ملیکا محمدی فخار - ۹۹۵۲۲۰۸۶

1.

مجموع کلمات پیکره: ۹۵۲۱
محاسبه احتمال یونیگرم‌ها:

$$p(\text{ما})=\frac{1872}{9521},\ p(\text{خواندیم})=\frac{1495}{9521},\ p(\text{دیروز})=\frac{2021}{9521},\ p(\text{امروز})=\frac{1943}{9521},$$

$$p(\text{داستان})=\frac{945}{9521},\ p(\text{کتاب})=\frac{1245}{9521}$$

در جدول بایگرام‌ها فرض می‌کنیم کلمه اول در ستون سمت راست و کلمه بعدی آن در ردیف بالا قرار داشته باشد.
seq: ما امروز کتاب خواندیم.

$$p(seq) = p(\text{ما}|\text{امروز}).p(\text{امروز}|\text{کتاب}).p(\text{کتاب}|\text{خواندیم})$$

$$p(\text{ما}|\text{امروز}) = \frac{452}{1872}\ ,\ p(\text{امروز}|\text{کتاب}) = \frac{231}{1943}\ ,$$
$$p(\text{کتاب}|\text{خواندیم}) = \frac{320}{1245}$$

حاصلضرب مقادیر فوق:

0.0074

seq: ما دیروز داستان خواندیم.

$$p(seq) = p(\text{ما}|\text{دیروز}).p(\text{دیروز}|\text{داستان}).p(\text{داستان}|\text{خواندیم})$$

$$p(\text{ما}|\text{دیروز}) = \frac{411}{1872}\ ,\ p(\text{دیروز}|\text{داستان}) = \frac{68}{2021}\ ,\ p(\text{داستان}|\text{خواندیم}) = \frac{345}{945}$$

حاصلضرب مقادیر فوق:

0.0027

2.
as we now:

$$P(B \mid A) = \frac{P(A, B)}{P(A)}$$

so we can say:

$$P(A, B) = P(A)P(B \mid A)$$

extending to more than two events:

$$P(A, B, C) = P(A)P(B \mid A)P(C \mid A, B)$$

SO:

- The probability of the first word w_1 appearing is simply its probability $P(w_1)$.
- The probability of the second word w_2 appearing, given the first word w_1, is $P(w_2 \mid w_1)$.
- This extends to any word w_n in the sequence, where its probability depends on all previous words, forming the product: $P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1^2) \ldots P(w_n \mid w_1^{n-1})$
- The general form of this product for any sequence length n is: $\prod_{k=1}^{n} P(w_k \mid w_1^{k-1})$

3.

a.

از آنجا که k = 2 می‌باشد، در مرحله اول هر دو کلمه را نگه می‌داریم:

- "neural" with a score of -0.65
- "network" with a score of -0.73

b.

در مرحله دوم هر یک از نودهای مرحله قبل را گسترش می‌دهیم:

From "neural" (-0.65):

- Expand to "neural neural“: Additional score from "neural": -0.80 (from -0.65 to -1.45)

- Expand to "neural network“: Additional score from "network": -0.60 (from -0.65 to -1.25)

From "network" (-0.73):

- Expand to "network neural“: Additional score from "neural": -0.60 (from -0.73 to -1.33)

- Expand to "network network“: Additional score from "network": -0.80 (from -0.73 to -1.53)

حال با توجه به k=2 بایست دو تا از نتایج مرحله قبل که از بقیه بهترند را حفظ کنیم:

- "neural network" (-1.25)
- "network neural" (-1.33)

c.

در مرحله سوم هر یک از نودهای مرحله قبل را گسترش می‌دهیم:

From "neural network" (-1.25)

- Expand to "neural network neural”: Additional score from "neural": -0.80 (from -1.25 to -2.05)
- Expand to "neural network network”: Additional score from "network": -0.60 (from -1.25 to -1.85)

From "network neural" (-1.33)

- Expand to "network neural neural": Additional score from "neural": -0.80 (from -1.33 to -2.13)
- Expand to "network neural network": Additional score from "network": -0.60 (from -1.33 to -1.93)

حال با توجه به k=2 بایست دو تا از نتایج مرحله قبل که از بقیه بهترند را حفظ کنیم:

- "neural network network" (-1.85)
- "network neural network" (-1.93)

d.

خیر. در این مثال بهترین دنباله‌ها "neural neural neural" با امتیاز ۱.۴۶- و "network network network" با امتیاز ۱.۵۴- است که beam search نتوانسته آن‌ها را پیدا کند.

درواقع beam search لزوماً "overall most-likely sequence" را برنمی‌گرداند، بلکه بهترین را در میان تعداد محدودی از دنباله‌هایی که ارزیابی می‌کند، نشان می‌دهد. این روش با استفاده از پارامتر k بین کارایی محاسباتی و کیفیت راه حل تعادل برقرار می کند. هرچه مقدار k را بالاتر در نظر بگیریم راه حل ما دقیق‌تر اما کندتر خواهد بود و بالعکس.

e.

تجزیه و تحلیل پیچیدگی زمان اجرا

تجزیه و تحلیل قدم به قدم:

در هر مرحله زمانی t برای هر یک از k دنباله‌هایی که در حال حاضر ذخیره شده‌اند، شبکه عصبی بازگشتی (RNN) امتیازات را برای همه m کلمه احتمالی بعدی تولید می‌کند.

این به معنای آن است که RNN برای محاسبه امتیاز برای هر گسترش احتمالی هر دنباله، k*m بار فراخوانی می‌شود.

پس از تولید امتیازات، بهترین k دنباله از این k*m امکان برای انتقال به مرحله بعدی انتخاب می‌شوند.

پیچیدگی کلی:
از آن‌جا که این فرایند برای هر یک از t مراحل دنباله تکرار می‌شود، تعداد کل ارزیابی‌ها یا محاسبات RNN مورد نیاز k*m برای هر یک از t مراحل است.
بنابراین، پیچیدگی زمان اجرای کلی O(k*t*m) است.

4.

a.

Forget Gate Alone: If you only have the forget gate, the LSTM can only decide which parts of the previous cell state to keep, but it cannot control what new information is added to the cell state.

No Input Gate: The input gate controls how much of the new information (from the current input and previous hidden state) is added to the cell state. Without the input gate, the LSTM loses its ability to add new information to the cell state. This means the cell state can only decrease or stay the same over time, as information is only ever forgotten, never updated or augmented with new inputs.

No Output Gate: The output gate controls how much of the cell state is used to generate the current hidden state (output). Without the output gate, the entire cell state would always be exposed as the output, which could lead to issues of excessive influence from the cell state and a lack of control over what part of the internal state is exposed to the subsequent layers or output.

How Output Changes:

The LSTM would likely become less capable of learning complex patterns over time since it cannot accumulate new information across inputs—only forget the old.

b.

Forget Gate Functionality: The forget gate in an LSTM controls the extent to which the previous cell state (from the last timestep) is retained in the current cell state. It does this by applying a gate multiplier (between 0 and 1) to the previous cell state.

Effect of Setting to Zero: When the forget gate is set to zero, it effectively blocks any information from the previous cell state from being retained in the current cell state. This means that the LSTM forgets everything it previously knew at each timestep, preventing any information from being carried over across timesteps.

Impact on Learning and Prediction:

Loss of Long-term Dependencies: One of the primary strengths of LSTMs is their ability to remember information over long periods. Setting the forget gate to zero removes this capability, thereby crippling the LSTM's ability to make use of historical context or dependencies that extend beyond a single timestep.

c.

1. Enhanced Learning Capabilities

- Adding more LSTM layers allows the network to learn more complex representations. Each layer can learn a different level of abstraction.

- More layers generally increase the network's capacity, allowing it to model more complex relationships and dependencies in the data.

2. Improved Contextual Understanding

- Deeper LSTM networks can potentially capture longer-term dependencies better than shallower ones, because the

information has more stages to be processed and refined, each adding a level of context understanding.

- Multiple LSTM layers can process information hierarchically. Lower layers can handle short-term dependencies, while higher layers can integrate these over longer spans, thus improving the overall sequence understanding.

Conclusion

Increasing the number of LSTM layers can enhance the model's ability to learn complex patterns and improve performance on tasks requiring understanding of long-term dependencies. However, this comes with increased computational cost, training complexity, and the potential for overfitting.