발화 내 페르소나 트리플 추출 방법 연구

장윤나[1○], 양기수[1,2], 허윤아[3*], 임희석[1,3*]
[1]고려대학교 컴퓨터학과, [2]바이브컴퍼니, [3]Human-inspired AI 연구소
{morelychee, willow4, yj72722, limhseok}@korea.ac.kr

A Method for Extracting Persona Triples in Dialogue

Yoonna Jang[1○], Kisu Yang[1,2], Yuna Hur[3*], Heuiseok Lim[1,3*]
[1]Department of Computer Science and Engineering, Korea University, [2]VAIV Company, [3]Human-inspired AI Research

요 약

본 논문에서는 대화 중 발화에서 페르소나 트리플을 추출하는 방법을 연구한다. 발화 문장과 그에 해당하는 트리플 쌍을 활용하여 발화 문장 혹은 페르소나 문장이 주어졌을 때 그로부터 페르소나 트리플을 추출하도록 모델을 멀티 태스크 러닝 방식으로 학습시킨다. 모델은 인코더-디코더 구조를 갖는 사전학습 언어모델 BART [1]와 T5 [2]를 활용하며 relation 추출과 tail 추출의 두 가지 태스크를 각각 인코더, 디코더 위에 head를 추가하여 학습한다. Relation 추출은 분류로, tail 추출은 생성 문제로 접근하도록 하여 최종적으로 head, relation, tail의 구조를 갖는 페르소나 트리플을 추출하도록 한다. 실험에서는 BART와 T5를 활용하여 각 태스크에 대해 다른 학습 가중치를 두어 훈련시켰고, 두 모델 모두 relation과 tail을 추출하는 태스크 정확도에 있어서 90% 이상의 높은 점수를 보임을 확인했다.

주제어: 페르소나 대화, 페르소나 트리플 추출, 페르소나 추출, 트리플 추출, 오픈 도메인 대화

1. 서론

일정한 페르소나를 가지며 일관된 발화를 생성해야 하는 페르소나 대화 [3, 4]에 대한 연구가 발전되며 기계는 사람과 같이 본인이 이전에 한 말에 모순되지 않고 일관된 발화를 하도록 학습이 되어왔다. 일관성을 위해 dialogue natural language inference (DNLI) [5]에서는 발화와 페르소나 문장에 트리플을 할당하고 발화-발화, 페르소나-발화 문장 사이의 함의 (entailment), 모순 (contradiction), 중립 (neutral) 관계에 대해 라벨링을 해 두었다. 이를 활용하여 페르소나 기반 대화의 발화 생성에 있어 본인의 페르소나 문장 혹은 이전 발화에 대해 높은 함의 관계를 가지는 발화를 생성할 수 있도록 한 연구들이 활발하게 진행되어 왔다. [6, 7, 8]

본 연구에서는 발화 문장과 그에 해당하는 트리플을 활용하여 발화 문장으로부터 페르소나 트리플을 추출하는 연구를 수행한다. 발화 내 페르소나 트리플을 추출함으로써 발화자에 대한 정보를 트리플 형태로 간단하게 저장해 둘 수 있다. 이는 발화자에 대한 지식 베이스 구축을 가능하게 하며 트리플 형태로 이루어져 관리 및 수정이 텍스트에 비해 용이하다. 또한 이러한 페르소나 트리플은 추후 대화 생성에 있어서도 도움을 줄 수 있다. 우리는 인코더-디코더 구조를 갖는 사전학습 언어모델 BART [1]와 T5 [2]를 활용하여 문장으로부터 트리플을 추출하는 태스크를 학습시킨다. 페르소나 트리플 추출은 relation 추출과 tail 추출로 이루어져 있다. 먼저 relation 추출은 61개의 기정의된 relation 클래스 중 하나를 고르도록 학습하며, tail 추출은 타겟이 되는 tail을 생성을 통해 학습하도록 한다. 실험을 통해 BART와 T5 모두 relation 예측 정확도와 tail 토큰 정확도에 있어서 90%가 넘는 높은 점수를 보임을 확인한다.

2. 관련 연구

기계에게 인간다운 대화를 학습시키기 위해 페르소나 대화 [3]가 등장하게 되었다. 페르소나 대화에서는 발화자에게 고유한 페르소나가 있으며 이에 기반하여 매 턴마다 변덕스러운 발화가 아닌 일관적인 페르소나를 유지하며 발화를 하는 양상을 보였다. BlenderBot [9, 10, 11, 12]에서는 이러한 페르소나 대화를 포함한 오픈 도메인 대화 모델링을 하고자 하는 시도를 보였다. 페르소나 대화에서 발화와 페르소나 문장에 트리플을 레이블링 해 두고 그 사이의 함의, 모순, 중립 관계에 대해 라벨링한 DNLI [5] 데이터셋을 활용한 연구도 활발하게 진행되어 왔다. [8, 6] 본 연구에서는 이전 연구 [13, 14]에서처럼 문장과 트리플 쌍을 활용하여 문장이 주어졌을 때 트리플을 추출하는 작업을 진행하나, 사전학습 언어모델인 BART [1]와 T5 [2]를 이용하여 relation과 tail을 추출한다.

3. 발화 내 페르소나 트리플 추출 방법

본 연구에서는 dialogue natural language inference (DNLI) [5] 데이터셋의 문장-트리플 쌍을 활용하여 발화 내 페르소나를 추출하고자 한다. 그림 1에서와 같이 발화 문장 혹은 페르소나 문장이 트리플 형태로 표현되어 있기에 이러한 문장-트리플 쌍을 활용하여 문장이 주어졌을 때 트리플을 추출하는 태스크를 인코더-디코더 구조를 가진 모델에 학습시킨다.

*교신저자 (Corresponding author)

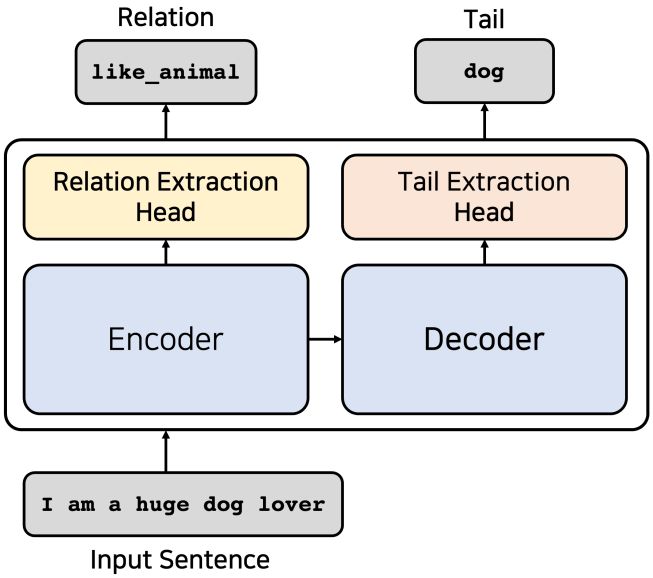

그림 1. 입력 문장이 들어가면 relation extraction head가 relation을 추출하며 tail extraction head는 tail을 생성하도록 학습함

학습은 두 가지의 하위 태스크로 이루어져 있는데 (1) relation 추출과 (2) tail 추출이다.

먼저 relation 추출에서는 문장이 주어졌을 때 표 1의 61개의 라벨 중 하나를 선택하는 작업을 학습한다. 문장의 트리플이 페르소나 추출과 관련이 없는 경우에는 라벨을 'none'으로 매핑시켜 학습하도록 한다. 모델의 입력에 페르소나 문장 혹은 발화 문장 S를 주고 모델 인코더 (encoder hidden states) 상단에 relation 추출을 위한 헤드를 추가하여 학습하도록 하였으며, 이 헤드는 다중 클래스 교차 엔트로피 손실 (multi-class cross entropy loss) 을 활용하여 여러 개의 relation 클래스 중 하나를 선택하는 법을 배우게 된다. 실제 relation label이 r이라 할 때 학습 손실은 다음과 같다:

$$\mathcal{L}_{rel} = -r \log p(\hat{r}|S), \tag{1}$$

다음으로 tail 추출에서는 트리플 중 tail에 해당하는 속성을 생성 (generation)을 통해 학습하도록 한다. 모델의 인코더에 문장이 주어졌을 때 디코더의 상단에 있는 language model head가 N의 토큰 길이를 갖는 tail $T = t_1, ..., t_N$을 생성하는 법을 학습하도록 한다. 방법은 일반적인 자동회귀 (auto-regressive) 언어 생성 방법과 같다:

$$\mathcal{L}_{tail} = -\frac{1}{N}\sum_{i=1}^{N} \log p(t_i|t_{<i}), \tag{2}$$

전체 학습은 그림 1과 같이 위의 두 가지 태스크에 대해 모델은 동시에 학습을 하는 멀티 태스크 학습 (multi-task learning)

Relation 라벨
'none', 'have_sibling', 'own', 'like_watching', 'favorite_hobby', 'teach', 'have_family', 'live_in_general', 'place_origin', 'like_food', 'favorite_place', 'has_degree', 'employed_by_general', 'employed_by_company', 'dislike', 'favorite_music', 'favorite_activity', 'gender', 'like_goto', 'have_children', 'member_of', 'like_movie', 'job_status', 'favorite_sport', 'like_activity', 'like_animal', 'marital_status', 'favorite_color', 'attend_school', 'favorite_animal', 'school_status', 'physical_attribute', 'favorite_food', 'nationality', 'like_general', 'like_music', 'favorite_book', 'favorite_music_artist', 'have_vehicle', 'want', 'has_hobby', 'like_drink', 'misc_attribute', 'have', 'other', 'has_profession', 'favorite_season', 'like_read', 'want_job', 'favorite_movie', 'previous_profession', 'favorite_show', 'live_in_citystatecountry', 'have_pet', 'not_have', 'has_ability', 'like_sports', 'want_do', 'favorite_drink', 'has_age', 'favorite'

표 1. Relation 추출 태스크에 사용된 전체 relation 라벨. 총 61개의 라벨로 구성되어 있음

을 수행한다. 각각 인코더와 디코더 상단의 헤드를 통해 두 가지 태스크를 배우게 되며 태스크에 대한 비중은 각 학습 로스에 서로 다른 값 ($\lambda_{rel}, \lambda_{tail}$) 을 부여함으로써 달라지게 된다. 전체 학습은 다음의 식을 최적화하며 이루어진다:

$$\mathcal{L} = \lambda_{rel}\mathcal{L}_{rel} + \lambda_{tail}\mathcal{L}_{tail} \tag{3}$$

4. 실험

4.1 데이터셋

실험에 사용한 데이터셋은 대화 내 발화와 페르소나 문장 사이의 함의 (entailment) 관계에 대해 표기해 둔 dialogue natural language inference (DNLI) [5]이다. 이 데이터셋은 페르소나 문장을 트리플 형태와 함께 병렬로 표기해 두었으며 발화 대 페르소나 문장, 페르소나 문장 대 페르소나 문장 사이의 함의 (entailment), 모순 (contradiction), 중립 (neutral) 관계에 대하여 라벨링이 되어있다. 전체 데이터는 학습, 검증, 테스트 셋 안에 각각 310,110, 16,500, 16,500 개의 문장 쌍과 라벨로 구성되어 있다. 우리는 그 중 모델이 relation과 tail 값을 추출하도록 학습시키기 위해 트리플의 헤드 부분의 값이 'i' 혹은 'my'가 아닌 경우 relation의 라벨을 'none'으로 할당하여 해

표 2. 실험 결과. Relation은 relation 추출 정확도를 나타내며 Tail은 tail token accuracy 정확도를 나타냄. 결과 값은 소수점 셋째 자리까지 반올림 되었음. 괄호 안의 값은 relation과 tail 추출 태스크에 대한 가중치 값을 나타냄

	Relation	Tail
BART (1:1)	0.904	0.955
BART (1:2)	**0.907**	**0.965**
BART (2:1)	0.904	0.941
T5 (1:1)	**0.901**	0.957
T5 (1:2)	0.898	**0.961**
T5 (2:1)	**0.901**	0.953

당 문장의 경우 페르소나 트리플을 추출하지 않도록 학습 및 평가에 사용한다.

4.2 실험 환경

실험 코드를 위해 Pytorch Lightning [15]과 Huggingface의 Transformers [16] 라이브러리가 사용된다. 학습에 사용된 모델은 인코더-디코더 구조를 가지는 사전학습 언어모델인 BART [1]와 T5 [2]의 베이스 모델을 미세조정하여 활용한다. 모델 학습은 AdamW [17] 옵티마이저와 5e-05의 학습률(learning rate) 을 사용하며 모델은 총 3에폭에 대해 학습된다. 평가를 위해 relation 추출에는 정확도, tail 추출에 있어서는 길이가 길지 않은 단어 혹은 구를 생성하기에 평가는 토큰 정확도(token accuracy)를 활용하여 측정한다.

4.3 결과

Relation 추출과 tail 추출에 대한 실험 결과는 표 2에서 확인할 수 있다. 먼저 여러 개의 relation 클래스 중 하나를 선택하는 relation 추출에서는 BART가 T5와 비슷하지만 조금 더 높은 성능을 보인다. Tail 추출에서 역시 BART가 T5보다 약간 높은 성능을 보이는 것을 확인할 수 있다. 각 태스크에 대한 학습 가중치는 괄호 안에 나와 있는데 BART 모델의 경우 relation 추출 태스크에 1, tail 추출 태스크에 2를 주었을 때 가장 좋은 성능을 보였다. T5의 tail 추출 태스크에서는 동일한 결과를 보이지만, relation 추출 태스크에서는 가중치가 1:1 혹은 2:1인 경우에 더 좋은 성능을 보이는 것을 확인할 수 있었다.

5. 결론

본 논문에서는 발화 문장 혹은 페르소나 문장에 연결된 트리플 쌍을 활용하여 문장이 주어졌을 때 문장으로부터 페르소나 트리플을 추출하는 기법을 연구했다. 인코더-디코더 구조를 가진 모델에 relation과 tail 추출을 위한 헤드를 추가하여 모델을 학습시켰고 정확도 기준 약 90%가 넘는 성능을 보이는 것을 확인하였다. 현재는 실험에 활용한 DNLI 데이터셋에만 학습이 되어 학습 데이터의 양적, 질적 측면에서 부족함이 있다. 따라서 일반 도메인 대화에서의 발화-트리플 쌍이 구축된다면 대화 중 상대방의 페르소나를 추출하는 기술이 보다 쉽게 가능할 것으로 보이며 이를 추후 연구로 남긴다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기술기획평가원의 대학ICT연구센터지원사업의 연구결과로 수행되었음 (IITP-2023-2018-0-01405). 이 논문은 2021년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (NRF-2021R1A6A1A03045425). 또한 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기술기획평가원의 지원을 받아 수행된 연구임 (No. 2020-0-00368, 뉴럴-심볼릭(neural-symbolic) 모델의 지식 학습 및 추론 기술 개발).

참고문헌

[1] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[2] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, Vol. 21, No. 1, pp. 5485–5551, 2020.

[3] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston, "Personalizing dialogue agents: I have a dog, do you have pets too?" *arXiv preprint arXiv:1801.07243*, 2018.

[4] J. Urbanek, A. Fan, S. Karamcheti, S. Jain, S. Humeau, E. Dinan, T. Rocktäschel, D. Kiela, A. Szlam, and J. Weston, "Learning to speak and act in a fantasy text adventure game," *arXiv preprint arXiv:1903.03094*, 2019.

[5] S. Welleck, J. Weston, A. Szlam, and K. Cho, "Dialogue natural language inference," *arXiv preprint arXiv:1811.00671*, 2018.

[6] Y. Cao, W. Bi, M. Fang, S. Shi, and D. Tao, "A model-agnostic data manipulation method for persona-based

dialogue generation," *arXiv preprint arXiv:2204.09867*, 2022.

[7] E. Mitchell, J. J. Noh, S. Li, W. S. Armstrong, A. Agarwal, P. Liu, C. Finn, and C. D. Manning, "Enhancing self-consistency and performance of pre-trained language models through natural language inference," *arXiv preprint arXiv:2211.11875*, 2022.

[8] Y. Nie, M. Williamson, M. Bansal, D. Kiela, and J. Weston, "I like fish, especially dolphins: Addressing contradictions in dialogue modeling," *arXiv preprint arXiv:2012.13391*, 2020.

[9] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, E. M. Smith, Y.-L. Boureau *et al.*, "Recipes for building an open-domain chatbot," *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 300–325, 2021.

[10] M. Komeili, K. Shuster, and J. Weston, "Internet-augmented dialogue generation," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8460–8478, 2022.

[11] J. Xu, A. Szlam, and J. Weston, "Beyond goldfish memory: Long-term open-domain conversation," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5180–5197, 2022.

[12] K. Shuster, J. Xu, M. Komeili, D. Ju, E. M. Smith, S. Roller, M. Ung, M. Chen, K. Arora, J. Lane *et al.*, "Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage," *arXiv preprint arXiv:2208.03188*, 2022.

[13] C.-S. Wu, A. Madotto, Z. Lin, P. Xu, and P. Fung, "Getting to know you: User attribute extraction from dialogues," *arXiv preprint arXiv:1908.04621*, 2019.

[14] L. Zhu, W. Li, R. Mao, V. Pandelea, and E. Cambria, "Paed: Zero-shot persona attribute extraction in dialogues," *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9771–9787, 2023.

[15] W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," Mar. 2019. [Online]. Available: https://github.com/Lightning-AI/lightning

[16] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Oct. 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.