrepresentation at this layer: This mechanism must be able to look broadly in thecontext, since words have rich linguistic relationships with words that can be manysentences away: Even within the sentence, words have important linguistic relation-ships with contextual words.Consider these examples, each exhibiting linguisticrelationships that we'Il discuss in more depth in later chapters:prior

In (10.1), the phrase The keys is the subject of the sentence, and in English andmany languages, must agree in grammatical number with the verb are; in this caseboth are plural.In Englishwe can't use asingular verb like is withaplural sub-ject likewe'Il discuss agreement more in Chapter 17.In (10.2) , the pronounit corefers to the chicken; it's the chicken that wants toto the other side.We 11discuss coreference more in Chapter 26. In (10.3), the way we know that bank refersto the side of a pond or river and not a financial institution is from the context, in-cluding words likeand water.We Il discuss word senses more inChapter 23.These helpful contextual words can befar way in the sentence Or paragraph;keys;getpondquite

so we need a mechanism that can look broadly in the context to help compute reresentations for words.

Self-attention is just such a mechanism: it allows us to look broadly in the con-text and tells us how to integrate the representation from words in that context fromlayer k1 to build the representation for words in layer k.

Figure 10.1The self-attention weight distribution & that is part of the computation of therepresentation for the word it at layer 6.In computing the representation for it,we attenddifferently to the various words at layer 5, with darker shades indicating higher self-attentionvalues. Note that the transformer is attending highly to animal, a sensible result; since in thisexample it corefers with the animal, and s0 we' d like the representation for it to draw on therepresentation for animal. Figure simplified from (Uszkoreit; 2017,

Fig. 10.1 Shows an schematic example Simplned trom a real transtormerUszko-reit; 2017,Here we want to compute a contextual representation for the word it, atlayer 6 of the transformer; and we'd like that representation to draw on the represen-tations of all the prior words, from layer 5.The figure uses color to represent theattention distribution over the contextual words: the word animal hasahigh atten -tion weight, meaning that as we are computing the representation for it, we will drawmost heavily o the representation for animal. This will be useful for the model tobuild arepresentation that has the correct meaning for it, which indeed is corefer-ent here with the word animal. (We say that a pronoun like it is coreferent withanoun like animal if they both refer to the same thing; we'Il return to coreference inChapter 26.)

L2Causal orbackward-looking self-attention

The concept of context can be used intwo ways in self-attention:In causal,Orbackward looking self-attention, the context is any of thewords.In generalbidirectional self-attention, the context can include future words.In this chapterwe focus on causal, backward looking self-attention;we Il introduce bidirectionalself-attention in Chapter 11.prior

10.2 thus illustrates the flow of information in a single causal, or backwardlooking, self-attention layer:As with the overall transformer;aself-attention layermaps input sequences (x1, -Xn\_to output sequences of the same length (a1,\_,an,When processing each item in the input, the model has access to all of the inputsup to andincluding the one under consideration, but no access to information aboutinputs beyond the current one. In addition, the computation performed for each itemis independent of all the other computations The first point ensures that we can usethis approach to create language models and use them for autoregressive generation,and the secondmeans thatwe can easilyparallelize both forward inferenceand training of such models.Fig:point

10.1.3Self-attention more formally

Figure 10.Information flow in a causal (Or masked) self-attention model.In processingach element of the sequence, the model attends to all the inputs up to, and including; theurrent one.Unlike RNNs, the computations at each time step are independent of all thether steps and therefore can be performed in parallel:

We've given the intuition of self-attention (as a way to compute representations of aword at a givenlayer by integrating information from words at the previous layer)and we ve defined context as all thewords in the input Let' snow introducethe self-attention computation itself:prior

The core intuition of attention is the idea of comparing an item of interest to &collection of other items in a way that reveals their relevance in the current context:In the case of self-attention for language, the set of comparisons are to other words(or tokens) within a given sequence The result of these comparisons is then used tocompute an output sequence for the current input sequence. For example, returningto Fig: 10.2\_the computation of a3 is based ona set ofcomparisons between theinput X3 and its preceding elements X1 and X2, and to X3 itself:

How shallwe comparewords to otherwords?Since our representations forwords are vectors, we Il make use of our old friend the dotproduct that we usedfor computingsimilarity in Chapter 6, and also played a role in attention inChapter 9.Let'$ refer to the result of this comparison between words1and j as ascore (we Il be updating this equation to add attention to the computation of thisscore):word

(10.4)

The result of a dotproduct is a scalar value ranging from~Cto 0, thelargerthe value the more similar the vectors that arecompared. Continuing with ourexample, the first step in computing Y3 would be to compute three scores:X3X1,X3X2 and X3X3. Then to make effective use of these scores, we Il normalize themwithasoftmax to createavector of weights;that indicates theproportionalrelevance of each input to the input element i that is the current focus of attention.beingOij,

Of course, the softmax weight will likely be highest for the current focus elementi,since vecxi is very Similar to itself, resulting inahigh dot product:But othercontext words may also be similar to i, and the softmax will also assign some weightto those words.

Given the proportional scores in &, we generate an output value ai by summing