

## < Hierarchical Attention Networks for Document Classification >

### ○ Abstract

- 문서 분류를 위한 계층적 어텐션 네트워크를 제안

### ○ Introduction

- 문서는 문장들로 이루어져 있고, 문장은 단어들로 이루어져 있음
- 이 논문에서는 문서를 분류할 때, 문서가 가진 계층적 구조적 특징을 활용하고자 함

### ○ Method

- 문서의 계층적 구조를 어텐션에 적용
- Word Level과 Sentence Level 두 가지 Attention Mechanism을 사용

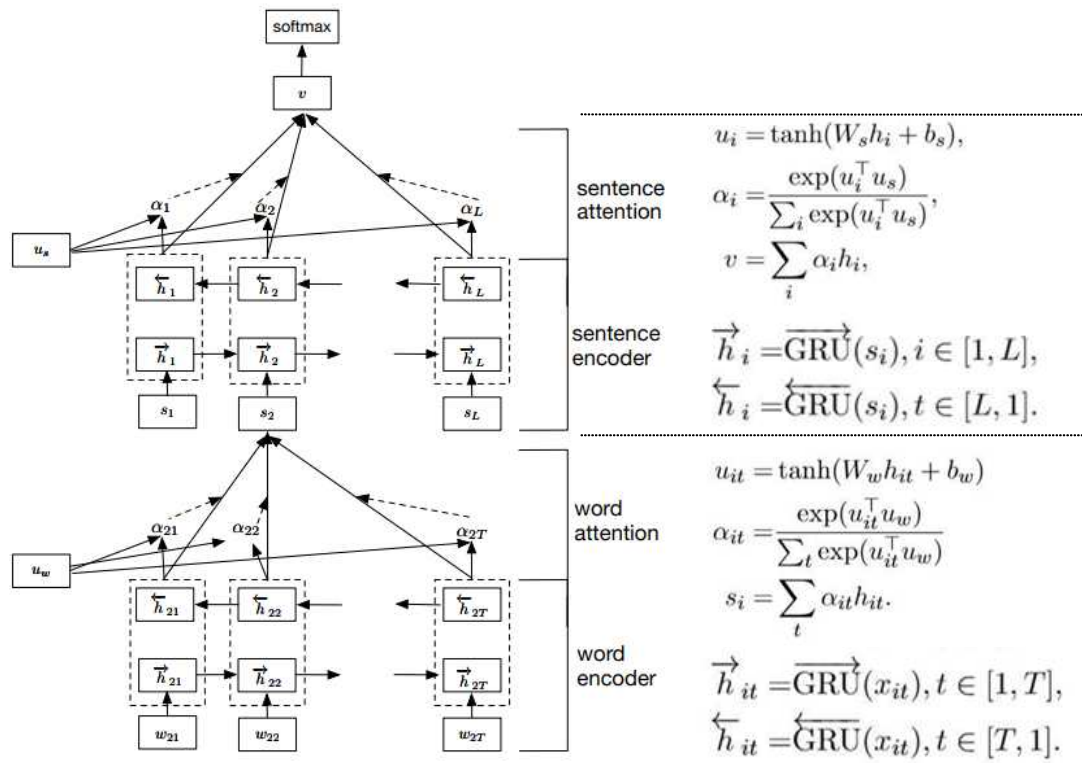
### ○ Conclusion

- 문맥을 파악해서 단어와 문장을 학습한 결과가 제일 좋은 결과가 나옴
- 어떤 문장이나 단어가 실제로 문서를 분류하는데 영향을 미치는지에 대한 분석이 가능

## ○ Notation

- $D = \sum_{i=1}^L s_i$  ,  $S = \sum_{i=1}^{T_i} w_{it}$
- A document L has sentences  $S_i$  and each sentence contains  $T_i$  words.
- $w, t \in [0, T]$  : the words in the  $i$ th sentence

## ○ Model



**Figure 2:** Hierarchical Attention Network.

## ○ Dataset

- 1) Yelp : 서비스 업장 리뷰, 5점 척도
- 2) IMDB review : 영화 리뷰, 10점 척도
- 3) Yahoo Answer : 질의 응답 주제 분류하는 데이터 셋, 10개 카테고리
- 4) Amazon review : 상품 리뷰, 5점 척도

## ○ Model configuration and training

- 1) Document -> sentence로 split
- 2) Tokenize using Stanford's CoreNLP
- 3) 5번 이상 등장한 단어들만 vocabulary에 추가, 아닌 것들은 UNK token으로 처리
- 4) Word embedding  $W_e$  : Initialized by using unsupervised word2vec
- 5) Hyperparameter 설정은 Validation set을 통해 조절
- 6) 논문 경험으로는
  - word embedding dimension : 200,
  - GRU dimension : 50 (bidirectional 이므로 100)
  - word / sentence context vector  $u_w, u_s$  dimension : 100 (Initialized randomly)
- 7) Mini batch 64, batch는 비슷한 길이의 문장들을 가지고 있는 문서들 끼리

## ○ Result

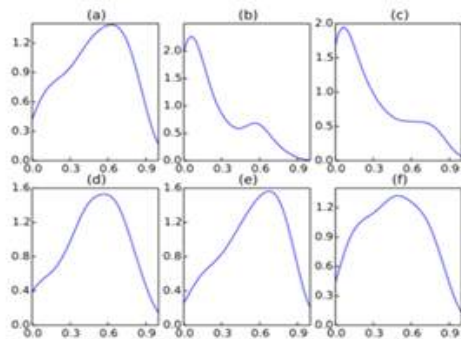
### (1) 다른 방법과 본 논문에서 제안한 구조 비교

|                           | Methods            | Yelp'13     | Yelp'14     | Yelp'15     | IMDB        | Yahoo Answer | Amazon      |
|---------------------------|--------------------|-------------|-------------|-------------|-------------|--------------|-------------|
| <b>Zhang et al., 2015</b> | BoW                | -           | -           | 58.0        | -           | 68.9         | 54.4        |
|                           | BoW TFIDF          | -           | -           | 59.9        | -           | 71.0         | 55.3        |
|                           | ngrams             | -           | -           | 56.3        | -           | 68.5         | 54.3        |
|                           | ngrams TFIDF       | -           | -           | 54.8        | -           | 68.5         | 52.4        |
|                           | Bag-of-means       | -           | -           | 52.5        | -           | 60.5         | 44.1        |
| <b>Tang et al., 2015</b>  | Majority           | 35.6        | 36.1        | 36.9        | 17.9        | -            | -           |
|                           | SVM + Unigrams     | 58.9        | 60.0        | 61.1        | 39.9        | -            | -           |
|                           | SVM + Bigrams      | 57.6        | 61.6        | 62.4        | 40.9        | -            | -           |
|                           | SVM + TextFeatures | 59.8        | 61.8        | 62.4        | 40.5        | -            | -           |
|                           | SVM + AverageSG    | 54.3        | 55.7        | 56.8        | 31.9        | -            | -           |
|                           | SVM + SSWE         | 53.5        | 54.3        | 55.4        | 26.2        | -            | -           |
| <b>Zhang et al., 2015</b> | LSTM               | -           | -           | 58.2        | -           | 70.8         | 59.4        |
|                           | CNN-char           | -           | -           | 62.0        | -           | 71.2         | 59.6        |
|                           | CNN-word           | -           | -           | 60.5        | -           | 71.2         | 57.6        |
| <b>Tang et al., 2015</b>  | Paragraph Vector   | 57.7        | 59.2        | 60.5        | 34.1        | -            | -           |
|                           | CNN-word           | 59.7        | 61.0        | 61.5        | 37.6        | -            | -           |
|                           | Conv-GRNN          | 63.7        | 65.5        | 66.0        | 42.5        | -            | -           |
|                           | LSTM-GRNN          | 65.1        | 67.1        | 67.6        | 45.3        | -            | -           |
| <b>This paper</b>         | HN-AVE             | 67.0        | 69.3        | 69.9        | 47.8        | 75.2         | 62.9        |
|                           | HN-MAX             | 66.9        | 69.3        | 70.1        | 48.2        | 75.2         | 62.9        |
|                           | HN-ATT             | <b>68.2</b> | <b>70.5</b> | <b>71.0</b> | <b>49.4</b> | <b>75.8</b>  | <b>63.6</b> |

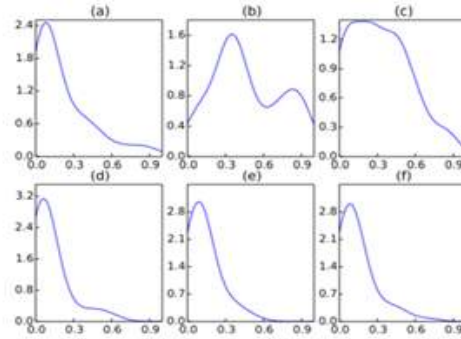
Table 2: Document Classification, in percentage

- 결국 문맥을 파악해서 단어와 문장을 학습한 결과가 제일 좋은 결과가 나왔음.
- 계층적 구조에 단순히 average나 max pooling을 하는 것보다 이 논문에서 제안한 Hierarchical Attention Model 이 모든 데이터 셋에서 성능이 좋았음

## (2) Attention 가중치 분포



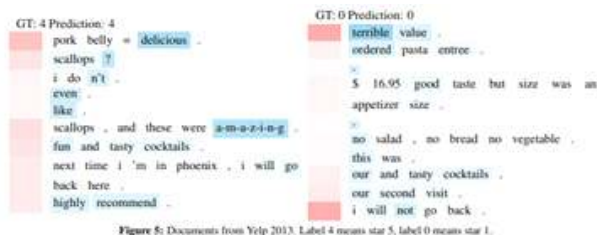
**Figure 3:** Attention weight distribution of *good*. (a) — aggregate distribution on the test split; (b)-(f) stratified for reviews with ratings 1-5 respectively. We can see that the weight distribution shifts to *higher* end as the rating goes higher.



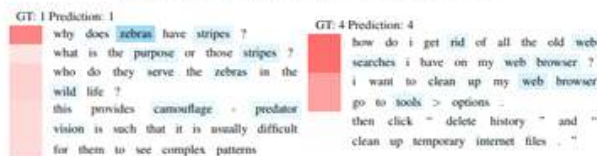
**Figure 4:** Attention weight distribution of the word *bad*. The setup is as above: (a) contains the aggregate distribution, while (b)-(f) contain stratifications to reviews with ratings 1-5 respectively. Contrary to before, the word *bad* is considered important for poor ratings and less so for good ones.

- 모델이 상황에 따른 단어의 중요성을 파악할 수 있는지 확인하기 위해 'good', 'bad' 단어의 attention 가중치 분포를 그림
- 분포에서 x축은 0에서 1까지의 단어에 attention 가중치를 의미
- Good 분포를 보면 별점이 1→5로 갈수록 분포가 왼쪽에서 오른쪽으로 치우침
- Good 분포는 정확히 오른쪽으로 크게 치우치지는 않는 것을 볼 수 있음
- Bad 분포를 보면 별점이 1→5로 갈수록 분포가 중간을 기준으로 왼쪽으로 치우침
- 'Bad' 라는 단어는 낮은 점수라고 판단하는데 더 중요하게 고려됨
- 같은 단어라도 다른 문맥에서 사용되었을 때 Attention Score의 분포가 다름

## (3) Attention Score



**Figure 5:** Documents from Yelp 2013. Label 4 means star 5, label 0 means star 1.



**Figure 6:** Documents from Yahoo Answers. Label 1 denotes Science and Mathematics and label 4 denotes Computers and Internet.

- \* 빨강 : Sentence attention score
- \* 파랑 : Word attention score

- 문서의 Class를 분류하는데 있어 중요한 정보를 담고 있는 문장과 단어를 attention score로 직관적으로 확인할 수 있음