CNN을 이용한 발화 주제 다중 분류

최경호 ○, 김경덕, 김용희, 강인호
Naver RND center, Clova Dialogue, Naver RND center, Clova NLP
{k.h.choi, kyungduk.kim, yong.hee.kim.0402, once.ihkang}@navercorp.com

Multi-labeled Domain Detection Using CNN

Kyoungho Choi ○, Kyungduk Kim, Yonghe Kim, Inho Kang
Naver RND center, Clova Dialogue, Naver RND center, Clova NLP

요 약

CNN(Convolutional Neural Network)을 이용하여 발화 주제 다중 분류 task를 multi-labeling 방법과, cluster 방법을 이용하여 수행하고, 각 방법론에 MSE(Mean Square Error), softmax cross-entropy, sigmoid cross-entropy를 적용하여 성능을 평가하였다. Network는 음절 단위로 tokenize하고, 품사정보를 각 token의 추가한 sequence와, Naver DB를 통하여 얻은 named entity 정보를 입력으로 사용한다. 실험결과 cluster 방법으로 문제를 변형하고, sigmoid를 output layer의 activation function으로 사용하고 cross entropy cost function을 이용하여 network를 학습시켰을 때 F1 0.9873으로 가장 좋은 성능을 보였다.

주제어: Multi-label classification, Domain detection

1. 서론

최근 딥러닝으로 인한 음성인식기술과 자연언어 처리 기술의 발달로 인해 가상 비서, 인공지능 스피커 등의 다양한 형태로 대화형 인터페이스 기술이 본격적으로 서비스에 적용되고 있다.

대화형 인터페이스의 핵심인 대화시스템의 목적은 별도의 정해진 명령어나, 정해진키워드 없이 자연언어 형태의 입력을 받아, 사용자의 요청을 이해하고, 해당 요청을 수행하거나, 사용자의 발화에 응답하는 것이다. 대화시스템은 자연언어 형태의 사용자 발화를 시스템이 이해할 수 있는 형태의 명령으로 번역하는 기술을 필요로 하며, 이를 자연언어 이해(Natural Langue Understanding)라 한다.

일반적으로 대화시스템을 이용하는 사용자의 발화는 주제에 따라 다른 어휘적, 언어적 이해를 필요로 할 수 있다. 예를 들어 동명이인이 발화에 등장하였을 경우 해당 발화의 주제를 확인 할 수 있다면, 실제 지칭하는 대상이 될 수 있은 인물들의 폭을 좁힐 수 있다. 때문에 발화의 주제를 찾아내는 domain detection을 자연언어 이해 과정 전반부에서 수행하고, 해당 task 정확도를 신뢰할 수준으로 높일 수 있다면, 대화시스템에 요구되는 자연언어 이해 과정의 정확성을 향상시킬 수 있다.

사용자의 발화는 여러 주제를 동시에 포괄할 수 있으며, 또한 주제가 모호하여 특정 주제라 확정하기 어려울 수 있다. 예를 들어 “내일은 어때?” 라는 발화가 주어졌을 때 해당 발화만으로 시스템이 “weather”, “schedule” 중에 주제를 확정하기 어렵다. 이러한 경우 한 발화를 하나의 주제로만 분류하지 않고, 여러 개의 주제로 다중 분류 할 수 있다.

그러나 전통적으로 연구되었던 CRF(Conditional Random Fields), SVM(Support Vector Machine)모델들은 다중 분류 task에 그대로 사용하기 어렵다[1]. 또한 모델의 입출력을 자유롭게 설계할 수 있는 deep neural network를 사용할 경우에도, 적절한 학습 방법과, cost function을 사용하지 않는다면, 시스템의 성능을 보장할 수 없다.

본 논문에서는 발화 주제 다중 분류 task를 CNN(Convolutional Neural Network)을 이용하여 수행하여 보고, 발화 주제 분류 task를 수행할 때 multi-label task를 다루기 위한 방법론과 cost function에 따른 성능을 비교한다.

2. 관련 연구

2.1 Domain detection

기존의 domain detection 연구는 단일 domain 분류를 중심으로 수행되었다. 또 발화를 연구의 목적으로 한 domain보다, 웹 상의 text를 대상으로 한 topic detection(topic modeling)이 주로 연구되어 왔다.

최근 연구 에서는 주제 간의 계층을 자동으로 구축하여, 주제 다중 분류 문제를 해결하기도 하였다[2].

2.2 Multi-label Classification

전통적으로 사용되는 기계학습 모델들은 입력 데이터를 하나의 class로 분류하는 모델이 일반적이다. 때문에 다중 분류 문제를 해결하기 위해서는 기존 기계학습 모델을 다중 분류를 가능하도록 수정하여 사용하거나, class들을 cluster로 만들어 모델이 단일 cluster로 분류하도

록 task를 수정하여 수행 할 수 있다. 전자의 경우, 다중 분류가 가능하도록 수정 될 수 있는 기계학습 모델이 많지 않을 뿐만 아니라, 해당 모델을 학습하는 전략에 있어, 분석과 계획이 필요하다. 후자의 경우, cluster가 일정하게 형성될 보장이 없는 task이거나, class들이 다양한 조합으로 cluster될 수 있는 경우, 모델이 필요한 계산 량이 커질 수 있다.

Neural network의 경우 모델의 output layer를 수정하여 multi-label 분류에 사용할 수 있다. 이때 cost function과, activation function, 학습 전략 등에 의해 모델의 성능이 크게 달라질 수 있다.

3. Convolutional neural network for domain detection

본 연구에서는 multi-label classification task에 적합한 방법과, 해당 방법과 함께 사용하였을 때 우수한 성능을 보이는 cost function을 확인하기 위하여, Convolutional Neural Network를 사용하였다.

3.1 Network

연구에 사용된 모델은 [3]에서 소개된 모델을 기반으로 하여 입력으로 추가적인 입력을 사용할 수 있도록 그림1과 같이 수정한 모델로, 2개의 projection layer를 이용하여, 두 종류의 토큰을 입력으로 사용한다. 하나는 lexicon이며 음절을 그대로 사용할 경우 음절 자체가 갖는 중의성을 줄이기 위하여, 음절 단위로 품사를 부착한 lexicon을 사용하였고, 다른 하나는 형태소 단위로 사전을 조회하여 얻은 type 정보를 첫번째 입력과 align 하고 BIO tag를 추가하여 만든 feature이다[4]. 이 feature는 lexicon만으로는 알 수 없는 외부 지식을 네트워크가 학습 할 수 있게 한다.

각 입력은 서로 다른 각각의 projection layer를 이용하여 vector representation형태로 만든다. 이때 feature 입력은 time-step 당 여러 개의 label이 입력으로 사용 될 수 있으며, 여러 개의 label이 입력으로 사용되는 경우 각 입력을 projection 하고, element-wise sum하여 vector representation을 만든다.

그림1은 “알루미늄 틀어줘”를 network의 입력으로 사용한 예시이다. 입력 받은 문장의 품사를 분석하여 “알/PROPER”과 같이 각 음절에 “/”을 경계로 부착한다. 음절과 품사가 부착된 형태의 token을 one-hot representation으로 하여 projection한다. 입력 문장을 각 DB에 조회하면, “알루미늄=[singer, song, metal]”, “루미=[singer]”, “미=[singer]”, “틀=[singer, song]”, “어=[song]”를 확인 할 수 있다. 이를 음절 단위로 분리하여, BIO tag를 추가하여, 각 lexicon에 일치하는 feature로 사용한다

두 vector representation을 concatenate하여, 3, 4, 5 개의 window를 갖는 convolution layer의 입력으로 사용한다. 세 convolution layer의 출력을 max-over-time polling 하고, concatenate 하여 dropout을 적용하고, output layer의 입력으로 사용한다. Output layer는 cost function 에 따른 성능평가를 위해 별도의 activation function을 사용하지 않는다.

각 cost function을 평가할 때 cluster 방법을 사용한 경우에는 Network의 출력 값을 argmax 하여 사용하였고, multi-label 방법은 threshold(0.5)를 넘긴 출력들만 사용하여 평가하였다.

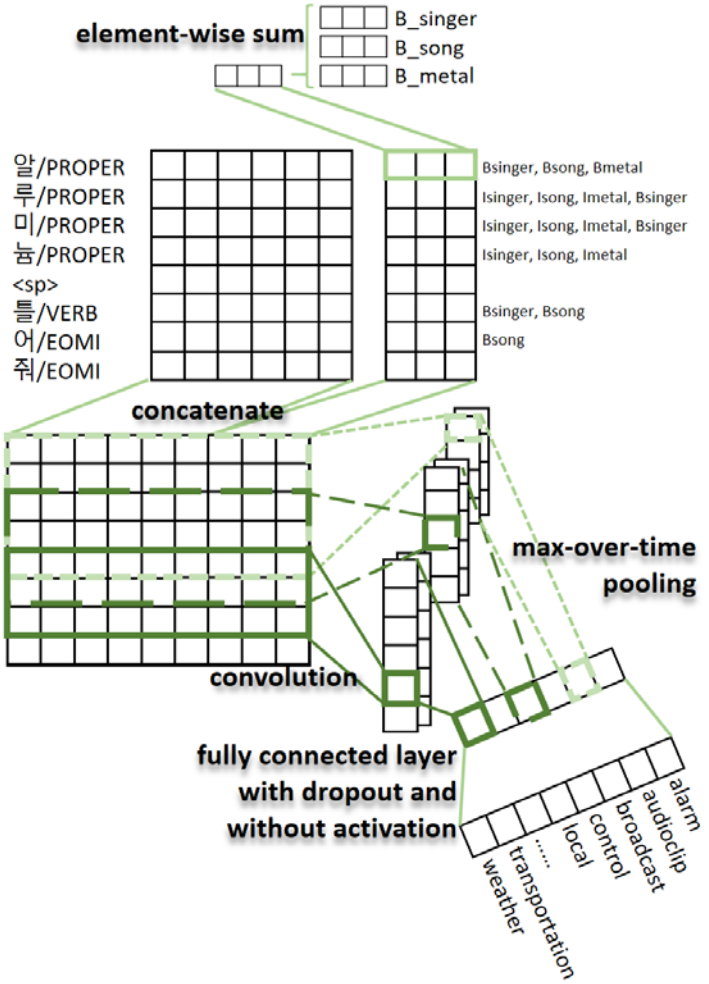

그림1. Model 구조

3.2 Cluster method

단 하나의 발화 만을 분석하여 주제를 추론할 때에 해당 발화의 주제를 명료하게 하나로 추론하기 어려울 수 있다. 이때 여러 개의 발화 주제를 동시에 해당 발화의 주제로 삼을 수 있다.

“매주 목요일 마다 알려줘”라는 발화가 있다면, 해당 발화의 주제는 해당 발화의 앞선 발화에 따라, alarm, control, reminder, memo모두 일 수 있다. 이때 이를 $cycletime 이라는 별도의 class를 만들어, 분류하는 방법을 cluster 방법이라 정의한다. Cluster 방법을 사용하였을 경우 앞서 그림1로 나타낸 network의 출력은 한 개한 class 만 가질 수 있도록 하며, 해당 network를 학습하고 평가할 때 사용할 데이터의 정답도 하나의 class 만 가질 수 있도록 작성한다.

Multi-label 방법론을 사용할 경우 그림1의 network의

출력은 임계값을 넘긴 여러 개의 class를 동시에 가질 수 있으며, 사용하는 데이터에서 발화 주제 또한 동시에 여러 개를 하나의 발화에 기록하여 사용한다.

3.3 Cost functions

본 연구에서는 총 세가지 cost function을 비교하여 보았다. Cross entropy cost function은 multi-class classification에 주로 쓰이는 함수로 다음 그림2의 수식으로 나타낼 수 있다. 그림2의 두 수식은 각각 앞서 소개한 network의 마지막 activation function을 포함하고 있으며, $\hat{y}$는 network의 출력을 뜻한다. L_{CE}는 network 마지막 layer의 출력($\hat{y}$)에 sigmoid 를 적용한 cross entropy cost이고, L_{SCE} 는 softmax 를 적용한 cross entropy cost이다. $\mathrm{sigmoid}(\hat{y})$ 는 각각의 output unit에 해당 class에 대한 0부터 1사이의 예측 값을 갖지만, $\mathrm{softmax}(\hat{y})$는 모든 class들 중 각 class가 예측 될 확률 값을 갖기 때문에($\sum_i \mathrm{softmax}(\hat{y}_i) = 1$), threshold를 이용한 multi-label task에 $\mathrm{L_{SCE}}(y, \hat{y})$는 사용 할 수 없다.

$$L_{\mathrm{CE}}(y, \hat{y}) = -\log\big(\mathrm{sigmoid}(\hat{y})\big)\, y - \log\big(1 - \mathrm{sigmoid}(\hat{y})\big)\,(1 - y)$$

$$\mathrm{L_{SCE}}(y, \hat{y}) = -\log\big(\mathrm{softmax}(\hat{y})\big)\, y - \log\big(1 - \mathrm{softmax}(\hat{y})\big)\,(1 - y)$$

그림2. Cross entropy cost functions

실험에 사용한 MSE(mean squared error)는 그림3의 수식으로 나타내었다.

$$\mathrm{L_{MSE}}(y, \hat{y}) = -(y - \mathrm{sigmoid}(\hat{y}))^2$$

그림3. Mean squared error

4. Data set

본 연구에서 사용한 말뭉치는 사내에서 다수의 연구원이 구축한 구어체의 발화 말뭉치로, 각 발화에 대해 적절한 도메인이 모두 기록되어 있는 multi-labeled 말뭉치이다. 총 14개의 도메인에 대해 33032 발화로 이루어져 있고, domain 별 발화의 개수는 다음 표1과 같다.

표1. Domain별 발화의 개수

Domain	Count	Domain	Count
alarm	2095	music	7720
audioclip	5720	news	5809
broadcast	4250	radio	5494
control	5907	reminder	593
local	1620	sports	2089
memo	848	transportation	2146
movie	2000	weather	4181

한 발화에 함께 표기된 domain들을 묶어 cluster로 만든 label별 발화의 개수는 다음 표2와 같다. 표 2에서 $media 는 music, radio, news, control, audioclip, broadcast 발화 주제를 묶어 표현한 cluster를 의미한다. $local은 local과 transportation을 동시에 표현한 cluster, $temp는 control과 weather, $cycletime은 alarm, control, reminder, memo, $domestic은 sport, news, transportation, local, weather 발화 주제를 함께 나타내는 cluster이다. 실험에 사용된 cluster들은 말뭉치 제작 단계에서 발견한 실생활에 사용 할 수 있다 판단한 발화들을 기준으로, 조합 가능한 다중 발화 주제를 산정하여 만들었다.

표2. Cluster별 발화의 개수

Domain	Count	Domain	Count
alarm	2000	reminder	498
audio clip	2000	sports	2000
broadcast	530	transportation	2000
control	2000	weather	4000
local	1474	$media	3720
memo	753	$local	57
movie	2000	$temp	92
music	4000	$cycletime	95
news	2000	$domestic	89
radio	1774		

5. 실험

입력 문장을 음절 단위로 나누고, 각 음절에 해당하는 품사 정보를 합쳐 하나의 token으로 사용하였다. Neural network의 입력으로는 token을 기준으로 별도의 pre-training을 수행하지 않은 100 차원의 word embedding과, Naver DB를 조회하여 얻은 type 정보를 16차원으로 embedding 하여 사용하였다. 데이터를 전처리 하였을 때 총 10회 미만으로 등장한 word token과, feature label은 unknown word로 치환하여 학습하였으며, 사용한 word token들의 사전 크기는 3274, feature label의 개수는 3313종이다.

Multi-label 방법과 cluster 방법에 대해 앞서 설명한 cost function들을 적용하여 실험하였다. Sigmoid-CE, MSE을 이용해서 양쪽 방법론의 성능을 측정하나, softmax-CE는 multi-label 방법에서는 사용할 수 없는 cost이므로 (3.2 cost function) Cluster 방법에 대해서만 성능을 측정하였다. 사용한 convolution layer의 window size는 각각 3, 4, 5 로 하였고, 각 convolution layer 의 filter 개수는 표3에 명시한 대로 64개와 128개씩 두어 평가하였다. CNN layer 출력에 0.5 확률의 dropout을 주었고, weight decay는 사용하지 않았다. 또한 batch size를 64로 고정하고, adam[5] 알고리즘을 이용하여 network를 학습하였으며, validation set을 바탕으로 learning rate decay를 사용하였다. 전체 발화를

8:1:1로 나누어 24644발화를 학습에 사용하고, 3219발화를 validation set으로 사용하였으며, 3219발화로 성능을 평가하였다.

시스템을 평가할 때에는 cluster 방법론을 사용하여 예측하더라도, 해당 cluster에 포함된 각 주제로 분리하여 평가하였다. 이때 하나의 발화에 대하여, 각 주제가 모두 일치할 경우에만 정답과 일치한다고 평가한 지표를, 표 3에 exact match 로 나타내었고, 한 발화에 대하여, 정답 주제들과, 예측 주제들을 xor 연산하여 합한 값을 전체 주제의 개수로 나누어 오류 비율을 나타낸 hamming error를 표3에서 Hamming으로 나타내었다. precision, recall, F1은 모두 macro 형식으로 측정하였다. 표3에서 epoch는 모델의 cost가 수렴하여 학습을 종료한 epoch을 의미한다.

Cluster 방법론을 사용한 모델을 평가할 때에는 각 cluster이 나타내는 발화주제들로 모두 치환하여 평가하였다. 가령 "아이유 노래 틀어줘"라는 발화를 $media로 분류하였을 때 해당 결과를 music, radio, news, control, audioclip, broadcast로 치환하여, 소개한 지표들로 평가하였다.

실험 결과 전반적으로 multi-label 방법을 사용 하였을 때 큰 성능 하락을 보였다. Cluster 방법을 사용하고, sigmoid를 output layer의 activation 함수로 사용, cross entropy 함수를 cost function으로 사용하였을 때 모든 metric에서 가장 우수한 score를 보였다.

6. 결론

실험에서 multi-label 방법과 cluster 방법론 간의 명료한 수준의 성능 차이를 발견하였다. 다만 현실 세계에서 실제로 입력으로 나타날 발화의 주제들이 cluster로 미리 정의되어 있지 않다면, cluster 방법론을 사용한 시스템은, 발화의 주제를 제대로 추론하지 못한다.

추후 기회가 된다면 모델이 cluster의 계층 관계를 학습할 수 없는 cluster방법론의 문제점을 개선하여, 발화의 주제를 여러 계층에 거쳐 표현하여 정의하고, 분류하는 방법인 hierarchy[2]를 이용하여 발화 주제 다중 분류 연구를 진행하고자 한다.

참고문헌

[1] Grigorios Tsoumakas, Ioannis Katakis, "Multi-Label Classification: An Overview" International Journal of Data Warehousing and Mining 3.3, 2006.

[2] Seonghan Ryu, Jaiyoun Song, Sangjun Koo, Soonchoul Kwon, Gary Geunbae Lee, "Detection Multiple Domain from User's Utterance in Spoken Dialog System", Natural Language Dialog Systems and Intelligent Assistants. Pp. 101-111, 2015.

[3] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882, 2014.

[4] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. "Exploiting diverse knowledge sources via maximum entropy in named entity recognition", In Sixth Workshop on Very Large Corpora New Brunswick, New Jersey. Association for Computational Linguistics., 1998.

[5] Kingma, D. P., & Ba, J. L. "Adam: a Method for Stochastic Optimization", International Conference on Learning Representations, 1-13, 2015.

표3. cost function에 따른 평가

	Filter	Multi-label						Cluster					
		Exact match	Hamming	Precision	Recall	F1	Epoch	Exact match	Hamming	Precision	Recall	F1	Epoch
Simoid Cross Entropy	64	88.78	0.040008	0.92513	0.8952	0.8999	39	98.1671	0.032805	0.9853	0.9862	0.9843	43
	128	**88.97**	**0.039342**	**0.93507**	**0.8985**	**0.9041**	32	**98.5089**	**0.031656**	**0.9887**	**0.9884**	**0.9873**	27
Softmax Cross Entropy	64							98.1671	0.038283	0.9847	0.9847	0.9835	26
	128							98.2914	0.035663	0.9863	0.9843	0.9853	27
MSE	64	88.78	0.041051	0.9089	0.8925	0.8952	28	97.0177	0.062545	0.98	0.9772	0.9756	34
	128	88.78	0.039875	0.9258	0.8953	0.9000	29	97.3284	0.056539	0.9812	0.9785	0.9777	**14**