Instructor : 김학수 교수님

2020 졸업작품 프로젝트

2nd Implementation Demo

김지은 | 채민형 | 최병규 | 최지원

목차
INDEX

01 프로젝트 개발동기

01 프로젝트 주제

: **Seq2Seq 모델**을 이용한 한국어 **개체명인식기** 및 **의도분석기** 구현

02 프로젝트 목표

- 한국어 개체명인식과 의도분석 기능을 동시에 수행할 수 있는 시스템을 구현한다.
- 각각의 기능으로 구성되어 있던 두 모델을 하나의 모델로 통합함으로써 시스템의 간편한 사용이 가능하도록 한다.

03 프로젝트 필요성

- 개체명 인식 : 문장으로부터 개체명을 추출하고 추출된 개체명의 종류를 분류하는 기능
  의도 분석 : 문장이 실제로 내포하는 의미를 파악하고 결정하는 기능
  ➡ 기계가 인간의 말을 이해하게 하는데 **핵심적인 기능**
- 한국어는 영어와 언어적 특질이 달라, 자연어처리 분야에서 비교적 낮은 성능을 보임
  ➡ 다양한 방면, 방식의 연구 필요성 대두

02 프로젝트 개요

01 모델 : Seq2Seq

: Seq2Seq, 즉 sequence-to-sequence는 임의 길이의 한 시퀀스를
다른 종류의 시퀀스로 변환하는 확률모델

02 기능 : 개체명인식 & 의도분석

- 개체명인식(NER : Named Entity Recognition)은 태깅 활용해 시퀀스 생성
- 의도분석(Intent Classification)은 다중 분류의 일종이나 시퀀스변환(번역)으로 접근

03 방법

: 하나의 **Encdoer**로 문장을 입력하고,
개체명인식과 의도분석 각각의 결과를 출력하는 두 개의 **Decoder**로 구성
(Encoder : 입력 문장 / Decoder 1 : 개체명인식 결과, Decoder 2 : 의도분석 결과)

04 데이터셋

: 건국 NLP LAB에서 제공받은 음식 주문 관련 데이터 클리닝, 가공하여 데이터 구성

05 기존 연구와의 차별성

- 하나의 모델로 두 가지의 결과를 도출한다.
- 태깅, 분류를 시퀀스 생성이라는 새로운 시각으로 바라본다.

03 요구사항 분석

01 인터페이스

1- 1. 사용자는 한국어 문장을 입력할 수 있다.

1- 2. 입력 후, “실행” 버튼을 마우스로 클릭하거나, 키보드의 “엔터키”를 눌러 예측을 진행한다.

02 기능적 요구사항

2-1. 데이터 가공

2-1-1. 입력된 모든 null 값은 제거한다.

2-1-2. 입력된 문장을 형태소 단위로 나눈다.

2-1-3. 학습 시 형태소를 word2vec 기법을 사용해 embedding한다.

2-1-4. 추가적으로, 개체명 인식 학습시 형태소의 품사를 사용한다.

2-2. Encoder

2-2-1. 전처리를 마친 문장을 Encoder의 입력값으로 넘겨준다.

2-2-2. 입력문장의 각 형태소의 모든 자질 및

Bi-LSTM Cell의 마지막 시점의 Hidden State 값을 Decoder로 넘겨준다.

03 요구사항 분석

2-3. 의도분석 Decoder

2-3-1. 학습 과정

2-3-1-1. Encoder에서 받은 Hidden State 값과, Output 값을 이용하여 정답값을 예측하고 정답을 학습하는 과정을 진행한다

2-3-2. 테스트 과정

2-3-2-1. Encoder에서 받은 Hidden State 값과, 시작 토큰만을 입력으로 받은 후에, 다음에 올 단어, 즉 의도를 예측한다.

2-4. 개체명인식 Decoder

2-4-1. 학습 과정

2-4-1-1. Encoder에서 받은 Hidden State 값과, Output 값을 이용하여 입력 받은 실제 정답 값이 output으로 나와야 한다는 것을 알려주는 교사 학습 과정을 진행한다.

2-4-2. 테스트 과정

2-4-2-1. Encoder에서 받은 Hidden State 값과, 시작 토큰만을 입력으로 받은 후에, 다음에 올 단어, 즉 개체명을 예측한다. 이후, 예측한 개체명을 다시 다음 입력으로 하여 다음 개체명을 예측한다.

03 요구사항 분석

2-5. Attention

2-5-1. Decoder 내 Bi-LSTM Cell의 output 값(Hidden State)을 transpose한 후,
Encoder의 모든 시점에서의 output 값을 내적하여 각각의 Attention 값을 구한다.

2-5-2. Attention 값에 Softmax 함수를 적용해 Attention Distribution을 구하고,
각각의 값은 Attenton 가중치가 된다.

2-5-3. 각 Encoder의 Attention 가중치와 output 값을 가중합하여 최종적인 Attention 값을 구한다.

2-5-4. Attention 값과 Decoder 시점의 output 값을 연결해 준 뒤,
Regression과 Softmax 함수를 적용해주고, 출력 층으로 내보낸다.

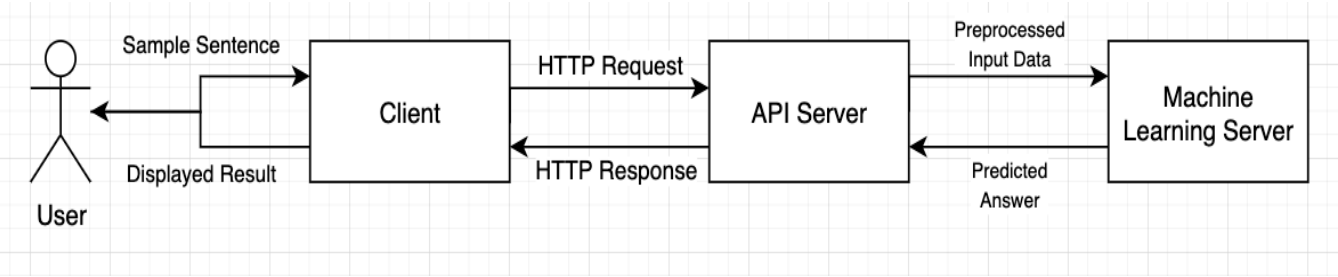
03 비기능적 요구사항

3-1. 사용자의 입력에 대해 5초 이내로 결과가 도출된다.

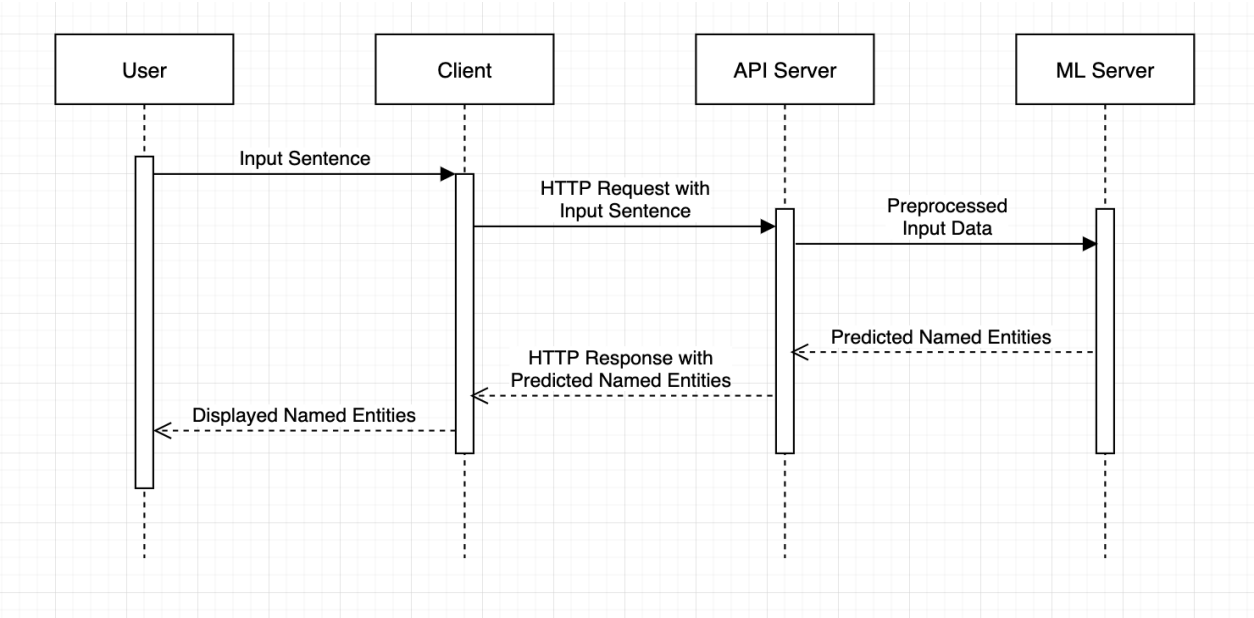
3-2. 결과는 70% 이상의 정확도를 갖는다.

04 디자인 : 하이레벨

01 시스템 아키텍처

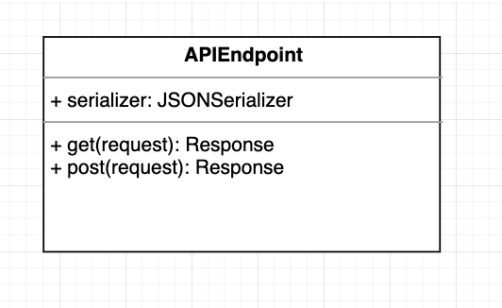


02 시퀀스 다이어그램

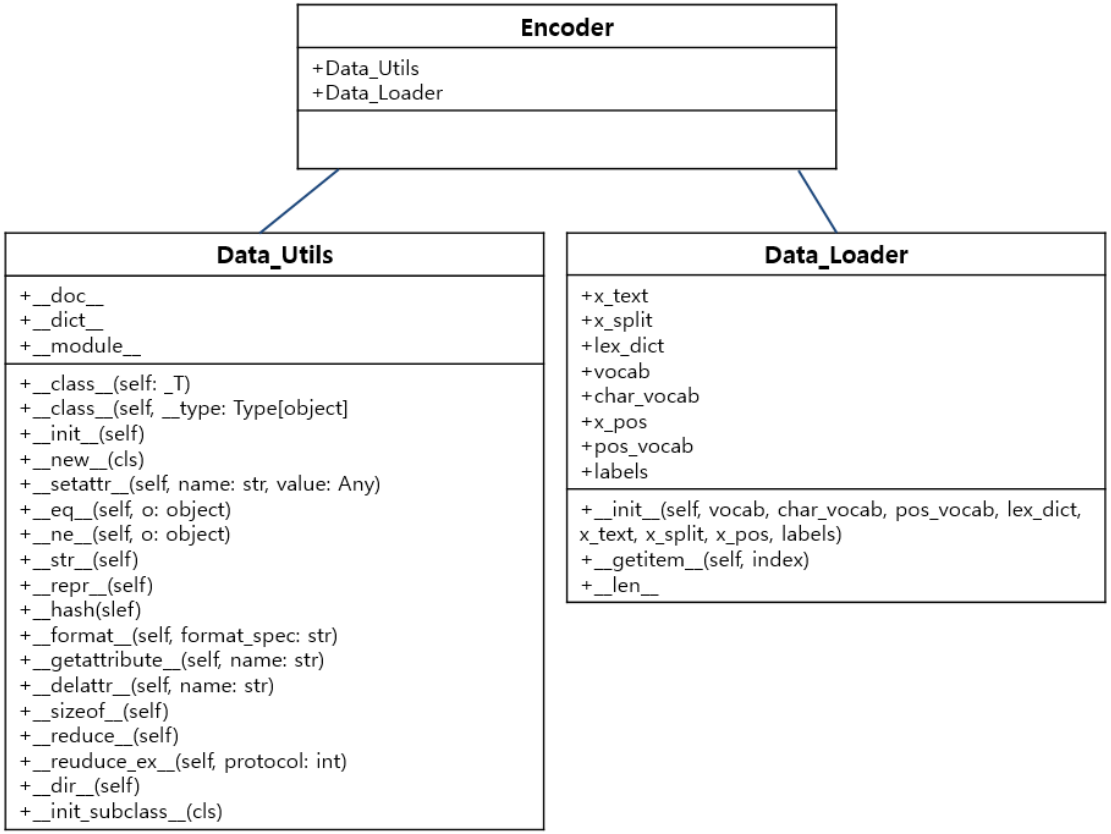


04 디자인 : 로우레벨

01 API 모듈



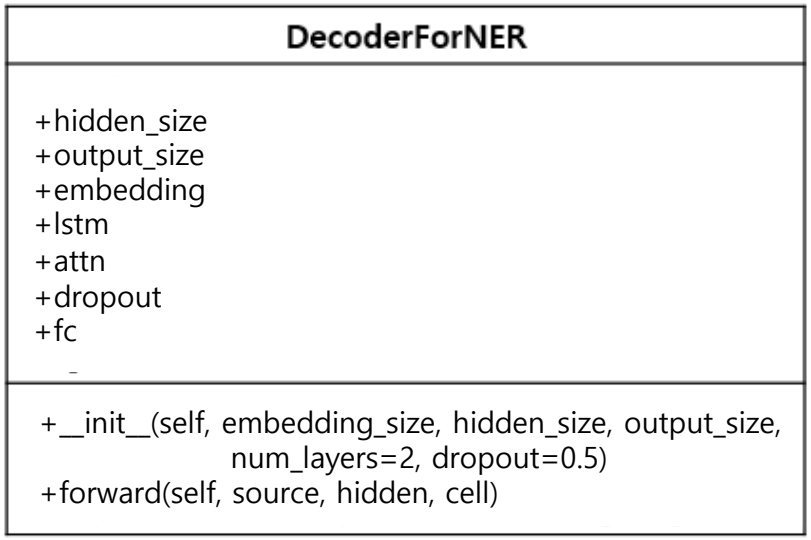
02 Encoder



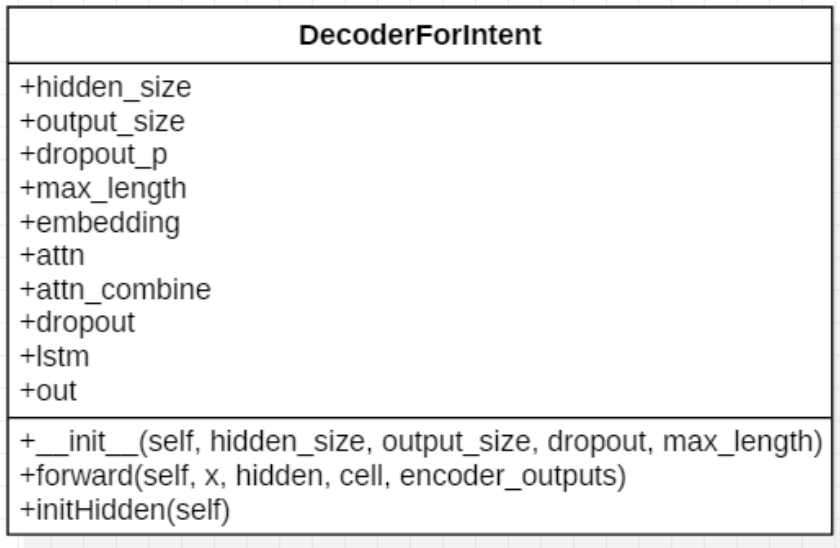
04 디자인 : 로우레벨

03 Decoder

1) NER (개체명인식)

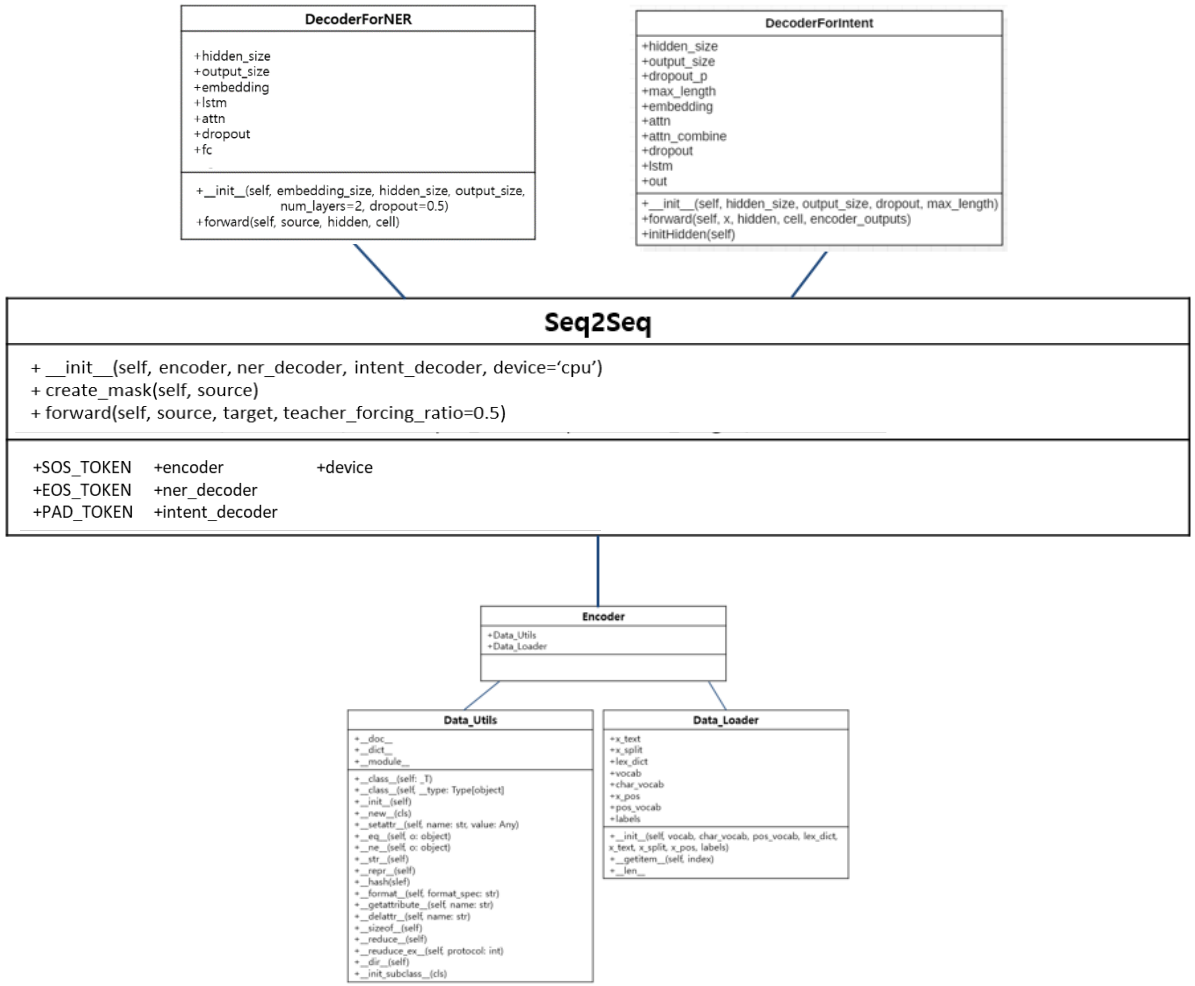


2) Intent (의도분석)



04 디자인 : 로우레벨

04 Seq2Seq 모델 : 인코더&디코더 통합 모델



05 시스템 테스트

01 테스트 케이스 – 기능적 요구사항

Case Type	테스트케이스 목표	진행/입력 내용	예상 결과	실행결과	Pass /Fail
1.1 Preprocessing	문장 단위의 입력 데이터에서 '일반 명사', '고유 명사', '의존 명사', '수사'에 대해 분리가 올바르게 되는지 테스트한다. 형태소를 분리한 결과와 사전 정보가 일치해야 한다.	기존 Mecab을 이용하여 형태소를 분리한 결과와 사전 정보(gazette)을 비교한다.	형태소를 분리한 결과와 사전 정보가 일치한다.	형태소 분석이 정확하다	P
1.2 Encoder	결과 벡터인 Context Vector을 정상적으로 추출하는지 테스트한다.	각 데이터를 입력하였을 때 Context Vector의 존재여부를 확인한다.	Context Vector를 정상적으로 뽑아낸다.	Vector가 정상적으로 나온다	P
1.3 의도분석 Decoder	정답 의도에 해당하는 숫자 값이 출력되는지 테스트한다. 모델이 예측한 의도와 실제 정답 의도가 일치해야 한다.	학습용 데이터를 학습 후, 테스트 용 데이터에 대하여 모델이 예측한 의도와 실제 정답의도를 비교하여 일치 여부를 확인한다.	모델이 예측한 의도와 실제 정답 의도가 일치한다. 전체 Test Data 에 대한 정확도가 70% 이상이다	소수점 첫째 자리에서 반올림한 결과 48%의 정확도	F
1.4 개체명인식 Decoder	형태소가 개체명에 해당할 경우 해당 개체명을 인식해 리턴하는지 테스트한다. 모델이 예측한 개체명들과 실제 정답 개체명들이 일치해야 한다.	학습용 데이터를 학습 후, 테스트 용 데이터에 대하여 모델이 예측한 개체명들과 실제 정답 개체명들을 비교하여 일치 여부를 확인한다.	모델의 예측한 개체명들과 실제 정답 개체명들이 일치한다. 전체 Test Data 에 대한 정확도가 70% 이상이다	소수점 첫째 자리에서 반올림한 결과 92%의 정확도	P

05 시스템 테스트

01 테스트 케이스- 비기능적 요구사항

Case Type	테스트케이스 목표	진행/입력 내용	예상 결과	실행결과	Pass /Fail
2.1	사용자가 입력한 문장이 서버로 전송되고, 예측한 개체명과 의도가 결과창에 올바르게 출력되는지 테스트한다.	Demo 웹사이트에서 입력란에 한국어로 이루어진 음식점 관련 문장을 입력한다.	해당 문장이 입력되어 서버로 전송 되고, 예측한 개체명과 의도를 결과창에 보여준다.	(데모 사이트 미완성)	F
2.2	입력 후 결과가 보여지기까지 측정되는 시간이 5초이내인지 테스트한다.	Demo 웹사이트에서 문장을 입력한 후, 예측한 결과를 보여주기 까지의 시간을 측정한다.	측정한 시간이 5초 이내이다.	(데모 사이트 미완성)	F
2.3	각각 Decoder의 테스트 정확도를 구하여, 평균값을 구하고, 평균값이 70%이상인지 테스트한다.	테스트한 데이터에 대하여 각각 Decoder의 테스트 정확도의 평균값을 구한다.	해당 평균값이 70% 이상이다.	(데모 사이트 미완성 및 의도분석 Decoder 에서의 성능미달)	F

* 한국어 이외의 언어의 문장과 음식점과 관련이 없는 문장에 대해서는 테스트 하지 않는다.

05 시스템 테스트

02 Pass / Fail 추적성 분석

2.1. Preprocessing	
Pass/Fail	형태소 분리 결과 == 사전 정보
Feature	'일반 명사', '고유 명사', '의존 명사', '수사'
Approach	기존 Mecab 을 활용하여 형태소를 분리한 결과와 사전 정보(gazette)를 비교

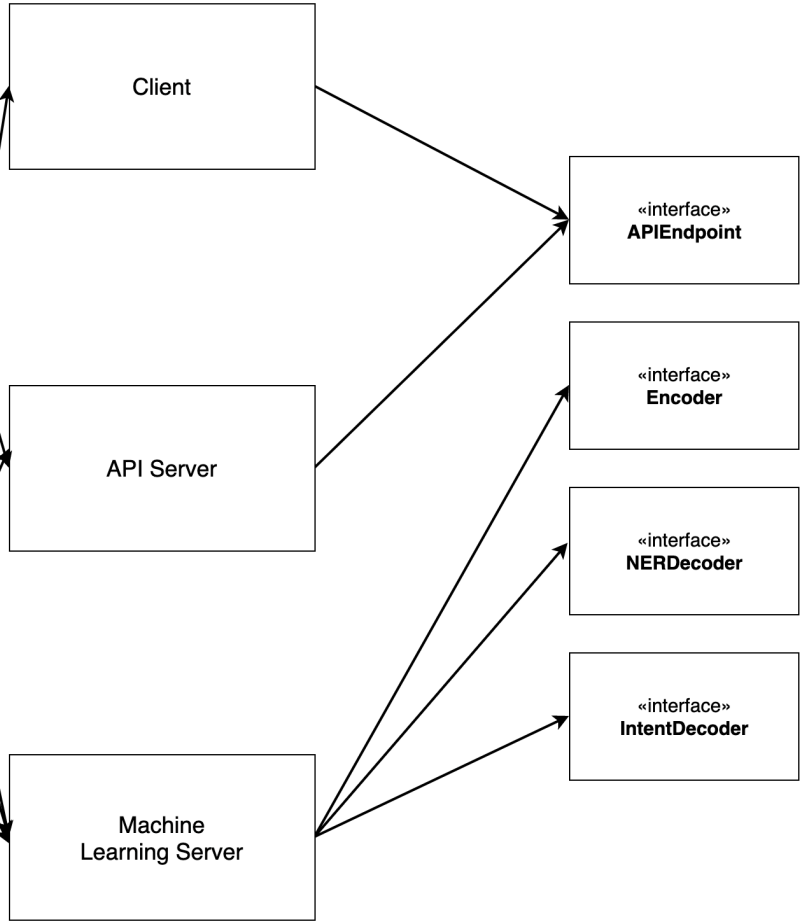
2.2. Encoder	
Pass/Fail	Context Vector 정상 생성
Feature	전처리 완료된 데이터
Approach	데이터를 모델에 입력하였을때 생성되는 벡터 확인

2.3. Intent Classification Decoder	
Pass/Fail	예측의도 == 정답의도
Feature	Encoder 에서 encoding 완료된 데이터
Approach	Train Data 로 모델 선 학습 후, Test Data 에 대하여 정답값과 예측값 비교

2.4. Named Entity Recognition Decoder	
Pass/Fail	예측 개체명 목록 == 정답 개체명 목록
Feature	Encoder 에서 encoding 완료된 데이터
Approach	Train Data 로 모델 선 학습 후, Test Data 에 대하여 정답값과 예측값 비교

3.1.1 Non-Functional Requirements	
Pass/Fail	HTTP Request 및 Response 가 정상적으로 전송 및 수신됨
Feature	사용자 입력 문장
Approach	해당 API Endpoint 로 HTTP Request 전송

3.1.1 Non-Functional Requirements	
Pass/Fail	실행시간 5초 이내
Feature	사용자 입력 문장
Approach	해당 API Endpoint 로 HTTP Request 전송



2nd Implementation Demo

06 시연 시나리오

시연용 웹사이트의 **URL을 공개**한다 (영상 및 포스터에 URL 추가)

웹사이트에서 **사람이 직접 문장을 입력**할 수 있게 하여,
모델이 **예측한 개체명과 의도를 직접 볼 수 있게** 한다

07 향후 과제

01 예측의 정확도 개선

- 기존 모델 구조를 유지한 채 Attention 방법을 조금씩 변화하거나, 레이어를 추가하는 방식으로 모델을 개선하며 예측의 정확도를 올린다.

02 학습 데이터 개선

- 데이터의 정확도를 개선하고, 더 다양한 문장을 수집하여 데이터셋을 구성하여 학습시킨다.

03 데모 사이트 구축

- 모델을 테스트하고 사용자가 직접 시연해볼 수 있는 웹사이트를 완성시킨다.

감 사 합 니 다