理解formula sheet上的所有公式

* Dot product 3 和 4的特点
  + W because s and h might not be in the same dimension
  + UV because of fast calculation
* Tf
* 
* IDF
* A black background with white text

  AI-generated content may be incorrect.
* Layer normalization
* A screenshot of a computer

  AI-generated content may be incorrect.

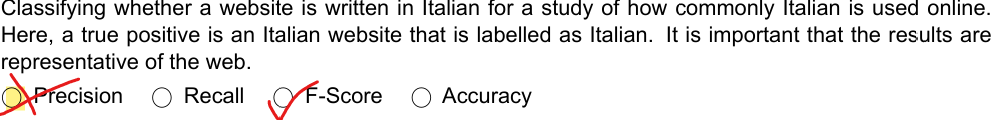
Fine-tuning

让模型更适应某一个特定的下游任务，比如情感分析、问答、文本分类等。



1. Vector, Sparse Vector, Map
   1. Space: vector > map > sparse vector
   2. Retrieval time: MAP >= Vector > sparse
   3. Comparing similarity: Map = sparse vector > vector
2. comparing vectors; 2 methods
   1. dot product: longer vector gets bigger score
   2. cos similarity: only consider about direction, so length does not matter **as long as 2 vector have the same length**
      1. **Not useful When vectors have the same direction but different magnitudes**
3. Word2vec algorithm; 2 methods
   1. CBOW: Using context to predict central word
      1. SoftMax (Mean of input word vector \* weight)
         1. SoftMax may have overflow mistake, 可以将输入向量中的每个值减去最大值
         2. softmax 会保留每个值的概率信息，不做“选择”，**只是提供所有选项的概率**
      2. **Continuous** 的原因是因为向量表示不再是整数，而是连续的浮点数
      3. Small context (+/- 2 words), more syntactically similar words
      4. Larger context (+/- 5 words), more semantically similar words
   2. Skip-gram: Using central word to predict context
4. TF-IDF
   1. TF: Term frequency, can use log to minimize the affect of very frequent word
   2. IDF: frequent word will have lower score
   3. TF-IDF: we want to find those really important word in each document, those words possibly are not frequent in other documents, that’s why we need to use TF-IDF
5. How to learn the relationship between words
   1. Word2Vec
   2. Glove: adjust word vector according to those words which usually appear with it at the same time
   3. Reducing the sparsity of vector, transfer sparse vector to dense vector, allow easy comparison
6. 5 keys components of NLP (Cheat sheet)
   1. Data: get the data, ~~like transfer token to vector~~, like tokenization, not every data can be easily split
   2. Model: **give score for (input, output) pairs**
   3. Inference method: find the highest score
   4. Metric: based on the prediction and real output to find the difference and potential way to improve the model
   5. Learning: model learning from metric’s suggestion
7. Evaluation — confusion matrix (Cheat sheet)
   1. FP: predict as positive, but reality is negative; **precision is important: you want to get less prediction error**
   2. FN: predict as negative, but reality is positive; **recall is important: you do not want to miss any positive class**
   3. Micro: regardless of class distribution, P and R are always equal when there is no null class
   4. Macro: minor class also have impact
8. ROC, AUC, PRC (Cheat sheet)
   1. ROC: Top left corner is better, high TRP, low FPR
   2. AUC: distinguish between positive and negative samples
   3. PRC: Top right corner is better; it mainly consider about the **positive** cases, so sometimes more accurate than ROC, such as unbalance data (many negative case)
9. MLP
   1. Non-linear by introducing activation function
   2. More layer means more complex model, may capture more info
10. Activation Function (Cheat sheet)
    1. Tan: [-1:1]; [0:1]
    2. Sigmoid: [0:1]; [0:0.25]
    3. Relu: [0:inf]; 分段函数，0之后的gradient都是0，但是总体而言，减少了gradient vanishing的概率
11. Loss Function Description
    1. Define the difference between prediction and real answer
    2. Towards negative means to optimize the model
12. Gradient
    1. Stochastic: read 1 data each time, avoid local min, not always converge, slower, not stable
13. Overfitting Solution (Cheat sheet)
    1. L1
    2. L2
    3. Dropout
    4. Early Stopping
14. Handle variable length input; 4 methods and what kind of question may lead
    1. Mean: may loss order information
    2. Truncate: may loss some information
    3. Padding: take many memory space; longer sentence still did not solve
    4. Concatenating: may still not have same length with other input
    5. 2 common problems
       1. Loss order information
       2. Need to relearn the information of certain word vector when it in different position
15. RNN
    1. 2个learning中的常见问题
       1. Exploding gradients: some numeric value is too big – Using ~~max~~ **min** (threshold, gradient) can solve it
          1. **主要是因为权重过大或激活函数的导数大于1**，使得反向传播过程中梯度不断放大，导致梯度值变得非常大。
       2. Gradient vanishing: through many layers, the gradient may become very small, one good example is the equation of MLP – Using LSTM is an efficient way
          1. 主要是因为non-linearly
    2. 双向RNN
       1. No loop
    3. 在一次RNN Model里面预测多个结果
       1. Use transducer to output many results, such as using it to output tag for each word
16. POS
    1. Morphological: similar ~~meaning~~ **affix**
    2. Syntactic: ~~was, is, be~~ **elephant, rabbit; can be replaced with another word and remain grammatical**
17. NER
    1. 对NER进行analyze: we use micro as consider the importance about the proportion of each class; many TN
18. Exhaustive
    1. k^n time complexity, always find global optimum, cost a lot
19. Greedy (see cheat sheet)- random sampling
    1. Top-1
    2. Random**要用exponential**
    3. Top-k**要用exponential**
    4. Top-p: **要用exponential**
    5. Contrastive
20. Beam (see cheat sheet)
21. Graph search: **A\* search**
    1. Estimate final score and use that to guide generation.
22. Viterbi(see cheat sheet)
    1. Can be used in many model as long as it has previous input (transition model) and current input (emission model)
    2. 1 previous label: |words|\*|labels|^2
    3. 2 previous labels: |words|\*|labels|^3
23. Graph Parsing — CKY
    1. Require CNF
    2. 为什么在有LLM的今天，我们还需要Graph Parsing
       1. That is just what we need, not costed
       2. LLM not good at certain task
24. Reference
    1. Identify entity
25. Coreference
    1. We need to find related entity inside the context, such he refer to Kumber Professor
    2. Antecedent: the entity appear before
26. Entity Linking / Wikification
    1. Link entity with external resources
27. 3 ways to reduce the reference space
    1. Identify all possible candidate for entity
    2. Find pair of entity
    3. ~~Link the same entity to the same external resources~~; **find the transitive closure**
28. How to get embedding tables; 3 ways
    1. Word2vec
    2. Glove
    3. Fast text: ~~according to n-grams to form the vector~~ **不直接为完整单词学习向量，而是为每个 n-gram 学习一个向量**; benefit is even if there are some words we did not know, we still can use its sub word to guess the word vector
29. What if my data is not the same as the data used for training? 3 ways
    1. Retrain the model only using the new data
    2. Retrain the last few layers of model, more efficient and works well
    3. Through back propagation to use new data train the model
30. **Understand word senses**;
    1. ~~Cos; not suit for certain cases, but can be use in many situations~~ **WordNet: A database of labeled relationships between words.**
    2. ~~Train a model; cannot detect untrained words~~; **Train a model with multiple word vectors, one per sense, Challenge is the data**
    3. Contextual Representations — **ELMo (Bidirectional LM): Vector of a word is dependent on the rest of the sequence.** From begin to end, end to begin
31. Decoder
    1. How to stop decoding? Fix length or using <stop> token
    2. How to train? Teacher Forcing: fast training speed, use correct output to replace prediction as next input. **Exposure Bias might be a serious problem: 一旦某步预测错了，后续输入就不再是“正确的”，错误会逐步累积，模型表现可能迅速下降**
32. We can also use Greedy Inference with a sequence-to-sequence model (see cheat sheet)
    1. How to stop beam searching encoder-decoder? Keep going until we have N outputs
    2. How to score different outputs with different lengths? Longer output will have higher length, so we need to normalize the score
33. Issues of Encoder-Decoder (RNN)
    1. Bottleneck: since we only output one vector to decoder, it may not capture long distance information
    2. Cannot parallel: RNN ask each token be processes one by one which not efficient enough.
34. Tokenization (see cheat sheet)- **Translation quality**
    1. Human evaluate: fluency, adequacy
    2. compare character ngrams, name is: chrF
    3. compare word ngrams, name is BLEU
       1. no 4-gram match will lead to 0, so works for multiple sentences
       2. Apply a penalty if the output is short
    4. word ngrams对于评分系统到底是好是坏？
       1. Word n-gram consider more about the sentence order meaning, such as “I love you” and “you love I” can recognized different
       2. Character n-gram can recognize different words with similar meaning, such as “like” and “liking”, **high tolerance to typo**
       3. So, we cannot say one is better than another
35. What if a sentence cannot be split on whitespace to get token? (see cheat sheet)
    1. start with small units, then combine units
       1. BPE
       2. WordPiece
    2. start with big and small units, then delete units
       1. Unigram
    3. 两者的区别，以及为什么?
       1. Second provide more meaningful sub word units as it keep removing low frequent words and less useful units, while BPE/WordPiece just keeps merging frequent ones (even if they’re less meaningful).
36. Attention
    1. **Allows the model to retain memory of long-distance dependencies**
    2. Dot product公式中s和h分别表示什么
       1. S means the current input; **query**
       2. H means previous hidden layer output; **key**
37. Self-Attention
    1. QKV的意思
       1. Q: the vector who want to compare with other vector
       2. K: the vector who was compared
       3. V: ~~the weight for K~~ **vector with actual content/info to retrieve**
    2. 如何像“RNN使用hidden layer”一样传递前面的信息
       1. **Every word compare with other words, and have its own contextual vector**
       2. **We can stack layers to allow it capture deeper information**
    3. 如何像RNN一样添加nonlinear calculation
       1. Non-linear Feedforward layer
    4. **如何像RNN一样考虑到position呢?**
       1. 使用sin和cos
          1. Need not to train
          2. Limit capability
       2. Learnable Positional Embedding
          1. can’t generalize to sequences longer than those seen in training!
       3. Rotary Positional Embeddings — RoPE
          1. similarity remains the same for two words the same distance apart!
    5. 如何像RNN一样进行backpropagation训练
       1. Casual mask, that is earlier token cannot see later token **enabling autoregressive training and RNN-like backpropagation.**
38. Transformer Encoder
    1. Multi-head: each head responsible for certain task, make our prediction can capture more information
    2. Scaled dot product: reduce the effect of vector dimension
    3. Residuals: make the gradient smoother, reduce vanishing gradient problem, **Improve training speed**
    4. Layer Normalization: fast training, stabilize training
39. Transformer Decoder
    1. Cross attention; QKV分别在哪
       1. Q: decoder
       2. KV: encoder
40. Benefit of transformer
    1. Small performance improve; huge efficiency improve which help scaling
41. Transformer的种类对比用到bench mark叫做什么
    1. Glue: BERT > GPT
42. Attention is quadratic, is that a problem?
    1. No memory bandwidth is the problem
    2. What is flash attention? Reduce memory bandwidth need
43. N-Gram Language Model
    1. **determine probabilities by dividing one count by another**
    2. Why is it a **strong assumption**? What problem will arise
       1. ~~Depending on the previous words; ?~~ **Surprising matching scores: a b c … x b z, if you switch bc and bz, you will get same score**
       2. **Cannot capture long-distance dependencies**
    3. How to make a less strong assumption
       1. Increase N
    4. 那么具体是如何计算P的呢？n=3
       1. P(C|AB)=P(ABC)/P(AB)
    5. 如何处理start和end在sequence中？
       1. Just add <start>, <end> token
          1. 这样模型才能**正确计算第一个词的概率，**因为N-Gram模型需要固定数量的前词作为上下文。
    6. 如何处理numerical issue when P is too small? log
44. Using LLMS 3 ways, adv and disadv
    1. Directly use; no further training needed; may not suit certain task **("zero-shot" or "few-shot" prompting)**
    2. Use its output as input for other models; cost a lot
    3. **Train on our own data – Fine tuning**
       1. **Feedforward**
       2. **adapter**
45. Evaluate LLM 2 ways, adv and disadv
    1. MRR: close to 1 means good; the earlier the prediction appear, the higher the mark is
    2. Perplexity: close to 1 is good, can be infinite, indicate how easy the model understand the sentence
46. Which tasks can be done by each model directly? (see cheat sheet)
47. Increasing Efficiency of LLM 4 Ways (see cheat sheet)
48. In-Context Learning – ICL (see cheat sheet)
    1. Perplexity
       1. Low perplexity prompts are better
    2. Order of examples matters
       1. Unsorted data is preferred
    3. number of examples
       1. more number of examples usually lead to better performance
    4. Label of examples
       1. Does not matter; Structure of data is more important
    5. Chain of Thought
       1. Bad explanation usually leads to the wrong answer
    6. What is verbalizer
       1. Certain template we use for ICL
49. Retrieval Augmentation 3 ways (see cheat sheet)
50. Data Sources
    1. Overfitting into one dataset: we need to make sure label what dataset our model is good at
    2. Shortcuts/validity: our model really understand the question
       1. Whether a model can do well without the question and/or context
       2. Collect data from real world
       3. ~~Modify the training data’s label slightly to see whether model can find it~~ **Wrong answers that have many matching words with the input question or true answer**
       4. ~~Modify the question to make it have many similar words to unrelated answer to see whether model can find it~~ **Make minimal edits to the true answer that make it wrong**
    3. Statistical power: does the data include enough difficult samples
51. Annotation
    1. Edit auto-generated labels
       1. High consistency, but may fail to misannotated a series of token
    2. Annotate from scratch
       1. Pilot Annotation: annotation people annotate a sample first and adjust the instruction based on their reply
       2. Cohen’s Kappa: ~~judge the correct on accident~~ **correcting for chance agreement**~~.~~, 1 is better
52. Instruction Tuning
    1. FLAN-T5: higher generalization
    2. LIMA: for alpaca, less quantity with high quality data is better than high quantity with low quality data
    3. Synthetic data
       1. Cheap
       2. Easy to get
       3. Cover corner cases
    4. Effect of training examples
       1. Increase training examples does not improve generation quality
    5. Remaining Issue
       1. Token-level does not work: some sentence may get similar score, but their meaning is different
       2. No absolute answer to certain questions
53. Preference Optimization
    1. Backprop the model, assign higher score to preferred answer
    2. RLHF: build a reward model
    3. DPO: directly train from preference data, might not reliable
    4. Alignment brittle: 我们要在transformer的just before output step解决
54. automatically find good prompts (Optimizing Prompts); 3 methods
    1. vector
    2. Reinforce learning
    3. Use as a system
55. Agent
    1. Reasoning: happened before generating prediction
       1. Self-consistency with chain of thought: ~~consider about the reason of model’s prediction~~ **voting with multiple outputs**
       2. Refletion: consider about previous part
       3. Tree of thoughts: **exploring multiple possible “thought paths**”
       4. what is planning? Temporary-prompt, permanent-database
    2. Acting: doing something beyond the generation; RAG
    3. ReAct



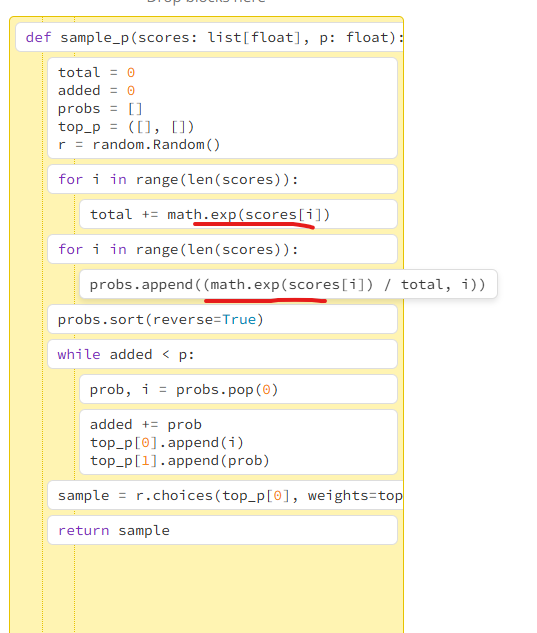


A white background with black text

AI-generated content may be incorrect.

A white background with black text

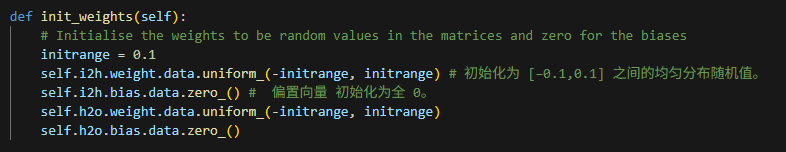
AI-generated content may be incorrect.



这里要用softmax



这是随机生成2个值

 A computer screen shot of a program code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

A computer screen shot of a program code

AI-generated content may be incorrect.

A computer screen shot of white text

AI-generated content may be incorrect.

Sum: -1

Unsqueeze: 1

Softmax: -1

A screen shot of a computer screen

AI-generated content may be incorrect.

A black background with white text

AI-generated content may be incorrect.

A black background with green and white text

AI-generated content may be incorrect.

Pinecorn: database

A white board with black writing

AI-generated content may be incorrect.

Vowpal wabbit: very fast, batch learning

A screenshot of a computer

AI-generated content may be incorrect.

A math test with numbers and a red mark

AI-generated content may be incorrect.

心里想想Macro对于每个class是怎么算的

A screenshot of a computer

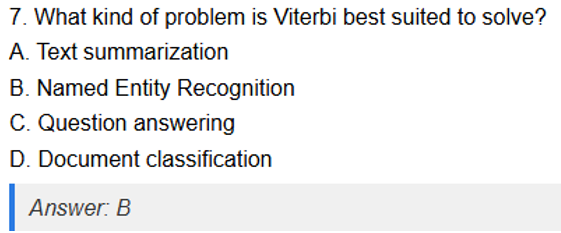
AI-generated content may be incorrect.

A white background with black text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



A screenshot of a computer screen

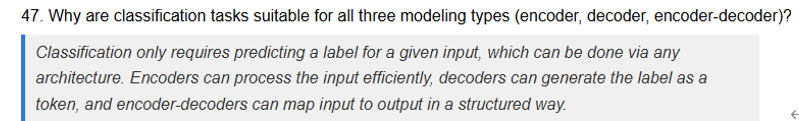
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



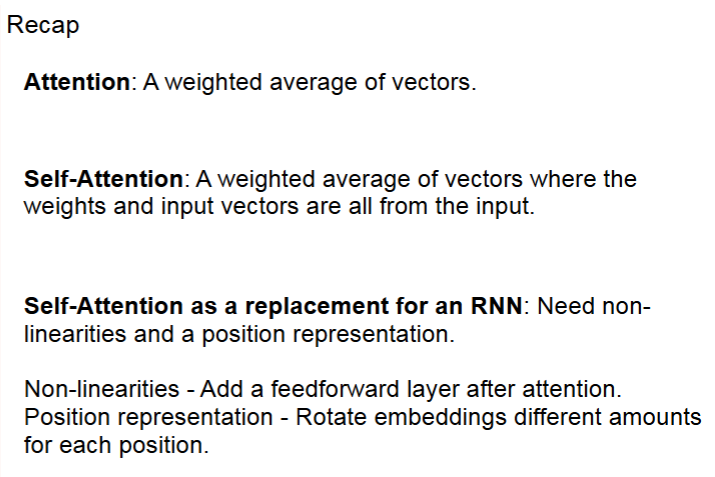
A white background with black text

AI-generated content may be incorrect.

ChrF

A close up of a text

AI-generated content may be incorrect.



A white text on a white background

AI-generated content may be incorrect.

A screenshot of a computer model

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A white text with black text

AI-generated content may be incorrect.

A screenshot of a white text

AI-generated content may be incorrect.

A white text on a white background

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A white background with black text

AI-generated content may be incorrect.

A white text with black text

AI-generated content may be incorrect.

A white background with black text

AI-generated content may be incorrect.

A close-up of a document

AI-generated content may be incorrect.

A close-up of a white background

AI-generated content may be incorrect.

A white text with black text

AI-generated content may be incorrect.

A white paper with black text

AI-generated content may be incorrect.

A white text on a white background

AI-generated content may be incorrect.

A white text with black text

AI-generated content may be incorrect.

A white background with black text

AI-generated content may be incorrect.