목차

서론

Sentence classification model을 구현하고 성능을 평가하여 기존 연구와 정량적 성능을 비교해 보기로 한다. 모델의 설계와 구현은 아래의 Yoon Kim의 논문과 그 구현 코드들을 참고하였다.

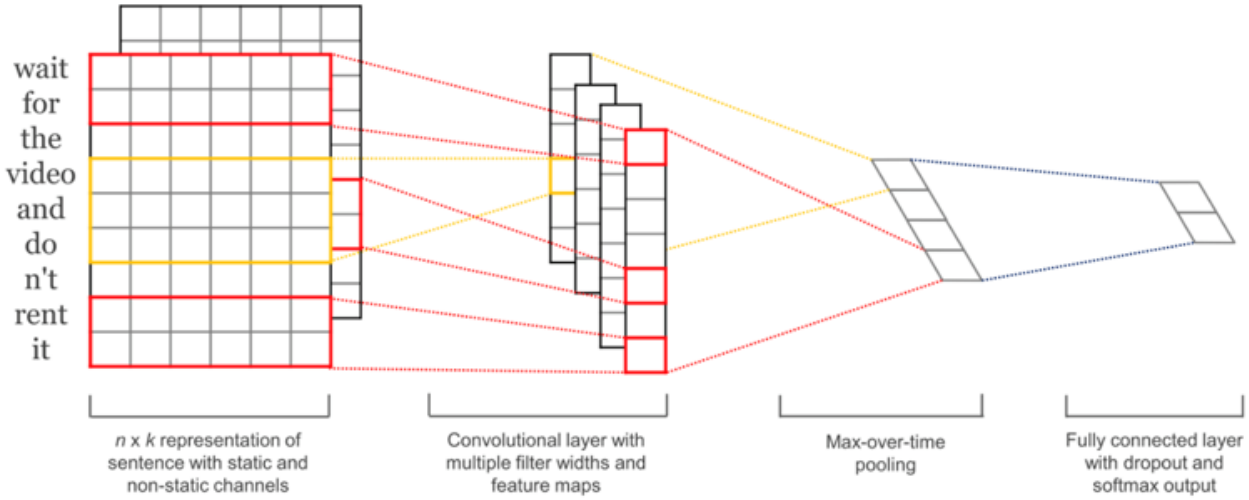
- Yoon Kim - Convolutional Neural Networks for Sentence Classification [1]
- Implementing a CNN for Text Classification in Tensorflow [2]
- yoonkim - CNN_sentence [3]
- dennybritz - cnn-text_classification-tf [4]

기존 연구는 문장의 단어들을 table lookup을 이용해 임베딩한 후 CNN 모델의 input으로 사용하였으나 본 과제에서는 CNN으로 Character-level word embeding을 수행한다. 그리고 기존 연구 모델에 더해서 RNN 기반 모델을 설계하여 기존 연구와 두 개의 모델간의 성능을 비교한다.

모델 설명

기존 연구

Yoon Kim (2014) 의 아키텍쳐



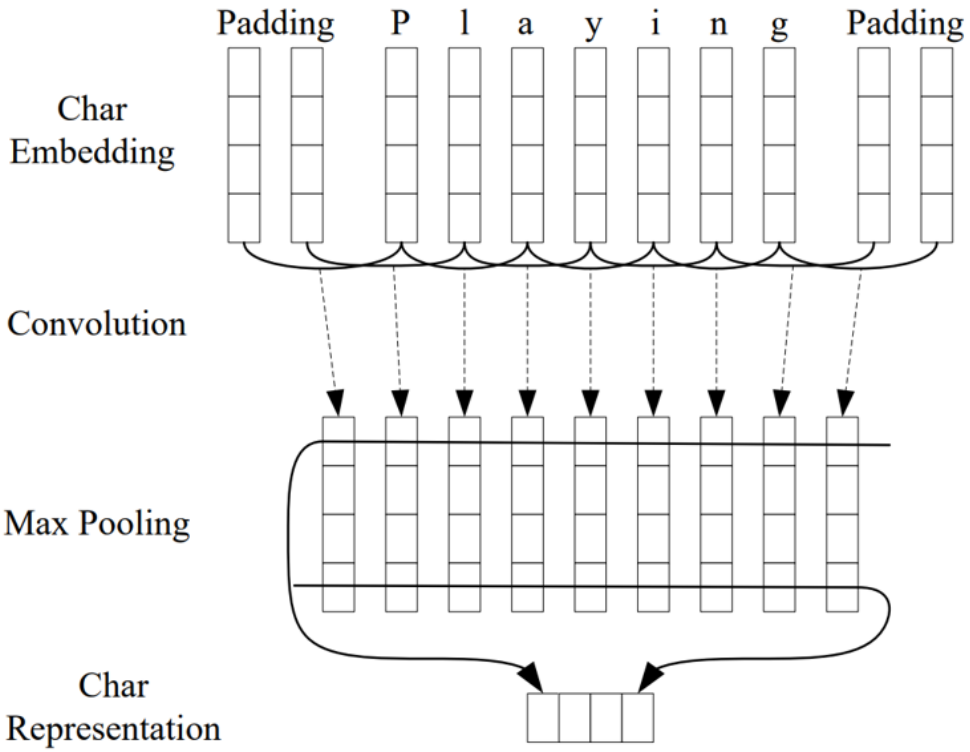
첫 번째 레이어는 단어를 저차원 벡터로 임베딩한다. 다음 레이어는 여러 사이즈의 필터로, 임베딩 된 단어 벡터에 대해 concatenation 을 수행한다. 다음에는 그것을 맥스 풀링하고, 드롭아웃을 적용하고, 소프트맥스 레이어의 결과로 분류를 수행한다.

과제 모델

Word Embeding

논문에는 Word2Vec 벡터를 사용했으나 본 과제에서는 Character-level embeding을 수행하여 얻은 벡터를 사용한다. Character-level embeding에는 CNN을 이용한다. 전반적인 성능 향상, 특히 Out-Of-Vocabulary 단어에 대하여 큰 성능 향상을 기대할 수 있다.

Character-level Feature - CNN



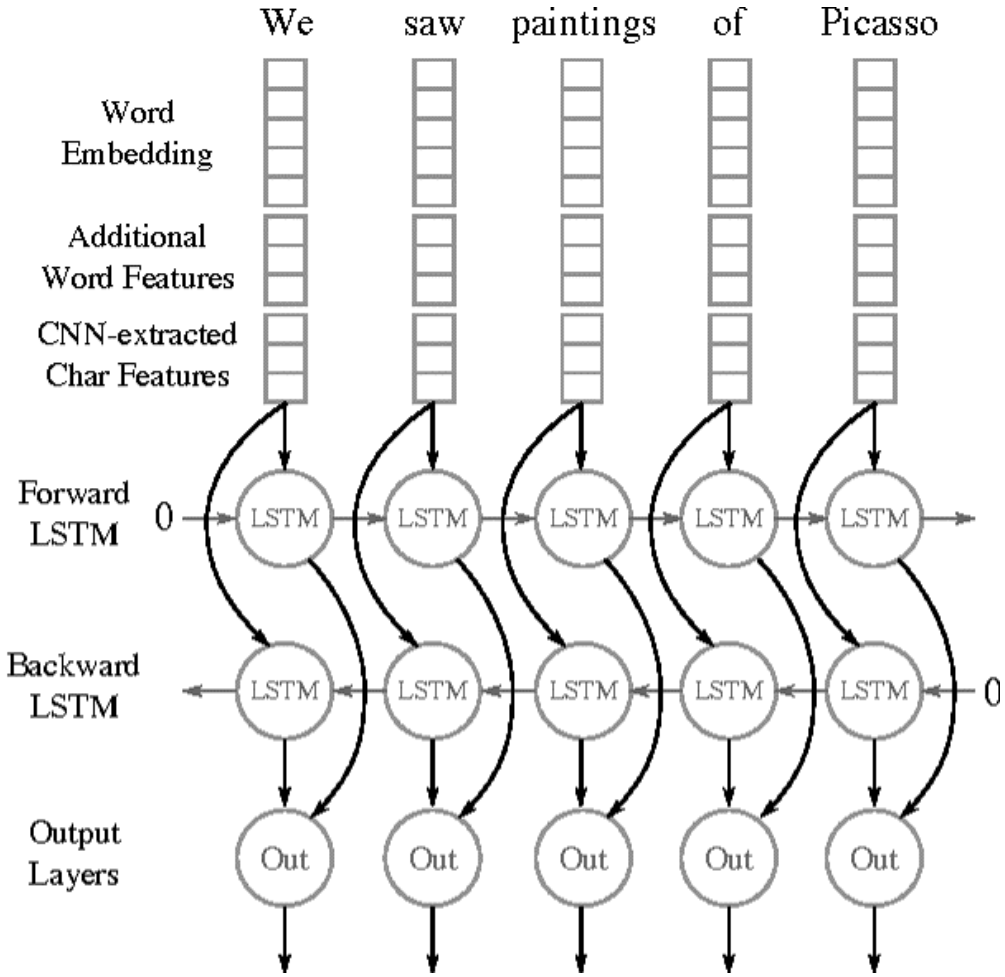
단어의 모든 문자들을 CNN 레이어의 input으로 넣고 output을 맥스풀링한 결과를 취한다.

Core Layer

CNN for Sentence Classification

위에서 설명한 기존 연구의 모델과 동일하다.

RNN for Sentence Classification



Character-level embeding 결과 벡터를 Bi-directional LSTM의 input으로 활용하여 그 output으로 분류를 수행한다. LSTM의 output은 3차원 텐서([pos, neg, neut]) 이며, 가장 값이 큰 요소를 결과로서 출력한다.

Hyper Parameters

실험 설계

평가 지표

Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	TP (True Positive)	FN (False Negative)
	Negative	FP (False Positive)	TN (True Negative)

1. Accuracy
    - 모델이 샘플을 정확히 분류한 비율.
    -
$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

2. Precision
    - 모델이 positive로 분류한 샘플 중 실제로 positive인 비율.
    -
$$Precision = \frac{TP}{TP + FP}$$

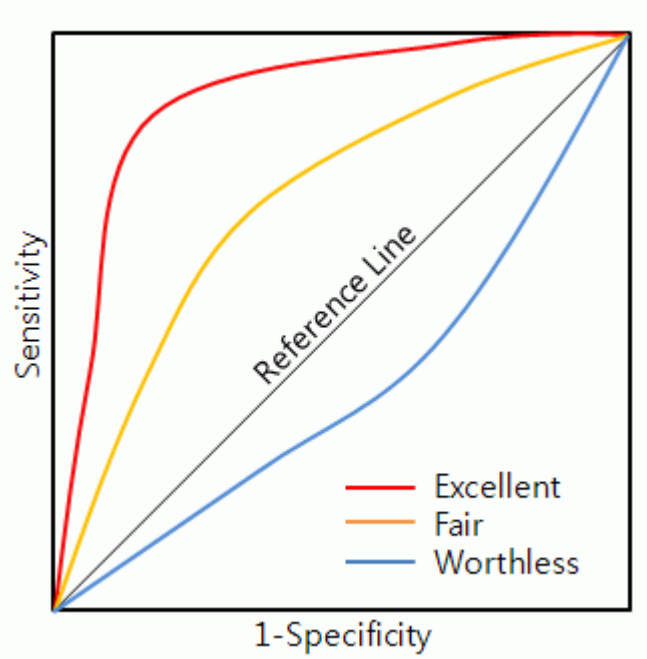
3. Recall
    - positive 샘플 중 모델이 positive로 분류한 비율.
    -
$$Recall = \frac{TP}{TP + FN}$$

4. Fallout
    - negative 샘플 중 모델이 positive로 잘못 분류한 비율.
    -
$$Fallout = \frac{FP}{TN + FP}$$

5. F1 score
    - Precision과 Recall의 조화 평균.
    -

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

6. ROC (Receiver Operating Characteristic) Curve
    - Fallout과 Recall의 변화를 시각화한 것.
    - Recall이 크고, Fallout이 작은 모형을 좋은 모형으로 생각할 수 있다.
    - 곡선이 왼쪽 위 모서리에 가까울 수록 모델 성능이 좋다.



7. AUC (Area Under the Curve)
    - ROC Curve의 밑면적을 계산한 값.
    - Fallout 대비 Recall값이 클 수록 AUC가 1에 가까운 값이며 우수한 모형이다.

실험 결과 및 분석

결론

레퍼런스

- Yoon Kim - Convolutional Neural Networks for Sentence Classification [1]
- Implementing a CNN for Text Classification in Tensorflow [2]
- yoonkim - CNN_sentence [3]
- dennybritz - cnn-text_classification-tf [4]

- 분류모델 성능 평가 지표[5]

---

1. http://www.aclweb.org/anthology/D14-1181↩↩
2. http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/↩↩
3. https://github.com/yoonkim/CNN_sentence↩↩
4. https://github.com/dennybritz/cnn-text-classification-tf↩↩
5. http://here.deepplus.co.kr/?p=24↩