**Improvement of context-based meaning selection using Bi-Sent2Vec**

2015104203 컴퓨터공학과 이예준

**개 요**

최근Netflix나 Youtube 등 다양한 매체들을 통해 K-POP이나 K-DRAMA등 한류가 해외로 퍼지면서 자연스럽게 한국어 공부를 희망하는 외국인들의 수요가 증가했다. 이러한 외국인들을 대상으로 미리내는 한국어 공부를 돕는 서비스를 제공한다. 이때 여러 서비스 중에서 문맥을 기반으로 한 다의어 의미 결정 기능의 문제점을 Bi-Sent2Vec을 통한 Cross-lingual numerical representation으로 개선함과 동시에 성능을 강화시키고자 한다.

1. **서론** 
   1. **연구 배경**

PSY와 BTS 가 한류 열풍을 일으켰던 것을 보아도 짐작할 수 있겠지만 아래의 통계 수치를 통해 이전부터 꾸준하게 한국어 공부를 희망하는 외국인들이 증가하고 있음을 알 수가 있다.

Timeline

Description automatically generated with low confidence

[그림 1]Ratio of language learner between 2006-2016

이러한 수요를 통해 미리내에서는 외국인들이 한국어 공부에 대한 어려움을 덜어주고 조금 더 쉽게 배울 수 있도록 다양한 서비스를 제공한다.

A picture containing graphical user interface

Description automatically generated

[그림 2] mirinae sentence analysis

위에서처럼 형태소 단위로 문장을 분석해주고 각 형태소를 기준으로 문장에서 Noun이나 Verb 등 어떤 역할을 수행하는지, 또 각 Phrase 를 구분해주고 해당 문장에선 어떤 의미로 쓰였는지 등에 대해 쉽게 알 수 있도록 한다.

* 1. **연구목표**

현재 입력된 문장에서 다의어인 형태소에 대하여 문장의 문맥을 기반으로 어떤 의미(영어)로 쓰였는지 알려주는 기능에 대해 문제점이 존재한다. 본 연구에서는 해당 문제점의 원인을 명확하게 파악하고 해결하여 해당 기능을 개선 및 성능 향상시키는 것을 목표로 한다.

Graphical user interface, application

Description automatically generated

[그림 3] 현재 문제점이 존재하는 다의어 의미 결정 기능

1. **기존 연구**
   1. **Word embedding by FastText**

NLP 분야에서 전처리 과정은 전체 개발 과정 중에서 가장 중요하다고 봐도 과언이 아니다. 아무리 훌륭한 Neural network나 encoder 및 decoder를 개발 및 구성했다고 해도 이를 통과하는 vector들이 연구의 목표에 알맞게 생성되지 않았다면 원하는 결과가 나오지 않을 수 있다. 기존 기능에 대한 전처리 과정에서의 Word embedding method로는 gensim의 FastText가 쓰였다. 다의어 의미 선택을 하는 부분의 output이 영어이고 영어 문장들을 통해서만 결과값을 가져오기 때문에 오직 영어만을 embedding 하는 과정을 거쳤다. 해당 기능을 구현하는 데에 있어서 단어 간 유사도를 파악하는 것이 핵심적인 포인트였기 때문에 대표적인 Distributed representation중 하나인 FastText를 선정한 것으로 분석된다. 또한 비슷한 분산표현 방식인 Word2Vec보다 FastText를 선호한 이유는 Vocab dictionary에 등록되지 않은 단어에 대해서도 유사도 계산을 진행시킬 수 있도록 하기 위해서 OOV처리에 특화된 FastText를 선정했다고 분석할 수 있다.

* 1. **Cosine similarity**

기존 기능의 동작 방식은 다의어를 가진 단어에 대하여 국립국어원에서 여러 영단어들을 가져온 후 이를 입력된 문장의 Papago 번역인 영어 문장과 Cosine similarity를 구해 가장 높은 유사도를 보이는 단어를 선택하는 간단한 과정으로 이루어져 있다.

Graphical user interface, application

Description automatically generated

[그림 4] 기존 기능의 동작 방식

위의 그림처럼 국립국어원에서 가져온 영단어 후보들을 각각 Vi로 매핑해준 뒤에 이를 Papago에서 가져온 전체 문장 벡터인 M과 Cosine similarity계산을 해준다.

* 1. **기존 연구의 문제점**

**2.3.1 Mistranslation of Papago**

간단히 생각해서Cosine similarity를 계산하는 데에 필요한 벡터 중 M 벡터가 잘못된 벡터인 경우이다. 단순 Word embedding 과정에서의 문제가 아닌 Papago로부터 가져온 문장의 영어 번역본이 오역인 경우 의도한 바와 다른 Embedding vector가 생성되는 것이다.

**2.3.2 Several homonym in one sentence**

아래와 같이 한 문장 내에서 동음이의어가 존재하는 경우 기존 연구를 통해 만들어진 의미 결정 기능으로는 하나의 영단어로밖에 결과물이 나오지 않는다.

Graphical user interface

Description automatically generated with medium confidence

[그림 5] 한 문장에 동음이의어가 존재하는 경우

위에서처럼 2가지 배가 존재하고 첫 번째 ‘배’에선 ‘boat’라는 결과물이 예측되길 원하지만 ‘pear’로 도출되는 것을 확인할 수 있다. 이는 같이 유사도를 구하는 과정에서 일반적으로 문장에서 Verb부분이 의미적으로 제일 큰 영향을 미치기 때문이라고 볼 수 있다.

1. **프로젝트** 
   1. **기존 연구와 차이점 및 해결방안**

위에서 제시한 2가지 문제점인 Papago의 오역과 문장 내에서 반복되는 동음이의어 문제를 동시에 해결할 수 있어야 한다. 그렇기 때문에 유사도를 구하는 과정에서 Papago의 번역을 사용하지 않고자 한다. 왜냐하면 전체 문장의 영어 해석이 없더라도 문제가 있었던 첫 번째 ‘배’ 단어로 보았을 때 일반적으로 그 뒤에 따라오는 ‘위에서’를 통해 충분히 유추할 수 있기 때문이다. 굳이 전체 영어 문장을 살필 필요가 없다. 또한 이렇게 의미를 결정하고자 하는 단어의 주변만 관찰한다면 두 번째 문제였던 문장 내 여러 동음이의어에 대해서도 처리해줄 수 있다.

Graphical user interface, application

Description automatically generated

[그림 6] Window size를 통한 부분 의미 결정 강화

이를 구현하기 위해선, 기존 연구 방식에서 ‘영어’만 Word embedding을 진행했던 것과는 달리 한국어-영어의 관계성을 맞추어 동시에 Embedding 하는 과정을 거쳐야 한다. 이를 통해 생성된 한국어-영어 Embedding vector들을 통해 Embedding method로 쓰이는CBOW방식과 유사하게 특정 window-size를 걸어주고 부분적으로 Cosine similarity를 구해줘야 한다.

* 1. **프로젝트 내용**

**3.2.1 Preprocessing with tagging from corpus**

현재 미리내에서 보유하고 있는 여러 corpus들을 한국어-영어 동시 Embedding 즉, Bi-Sent2Vec에 학습시키기 위한 형태로 \_ko, \_en tag들을 부여해주어 pairs를 만들어줘야 한다. 한국어 문장은 Kakao의 CNN을 이용해 개발된 오픈소스인 Khaiii를 통해 형태소 단위로 분해하고 영어 문장은 공백을 기준으로 분해해준다. 또한 이때 Embedding을 위한 학습에 단순한 공백이라도 결과에 크게 영향을 미칠 수 있으므로 정규표현식을 통해 필요없는 값과 필요한 값을 제대로 여과해줘야 한다.

**3.2.2 Training & Experiment with Bi-Sent2Vec**

**Text

Description automatically generated**

[그림 7] Hyper-parameter for training with Bi-Sent2Vec

전처리 과정이 끝난 뒤엔 본격적으로 Bi-Sent2Vec model을 통해 한국어-영어 문장을 Training시켜주고 이를 통해 형태소 각각의 Vector를 구해준다. Bi-Sent2Vec 역시 FastText를 기반으로 개발된 model이기 때문에 OOV(Out of vocabulary)에 대해서도 새롭게 벡터를 생성해줄 수 있고 당연히 Distributed representation이라 문맥 기반 해석에 탁월한 효율을 보인다. 이후엔 위의 그림처럼 hyper-parameter들을 조정해가며 최적의 효율을 낼 수 있는 parameter들을 추려낼 계획이다.

**3.2.3 Preparing Demo with React.js**

기본적인 UI/UX 등은 위의 예시 그림들처럼 현재 미리내 서비스와 유사하게 구현할 것이며, 데모에 있어서 가장 중점적인 부분은 기존 기능의 문제점들과 그에 대한 문제점들이 해결되어 성능이 더욱 개선된 문맥 기반 의미 결정기를 보이는 것이다.

1. **추후 진행 일정**

|  |  |  |
| --- | --- | --- |
| 진행 주차 | 담당 | 내용 |
| 4월 2주차 | 이예준 | Preprocessing by Khaiii&tagging for training |
| 4월 3주차 | 이예준 | Training with Bi-Sent2Vec - 1 |
| 4월 4주차 | 이예준 | Training with Bi-Sent2Vec - 2 |
| 4월 5주차 | 이예준 | Handling any potential issues with training |
| 5월 1주차 | 이예준 | Experiment for training Bi-Sent2Vec with Hyper-parameters - 1 |
| 5월 2주차 | 이예준 | Experiment for training Bi-Sent2Vec with Hyper-parameters - 2 |
| 5월 3주차 | 이예준 | Test performance comparing to previous service |
| 5월 4주차 | 이예준 | Handling any potential issues with test & Preparing Client-side(React.js) for demo |
| 6월 1주차 | 이예준 | Development of Client & preparing presentation |
| 6월 2주차 | 이예준 | Demo presentation |
| 6월 3주차 | 이예준 | Demo presentation |

1. **결론**

미리내의 문맥 기반 의미 결정 기능의 궁극적인 목표는 어떠한 문장에서도 각각의 단어가 해당 문장에서 어떤 의미로 쓰였는지 정확하게 선택해주는 것이다. 기존 연구에서는 각 단어가 문장 내에서 고유한 경우 효율적인 선택이 가능했지만 중복해서 나타나는 동음이의어가 존재할 경우 그 기능이 쉽게 무너진다. 따라서 이번 프로젝트에서 Bi-Sent2Vec을 통해 얻어진 벡터들을 통해 특정 window-size에 따라 얼마나 정확하게 의미 선택을 가려줄 것인지는 시도해봐야 알겠지만, 만약 이 프로젝트에서 유의미한 성과를 거두게 된다면 여러 복합 문장에 대해서도 다의어 및 동음이의어를 각 단어에 알맞게 매칭시켜줄 수 있는 훌륭한 서비스가 될 것이라 확신한다.

**참고 문헌**

<https://mirinae.io/>

<https://github.com/epfml/Bi-Sent2Vec>

<https://github.com/facebookresearch/fastText>

<https://papago.naver.com/>

<https://stdict.korean.go.kr/openapi/openApiInfo.do>

<https://www.statista.com/>