Noisy 텍스트 임베딩을 이용한 한국어 감정 분석

이현영○, 강승식
국민대학교 컴퓨터공학과
le32146@gmail.com, sskang@kookmin.ac.kr

Korean Sentiment Analysis by using Noisy Text Embedding

Hyun-Young Lee○, Seung-Shik Kang
Dept. of Computer Science, Kookmin University

요 약

신문기사나 위키피디아와 같이 정보를 전달하는 텍스트와는 달리 사람의 감정 및 의도를 표현하는 텍스트는 다양한 형태의 노이즈를 포함한다. 본 논문에서는 data-driven 방법을 이용하여 노이즈와 단어들 사이의 관계를 LSTM을 이용하여 하나의 벡터로 요약하는 모델을 제안한다. 노이즈 문장 벡터를 표현하는 방식으로는 단방향 LSTM 인코더과 양방향 LSTM 인코더의 두 가지 모델을 이용하여 노이즈를 포함하는 영화 리뷰 데이터를 가지고 감정 분석 실험을 하였고, 실험 결과 단방향 LSTM 인코더보다 양방향 LSTM인코더가 우수한 성능을 보여주었다.

주제어: 노이즈 문장 벡터, LSTM, 감정 분석

1. 서론

분산 표현(distributed representation) 또는 임베딩(embedding)은 희소한 차원 문제를 해결하고 유사한 의미를 가진 단어를 저차원의 벡터로 연속적인 벡터 공간에 표현한다[1, 2]. 신경망 기반의 자연어 처리 모델에서 단어 임베딩은 입력으로 사용되는 만큼 임베딩의 성능은 모델에 성능에 영향을 미친다[3]. 기존의 임베딩은 신문기사나 위키피디아(wikipedia)와 같이 글의 의도를 명확히 전달하기 위해 쓰여진 노이즈가 제거된 말뭉치를 기반으로 학습한다[3,4,5,6,7].

감정 분석은 자연어 처리의 텍스트 분류문제로서 텍스트에 포함된 감정의 극성을 분석하는 기술이다. 기존의 감정 분석은 Naive Bayes, maximum entropy, 그리고 support vector machine(SVM)과 같은 전통적인 기계 학습 방법을 많이 사용하였다[8]. 전통적인 기계 학습 방법은 연구자가 데이터에서 특징을 추출한다. 하지만 최근 신경망을 이용한 모델들은 데이터로부터 특징을 스스로 추출하고 우수한 성능을 보여주고 있다[3].

하지만 인터넷 댓글, 영화 리뷰, 상품 리뷰, 그리고 tweet과 같은 텍스트 감정 분석의 경우에는 사용자들은 자신의 의도와 감정을 표현하기 위해 이모티콘, 축약어, 오타, 띄어쓰기 오류, 그리고 문장 부호(.,!)와 같은 노이즈를 포함하는 텍스트를 사용한다. 그리고 영어와 달리 교착어인 한국어의 경우에는 형태학적으로 단어를 표현하는 방식이 풍부하기 때문에 영화를 평가할 때 자소 단위로도 리뷰어들는 영화에 대한 의도나 감정을 표현한다[3]. 본 논문에서는 영화평과 같은 noisy 문장에 포함된 노이즈와 단어들의 사이의 문맥적인 정보를 하나의 벡터로 요약하는 모델을 제안한다.

2. 관련 연구

단어를 연속적인 벡터 공간에 표현하는 이웃하는 단어들이 함께 출현하는 정보를 이용하여 단어를 연속적인 벡터 공간에 표현한다[4,5,6]. 하지만 이러한 정보는 사전에 없는 단어를 다루는데 어려움이 발생하는 out-of-vocabulary 문제가 발생한다. 그리고 단어를 구성하는 sub-character를 이용하여 out-of-vocabulary를 해결하는 방법이 제안되었다[7]. [3]은 한국어의 단어를 구성하는 문자(character)와 자소 단위의 정보를 포함하여 단어 벡터를 표현한다.

하나의 문장을 하나의 벡터로 요약하는 RNN 기반의 번역 모델에서 소스 문장(source sentence)을 RNN 인코더의 입력으로 사용하여 하나의 벡터로 요약을 하고 타겟 문장(target sentence)의 생성을 위해 RNN의 디코더를 사용한다[9,10,11]. 본 논문에서는 LSTM 기반의 입력 문장의 문맥정보를 하나의 벡터로 요약하는 모델을 제안한다.

3. Noisy 텍스트 임베딩

Recurrent neural network(RNN) 기반의 모델은 sequence-to-sequence 번역 모델에서 인코더와 디코더를 구성한다[9,10,11]. 이때 인코더는 입력 문장을 문맥정보를 하나의 벡터로 요약하고 디코더는 하나로 요약된 벡터로부터 입력 문장에 대응되는 출력 문장을 생성한다. 본 논문에서는 번역의 인코더와 같이 RNN의 장기 의존성의 문제를 해결한 LSTM을 이용하여 140자로 구성된 영화평을 하나의 벡터로 요약한 노이즈 문장 벡터를 연속적인 벡터 공간에 표현한다[12,13].

Long short-term memory(LSTM)은 장기 메모리 c_t과 단기 메모리 h_t을 아래의 식과 같이 input 게이트 i_t, forget 게이트 f_t, 그리고 output 게이트 o_t와 상호작용

을 하면서 입력 문장을 하나의 벡터 h_t로 요약한다.

$$f_t = \sigma(w_f x_t + u_f h_{t-1} + b_f)$$
$$i_t = \sigma(w_i x_t + u_i h_{t-1} + b_i)$$
$$o_t = \sigma(w_o x_t + u_o h_{t-1} + b_o)$$
$$\tilde{c_t} = \tanh(w_{\tilde{c}} x_t + u_{\tilde{c}} h_{t-1} + b_{\tilde{c}})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$
$$h_t = o_t \odot \tanh(c_t)$$

여기서, σ는 sigmoid 함수, $\odot$는 element-wise 곱셈을 의미한다. $w_{\{i,f,o\}}$, $u_{\{i,f,o\}}$, 그리고 $b_{\{i,f,o\}}$ 는 LSTM의 파라미터이고 역전파에 의해 학습된다. 본 논문에서는 두 가지 방식의 노이즈 문장 벡터의 임베딩 방법을 제안한다. 3.1에서는 단방향 LSTM 인코더를 이용한 노이즈 문장 벡터를 설명하고 3.2에서는 양방향 LSTM 인코더를 이용한 노이즈 문장 벡터를 설명한다.

3-1. 단방향 LSTM 인코더를 이용한 노이즈 텍스트 임베딩

단방향 LSTM은 장기 메모리 c_t과 단기 메모리 h_t를 이용하여 과거와 현재의 의존성 정보를 하나의 벡터 h_t로 요약한다. 본 논문에서 제안한 noisy 텍스트 임베딩을 위한 단방향 LSTM 인코더는 영화평을 하나의 벡터로 표현하기 위해 이모티콘, 자소 단위의 감정 표현 텍스트와 같은 노이즈를 포함한 문장을 unigram 문자열로 보고 단방향 LSTM의 입력값으로 사용한다. 그리고 노이즈와 단어들 사이의 문맥적인 정보를 하나의 벡터로 요약한다.

그림 1과 같이 단방향 LSTM 인코더 모델은 노이즈를 함하는 입력 문장의 unigram 문자열을 단방향 LSTM 입력으로 한다. 예를 들어, “ㄴㄴ 영화가 맘에 들어 ^^”라는 입력 문장은 공백 문자를 “<SPACE>”로 대체하여 [“ㄴ”, “ㄴ”, “<SPACE>”, “영”, “화”, “가”, “<SPACE>”, “맘”, “에”, “<SPACE>”, “들”, “어”, “^”, “^”]을 단방향 LSTM 인코더의 입력으로 사용한다. 단어를 자소 단위로 나눈 표현 “ㄴㄴ(나는)”, 준말 “맘(마음)”, 그리고 이모티콘 “^^” 와 같은 노이즈를 포함하는 문장의 문맥정보를 단방향 LSTM을 이용하여 하나의 벡터로 표현한 노이즈 문장 벡터를 출력한다. 이 노이즈 문장 벡터는 활성화 함수가 없는 단층 전방향 신경망과 softmax 층을 거쳐서 노이즈를 포함한 영화평이 긍정인지 부정인 감정 분류를 한다.

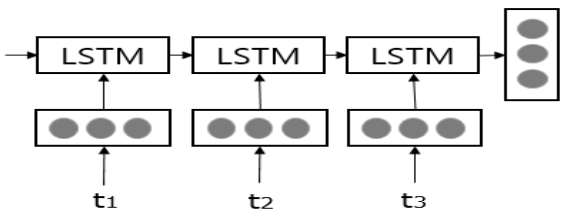

그림 1. 단방향 LSTM 인코더 모델 구조

3-1. 양방향 LSTM 인코더를 이용한 노이즈 텍스트 임베딩

양방향 LSTM은 정방향 LSTM과 역방향 LSTM으로 구성되어 있다. 정방향 LSTM은 하나의 열을 LSTM은 과거에서 현재의 순으로 입력받아 과거와 현재의 의존성 정보를 하나의 벡터로 요약한다. 그리고 역방향 LSTM은 정방향 LSTM과 역순으로 하나 열을 입력받아 미래와 현재의 의존성 정보를 하나의 벡터로 요약한다. 본 논문에서 제안하는 양방향 LSTM 인코더는 노이즈를 포함한 영화평의 unigram 문자열을 그림 2와 같이 양방향 LSTM 입력값으로 사용한다. 정방향 LSTM은 입력 문장의 과거와 현재의 의존성 정보를 하나의 벡터로 요약하고, 역방향 LSTM은 현재와 미래의 의존성 정보를 하나의 벡터로 요약한다. 그리고 정방향 LSTM과 양방향 LSTM이 출력한 각각의 특징 벡터를 더하거나 이어붙이는 연산을 이용하여 과거, 현재, 미래의 의존성 정보를 하나의 벡터로 요약한 노이즈 문장 벡터를 연속적인 벡터 공간에 표현한다. 이 노이즈 문장 벡터는 활성화 함수가 없는 단층 전방향 신경망과 softmax를 거쳐서 노이즈를 포함하고 있는 영화평이 긍정 또는 부정인지 감정 분류한다.

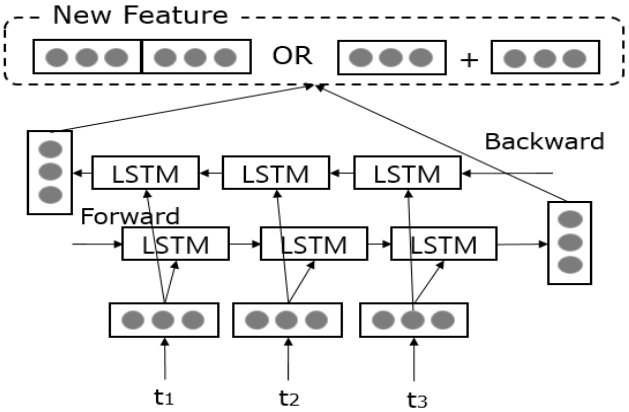

그림 2. 양방향 LSTM 인코더 모델 구조

3-1. Sentiment Classification

본 논문에서 제안한 Multi-sequence 단양향 LSTM 인코더와 양방향 LSTM 인코더의 학습을 위해 아래의 식과 같이 softmax 함수는 노이즈 문장 벡터를 긍정인지 부정인지에 대한 정규화된 확률 $p(\tilde{y}_n)$을 계산한다. 이를 통하여 정답 레이블의 네거티브 로그 가능도(negative log-likelihood)를 최소화하는 방향으로 모델을 학습한다.

$$p(\tilde{y}_n) = softmax(s_t) = \frac{\exp(s_t)}{\sum_{k=1}^{n} \exp(s_k)}$$

표 1. 모델 성능

Model			Acc.	Positive			Negative		
				Pre.	Rec.	F1	Pre.	Rec.	F1
단방향 LSTM	word(baseline)		78.420	78.729	78.288	78.508	78.111	78.554	78.332
	unigram		84.075	84.817	83.274	84.038	83.349	84.886	84.110
양방향 LSTM	word (baseline)	add	78.408	79.271	77.335	78.291	77.575	79.497	78.524
		concat	78.592	79.000	78.288	78.643	78.186	78.901	78.542
	unigram	add	**84.629**	**86.137**	82.793	84.432	83.215	**86.489**	**84.820**
		concat	84.363	84.341	**84.656**	**84.499**	**84.384**	84.065	84.224

$$Loss = -\frac{1}{N}\sum_{k=1}^{n}\ln p(\tilde{y_k})$$

위의 식에서 s_t는 전방향 신경망을 거친 긍정과 부정의 정규화 되지 않은 로그 확률이다. t는 입력 영화평의 레이블이다.

4. 실험 및 결과

우리의 모델의 실험을 위해 노이즈 텍스트를 포함하고 있는 네이버 영화 리뷰 말뭉치(v1.0)[1]을 이용하여 모델을 학습하고 성능 평가를 하였다. 이 말뭉치는 긍정과 부정의 영화평으로 구분이 되어 있다. 긍정과 부정의 영화평을 균형적으로 분포되어 있다. 하나의 영화평의 경우 140자 문자로 구성되어 있다. 그리고 학습 15만과 평가 5만의 영화평으로 구성된다. 자소 단위의 표현 “ㄹㅇ”, “ㅋㅋㅋ”, 이모티콘 “^^”, “ㅡ0ㅡ”, 그리고 문장 부호(!) 등을 통해 표 2과 같이 영화에 대한 리뷰어의 감정 및 의도 표현하고 있다.

표 2. 네이버 영화평 예시

영화평	감정
언제봐도 좋아좋아..^^	긍정
정말 따뜻했던 드라마ㅠㅠ	긍정
감명깊고 재밌다.굿!!!!!!!!!!!!!	긍정
넘 재밌다.ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ	긍정
총격전 두세발이 전부네 ㅋㅋㅋㅋ	부정
ㄹㅇ 노-잼 절대 보지마라	부정
완전 재밌음^^ 기대이상이였음 1점!!	부정
배트맨+스파이더맨=할람 포 ... 암튼 지루해 지루해 ...ㅡ0ㅡ	부정

노이즈 문장 벡터의 성능 평가를 위해 긍정과 부정으로 분류하는 신경망은 활성화 함수가 없는 단층 전방향 신경망(feedforward neural network)로 구성하였다. 그리고 단방향 LSTM과 양방향 LSTM의 층수는 단층으로 구성하였다. LSTM에 의한 노이즈 문장 벡터에 긍정과 부정으로 분류하기 모델의 하이퍼 파라미터는 음절 벡터 차원 수와 LSTM의 유닛 수는 300, 학습률 0.001, 그리고 배치 크기 1로 설정하였으며 경사 하강법으로 모델을 학습한다. 우리의 실험 모델을 다음과 같다.

표 3. 영화평 감성 분석 성능 비교

Model			Acc.
Park[3]	SG		76.15
	SISG(Ch)		76.26
	SISG(jm)		76.53
	SISG(ch4+jm)		76.28
	SISG(ch6+jm)		76.54
단방향 LSTM	word(baseline)		78.420
	unigram		84.075
양방향 LSTM	word (baseline)	add	78.408
		concat	78.592
	unigram	add	**84.629**
		concat	84.363

- Park (2018): 노이즈가 없는 말뭉치에서 학습된 벡터와 LSTM 기반의 감정 분류기에서 입력 벡터를 벡터를 skip-gram[5], 문자 n-gram를 사용한 경우(SISG(ch)), 자모 n-gram를 사용한 경우 (SISG(jm)),문자 n-gram와 자모 n-gram를 함께 사용한 경우(SISG(ch4+jm), SISG(ch6+jm)) 에 대하여 우리의 실험과 같은 네이버 영화평에 대하여 감정 분류 실험을 하였다.

- 어절 기반 단방향 LSTM과 양방향 LSTM (baseline): 노이즈를 포함하고 있는 네이버 영화평 문장을 어절 단위로 단방향과 양방향 LSTM의 입력으로 사용하였다. 양방향 LSTM의 경우에는 정방향 LSTM과 역방향 LSTM의 입력을 고정된 사이즈로 출력한 벡터의 결합 방법을 두 가지(add, concat)로 나누어 실험하였다.

- unigram 문자열 기반 단방향 LSTM과 양방향 LSTM: 어절 기반 모델과는 달리 네이버 영화평 문장을 unigram

1) https://github.com/e9t/nsmc

문자열을 단방향과 양방향 LSTM의 입력으로 사용하였다. 양방향 LSTM의 경우 어절 기반처럼 정방향 LSTM과 역방향 LSTM의 출력된 특징 벡터를 더하거나 이어붙이는 형태의 두 가지 방법을 실험하였다.

표 1는 본 논문에서 제안한 노이즈 문장 벡터 정략적인 성능 평가를 위해서 감정 분류의 정확도(accuracy)와 감정별 분류 성능을 측정하기 위해 긍정과 부정에 대하여 precision, recall, f1을 사용하였다. 표 1에서 볼 수 있듯이 그 결과 노이즈와 단어들 사이의 정보를 포함한 노이즈 문장 벡터 분류 성능에서는 단방향 LSTM보다 양방향 LSTM이 우수함을 보여주었다. 그리고 단방향 LSTM과 양방향 LSTM의 두 개 모델에서도 어절 기반보단 unigram 문자열을 입력으로 사용한 경우가 더 우수함을 보여주었다.

본 논문에서 제안한 노이즈 문장 벡터를 출력하는 단방향 LSTM은 어절 기반 모델도 [3]에서의 제안한 모델보다 우수함을 보여주었다. 그리고 단방향 LSTM에서 입력을 unigram 문자열로 하는 경우에는 약 7.5% 향상된 결과를 보여주었다.

5. 결론

영화평에서는 자소 단위의 단어 표현, 이모티 콘, 줄임말과 같은 표현도 사용자의 의도 및 감정을 표현하는 수단으로 사용할 수 있음을 확인하였고, 이러한 노이즈와 단어들 사이의 의존성을 하나의 벡터로 표현하는 LSTM을 이용하여 노이즈 문장 벡터를 표현하는 방법을 제안하였다. 노이즈를 없는 말뭉치에서 학습된 벡터를 사용한 경우보다 노이즈 정보를 사용한 경우가 우수함을 보여주었다. 영화평에서 사용된 노이즈는 사람의 주관을 파악하는데 중요한 요소일 수 있다는 것을 확인하였다.

감사의 글

이 논문은 2017 년 정부 (과학기술정보통신부) 의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2017M3C4A7068186)

참고문헌

[1] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, "Natural language processing (almost) from scratch", Journal of machine learning research, vol 12, pp. 2493-253, 2011

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin, "A neural probabilistic language model", Journal of machine learning research, vol. 3, 1137-1155, 2003.

[3] Sungjoon Park, Jeongmin Byun, Sion Baek, Yongseok Cho, and Alice Oh. "Subword-level Word Vector Representations for Korean", In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 2429-2438, Melbourne, Australia, Jul. 2018.

[4] Tomas Mikolov, Ilya Sutskever,Kai Chen, Greg Corrado, and Jeffrey Dean, "Distributed Representations of Words and Phrases and their Compositionality", In NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems, vol 2, pp. 3111-3119, Lake Tahoe, Nevada, Dec. 2013

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space", arXiv preprint arXiv:1301.3781. 2013.

[6] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, "GloVe: Global Vectors for Word Representation", In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532-1543, Doha, Qatar, Oct. 2014.

[7] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, "Enriching Word Vectors with Subword Information", Transactions of the Association for Computational Linguistics, vol. 5, pp. 135-146, 2017

[8] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques", In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79-86, Philadelphia, Jul 2002.

[9] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks", In Advances in neural information processing systems. pp. 3104-3112, 2014.

[10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473, 2014.

[11] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025, 2015.

[12] Sepp Hochreiter, and Jürgen Schmidhuber. "Long short-term memory", Neural computation, vol. 9, Issue 8, pp. 1735-1780, 1997.

[13] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization", arXiv preprint arXiv:1409.2329, 2014.