

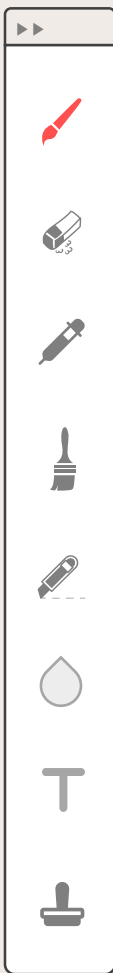


NEUSUM

Neural Document Summarization by Jointly Learning
to Score and Select Sentences

김사무엘 박새란 이승언





1.Introduction

2.Problem Formulation

3.Neural Document Summarization

4.Experiments

5.Discussion



1.Introduction



1. Introduction

자동 요약(automatic text summarization)에서 핵심은 문장 단위의 중요 콘텐츠를 식별하는 것임

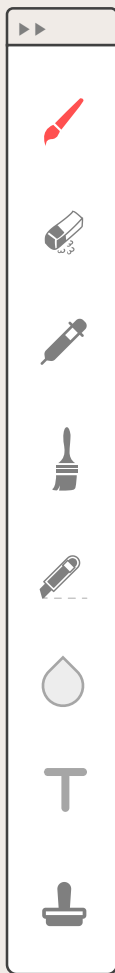
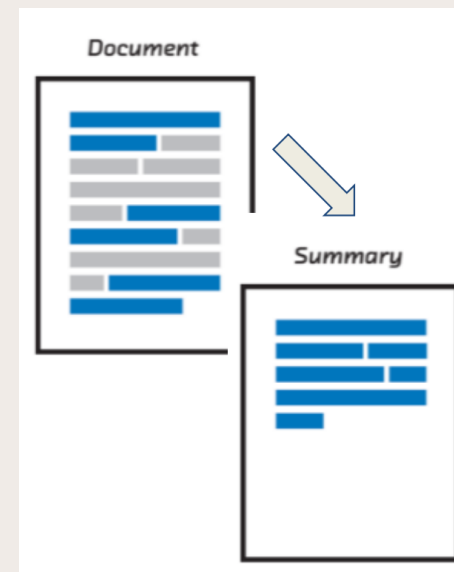
1. 이전의 Extractive summarization은?

STEP 1. sentence scoring : 각 문장의 중요도를 계산

1. Feature 기반 : 단어 확률(word probability), TF-IDF, 문장 위치 등 활용
2. Graph 기반 : TextRank, LexRank
3. Neural Network 기반

STEP2. sentence selection :scoring 기반으로

1. Maximal Marginal Relevance(MMR) : 최대 점수, 최소 중복
2. Integer Linear Programming (ILP)
3. Submodular functions
4. Neural Network 기반





1. Introduction

2. NeuSUM은?

1. [문장 평가(sentence scoring) + 문장 선택(sentence selection)]
 - ☞ 하나로 결합된 end-to-end model
2. 앞서 추출된 요약 문장들의 정보를 함께 고려해 각 요약 문장을 선택하는 특징을 가짐
3. Without handcrafted features
4. CNN/Daily Mail 데이터셋에서 최고의 성능을 기록함(state-of-the-art)





1. Introduction

3. NEUSUM 모델의 구조는

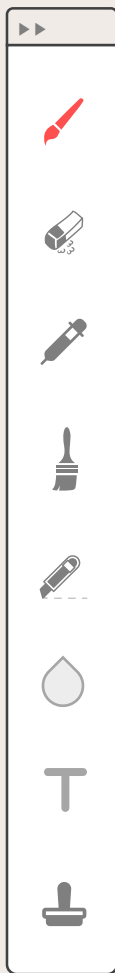
1. 문서 인코더(document encoder)

- 문서 내의 모든 문장을 벡터로 변환하는 부분
- 자연어 문장의 실제 구성처럼 계층적 구조를 가짐 (단어 → 문장 → 문서)

2. 문장 추출부(sentence extractor)

- 순환신경망(RNN) 기반
- 매 시점마다 과거에 추출된 문장들(요약문)에 대한 정보 저장/유지
- 매 시점마다 문장 추출 상태(state) 제공 → 문장 평가(scoring)에 활용





2. Problem Formulation



2. Problem Formulation

1. 추출 요약은 문서의 주요 의미를 표현(represent)하기 위해 유용한 정보를 제공하는(informative) 문장들을 추출하는 문제임

1) L개의 문장으로 이루어진 document

$$\mathcal{D} = (S_1, S_2, \dots, S_L)$$

2. 문장 평가 함수(scoring function) $f(S)$ 를 학습

2) 목적함수 : 문장 평가 함수를 최대화

• L개의 문장을 가진 문서 D 가 있을 때, 추출 요약 시스템은 문장 평가 함수(scoring function) $f(S)$ 를 학습하는 것을 목표로 함

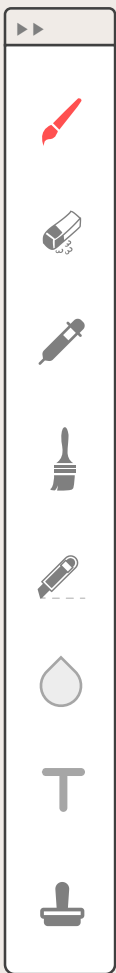
$$\begin{aligned} & \arg \max_{\mathcal{S}} f(\mathcal{S}) \\ \text{s.t.} \quad & \mathcal{S} = \{\hat{S}_i | \hat{S}_i \in \mathcal{D}\} \\ & |\mathcal{S}| \leq l. \end{aligned}$$

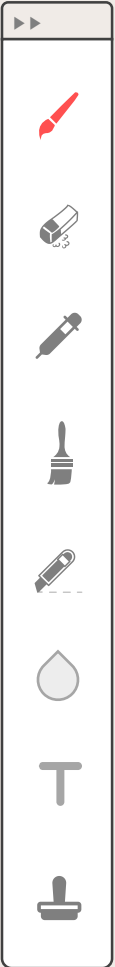
3. Rouge F1 지표 사용

3) Rouge F1 gain (scoring function)

• 본 연구에서는 모델이 더 긴 문장을 요약문에 포함시키려는 경향을 방지하기 위해, 요약문을 평가하는 데 ROUGE F1 지표를 사용하고 문장 수를 l 로 제한함

$$g(S_t | \mathcal{S}_{t-1}) = r(\mathcal{S}_{t-1} \cup \{S_t\}) - r(\mathcal{S}_{t-1})$$





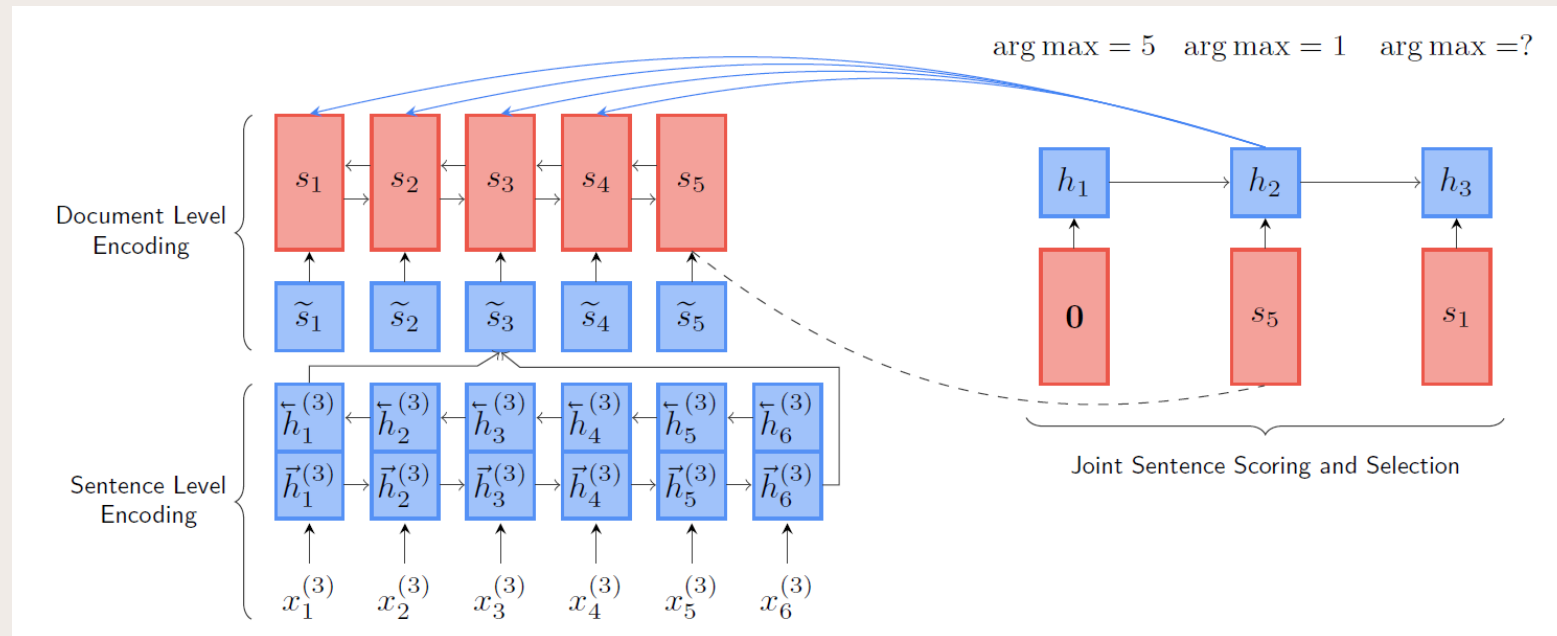
3. Neural Document Summarization



3. Neural Document Summarization

1. NEUSUM 모델의 구조

- NEUSUM 모델의 구조는 계층적인 문서 인코더(hierarchical document encoder)와 문장 추출부(sentence extractor)로 구성됨
- 문서 인코더에서 문서 전체의 문장들이 인코딩되어 전달되면, 문장 추출부에서 매 시점마다 각 문장을 평가하고 문장 수 제한에 도달할 때까지 요약문을 추가함





3. Neural Document Summarization

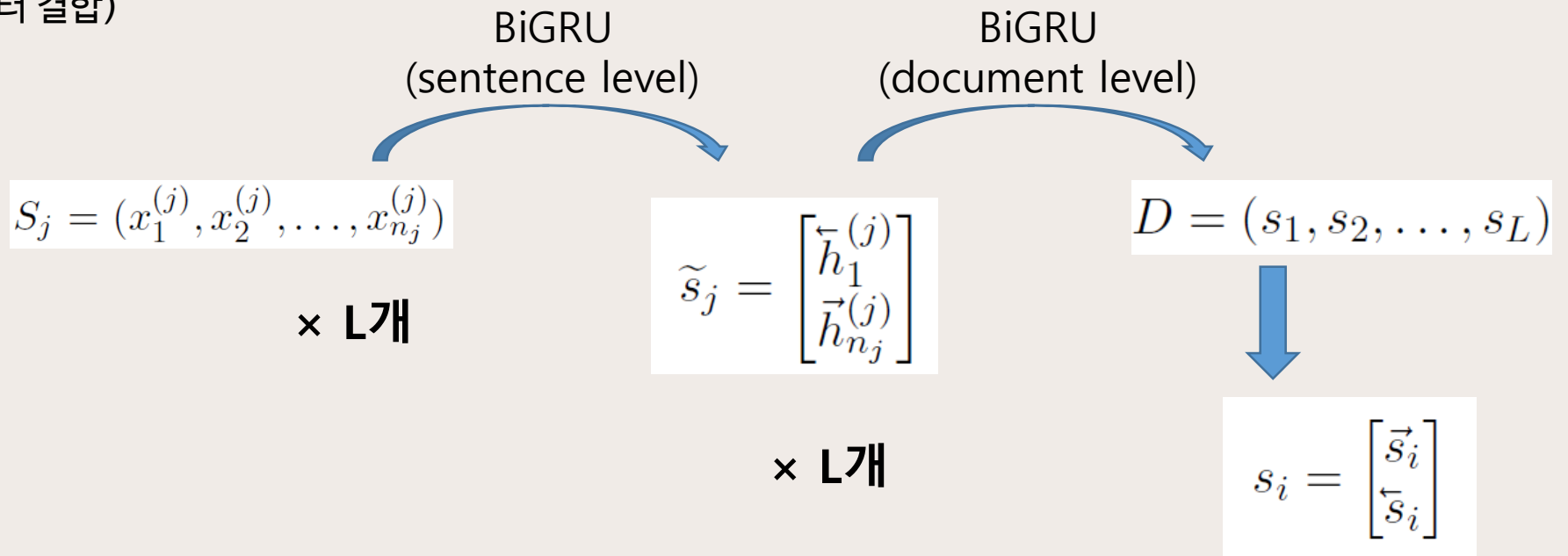
1-1. 문서 인코더(document encoder) : 두 단계의 계층적인 BiGRU 층으로 구성됨

1) 문장 단위의 인코딩층

각 토큰을 BiGRU 층에 통과시켜 문장의 임베딩 벡터를 생성함 (마지막 시점의 히든 벡터 2개 결합)

2) 문서 단위의 인코딩층

각 문장의 임베딩 벡터들을 인코딩 BiGRU 층에 통과시켜 문서의 임베딩, 즉 문장 임베딩 벡터들의 집합을 생성함
(각 시점의 히든 벡터 결합)





3. Neural Document Summarization

1-2. 문장 추출부(sentence extractor) : GRU 층과 MLP 층으로 구성됨

- 문장 추출부는 최종적으로 문장을 추출해 요약문을 생성할 때

1) 이전에 선택된 문장들의 정보를 기억하고 (to remember)

2) 아직 선택되지 않은 남은 문장들의 중요도를 평가함 (to score)

1) GRU 층에서는 직전 시점($t-1$)에 선택된 문장의 임베딩 벡터를 통과시켜 현재 시점(t)의 히든 벡터(h_t)를 생성함

2) MLP 층에서는 현재 시점의 히든 벡터(h_t)와 현재 시점 문장의 임베딩 벡터를 통과시켜 문장 중요도를 계산함

- 이상의 과정을 통해 t 시점에 계산된 모든 문장들의 중요도를 바탕으로, 가장 중요도가 높은 문장을 요약문으로 추출함

GRU -

$$h_t = \text{GRU}(s_{t-1}, h_{t-1})$$

MLP -

$$\delta(S_i) = \mathbf{W}_s \tanh(\mathbf{W}_q h_t + \mathbf{W}_d s_i)$$

$\times L$ 개

$$\hat{S}_t = \arg \max_{S_i \in \mathcal{D}} \delta(S_i)$$

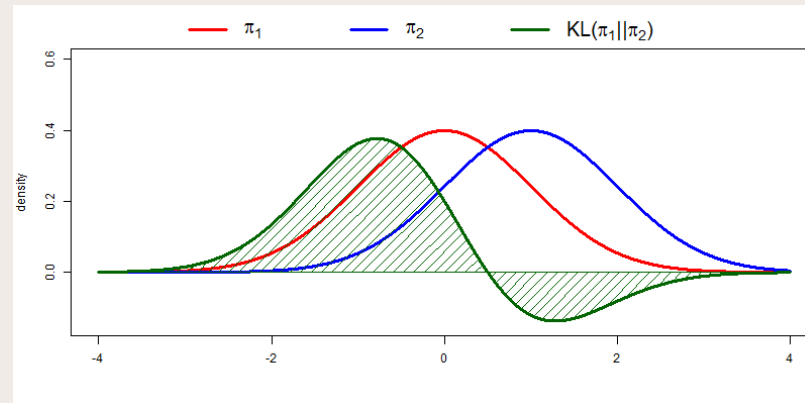




3. Neural Document Summarization

2. 목적함수

- NEUSUM 모델의 목적 함수는 모델의 예측값 P 와 학습데이터의 라벨값 Q 간의 Kullback-Leibler (KL) divergence를 사용함
 - P 계산하기 위해, 모델이 예측한 각 문장들의 중요도를 softmax 함수로 정규화함
 - Q 를 계산하기 전에, 각 문장의 ROUGE F1 gain 점수가 음수(negative)일 수 있음을 고려하여 $[0,1]$ 값으로 스케일링하기 위해서 Min-Max 정규화를 적용함
 - Q 계산하기 위해, 각 문장의 ROUGE F1 gain 점수를 softmax 함수로 정규화함



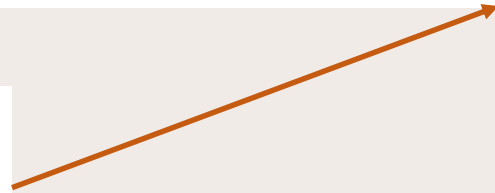


2. 목적함수

Model prediction distribution(P)

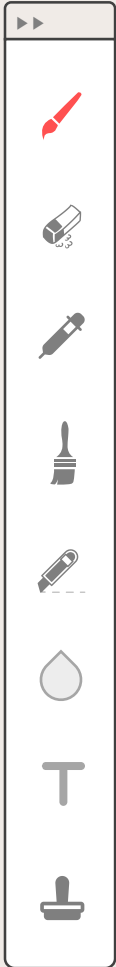
Reference distribution(Q)

$$J = D_{KL}(P \parallel Q)$$





4. Experiments





4. Experiments

1. Dataset

1. CNN/Daily Mail Dataset 사용

기사와 기사의 하이라이트부분을 포함하고 있는 데이터셋이지만 추출 요약으로 사용하기에는 부족하여 추출요약문 직접 생성

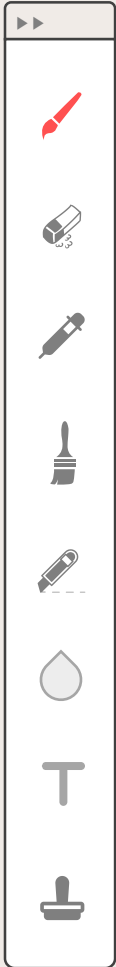
2. 각 문장과 기존 요약문을 대조하며, ROUGE-2 F1 점수가 가장 높은 문장 조합을 추출 요약문으로 생성함

문장 조합의 경우의 수가 많아 계산 비용이 큼 -> Greedy 접근법

- 1) 1문장의 조합들에 대해 ROUGE-2 F1 점수를 계산함
- 2) 1문장씩 늘려가며 k문장의 조합들에 대해 ROUGE-2 F1 점수를 계산함
- 3) k문장의 조합들에 대한 ROUGE-2 F1 점수들 중 최댓값이 이전 k-1까지의 최댓값보다 작을 경우 중단함

3. Data preprocessing

- 고유명사(named entity)들은 익명화된 식별자로 대체됨 (ex. Entity4)



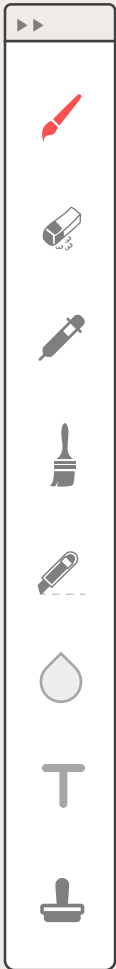


4. Experiments

2. Model Training

- 각 문서는 80문장으로 제한했고, 각 문장은 100 단어로 제한함

| Vocab size | 100,000 |
|--------------------|--|
| embedding | 50차원, GloVe(cover 90.39% of NeuSUM vocab) |
| Sentence encoder | BiGRU 256차원 |
| Sentence extractor | GRU 256차원 |
| Initialization | Gaussian distribution with Xavier |
| optimizer | Adam optimizer |
| Learning rate | 0.001 |
| Momentum | 0.9, 0.999 |
| Dropout | 0.3 (after sentence-level encoder) 0.2 (after document-level encoder) |





4. Experiments

3. Model Testing

- 모델 성능비교에 사용된 모델들
 - LEAD3 (선행하는 3개 문장 추출)
 - TextRank (Mihalcea and Tarau, 2004) from Gensim
 - CRSUM (Ren et al., 2017)
 - NN-SE (Cheng and Lapata, 2016)
 - SUMMARUNNER (Nallapati et al., 2017)
 - PGN (See et al., 2017)
- 모델 간의 비교 평가에는 ROUGE-1, ROUGE-2, ROUGE-L(LCS) 지표가 사용됨
- 생성된 요약문에 대해 사람이 정성평가를 추가로 수행함

| Models | ROUGE-1 | ROUGE-2 | ROUGE-L |
|----------------------------|--------------------|--------------------|--------------------|
| LEAD3 | 40.24 ⁻ | 17.70 ⁻ | 36.45 ⁻ |
| TEXTRANK | 40.20 ⁻ | 17.56 ⁻ | 36.44 ⁻ |
| CRSUM | 40.52 ⁻ | 18.08 ⁻ | 36.81 ⁻ |
| NN-SE | 41.13 ⁻ | 18.59 ⁻ | 37.40 ⁻ |
| PGN [‡] | 39.53 ⁻ | 17.28 ⁻ | 36.38 ⁻ |
| LEAD3 [‡] * | 39.2 | 15.7 | 35.5 |
| SUMMARUNNER [‡] * | 39.6 | 16.2 | 35.3 |
| NEUSUM | 41.59 | 19.01 | 37.98 |

| Models | Info | Rdnd | Overall |
|---------------|-------------|-------------|-------------|
| NN-SE | 1.36 | 1.29 | 1.39 |
| NEUSUM | 1.33 | 1.21 | 1.34 |



5. Discussion





5. Discussion

1. 문장 선택의 정확도

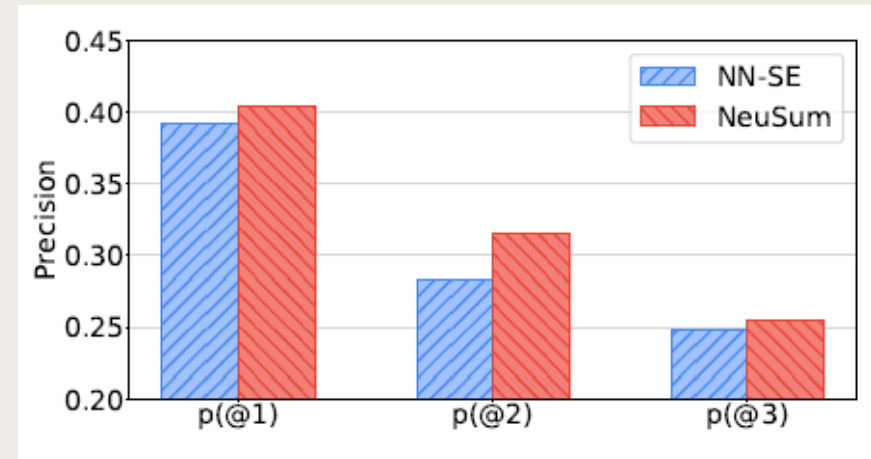
- 본 논문에서는 각 시점별 예측값에 대한 정확도 (precision at step t)를 계산해, NEUSUM 모델과 NN-SE 모델을 비교함
- 모든 시점에서 NEUSUM이 NN-SE보다 정확했으며, 특히 $t=2$ 에서 간극이 커짐

$$1_{\mathcal{O}}(x) = \begin{cases} 1 & \text{if } x \in \mathcal{O} \\ 0 & \text{if } x \notin \mathcal{O} \end{cases}$$

$$p(@t) = \frac{1}{|\mathcal{T}|} \sum_{D \in \mathcal{T}} 1_{\mathcal{O}}(\mathcal{S}[t])$$

Document D in testset

Predicted summary(\mathcal{S})





5. Discussion

2. 문장 포지션

- 본 논문에서는 요약문으로 추출된 문장들의 위치 분포를 통해, NEUSUM 모델과 NN-SE 모델을 비교함
- NEUSUM의 위치 분포는 NN-SE보다 넓게 퍼져 있으며, 정답 요약문과도 더 분포가 유사함

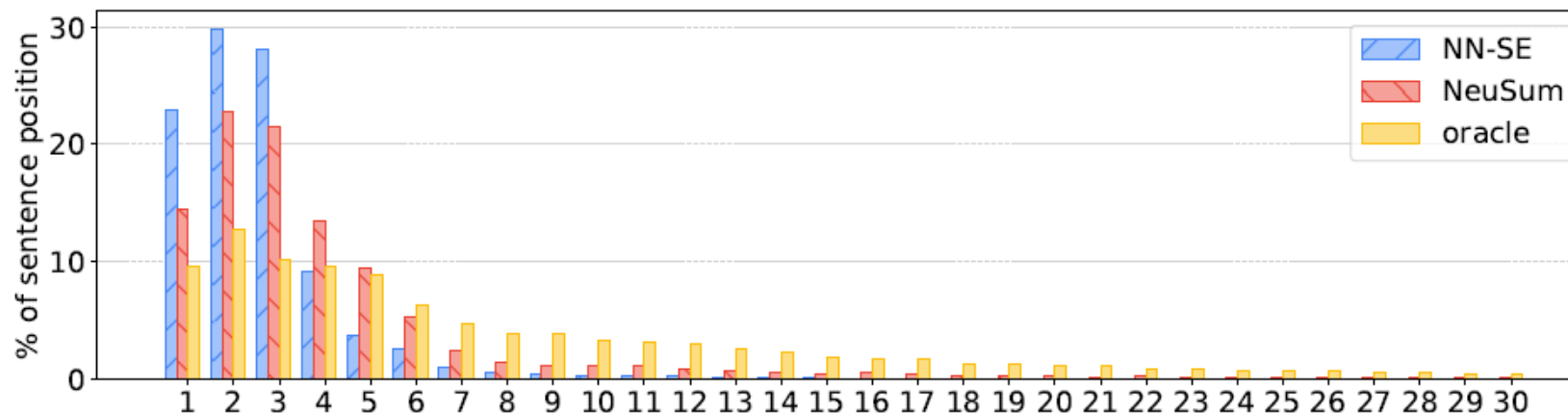


Figure 2: Position distribution of selected sentences of the NN-SE baseline, our NEUSUM model and oracle on the test set. We only draw the first 30 sentences since the average document length is 27.05.

