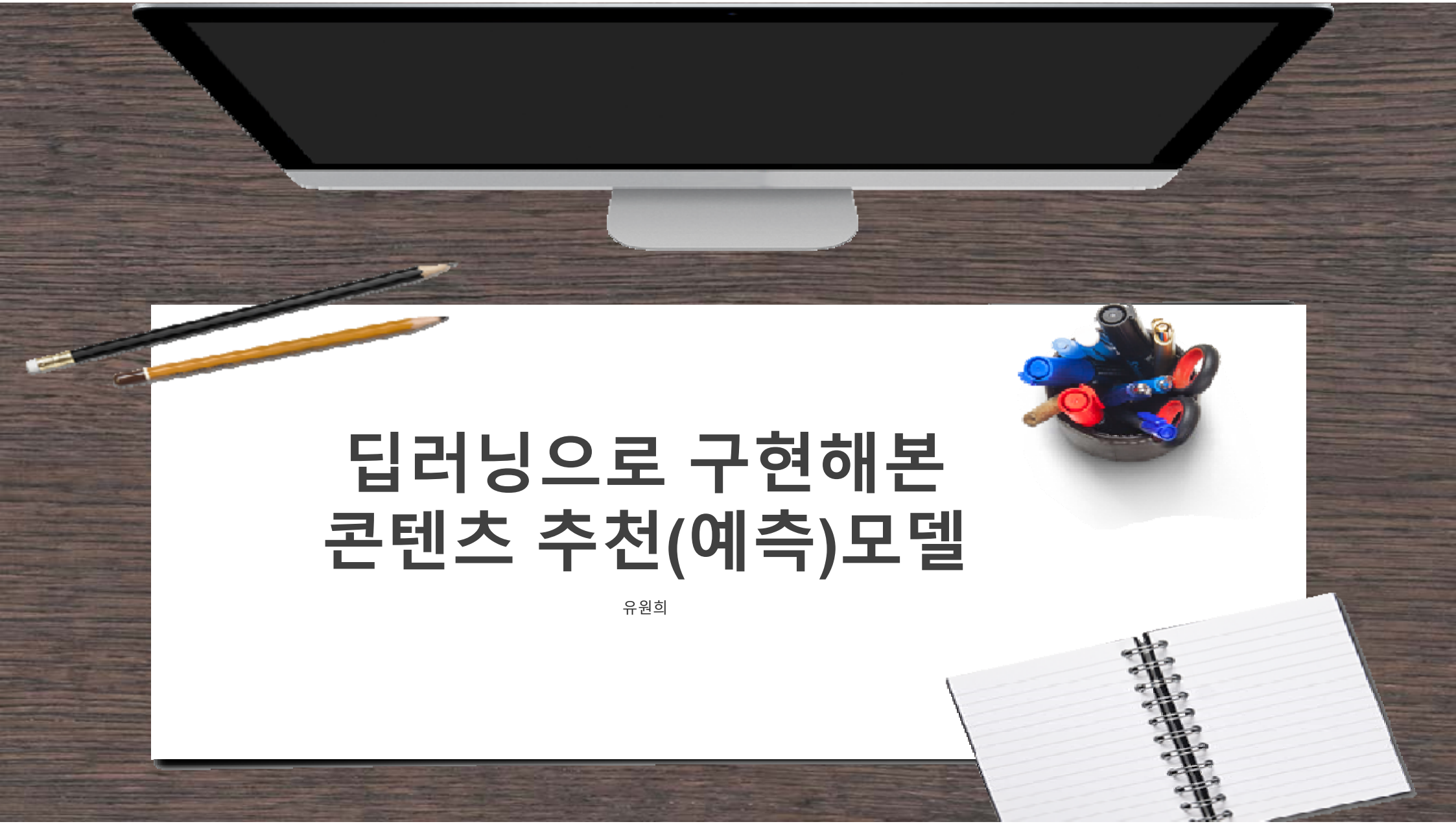


딥러닝으로 구현해본 콘텐츠 추천(예측)모델

유원희

Contents

- 콘텐츠 추천 문제
  - Matrix Completion 문제점
    - Matrix factorization
- 모델링
- 실험
- 평가
- 추가연구

Recommendation Problem in Netflix

- 사용자가 item에 대해 evaluate한 history data를 기반으로 사용자가 아직 사용하지 않은 item에 대한 사용자의 평가를 예측하는 문제

$$
\begin{array}{c} movie. \\ \\ user1 \\ user2 \\ user3 \\ user4 \\ user5 \\ user6 \\ user7 \\ user8 \end{array}
\begin{array}{c}
\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
\left\{\begin{bmatrix}
3 & 5 & * & 4 & 1 & * & * & 2 \\
* & 3 & 5 & 1 & 2 & * & * & 3 \\
4 & 1 & * & 4 & 1 & * & 3 & 2 \\
5 & 2 & * & * & 2 & 3 & * & * \\
* & 2 & 4 & 2 & * & * & 1 & 2 \\
5 & * & * & 5 & 4 & * & * & 4 \\
1 & * & 5 & 2 & 3 & 1 & 5 & 3 \\
* & 3 & 2 & 1 & 4 & * & * & *
\end{bmatrix}\right.
\end{array}
$$

Netflix Prize

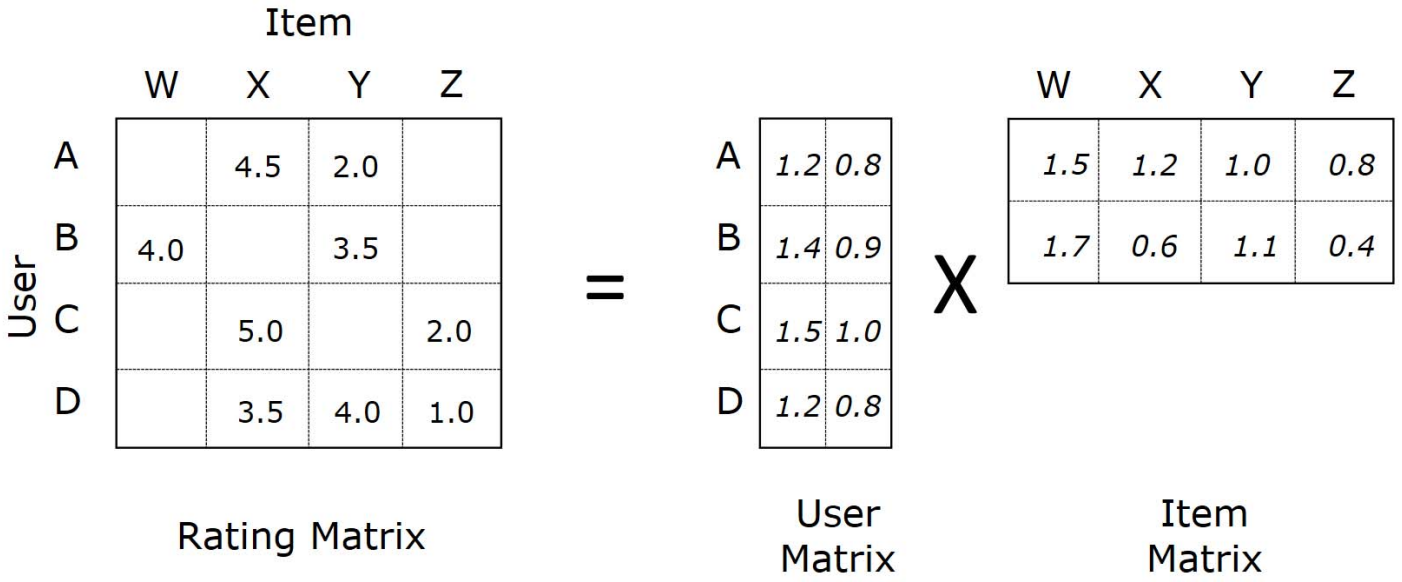
- Collaborative filtering
- SVD++(Matrix Factorization)

- Data
  - User, Movie, rating
- Evaluation
  - RMSE(root mean square error)
    - Netflix Cinematch : 0.9525
    - Cinematch 보다 10%이상 향상시켜라.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

  - TopN precision

Matrix Factorization



Overfitting Problem
Lack of Data

해결책은?

- 콘텐츠 추천방식이 기계학습 방식으로 이동

제안모델#1

- **The mathematics of language models**
  - The probability of a sequence of words can be obtained from the probability of each word given the context of words preceding it, using the chain rule of probability (a consequence of Bayes theorem):

$$P(w_1, w_2, \ldots, w_{n-1}, w_t) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\ldots P(w_t|w_1, w_2, \ldots, w_{t-1})$$

  - Most probabilistic language models (including published neural net language models)
  - approximate $P(w_t|w_1, w_2, \ldots, w_{t-1})$ using a fixed context of size n−1
  - as in n-grams(i.e. using $P(w_t|w_{t-n+1}, \ldots, w_{t-1})$)

제안모델#2

- **The mathematics of neural net language models**

$P(w_t|context)$

$P(w_t|w_{t-n+1}, \dots, w_{t-1})$

– The neural network is trained using a gradient-based optimization algorithm to maximize the training set *log-likelihood*

$$L(\theta) = \sum_{t=1}^{T} \log P(w_t|w_{t-n+1}, \dots, w_{t-1})$$

제안모델#3

- **The mathematics of item recommendation models**

$P(i_t|context)$

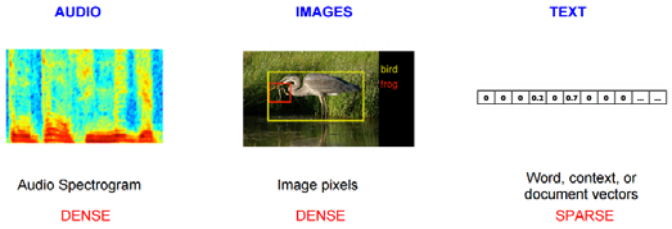
$P(i_t|i_{t-n+1}, \ldots, i_{t-1})$

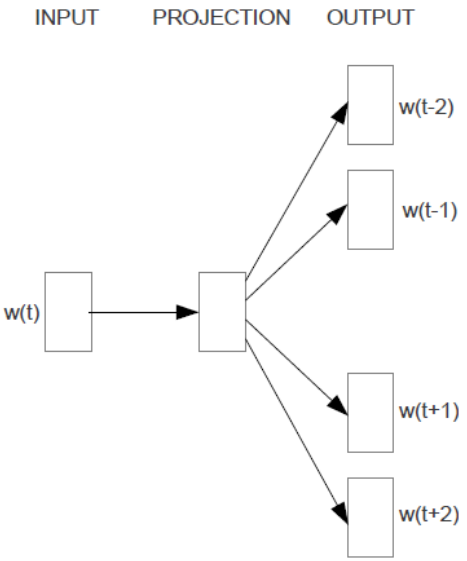
  - The neural network is trained using a gradient-based optimization algorithm to maximize the training set *log-likelihood*

$$L(\theta) = \sum_{t=1}^{T} \log P(i_t|i_{t-n+1}, \ldots, i_{t-1})$$

제안 모델#4 - 입력 설계

- Item Embedding





Skip-gram

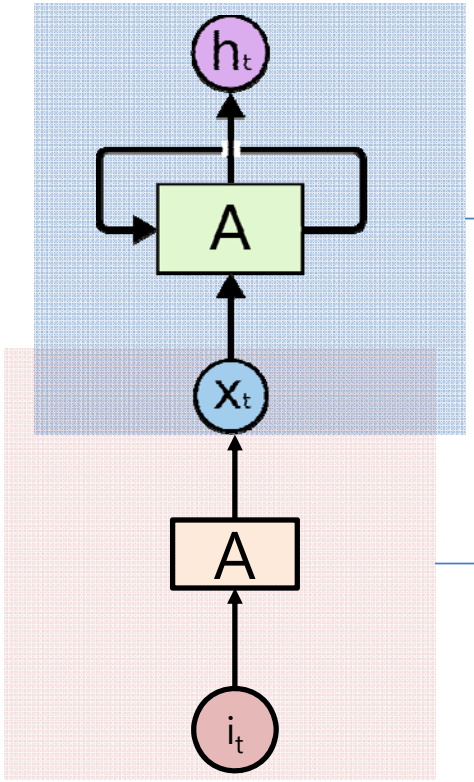
$$P(w_t|context)$$

$$P(w_t|w_{t-k},\dots,w_{t-1},w_{t+1},\dots,w_{t+k})$$

$$L(\theta) = \sum_{-k\le j-1, j\le k} \log P(w_{t+j}|w_t)$$

to maximize

제안모델#5 - 그림



RNN(LSTM)
(Next item prediction)

NN
(Item Embedding)

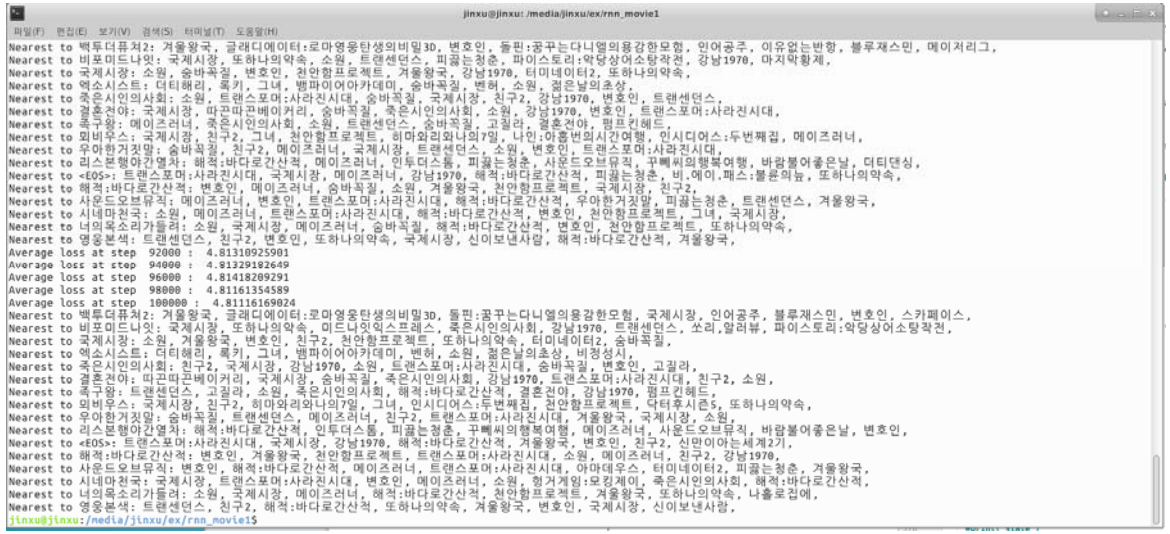
데이터

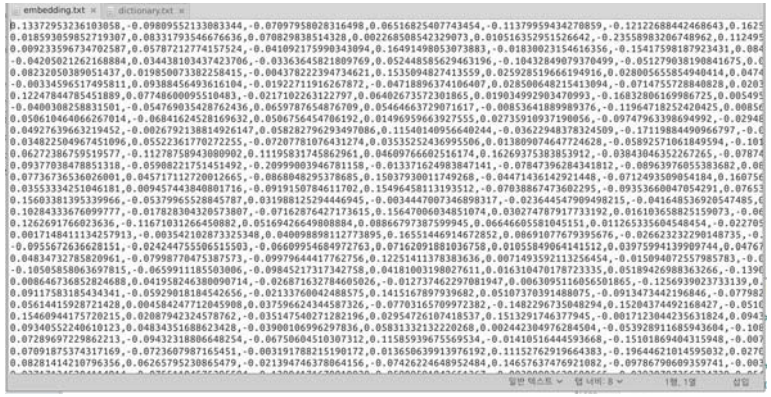
- 영화 : 70종
- 사용자: 89,158명
- 시청기록 : 14,080,936개

- Train data set
  - 71,319명
- Test data set
  - 17,939명

평가#1

- 2-D embedding of the learned item representation

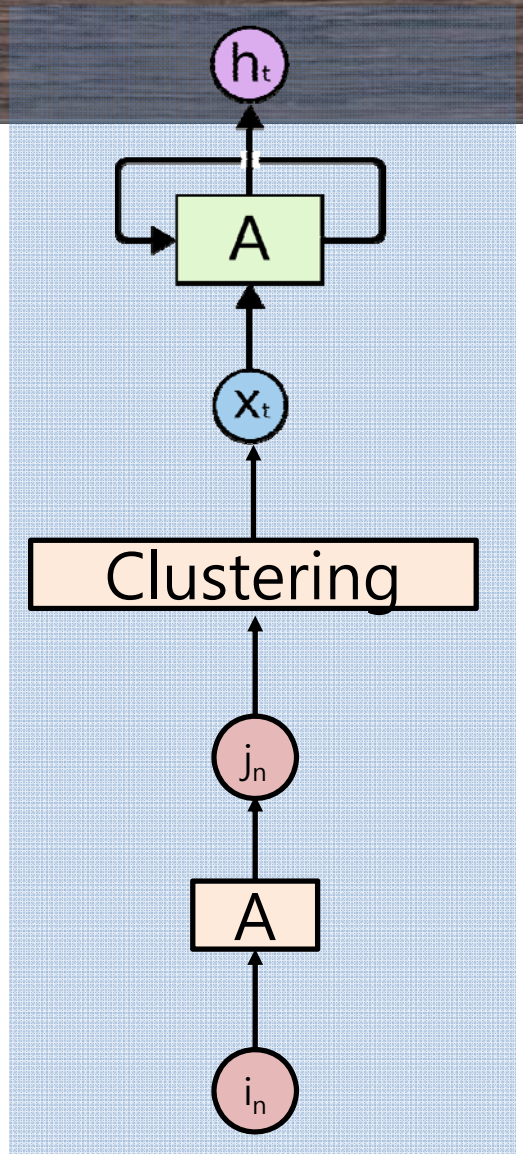


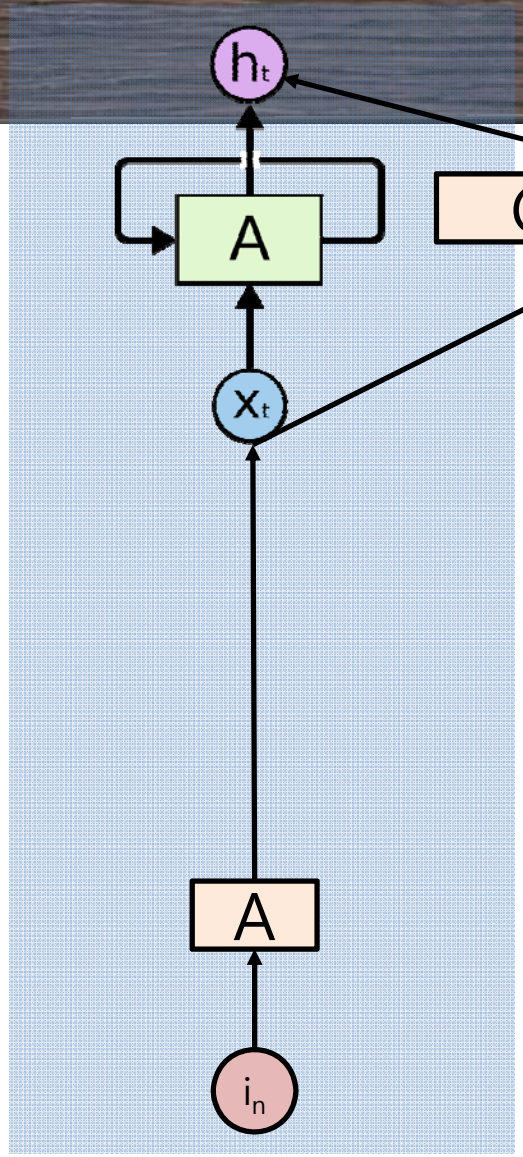


평가#2

	Embedding + RNN	RNN	baseline
accuracy	65.35%	3.4%	1.25%
perplexity			

추가연구





TOP-K
evaluation