Incorporating Behavioral Hypotheses for Query Generation

Ruey-Cheng Chen (SEEK Ltd)
Chia-Jung Lee (Microsoft)

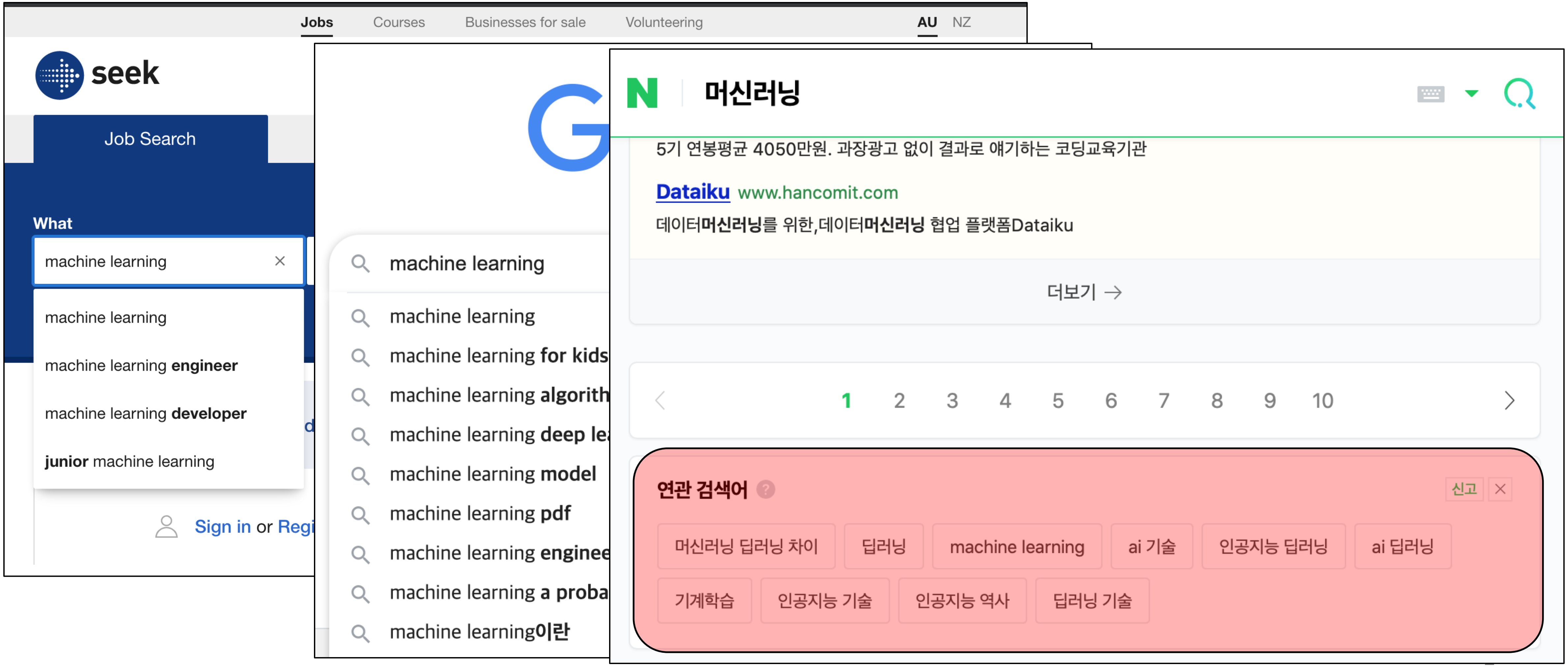
EMNLP 2020 / Short Paper

paper / talk

집현전 중급반
발표자 송일현

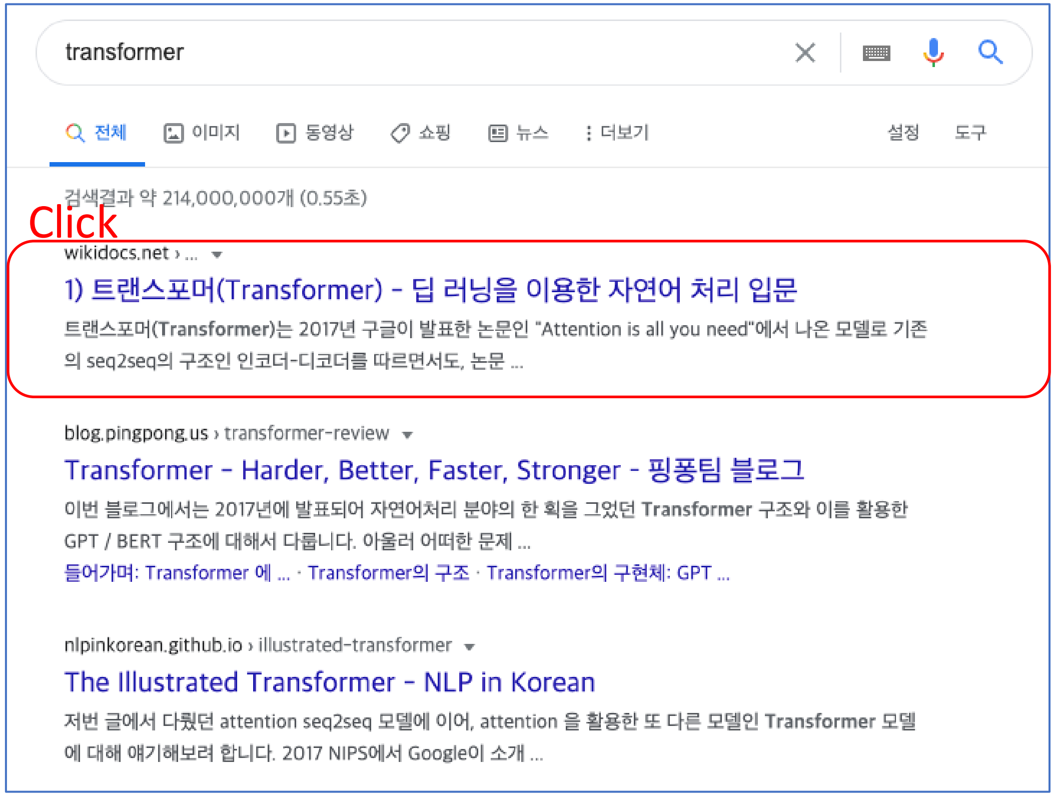
Incorporating Behavioral Hypotheses for Query Generation

≈ Query Suggestion, Related Query



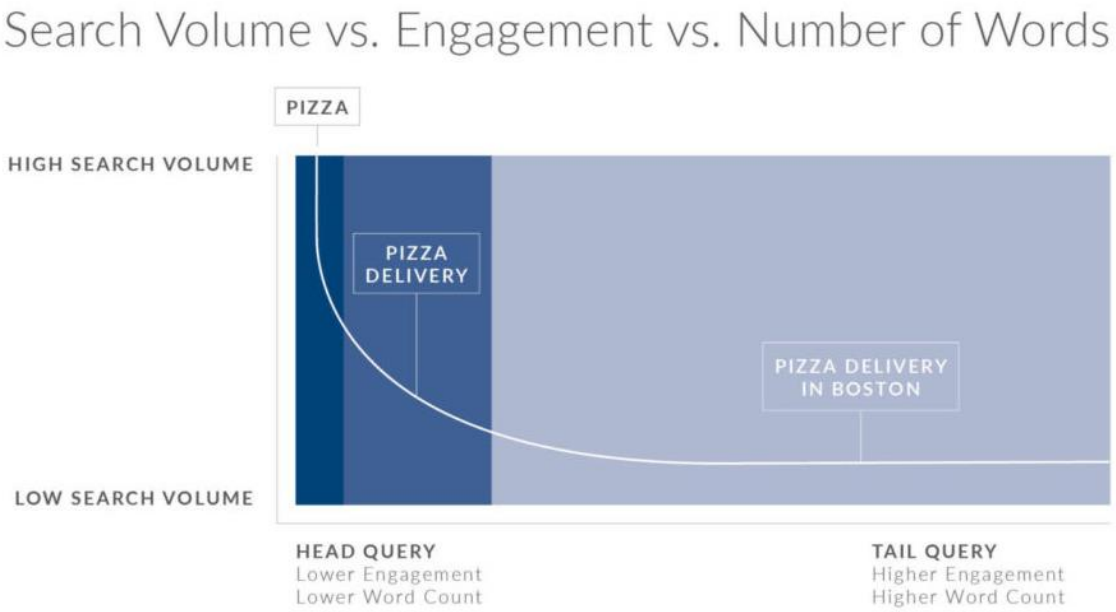
Incorporating Behavioral Hypotheses for Query Generation

- User Behavior
  - Query -> Click (Document : title)
    - Transformer
      -> "1) 트랜스포머(Transformer) – 딥러닝을 이용한 자연어처리…"
  - Query -> Query
    - Machine Learning Engineer
      -> Machine Learning Developer



Introduction

- Query Suggestion
  - Discriminative Approach
    - 사용자 로그에서 다음에 발생할 확률이 높은 질의를 선택 (Query Co-occurrence)
      - 로그에 있는 질의만 처리 가능, tail query 에 penalty

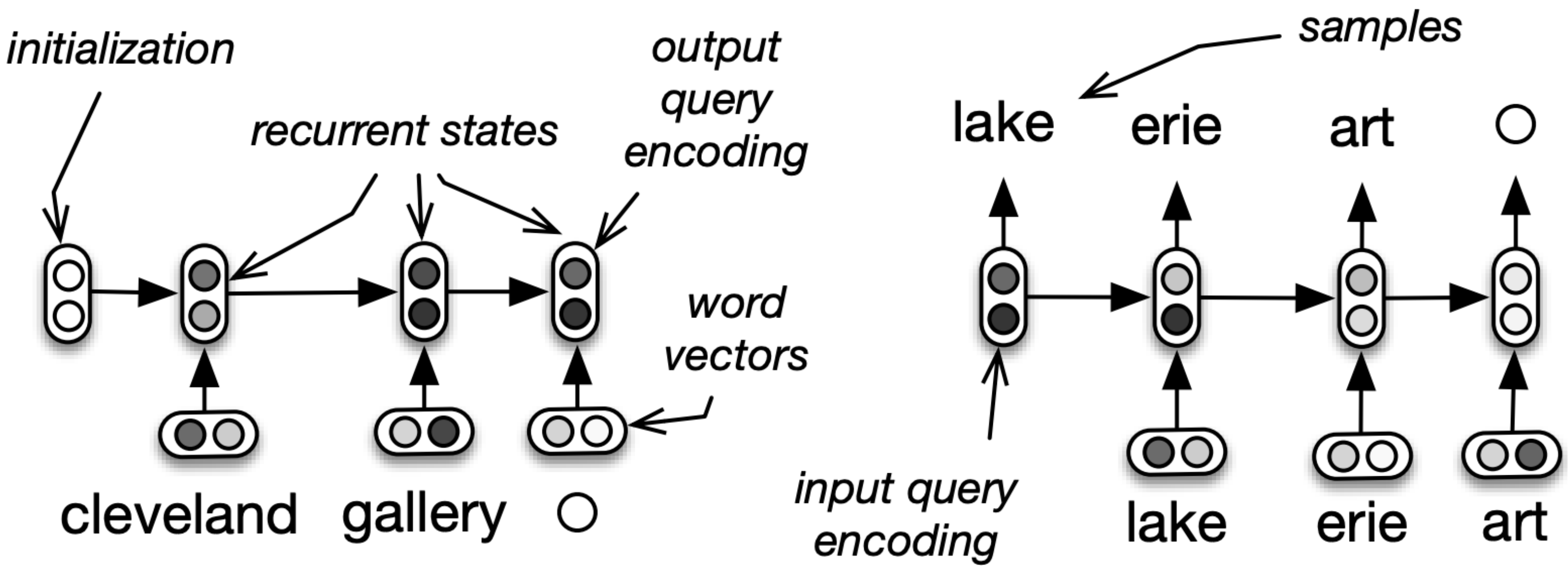


https://lucidworks.com/ai-powered-search/head-tail-analysis/

Introduction

A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion (Sordoni et al. 2015)

- Query Suggestion
  - Generative Approach
    - 질의를 생성 (ex. RNN Encoder-Decoder)

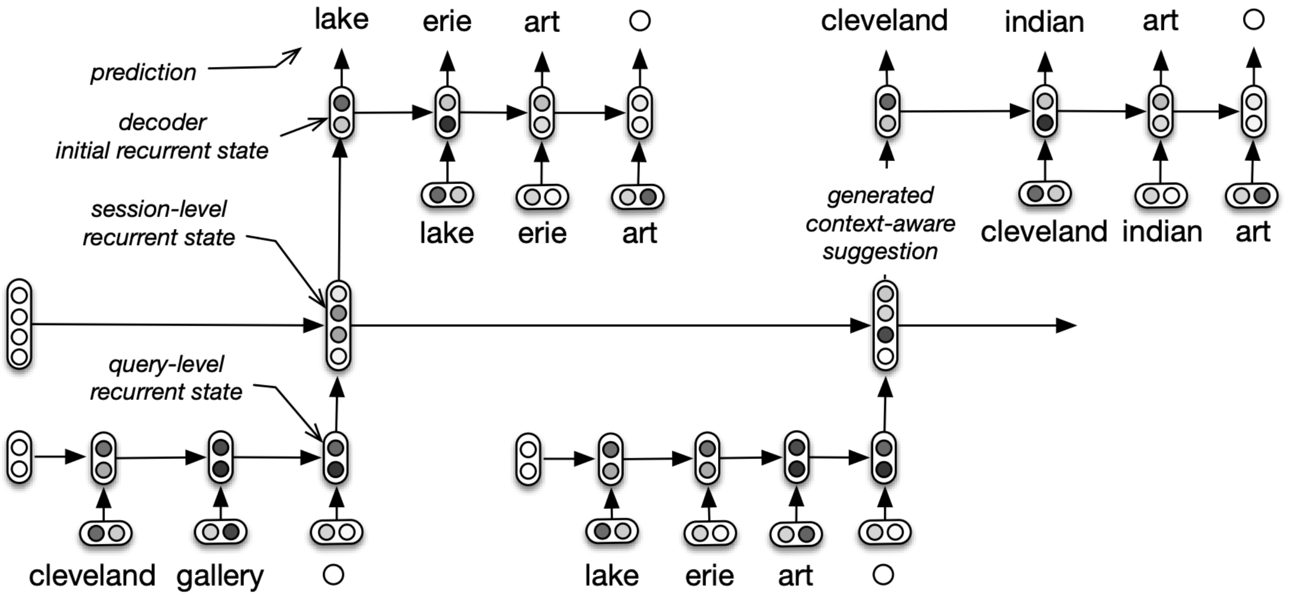


(a) (b)

cleveland gallery -> lake erie art

Introduction

- Query Suggestion
  - Generative Approach
    - HRED (Hierarchical Recurrent Encoder-Decoder)



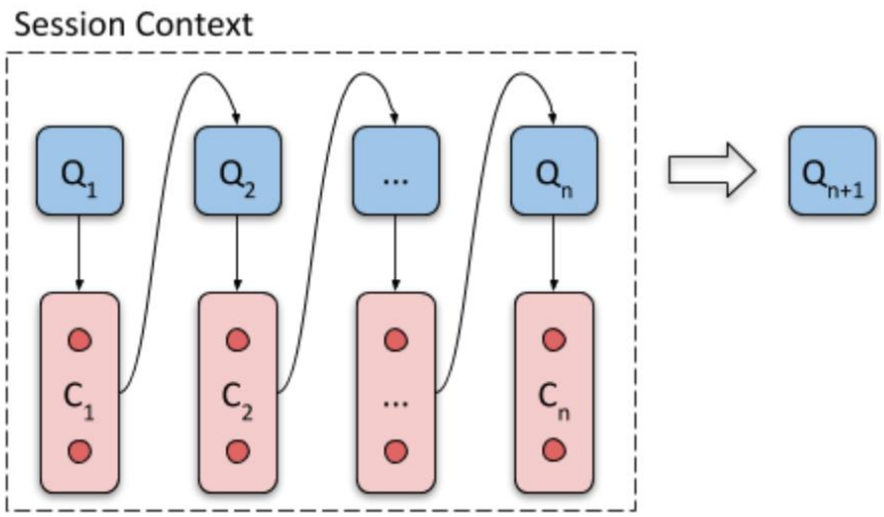
[cleveland gallery -> lake erie art] -> cleveland indian art

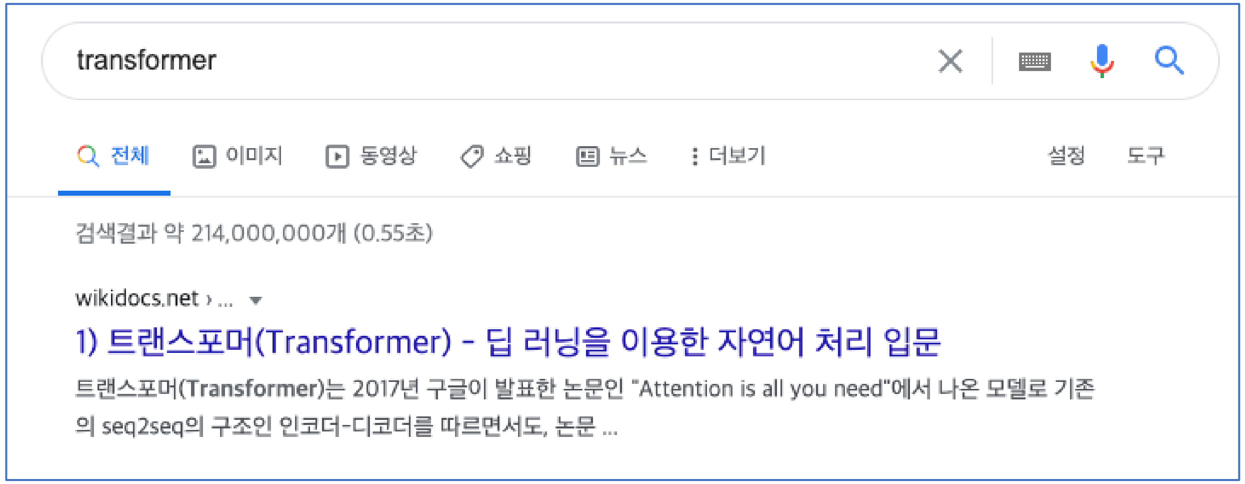
한계 : 질의-질의 간의 관계만 반영
=> User의 Implicit 한 Intent(click) 정보도 포함 해보자

Approach

- $Q = (Q_1, Q_2, \ldots, Q_n)$: 사용자가 입력한 질의 Sequence (in 세션)
- $C = (C_1, C_2, \ldots, C_n)$: C_i -> Q_i 의 결과에서 사용자의 Interaction
  ex) 사용자가 클릭한 문서의 제목
  Q_i : Transformer
  C_i : 트랜스포머(Transformer) – 딥러닝을 이용한 자연어 처리 입문

- Q, C 를 이용해서 Q_{n+1} (다음 질의) 생성





Approach : Behavioral Hypotheses

- Qn+1 생성하는데 무엇이 중요할까? (가정)

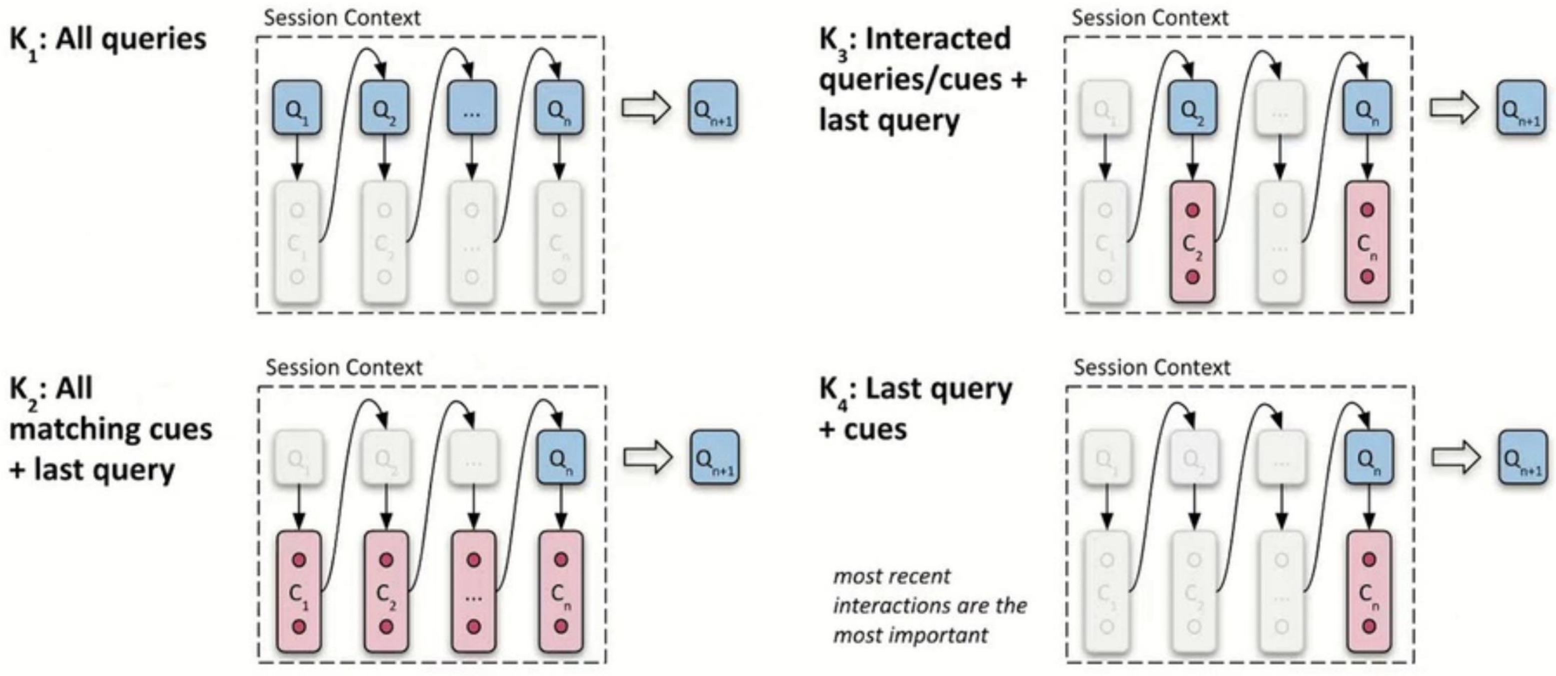
K_1 : 사용자가 입력한 질의들이 중요할 것이다.

K_2 : 사용자가 반응(Click)한 문서들과 직전 질의가 중요할 것이다.

K_3 : 사용자가 반응한 질의와 문서들이 중요할 것이다.

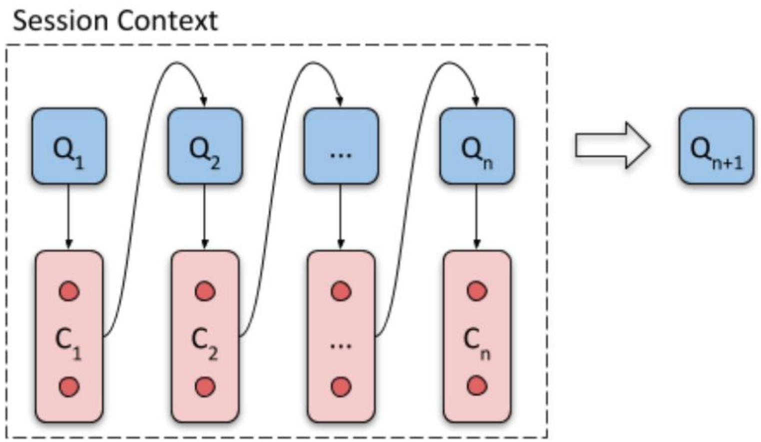
K_4 : 사용자의 직전에 입력한 질의와 반응한 문서가 중요할 것이다.

$$K_1 = (Q_1, Q_2, \ldots, Q_n)$$
$$K_2 = \bigoplus_{i<n} \{(t)_{t \in C_i}\} \oplus (Q_n)$$
$$K_3 = \bigoplus_{Q_i : C_i \neq \varnothing} \{(Q_i) \oplus (t)_{t \in C_i}\} \oplus (Q_n)$$
$$K_4 = (Q_n) \oplus (t)_{t \in C_n}$$



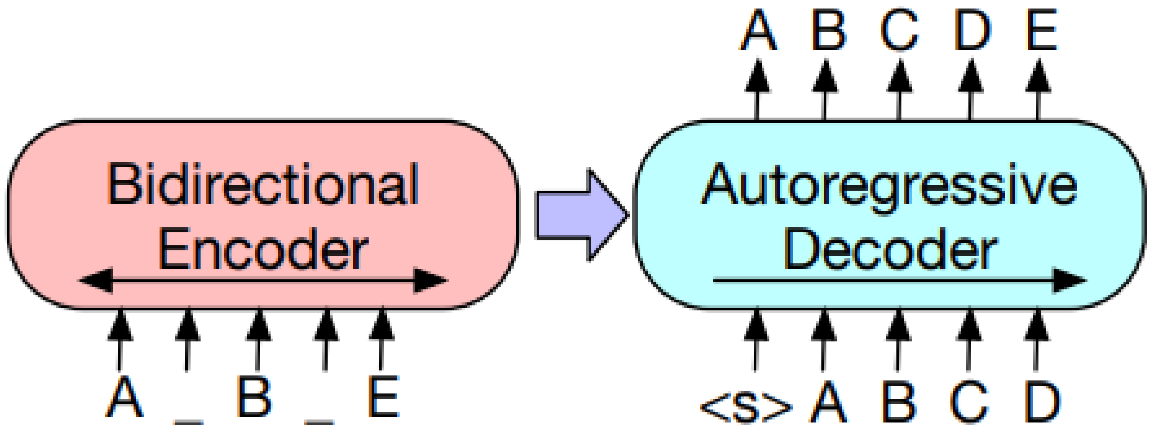
Approach : Vanilla Encoder-Decoder Transformer

- 이 가정들을 어떻게 사용할까?
  - Vanilla Encoder-Decoder Transformer
    - User Behavior 를 Concatenate 한 입력에서 Q_{n+1}을 생성 (번역 모델과 유사한 Seq2Seq)
    - $Q_1 \oplus C_1 \oplus Q_2 \oplus C_2 \oplus \ldots \oplus Q_n \oplus C_n$ -> Q_{n+1}



  - **BART** 사용

Q_{n+1} [t1, t2, ..., tn]



$Q_1 \oplus C_1 \oplus \ldots \oplus Q_n \oplus C_n$

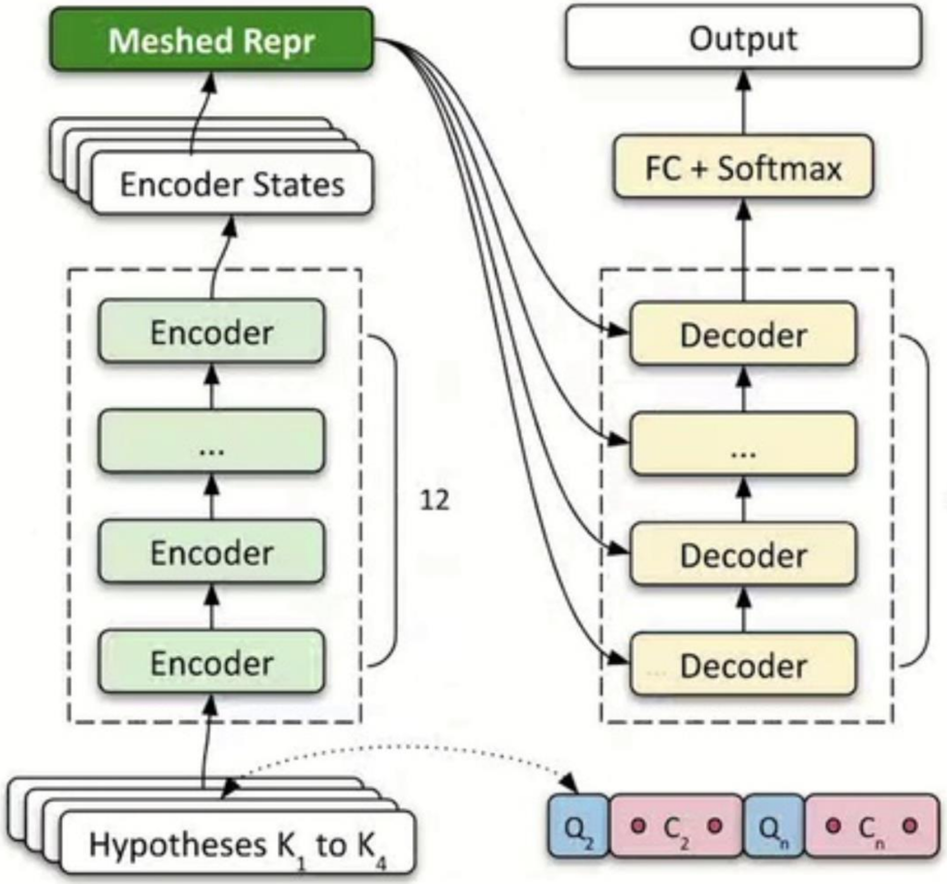
Approach : Meshed Representations

- 이 가정들을 어떻게 사용할까?

- Meshed Representations

  - K_1~K_4 의 encoding된 token 들을 Token wise Attention 를 통해 Mesh

  - Meshed Repr의 결과를 Decoder로 전달



Token wise Attention

https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

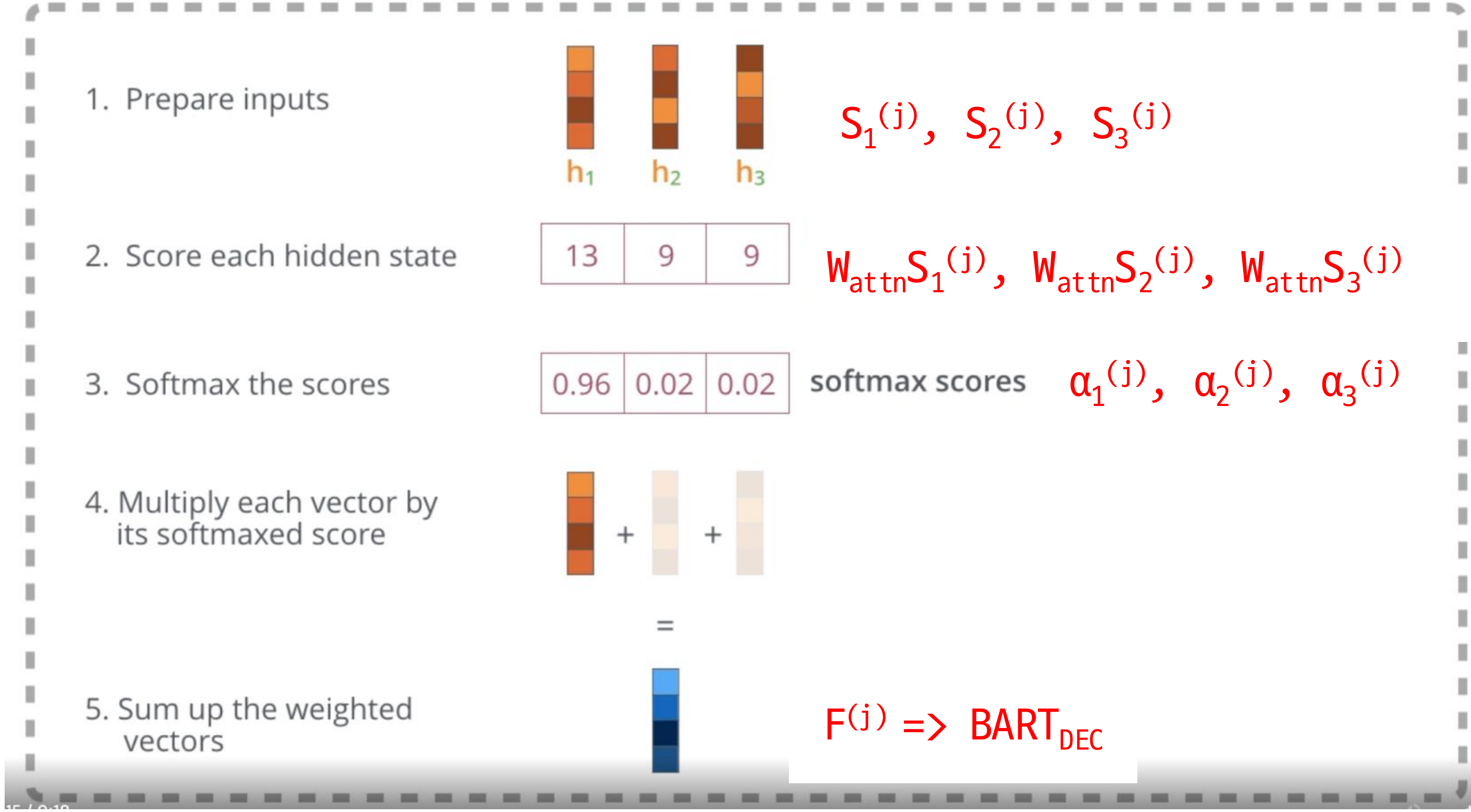
$S_1^{(j)}$ -> K_1의 j번째 token의 BART Encoding -> h_1
$S_2^{(j)}$ -> K_2의 j번째 token의 BART Encoding -> h_2
$S_3^{(j)}$ -> K_3의 j번째 token의 BART Encoding -> h_3
$S_4^{(j)}$ -> K_4의 j번째 token의 BART Encoding -> h_4 (그림생략)

$$K_1 = (Q_1, Q_2, \ldots, Q_n)$$
$$K_2 = \bigoplus_{i<n} \{(t)_{t\in C_i}\} \oplus (Q_n)$$
$$K_3 = \bigoplus_{Q_i:C_i\neq\varnothing} \{(Q_i) \oplus (t)_{t\in C_i}\} \oplus (Q_n)$$
$$K_4 = (Q_n) \oplus (t)_{t\in C_n}$$

The procedure can be described as follows. Let $[S_i^{(1)}; \ldots; S_i^{(T)}] = \mathrm{BART}_{\mathrm{enc}}(K_i)$ for all K_i, and T is the sequence length. We have:

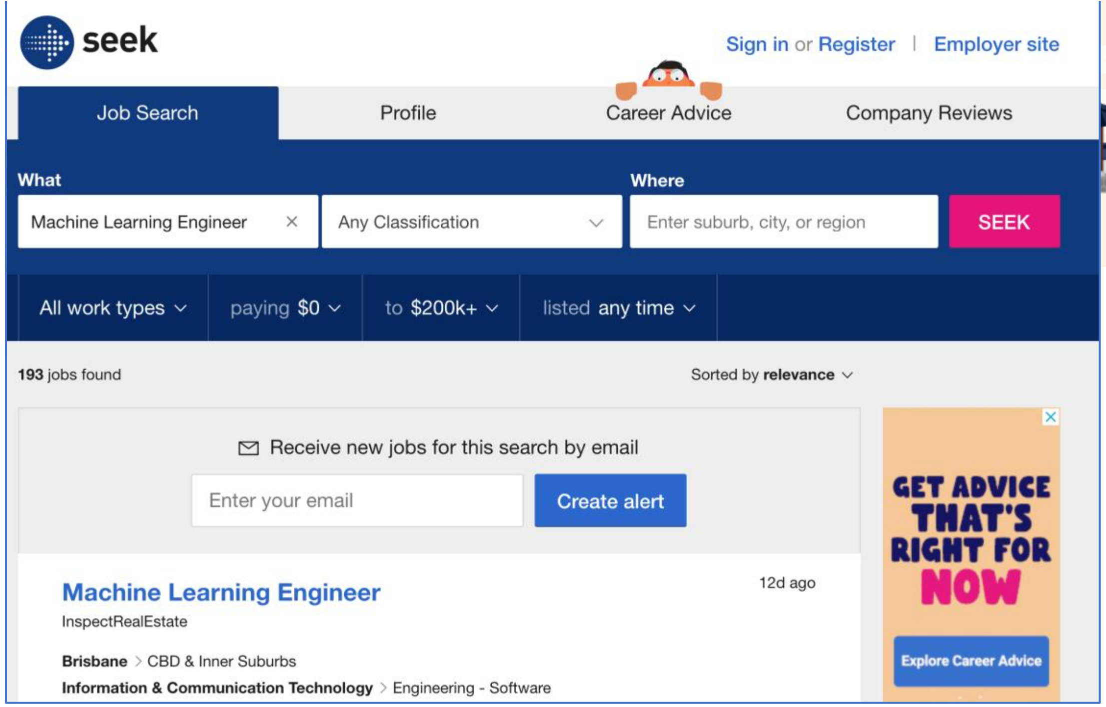
$$\alpha_i^{(j)} \propto \exp(W_{\mathrm{attn}} S_i^{(j)})$$
$$F^{(j)} = \sum_i \alpha_i^{(j)} S_i^{(j)} \qquad (2)$$
$$O = \mathrm{BART}_{\mathrm{dec}}([F^{(1)}; \ldots; F^{(T)}])$$

where W_{attn} is the attention weight matrix to be learned and O the output. On the decoder side, the



Dim($F^{(j)}$) == Dim($S_i^{(j)}$) : Pretrained Model 사용 가능!

Experiment : Data



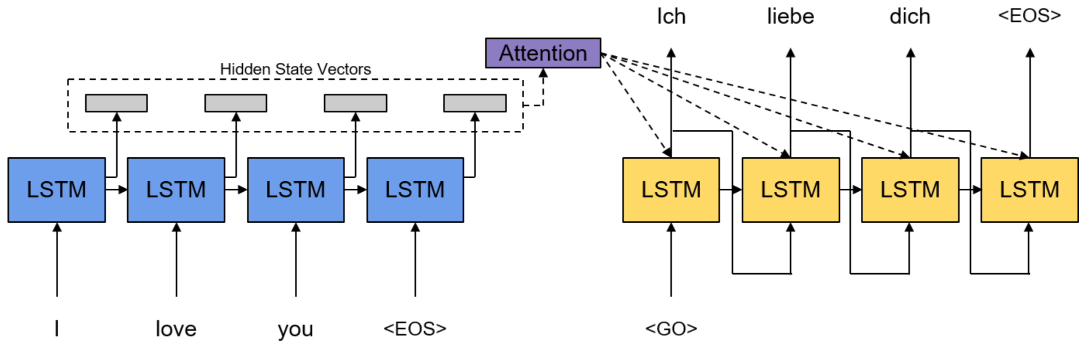
- [SEEK] : Job search engine in AU
  - Role Title, Skill, Company Name, Geo Location
  - Search Session log

- Q_i : Query
- C_i : Title of documents that were clicked on in response to Qi
- Session boundary: inactivity of 30 minutes or more between two consecutive actions.

- In each session the latest query was held out as the ground truth.
- Training Session (500K) + DEV(1K) : 2019'10 월 2주 / Test (100K) : 그 후 2주
- Data Cleansing
  - BART의 maximum sequence length 넘어가는 15% 세션 제거
  - 10회 이하 출현한 noisy query 제거
  - Singleton session (contain only one query) 제거

Experiment : 비교 모델

https://hcnoh.github.io/2018-12-11-bahdanau-attention



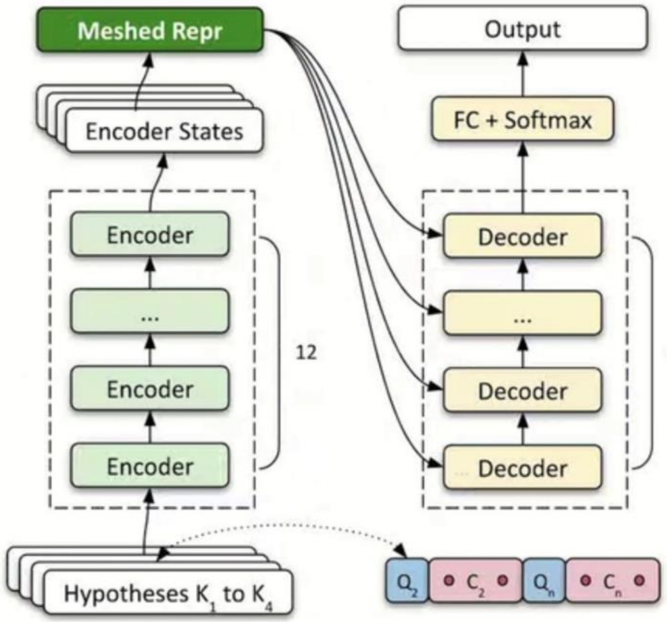
- **Seq2Seq-Attn** (Bahdanau et al., 2014)
  - GRU, 1000 hidden dim, byte-pair encoding

- **MPS (Most Popular Suggestions)**
  - Co-occurrence Frequencies of the last query in the search context and all candidate queries.

  ML Scientist -> ML Engineer (10번),
  ML Scientist -> Data Scientist (4번)

- **BART - Vanilla BART**
  - Full concatenate search context as INPUT
  - Q1 ⊕ C1 ⊕ Q2 ⊕ C2 ⊕ … ⊕ Qn ⊕ Cn -> Qn+1

- **Mesh BART**- 제안 모델 (p.10)



Experiment : Metric (1/2)

- **Word Error Rate** $$\text{WER@}k = \min_{i=1,\ldots,k} \text{EditDist}(ref, hyp^{(i)})/|ref|$$

  EditDist("Lead Data Engineer", "Data Scientist") = 2

  (Delete:Lead, Change:Engineer -> Scientist)

- **Mean Reciprocal Rank (MRR@K)** - https://en.wikipedia.org/wiki/Mean_reciprocal_rank
  Rank 정답이 있는 순위 (1~K)

  Reciprocal Rank = 1/rank

  정답이 1등에 있음 - 1/1 = 1
  2등에 있음 1/2 = 0.5
  3등에 있음 1/3 = 0.333
  K등안에 없음 0

- **Success at K (S@K)** : K등 안에 정답 있음

Experiment : Metric (2/2)

- **BERT F1**- BERTScore: Evaluating Text Generation with BERT / https://arxiv.org/abs/1904.09675

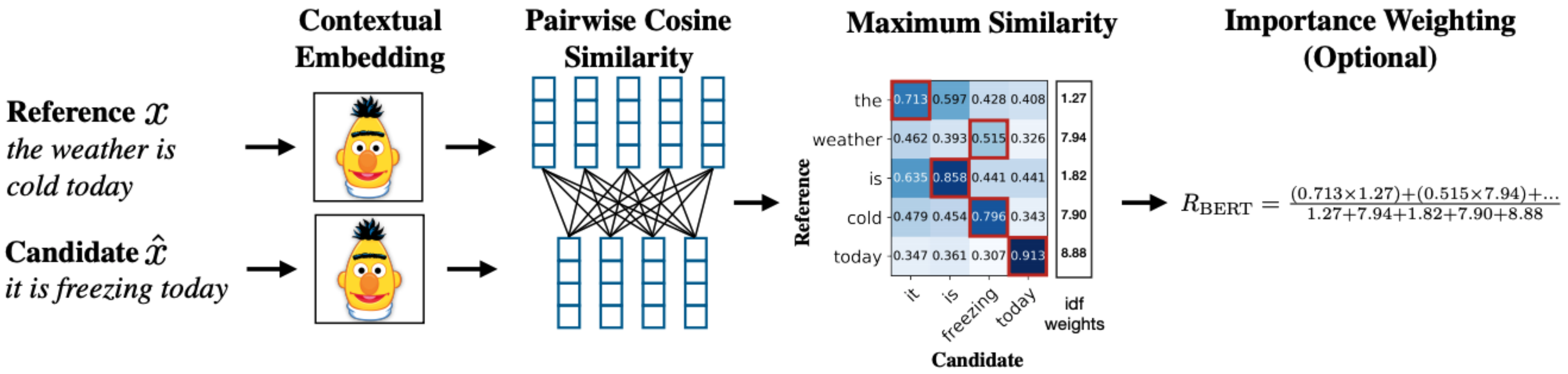


Figure 1: Illustration of the computation of the recall metric R_{BERT}. Given the reference x and candidate $\hat{x}$, we compute BERT embeddings and pairwise cosine similarity. We highlight the greedy matching in red, and include the optional idf importance weighting.

BERTSCORE The complete score matches each token in x to a token in $\hat{x}$ to compute recall, and each token in $\hat{x}$ to a token in x to compute precision. We use greedy matching to maximize the matching similarity score,[2] where each token is matched to the most similar token in the other sentence. We combine precision and recall to compute an F1 measure. For a reference x and candidate $\hat{x}$, the recall, precision, and F1 scores are:

$$R_{\mathrm{BERT}} = \frac{1}{|x|}\sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \ , \quad P_{\mathrm{BERT}} = \frac{1}{|\hat{x}|}\sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \ , \quad F_{\mathrm{BERT}} = 2\frac{P_{\mathrm{BERT}} \cdot R_{\mathrm{BERT}}}{P_{\mathrm{BERT}} + R_{\mathrm{BERT}}} \ .$$

Results : Quality of Generated Queries (1/3)

	WER	BertF1	MRR@3	S@3
Seq2Seq+Attn	88.3	53.1	9.8	14.1
MPS	49.7	72.6	33.9	47.1
BART	41.5	76.1	42.5	54.6
MeshBART	**40.9**	**76.5**	**42.7**	**55.0**

Table 1: Top-3 test performance. Differences between BART and MeshBART on WER and BertF1 are significant ($p < 10^{-4}$) on the Wilcoxon sign-rank test.

- MPS (직전 질의와 Co-ocurrence 가장 높은 질의 선택)도 나름 괜찮은 결과
- MPS < BART < MeshBART

Results : Quality of Generated Queries (2/3)

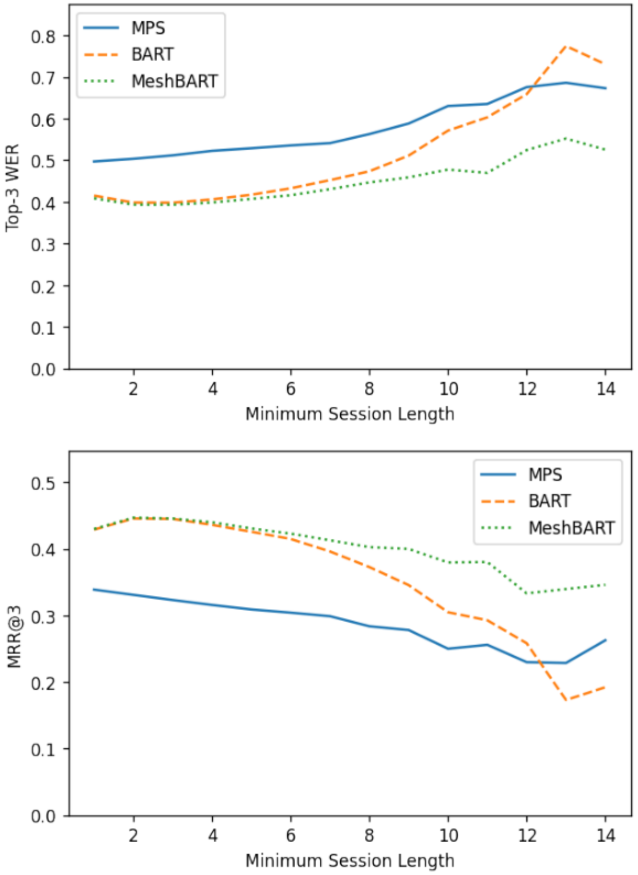


Figure 2: A breakdown of top-3 word error rate and MRR@3 by minimum session length. Each bucket on x-axis indicates a sub-population of test sessions that contain at least X queries.

Minimum Session Length
(한 세션에서 입력된 질의 수) 구간 별로 나누어서 보면,

긴 Session 에서 Mesh BART가 잘한다.

-> 긴 Session일수록
Diverse and Complex intents 를 갖고 있기때문에, Next Query 예측이 어렵다.

Results : Quality of Generated Queries (3/3)

- MeshBART vs MPS (Generative vs Discriminative)
- W/T/L (Wins, Tie, Loose) Analysis : MeshBART vs MPS
  : (30% Wins / 52% Tie / 18% Loose) on MRR@3
- Sessions with only one preceding query (for 39.2% of test session)
  - MeshBART can produce at least one novel suggestions
    (i.e. queries not seen in the candidate pool)
    (MPS는 후보가 없으면 생성이 불가능)

Preceding Query (Q_n)	Candidates	Generative Suggestions (MeshBART)
environmental technology	environmental, environment	environmental, environmental science, sustainability, environmental scientist
part time adobe	part time, part time marketing	part time marketing, marketing, digital marketing, graphic design
aviation security adelaide	adelaide airport security	aviation security, security, security officer, airport security

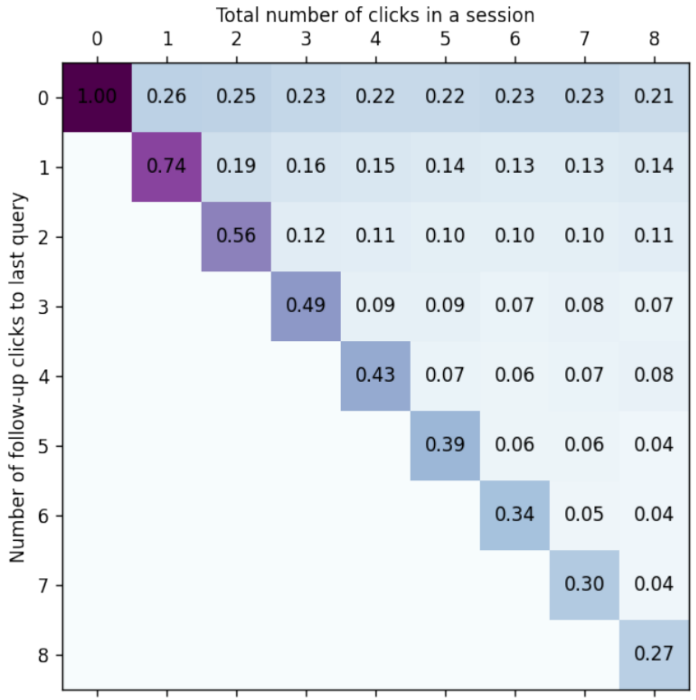
*Adelaide
: 호주도시명

Table 2: Query generation examples on tail queries, based on test sessions with only one preceding query that the logs fail to produce enough candidates for due to scarcity. Generative models such as MeshBART can produce reasonable suggestions regardless of candidate pool coverage.

Results : Analysis of Behavioral Hypotheses (1/3)

- Figure 3(a) : 세션의 마지막 질의에 대한 사용자 클릭의 강도
  -> 대부분의 클릭이 마지막 질의에서 발생
  -> 사용자가 원하는 결과가 나올 때 까지
  질의를 명확히 표현하려고 노력하고,
  검색 세션 종료 전에 대부분 검색결과를
  소비(클릭)함

  -> K4의 가설을 반영
  K_4 : 사용자의 직전에 입력한 질의와 반응한 문서가 중요할 것이다.



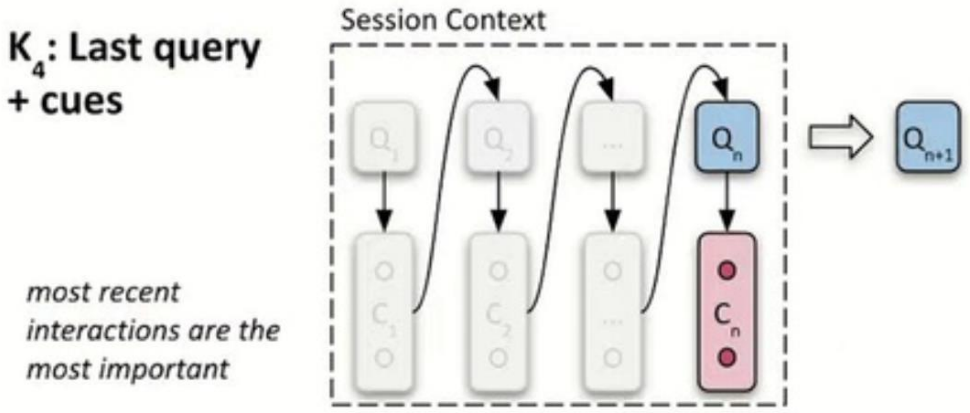
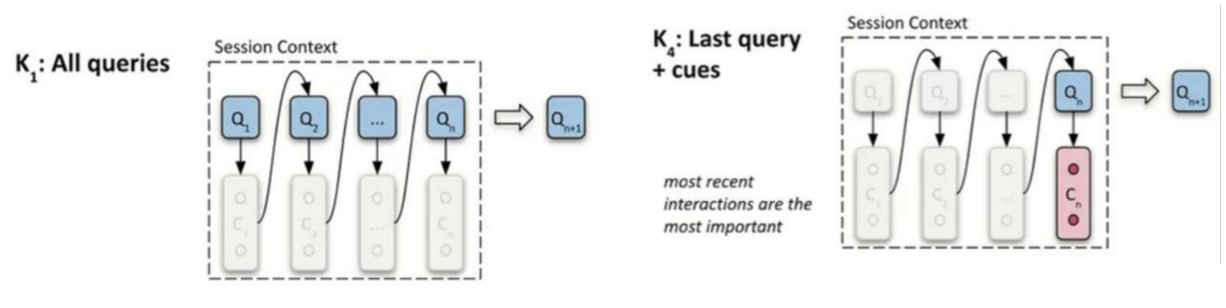
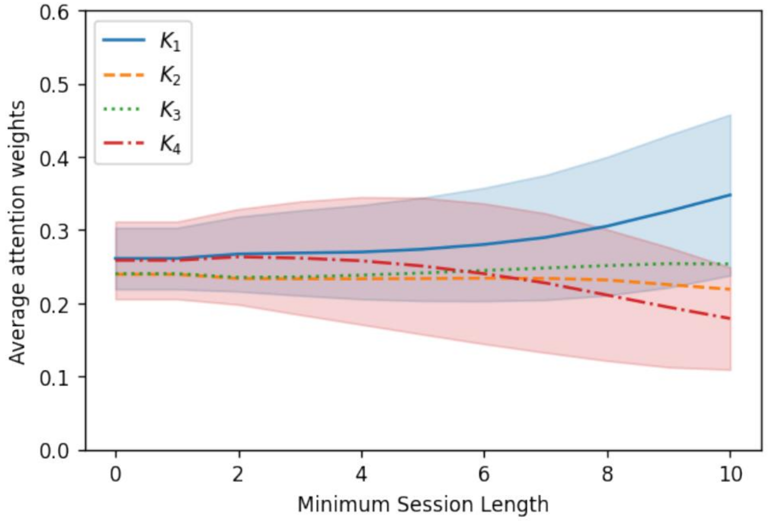


Figure 3: (a) Column-normalized contingency table illustrating clicking behavior.

Results : Analysis of Behavioral Hypotheses (2/3)

- Attention Weight 을 통한 분석
- X: 세션 길이



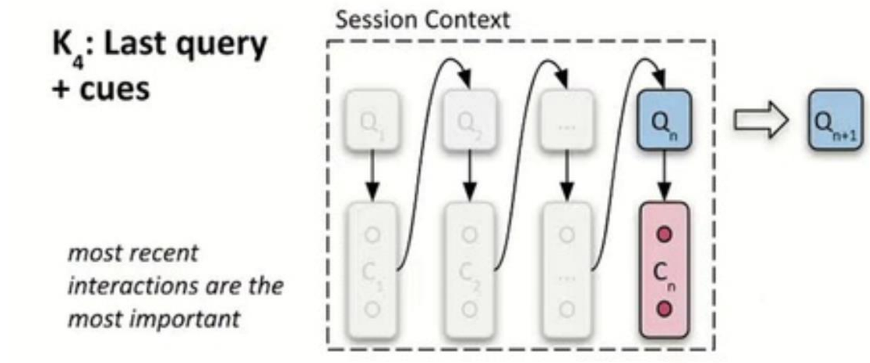


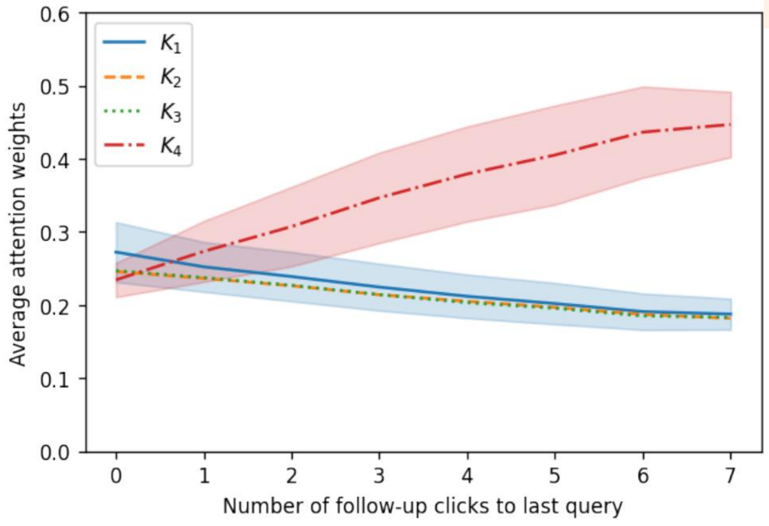
(b) Average attention weights for all four behavioral hypotheses
Shaded areas for K1 and K4 indicate the standard error.

- K1(All preceding queries) 의 Attention이 Positively Correlated.
  : 더 긴 세션 -> 사용자가 적극적으로 exploring.
  Explicit Search Intent(=Query)가 중요하다는 signal
- K4 는 길어질수록 Attention 이 줄어듦 => Most recent interaction 의 중요도는 떨어짐

Results : Analysis of Behavioral Hypotheses (3/3)

- Attention Weight 을 통한 분석
- X: 마지막 질의에서의 클릭 수





(c) Average attention weights for all four behavioral hypotheses
Shaded areas for K1 and K4 indicate the standard error.

- 마지막 질의에서 클릭이 많았던 질의에서 K4에 대한 가중치가 높음.

- (b),(c) => flexibility to draw information from different hypotheses in a unified query generation process

Conclusion

- an effective approach for incorporating user induced interaction patterns as behavioral hypotheses into the query generation process
- Under an encoder-decoder Transformer framework, the proposed tokenwise attentions demonstrate the desirable modeling working by placing emphasis on different behavioral hypotheses at different occasions.
- In future work, we will focus on producing novel continuations of the user's search intent, extending the approach to other domains, and automating the design of behavioral hypotheses.

Reference

[0] https://slideslive.com/38939310/incorporating-behavioral-hypotheses-for-query-generation

[1] https://lucidworks.com/ai-powered-search/head-tail-analysis/

[2] SORDONI, Alessandro, et al. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. 2015. p. 553-562.

[3] LEWIS, Mike, et al. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461, 2019.

[4] https://youtu.be/VmYMnpDLPEo / 딥러닝논문읽기모임 : BART paper review

[5] https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

[6] https://hcnoh.github.io/2018-12-11-bahdanau-attention

[7] BAHDANAU, Dzmitry; CHO, Kyunghyun; BENGIO, Yoshua. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[8] https://en.wikipedia.org/wiki/Mean_reciprocal_rank

[9] ZHANG, Tianyi, et al. Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675, 2019.