|  |
| --- |
| **Mirinae: 한국어 NLP를 이용한 Reconjugator**  ML based Korean Reconjugator using seq2seq  **요 약**  한국어는 발음을 배우는 시간은 다른 언어에 비해 짧으나, 한국어의 문법의 난이도는 다른 언어에 비하여 매우 고난도이면서 동시에 상당한 학습 시간을 요구한다. Mirinae는 이를 위해 한국어 nlp를 이용하여 사용자들에게 사용자 친화적이며 쉽게 사용할 수 있는 한국어 학습 서비스를 개발하고 있으며, 학습 서비스 기능 중 시제변환, 높임/낮춤말 변환을 지원하기 위한 reconjugator를 개발하고자 한다. |

1. **서론**

과거의 한류문화는 아시아권에서만 시장성이 있다고 평가되었으나, 2012년 싸이의 강남스타일이 빌보드 차트에 차트인 한 것을 계기로, 한류문화의 아시아권 밖에서의 시장성도 잠재력이 있다는 평가를 받아왔으며, 2010년대 후반 BTS라는 세계적인 열광을 일으키는 그룹이 등장함으로써, 한류문화가 세계적으로도 경쟁력이 있다는것이 입증되었다. 이러한 흐름에 발맞추어 한류음악과 함께 한류드라마와 영화도 전세계 시장에서 매우 긍정적인 반응을 이끌어내고 있다. 한류문화의 성공은 한국 엔터테인먼트 시장의 세계 시장으로의 진출 뿐만 아니라, 한국어 학습 및 교육 산업의 세계 시장 진출 역시 가져오게 되었다. 실제로 유명 언어 학습 어플리케이션 Duoligo의 통계에 따르면, 400만 사용자가 한국어 학습을 시작했으며 한국어의 TOEIC시험으로 알려진 TOPIK의 평균 시험 준비생의 수는 1년에 약 110만명이라고 알려져 있다.

한국어는 표음문자이며 동시에 과학적으로 설계되었기 때문에, 발음을 하기 쉬워 발음을 공부하는데는 쉽다고 평가되고있다. 하지만 그에 비해 한국어 문법은 난이도가 매우 어려운 것으로 평가되고있다. 한국어 문법은 불규칙한 문법이 많고, 높임말 개념이 존재하며, 하나의 단어가 여러가지 의미를 가지는 경우가 많아 문법을 공부하는 도중에 포기하는 경우가 많은 것으로 알려져 있다. 따라서 한국어 학습에서의 어려움 해소를 중점으로 한국어 학습을 도와주는 서비스가 등장한다면, 한류문화가 세계 시장에서 위상을 펼치고 있는 현 상황에서 훌륭한 시장성을 보일 것으로 예상된다.

우리는 한국어 공부를 어렵게 하는 여러 요인중, 높임말 개념과 시제변환에 초점을 맞추어 기계학습 기반의 sequence to sequence 방식을 사용한 reconjugator를 개발하고자 한다.

reconjugator의 핵심 기능은 입력된 문장의 시제를 사용자가 원하는 시제로 바꾸는 것이다. 미리내는 KAIST, 국립국어원 등에서 머신러닝에 활용할 수 있는 text corpus dataset을 제공받았다. 이 데이터셋을 활용하여 원래의 문장과 시제가 변환된 문장을 라벨링하여 기계학습을 할 수 있도록 가공한뒤 sequence to sequence 기법을 이용하여 사용자가 입력한 문장의 시제를 변환할 수 있는 기능을 제공한다.

**2. 관련연구**

**2.1. NLP**

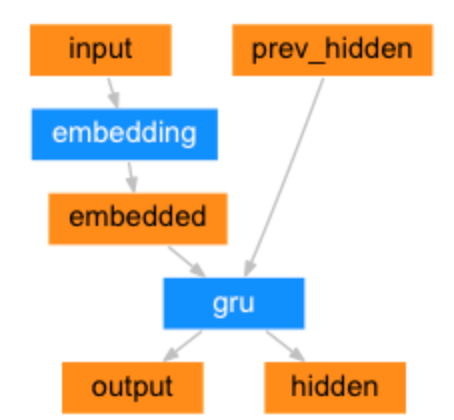
인간의 언어 현상을 컴퓨터와 같은 기계를 이용해서 모사 할 수 있도록 연구하고 이를 구현하는 인공지능의 주요 분야 중 하나이다. 미리내가 외국인을 대상으로 한 한국어 학습기 프로젝트이기 때문에 미리내 서버에서 머신러닝 부분의 거의 대부분을 차지한다.

**2.1.1. Morphological analysis**

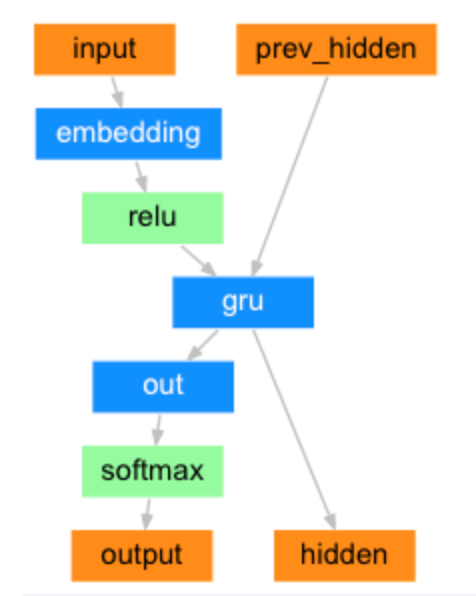
미리내 서비스의 대표적인 기능으로는 한국어 문장을 의미를 가지는 최소 단위인 형태소(Morpheme)로 분할해서 각 부분의 의미나 기능 등을 학습할 수 있게 하였다. 이때 형태소 분석기에 사용되는 기술은 Kakao Labs에서 CNN을 base로한 Khaiii(Kakao Hangul Analyzer III)라고 하는 형태소 분석기를 사용한다.

**2.1.2. sequence-to-sequence**

이번 캡스톤 디자인1 미리내 팀의 메인 주제인 reconjugator 구현에 있어서 사용되는 핵심적인 모델이다. reconjugator는 morpheme의 단위로 잘린 일종의 sequence를 높임말 혹은 시제 변환을 해줌으로써 새로운 morpheme의 sequence를 만들어내야 한다. 따라서 sequence를 input으로 받고 output으로 역시 sequence를 반환하는 sequence-to-sequence model을 사용하게 되었다. 일반적인 인코더와 디코더는 다음과 같다.



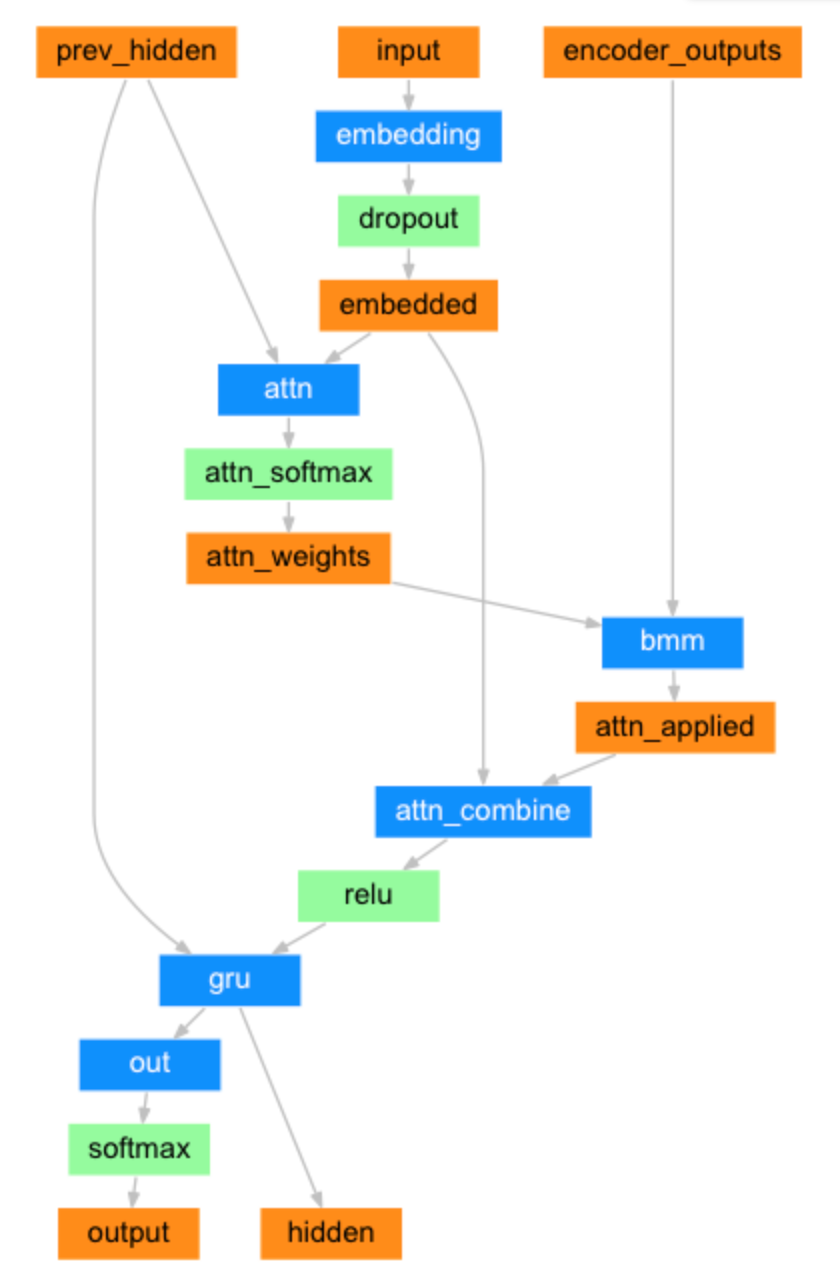
< 그림1 encoder >



< 그림 2 decoder >

**2.1.3.** **Attention mechanism**

위에서 처럼 일반적인 인코더와 디코더 관계에서는 인코더에서 context vector만을 넘겨주게 되어있다. 하지만 이는 인코더에서의 가장 마지막 cell의 정보에 가장 큰 영향을 미쳤으며 이 vector 하나만으로 디코더의 모든 부분을 예측하기엔 정확성이 떨어질 수 있다. 따라서 Attention mechanism을 도입하여 디코더 네트워크가 자기 출력의 모든 단계에서 인코더 출력의 다른 부분에 집중할 수 있게 도와줍니다. 따라서 attention이 적용된 디코더는 다음과 같습니다.



< 그림 3 att-decoder >

**2.2. FRONT\_END**

산업체 주제로 진행하기 때문에 전체적인 소스코드를 공개하는 것이 불가능하다. 따라서 연구 주제로 삼은 reconjugator를 demo하기 위한 mirror code를 ReactJS를 통해 간단히 구현할 계획이다.

**2.3. 기존 연구의 문제점 및 해결 방안**

**2.3.1. 연구의 문제점**

translator에 관해서는 한국어도 seq2seq로 개발된 사례가 존재하지만 현재 미리내에서 계획하는 conjugating하는 비슷한 자료를 찾아보기가 정말 힘들다. 따라서 기본적인 모델을 가지고 미리내에서 자체 개발중이다. 현재까지 진행된 개발상황으로 보아 발견된 문제점은 새로운 단어 즉, 형태소에 대한 예측이 불가능하다는 것이다. 기존에 의미를 구분할 수 있는 형태소 단위로 vocab-dictionary에 저장하고 인코더와 디코더를 통해 학습을 진행했었는데 이렇게 진행되면 학습된 word들에 한해선 제대로 된 예측을 보이나, 새로운 word에 대해선 indexing이 처리되어 있지 않아 에러가 난다. 또한 embedding size를 임의로 늘리고 모델을 evalute하기 전에 새로운 word에 대해서 미리 dictionary에 추가하면서 indexing을 해주어도 에러가 발견되지 않지만 전혀 다른 의미를 가진 word에 대해서 reconjugate된 결과값이 나타나게 된다.

**2.3.2. 해결 방안**

**2.3.2.1. Adding Training Set**

학기 초에 whitespace 기준으로 split하여 word를 지정해 vocab-dictionary를 만들고 이를 통해 sequence to sequence model에 넣어 학습을 진행했었다. 이때 사용된 training pairs는 1,064개였고 evaluate함수를 통과시켰을 땐 over fitting이 의심될 정도로 지나치게 높은 정확성을 보였다. 그리고 vocab-dictionary에 등록되지 않은 word들에 대해선 학습을 진행하지 못 하는 결과를 보였다. 이후 822,942개의 training pairs로 학습을 진행했을 땐 over fitting이 없어진 결과를 확인할 수 있었지만 그렇다고 유의미한 결과값을 얻을 수 있었던 것은 아니다. 하지만 vocab-dictionary에 등록되는 word의 수가 많아야 하는 것은 자명하기에 앞으로의 모델 학습의 개선을 위해서라도 충분한 training set은 필요하다.

**2.3.2.2. Jamo decomposition**

굳이 어렵게 현존하는 모든 단어들을 학습시킬 필요가 없이 아예 의미조차 구분할 수 없는 자음과 모음 단위로 한글을 분해한 뒤에 자음과 모음들을 vocab-dictionary에 저장하는 word로 지정해놓으면 자음과 자음이 합쳐지거나 이중 모음까지 포함하여 총 40개의 자음+모음 조합의 수로 dictionary에 입력될 것이기 때문에 은어나 신조어 등에 대한 대비도 될 것이란 가능성이 있다. 하지만 분해시킨 자음과 모음들을 output을 통해 받고 다시 reconstruct 하는 model을 생성하여 실험적으로 앞서 시도했던 1,064개의 training pairs로 학습을 진행시켜보았으나whitespace단위로 문장을 쪼개었을 때와 같은 over fitting이 의심되는 현상이 나타났다. 따라서 완전한 자모 분해는 좋은 결과값을 기대하기 힘들 것 같다는 결론이 나왔다.

**2.3.2.3. Jamo wild-cards decomposition**

앞서 진행했던 자모분해를 한 문장에 대해서 전체적으로 진행하지 않고, 현재 우리 팀이 주요하게 보는 부분 즉, conjugating하는 경우에 inflectional한 어미 활용의 변화가 있으며 문장의 의미에 가장 큰 영향을 주는 부분인 Verb부분에 한해서 encoder와 decoder를 통과했을 때 fix를 시키기 위해 나름의 wild-card를 적용하는 방법이다. 이를 적용하면 reconjugator를 통과했을 때 의미가 크게 뒤바뀌는 일이 현저하게 줄어들 것으로 예상된다.

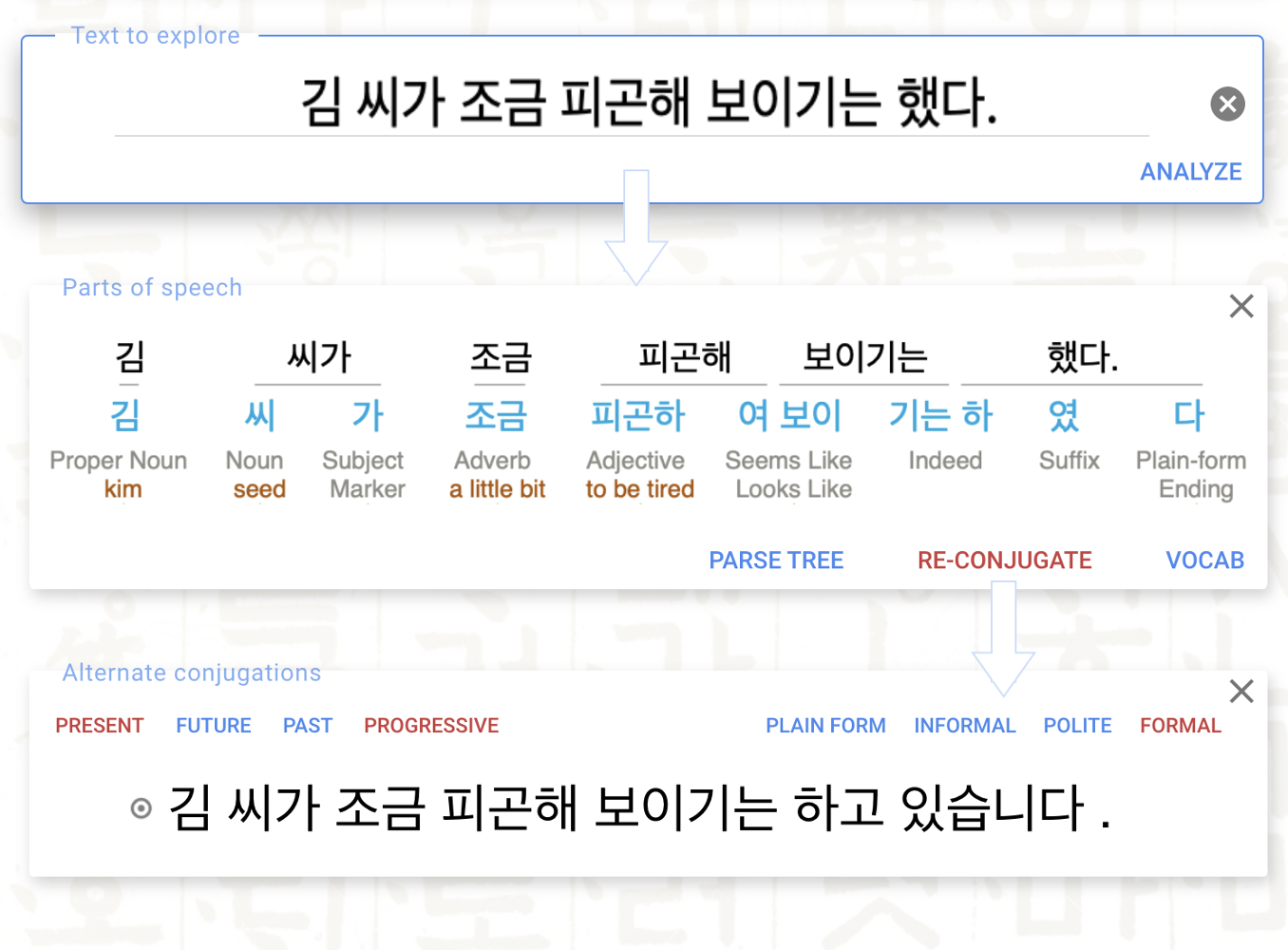
**2.3.2.4. Morpheme embedding**

위에서822,942개의 training pairs로 꽤 많은 훈련 데이터를 통해 학습을 진행했는데도 좋은 결과값이 나오지 않는다는 것을 확인했다. 물론 over fitting이 사라지긴 했지만 문장 의미에 있어서 큰 영향을 미치는 verb phrase의 서술어 자체가 뒤바뀌는 등 문제가 심했고 높임이나 시제 자체도 정확성이 높지 않았다. 이는 whitespace단위로 문장을 쪼개어서 word를 등록했기 때문이라는 것을 여러 모델을 테스트해본 결과 알게 되었다. 따라서 정확하게 형태소 단위만으로 문장을 쪼개어서 embedding과정을 거치고 이를 통해 나온 word dictionary를 가지고 학습을 진행하였다. 결과는 상당히 좋았고 약간의 오차가 물론 존재하긴 했지만 높임이나 시제 변환 부분에 있어서 정확성은 좋았다. 하지만 앞의 verb부분을 fix할 필요성을 느꼈고 이 부분에 대해서 위에서 제시한 wild-card method를 자체 개발하여 적용시키면 좋은 결과가 있을 것이라 예상한다.

**3. 프로젝트 내용**

**3.1. 시나리오**

**3.1.1. Default UI**

****

< 그림 4 메인화면 >

<그림 4>는 어플리케이션의 메인 화면이다. 중앙에 한국어 문장 입력 텍스트 박스가 위치하며, ANALYZE 버튼을 누르면 미리내에서 제공하는 다른 기능인 형태소 분석과 함께 시제가 변환된 예제가 Alternate conjugations 텍스트 박스에 출력된다.

텍스트 박스에서는 과거형, 현재형, 미래형 등의 시제변환이나 높임이나 평서문 등 변환된 문장을 제공한다.

**3.2. 요구사항**

**3.2.1. Sequence to sequence model 개선**

마지막 해결방안으로 고안된 Morpheme embedding을 거친 후의 벡터들을 인코더와 디코더의 초기 embedding vector로 쓰기위해 weight를 제공하고 앞의 verb phrase 부분을 fix하기 위해 wild-card method로 정의해서 model의 정확도를 높여야 한다 한국어에는 예외규칙이 꽤 많기 때문에 wild-card rule을 정하는 데에 시간 소요가 클 것으로 예상된다. 또한 rule을 설정함과 동시에 training pairs의 수도 늘려야 하기에 corpus들을 추가적으로 모을 필요가 있다. 그리고 model 자체의 layer들도 조금씩 바꿔가며 성능을 테스트해봐야 한다.

**4. 향후 일정 및 역할 분담**

* 이예준 : wild-card rule 및 morpheme embedding
* 박기홍 : data preprocessing 및 model test

**5. 결론 및 기대효과**

현재 미리내에서 개발중인 한국어 학습 서비스는 MVP 단계이며, 소수의 유저를 대상으로 테스트를 진행하고 있다. 테스팅 결과 매우 긍정적인 결과를 얻을 수 있었으며, 테스트에서 얻어진 피드백을 보완하는 작업을 진행중이다. Reconjugator 뿐만 아니라 다양한 한국어 학습에 필요한 기능들을 개발 중에 있으며 완성된다면 많은 수의 한국어 학습자들을 만족시킬 수 있는 서비스가 될 것으로 보인다.

**6. 참고문헌**

[1]Github Reference: <https://github.com/jonghwanhyeon/hangul-jamo>

[2] Github Reference: <https://github.com/kakao/khaiii>

[3] Pytorch homepage: <https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html>

[4] RNN 이해하기: <https://dreamgonfly.github.io/blog/understanding-rnn/>

[5] 미리내: Mirinae Co.

[6] KAIST text corpus: http://semanticweb.kaist.ac.kr/home/index.php/KAIST\_Corpus