

S2W 데이터셋 학습 실험 결과 보고 - 2



20151322 이한용

학습 세팅

- 모델
 - Flair 에서 권장하는 모델을 사용(<https://huggingface.co/flair/ner-english>)
 - Flair embeddings 와 LSTM-CRF을 사용한 모델
 - 단, FlairEmbedding 은 기존의 것에 S2W corpus를 fine-tuning 한 것을 사용함 (다음 슬라이드 참고)
- dataset
 - corpus에 중복된 문장이 몇 개 포함되어 있었음 (51쌍). 해당 문장의 중복을 제거
 - 두개 이상의 랜섬웨어 이름이 들어간 문장을 제외. (제외 후 6540문장)
 - train, dev, test을 약 4:1:1의 비율로 나눔
 - 따라서 4360 train + 1090 dev + 1090 test sentences 으로 구성되었음.
- 10번 반복하여 실험을 진행했고, 매 학습마다 다른 dataset과 해당 dataset으로 fine-tuning한 embedding을 사용함

학습 세팅

- FlairEmbedding fine-tuning 세팅

- 기본 설정 :

- sequence_length = 미정
 - mini_batch_size = 100
 - learning_rate = 20
 - max_epochs = 1000
 - patience = 10
 - num_workers = 2

- 사용 설정 :

- sequence_length = 250
 - mini_batch_size = 100
 - learning_rate = 20
 - max_epochs = 100
 - patience = 3
 - num_workers = 16

S2W 데이터셋 학습 실험 결과 보고 - 2

이번 실험 결과 – embedding을 fine-tuning 했을 때

F1-scores	F1-score (micro)	F1-score (macro)	LOC	MISC	ORG	PER	RAN
실험1	0.8603	0.8540	0.9259	0.8823	0.8083	0.8148	0.8389
실험2	0.8600	0.8766	0.9516	0.8826	0.8242	0.9062	0.8184
실험3	0.8450	0.8609	0.9737	0.8798	0.8291	0.8406	0.7813
실험4	0.8989	0.8918	0.9296	0.9122	0.8590	0.8710	0.8872
실험5	0.8712	0.9065	0.9639	0.8742	0.8622	0.9730	0.8593
실험6	0.8543	0.8780	0.9620	0.8844	0.8204	0.9231	0.8004
실험7	0.9047	0.9096	0.9677	0.9011	0.8786	0.8889	0.9117
실험8	0.9198	0.8970	0.9388	0.9363	0.8418	0.8462	0.9220
실험9	0.8332	0.8554	0.9635	0.8452	0.8491	0.8276	0.7914
실험10	0.8764	0.8971	0.9811	0.8935	0.8418	0.9184	0.8508
평균 ± 분산	0.8724 ± 0.0008	0.8827 ± 0.0004	0.9558 ± 0.0004	0.8892 ± 0.0006	0.8415 ± 0.0005	0.8810 ± 0.0025	0.8461 ± 0.0024

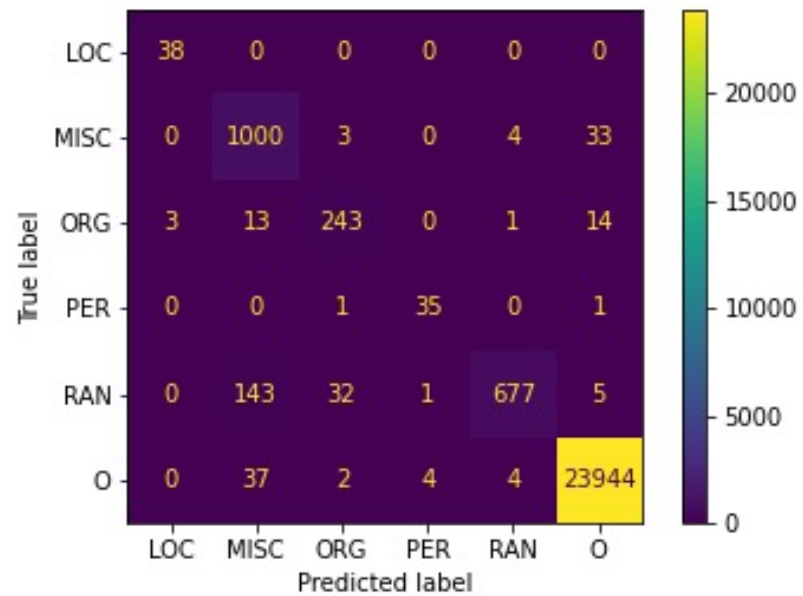
S2W 데이터셋 학습 실험 결과 보고 - 2

(비교) 저번 실험 결과 – embedding을 fine-tuning 하지 않았을 때

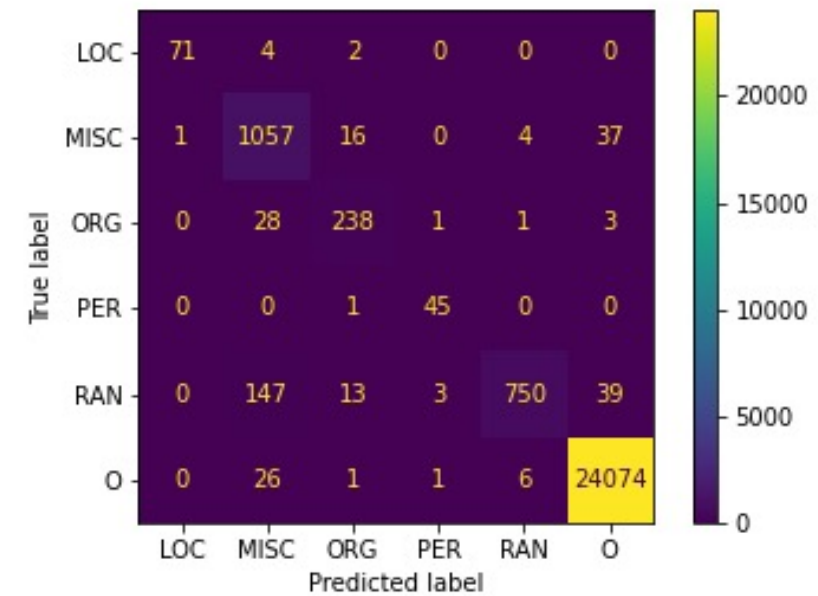
F1-scores	F1-score (micro)	F1-score (macro)	LOC	MISC	ORG	PER	RAN
실험1	0.8934	0.9150	0.9890	0.9115	0.8571	0.9538	0.8635
실험2	0.8687	0.8946	0.9484	0.8906	0.8753	0.9492	0.8098
실험3	0.8728	0.8839	0.9697	0.9121	0.8021	0.9167	0.8190
실험4	0.9221	0.9261	0.9933	0.9294	0.8805	0.9143	0.9130
실험5	0.8705	0.8947	0.9565	0.9001	0.8571	0.9474	0.8126
실험6	0.8853	0.8899	0.9630	0.8919	0.8659	0.8571	0.8714
실험7	0.9193	0.9194	1.0000	0.9311	0.8458	0.9057	0.9145
실험8	0.8612	0.8387	0.9512	0.8800	0.8641	0.6667	0.8316
실험9	0.8948	0.8980	0.9877	0.9167	0.8244	0.8889	0.8723
실험10	0.8762	0.8973	0.9552	0.9076	0.8446	0.9524	0.8268
평균 ± 분산	0.8864 ± 0.0004	0.8958 ± 0.0006	0.9714 ± 0.0004	0.9071 ± 0.0003	0.8517 ± 0.0006	0.8952 ± 0.0074	0.8535 ± 0.0016

S2W 데이터셋 학습 실험 결과 보고 - 2

실험 결과 - confusion matrix



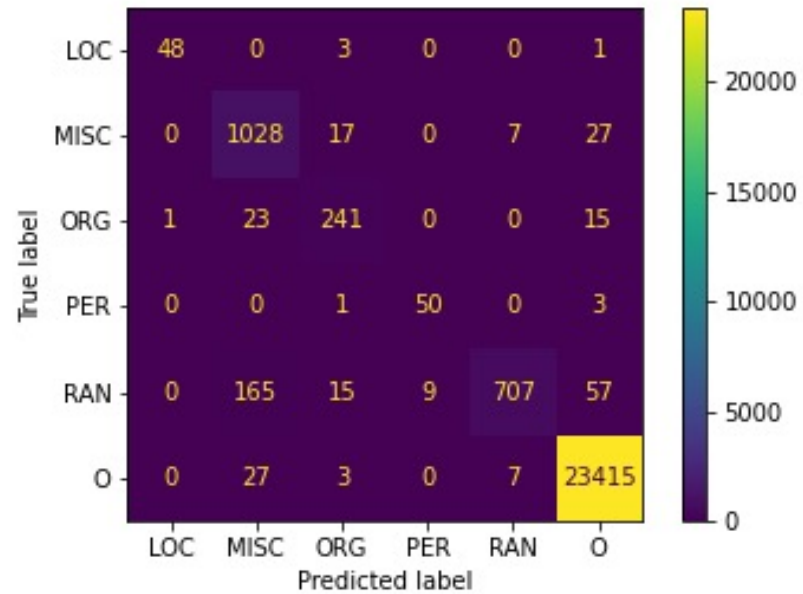
실험 1



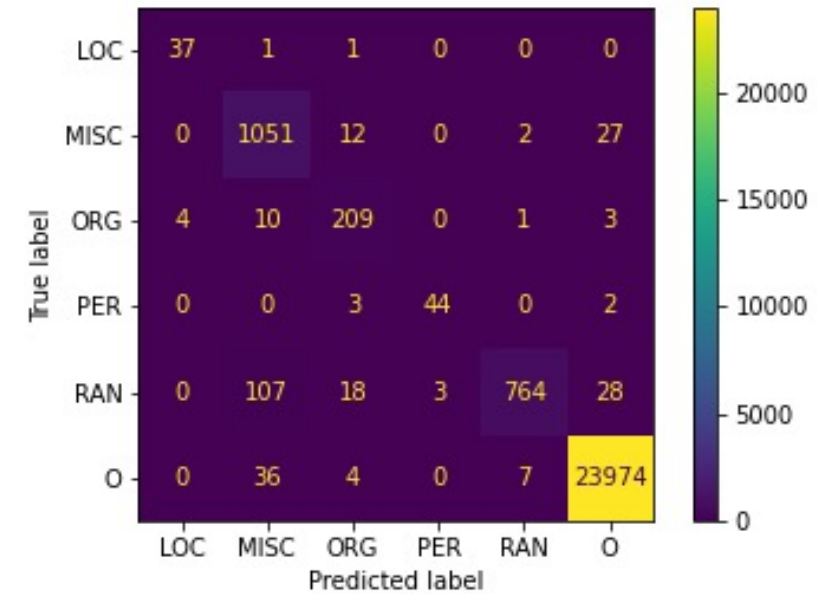
실험 2

S2W 데이터셋 학습 실험 결과 보고 - 2

실험 결과 – confusion matrix



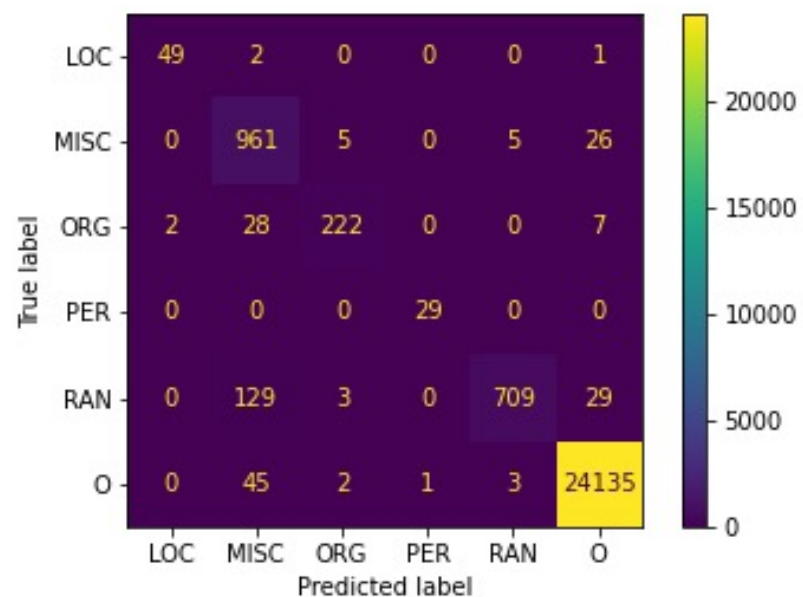
실험 3



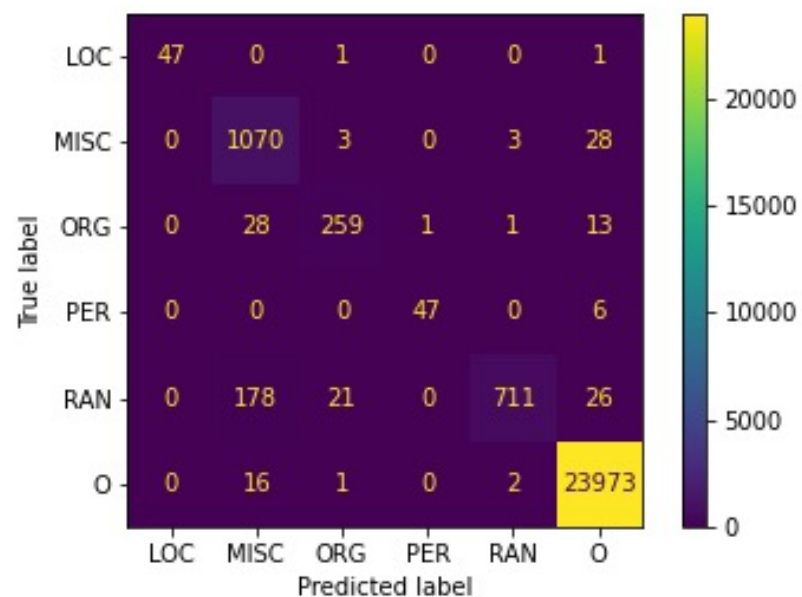
실험 4

S2W 데이터셋 학습 실험 결과 보고 - 2

실험 결과 - confusion matrix

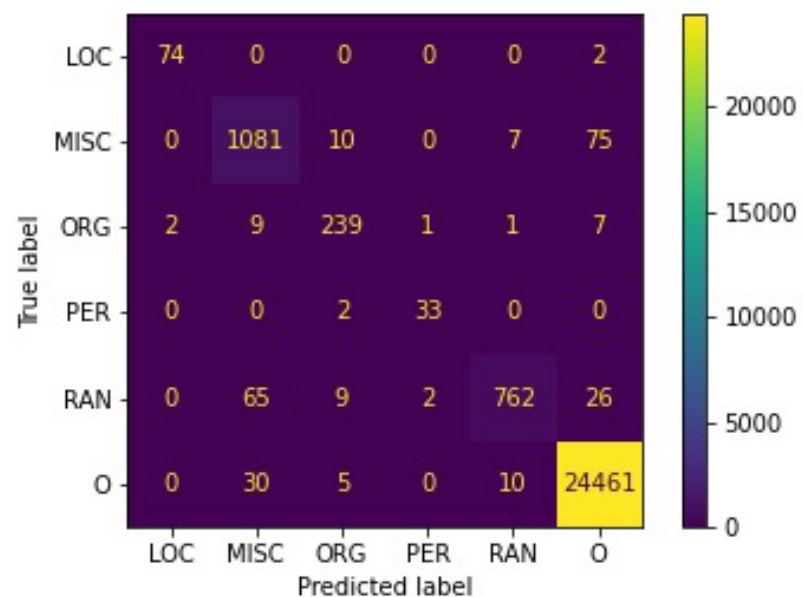


실험 5

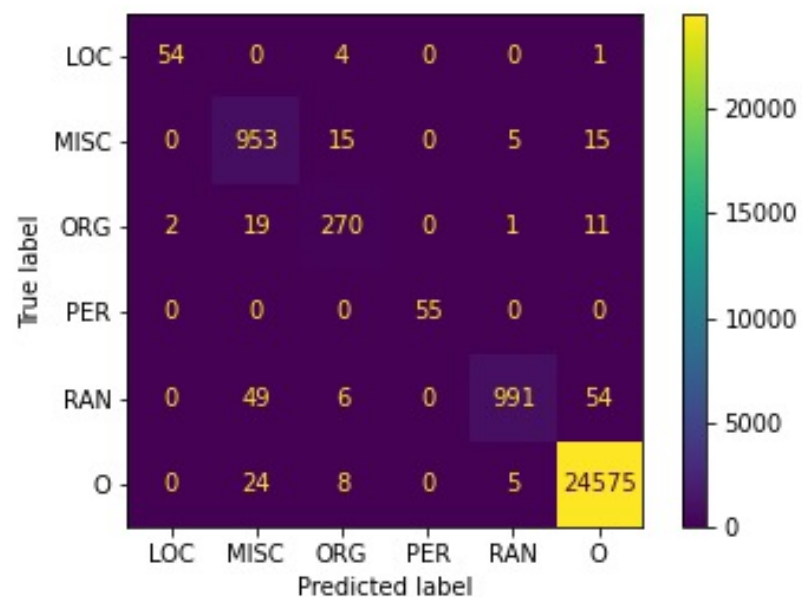


실험 6

실험 결과 - confusion matrix

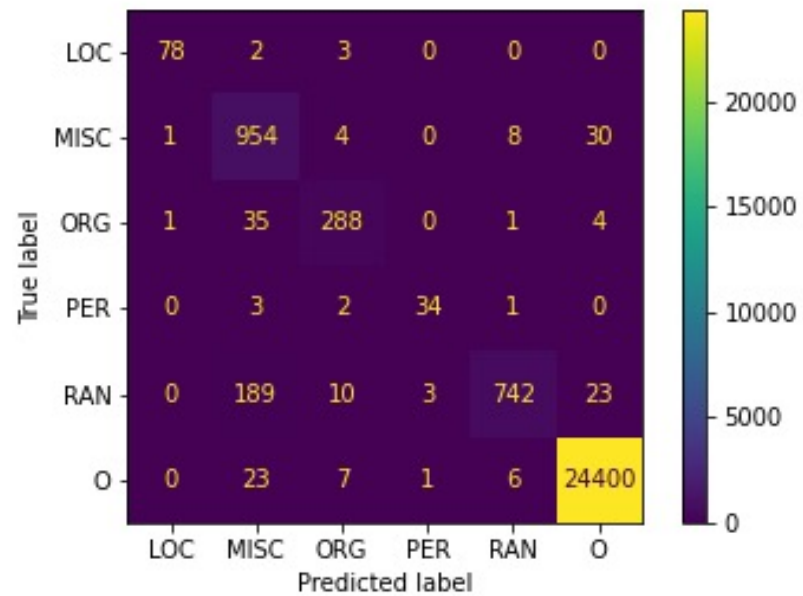


실험 7

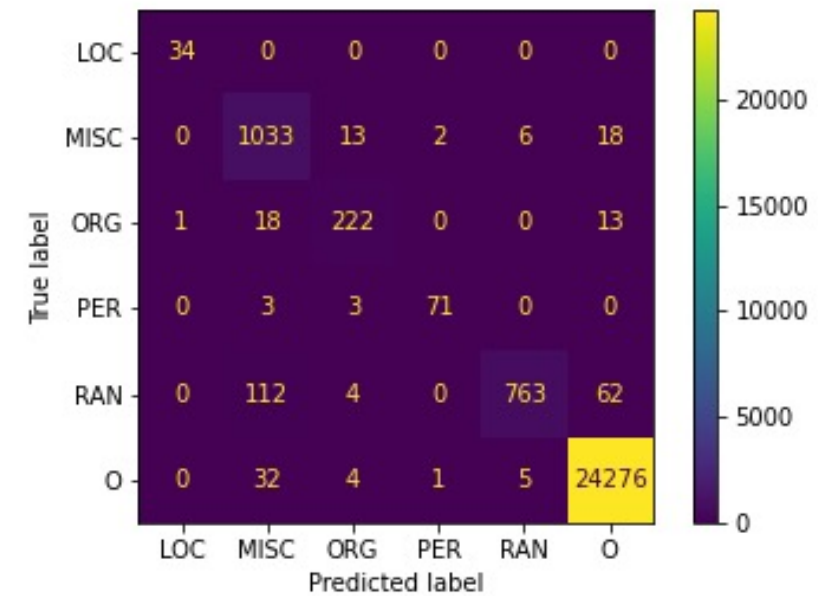


실험 8

실험 결과 – confusion matrix



실험 9



실험 10

결과 분석

1. Flair Embedding을 fine-tuning 했을 때 F1-score가 전반적으로 낮아짐

- 저번 실험에서는 pretrained model 을 fine-tuning 했기 때문에 RAN tag를 제외한 나머지 4개 tag에 대한 F1-score가 이번 실험보다 높을 수 있음
- **문제에 대한 해결책**
 1. 저번 실험의 pretrained model이 학습한 conll 03 dataset을 추가로 학습한다.

2. RAN -> MISC으로 오판한 경우의 수가 줄어들지 않음

- FlairEmbedding 을 s2w corpus에 대해 fine-tuning한 효과가 거의 없었다고 판단됨.
- **문제에 대한 해결책**
 1. max_epoch을 증가시켜 embedding을 재학습