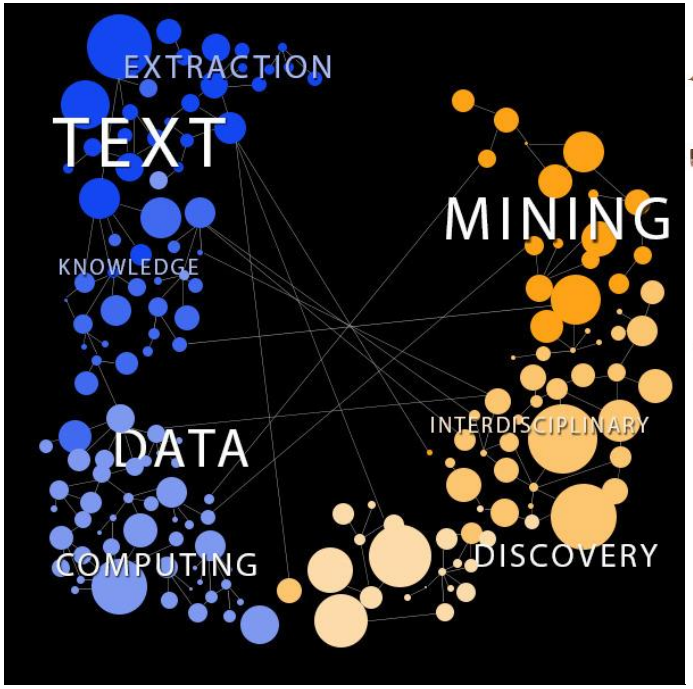
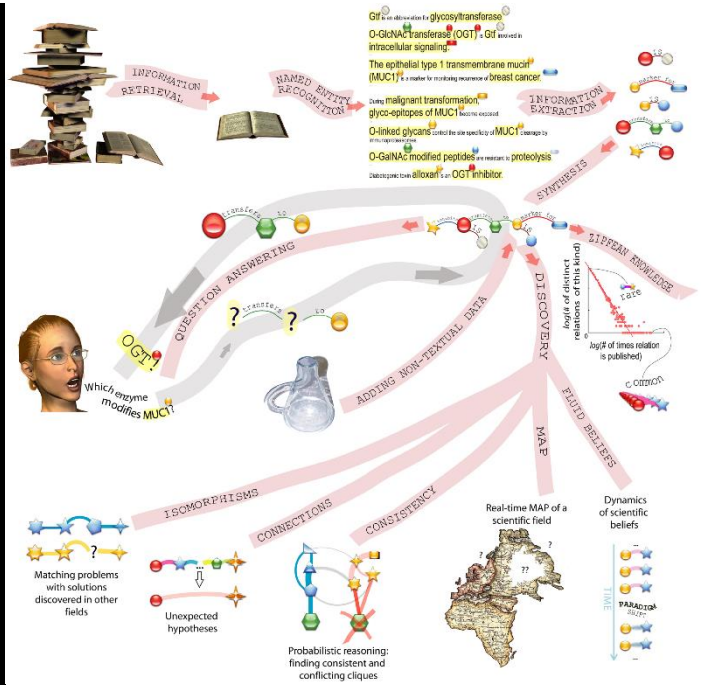




Lecture 1: Introduction to Text Analytics (QA)

Pilsung Kang

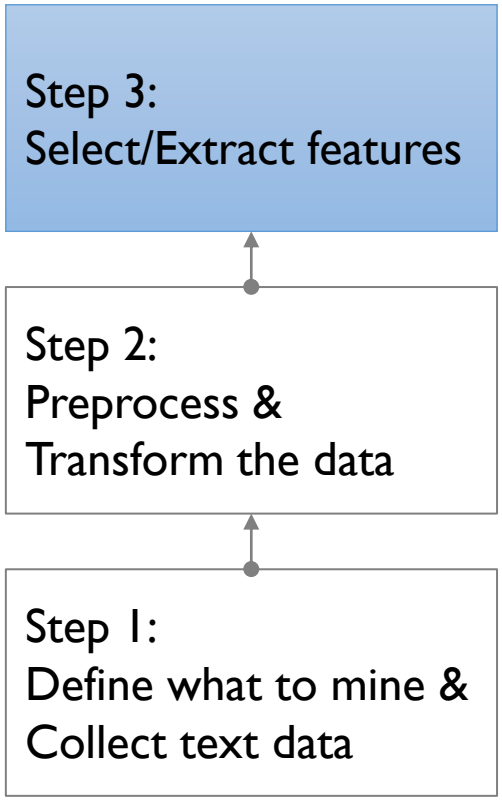
School of Industrial Management Engineering

Korea University

Q1 (최정우)

- Introduction to Text Anlaytics:Part2 강의에서 Feature Selection과 관련된 질문입니다. Label이 있는 Supervised Learning Task 에서 Feature Selection 방법은 강의 중에 말씀해주신 방법들이 있음을 이해하였습니다.
- Q. Label이 없는 Unsupervised Learning에서 Feature Selection 방법 관련해서도 알고 싶습니다. (관련된 몇 가지 방법론과 관련 논문이 있는 것으로 아는데, Unsupervised Learning Task에서는 일반적으로 Feature Selection이 아닌 Feature Extraction 방법으로 접근 하는지 궁금합니다.)

A Simplified Process of Text Analytics



Reduce the number of features

Word	S1	S2
John	1	1
Likes	2	1
To	1	1
Watch	1	1
Movies	1	0
Also	0	1
Football	0	1
Games	0	1
Mary	1	0
too	1	0

Word	S1	S2
Likes	2	1
Watch	1	1
Movies	1	0
Football	0	1
Games	0	1

AI-I (허종국)

- Label 이 없는 상황에서 feature 를 추출하는 방법이 궁금하신건가요??(해당 질문에 적절한 답변인지는 모르겠지만, 적용 분야를 **NLP**로 한정 지으신 것 같지 않아서 답변 달아봅니다..) 제가 본 경우 보통 selection 보다는 extraction 을 쓰는 경우를 더 많이 본 것같습니다. **Label** 이 없는 경우, **feature** 를 잘 뽑았는지에 대한 평가 지표를 사용하기 위해 **pseudo label** 을 직접 만드는 경우가 있습니다. 예를 들어, (NLP task는 아니지만 CV 분야의 경우), 원본 이미지를 돌리거나(Rotation), 직소 퍼즐(Jigsaw)을 만들어 이러한 augmentation의 결과 값을 pseudo label로 둡니다. 기존 이미지에서 몇 도를 회전하였는지, 퍼즐 이미지를 어떻게 풀어야 원본 이미지가 나오는 지를 f(x) 라는 인코더가 학습하게 하면 좀 더 feature 를 잘 추출하게 됩니다. (이러한 feature extraction 방식을 Self-supervised learning 에서 pretext task라고 칭합니다.) pseudo label 을 만들지 않는 경우(Pretext task가 없는 경우) 는 데이터에 서로 다른 어그멘테이션이나 인코딩을 진행한 후, 같은 데이터로부터 나온 것은 유사도가 높게, 다른 데이터로부터 나온 것은 유사도가 낮게 학습하는 Contrastive Learning이 있습니다. CV 분야에서 예시를 든 것이지만, 해당 내용들을 찾아보신다면 NLP task에서도 적용할 수 있거나, NLP task에 맞는 representation learning을 접하실 수 있을 것 같습니다.

AI-I (허종국)

- Word2Vec etc도 unsupervised learning 으로 볼 수 있을 것 같습니다. Word2Vec의 skip-gram 의 경우를 예로 들면, 문장 내의 인접한 단어들끼리는 유사도가 높을 것이라는 가정 하에 len(corpus) 보다 적은 차원의 vector(feature)로 임베딩 (feature extracting) 하는 것이니까요

A2-1 (김탁영)

- **[1] "Unsupervised Learning Task에서는 일반적으로 Feature Extraction이 사용되는지"에 대한 답변**
- 제 개인적인 의견은 "맞다" 입니다. Feature Extraction은 별도의 label이 없어도 데이터 자체의 구조가 나타내는 분산, 거리정보 등을 사용하여 데이터의 본래적 특징을 잘 보존하면서 저차원으로 나타낼 수 있기 때문입니다. 강필성 교수님의 2학기 수업인 Business Analytics에서 다루는 Feature Extraction 기법들도 class label을 사용하지 않는 방법들이기 때문에 관련해서는 유튜브 영상을 확인하시면 더 도움이 될 것 같습니다.그런데 Unsupervised Feature Selection 키워드로 찾아보니 최근까지도 논문이 출판되고 있어서, 실무 관점에서는 다를 수 있지만 연구는 지속적으로 이루어진다고 할 수 있겠습니다.

A2-1 (김탁영)

- **[2] "Label이 없는 Unsupervised Learning에서 Feature Selection 방법을 알고싶음"에 대한 답변저도 질문을 듣고** Unsupervised Feature Selection**이 있다는 것을 처음 알게 되어 자료를 찾아보았습니다. 지금부터 설명드리는 내용은 아래 적어놓은 논문을 기반으로 전달드리는 점을 미리 밝힙니다.**
- - **논문 제목** 1: An efficient unsupervised feature selection procedure through feature clustering (2020)
  - **링크** 1: https://www.sciencedirect.com/science/article/pii/S0167865519303976
- - **논문 제목** 2: Unsupervised Feature Selection based on Feature Relevance (2009)
  - **링크** 2: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5212453
- - **논문 제목** 3: Efficient Feature Selection via Analysis of Relevance and Redundancy (2004)
  - **링크** 3: https://www.jmlr.org/papers/volume5/yu04a/yu04a.pdf

A2-1 (김탁영)

- **[2-1] Feature Selection의 접근 방법**
- Feature Selection 기법은 크게 ① feature relevance ② feature redundancy 관점으로 접근합니다. ①은 feature와 어떠한 지표(class 정보 또는 기타 다른 성능지표가 될 수 있음)와의 관계를 의미하여, 기대하는 수치가 나오는 데 영향을 많이 미치는 변수를 중요하다고 판단할 것입니다. ②는 feature끼리 관련성을 측정하여 많이 겹치는(redundant) feature는 삭제하는 방식으로 접근할 것입니다. 위의 2004년 논문이 feature의 redundancy를 처음으로 정의한 것 같은데, 자세한 내용은 생략하겠습니다.

A2-1 (김탁영)

- **[2-2] Label 유무에 따른 Feature Selection 분류**
- 그리고 label의 유무에 따라 Supervised/Unsupervised Feature Selection으로 구분됩니다. Supervised FS는 출력값(label)이 존재하기 때문에 특정 feature를 추가/제거하였을 때 예측성능에 영향을 미치는 정도를 판단하여 selection을 수행하는 방법입니다. 즉 Supervised FS는 위의 ① feature relevance 관점으로 접근한다고 할 수 있습니다.
- 반면 Unsupervised FS는 별도의 label이 없기 때문에 다양한 비지도학습 방법론(예: 클러스터링)을 사용하여 데이터 자체가 갖고 있는 특징을 파악하고 relevance를 측정합니다. Label이 없는데 어떻게 relevance를 측정하는지 감이 잘 오지 않을 수 있어서 위에 언급한 2009년 논문의 내용을 설명드리자면, unsupervised 상황에서 relevance가 높은 feature는 전체 n개의 feature 중 가장 "informative"한 k개의 feature입니다. 해당 논문에서는 relevance 측정을 위한 지표로 Mutual Information을 활용한 Relevance Degree(이하 RD)를 사용하는데, 절차를 간단하게 설명하면 먼저 클러스터링을 통해 특정 개수의 클러스터를 만듭니다. 그 후 각 클러스터마다 같은 클러스터 내의 feature와는 relevant하고(RD가 크고), 다른 클러스터 내의 feature와는 relevant하지 않은 k개의 feature를 선택합니다. 그렇다면 클러스터마다 가장 informative한 몇 개의 feature들이 선택되는 것이죠. 논문에서 언급은 없었지만 redundancy 관점에서는 같은 클러스터 내의 feature들이 겹치는 정도를 고려하지는 않은 것 같아 해당 방법론은 relevance만 고려한 unsupervised feature selection이라 생각합니다.
- UFS의 초기 방법론들은 주로 ① feature relevance에만 집중하여 FS를 수행하다가 시간이 흐르면서 ② feature redundancy를 함께 고려하는 추세인 것 같습니다. 위에 적어놓은 2020년의 논문은 (유전알고리즘의 아이디어를 약간 차용한) 클러스터링 기법을 사용하여 feature redundancy도 함께 고려하는 selection을 수행합니다.

Q2 (김정원)

- Part 2 - Lexical Analysis에서 3번째 단계로 언급된 Stemming과 Lemmatization에 대해 질문 드립니다.
  두 작업의 장단점이 있고 용도에 따라 어떤 작업을 할지 선택하면 된다는 점까진 이해했는데요. 이를 한국어 데이터에는 어떻게 적용하게 되는지 궁금합니다. 한국어는 특히 동사나 형용사 같은 경우 영어에 비해 변형이 심하기 때문에 stemming이 매우 위험한 방식일 것 같아서요. 예를 들어 '먹다'는 먹었다/먹고 있다/먹을 수 있다 등으로 변형되기 때문에 stemming을 하면 '먹' 밖에 남지 않고. '예쁘다'도 예뻤다/예쁘겠다/예쁘구나 등으로 '예'만 남는 식일 듯 한데요.
  제가 추측한 것이 맞다면 한국어는 대부분 lemmatization만 사용하게 되는 건지, 아니면 그 외 한국어에 적합한 정규화 방법이 있는 것인지 궁금합니다. 답변 미리 감사합니다!

A2-1 (김지은)

- 제가 이해한 내용 기준으로 답변 드리겠습니다. 혹시 추가할 말씀이 있으시면 답변주시면 감사하겠습니다.질문해주신 내용에 대해 아래의 깃허브에 정리된 내용을 참고하였습니다.
- 제가 이해한 바로는 어간의 형태가 유지되고, 어미만 다른 어미로 교체되는 규칙 활용의 경우 stemming 방식이 적용되기에 용이하지만 그 외의 불규칙 활용 방식으로 어간, 어미 형식이 변하는 경우 lemmatizer을 만들어서 음운 변동 규칙에 따라 원형을 찾아내는 것으로 알고 있습니다. 한국어의 경우 음운변동 규칙이 정리되어 있기 때문에 해당 문법 규칙을 기준으로 용언의 어간과 어미를 인식한 lemmatization 방식을 사용하여 한국어 데이터에 적용되는 것으로 이해하였습니다.

A2-2 (이윤승)

- 한국어에서 lemmatization이 stemming보다 많이 활용되는 것 같습니다. 앞서 김지은님이 첨부해주신 참고자료를 보면, 규칙활용이 적용되는 단어는 stemming만으로도 용언의 의미 표현이 가능하지만 불규칙활용이 적용되는 단어는 stemming을 활용하면 의미를 반영하지 못하여 이때 lemmatization을 활용하게 됩니다.
- KoNLPy에서 Okt(Open Korea Tex)라는 형태소 분석기에는 morphs라는 함수가 존재하는데, stem 옵션을 True/False로 선택할 수 있습니다. Stem=False 로 둔다면, lemmatization만을 사용할 수도 있고, 사용자가 원한다면 stem=True를 통해 stemming을 활용한 어간 추출하는 방법도 활용할 수 있습니다.(참고자료)
  https://dacon.io/competitions/official/235658/codeshare/1808
  https://cheris8.tistory.com/6

A2-2 (이윤승)

- 이 질문을 보고 검색을 하다가, '김기현의 자연어 처리 딥러닝 캠프'에서 stemming/lemmatization 관련한 새로운 내용을 찾았는데, 같이 공유하면 좋을 것 같아 아래 상세하게 적어두었습니다.
- p.248를 참고하면, 한국어 데이터를 활용한 프로젝트에서 stemming/lemmatization이 무조건 필요한 것은 아니라는 내용이 있습니다. Stemming과 lemmatization은 딥러닝 이전의 전통적인 머신러닝 방법에서 희소성 문제를 해결하기 위해 주로 사용이 되었지만, 현재 딥러닝 모델을 사용하면서 여러 차원축소 기법을 사용할 수 있기 때문에 stemming/lemmatization을 반드시 사용하지는 않는다고 합니다. 실제로 감성분석과 같은 text classification을 할 때에는 stemming/lemmatization을 적용하지 않고 베이스라인 성능을 확보한다고 합니다.
- 예를 들어, "나는 학교에 가요."라는 문장은 stemming/lemmatization을 적용하면, "나 학교 가 ." 라고 추출됩니다. 이외에도 "나만 학교에 가요.", "나는 학교를 가요."의 문장은 모두 "나 학교 가 ."로 표현되게 됩니다. 이 예시에서는 stemming/lemmatization이 오히려 문장이 내포하는 의미를 삭제하기 때문에 stemming/lemmatization을 필수적으로 사용해야 한다는 것은 오해라고 책에서 밝히고 있는 것 같습니다.

A2-3 (강필성)

- Word Piece Model

Word piece, units of words

Word Piece Model 은 제한적인 vocabulary units, 정확히는 단어를 표현할 수 있는 subwords units 으로 모든 단어를 표현합니다. 수백만개의 단어를 포함하는 데이터를 표현하기 위해서 bag of words model 는 단어 개수 만큼의 차원을 지닌 벡터 공간을 이용합니다. RNN 처럼 word embedding vectors 를 이용하는 모델은 단어 개수 만큼의 embedding vector 를 학습합니다. 단어의 개수가 많을수록 차원이 크거나 모델이 무거워집니다. 이를 해결하기 위해서는 제한된 (finite) 개수의 단어를 이용해야 합니다. 그러나 자주 이용되지 않는 수 많은 (long-tail) 단어들을 무시하면 미등록단어 문제가 발생합니다.

언어는 글자 (characters)를 subword units 으로 이용합니다. 영어는 알파벳을 유닛으로 이용합니다. 대부분의 영어 단어는 몇 개의 글자가 모여 하나의 단어를 구성합니다. 즉, 하나의 유닛이 어떤 개념을 지칭하기는 어렵습니다. 유닛이 모호성을 지닙니다. 중국어는 한자를 units 으로 이용합니다. 중국어도 여러 글자가 모여 하나의 단어를 이루기도 하지만, 한 글자로 구성된 단어도 많습니다. 영어보다는 유닛의 모호성이 줄어듭니다. 동음이의어의 문제가 남아있지만, 가장 모호성이 적은 방법은 모든 단어를 단어의 유닛으로 이용하는 것입니다.

https://lovit.github.io/nlp/2018/04/02/wpm/

A2-3 (강필성)

- Word Piece Model

그러나 토크나이징 방법에 따라 모호성이 적은 최소한의 유닛을 만들 수도 있습니다. 아래의 세 문장을 다음의 요소들로 나눈다면 이 유닛들은 의미를 보존하면서도 재활용이 될 수 있습니다.

```
공연은 끝났어 -> ['공연-' + '-은' + '끝-' + '-났어']
공연을 끝냈어 -> ['공연-' + '-을' + '끝-' + '-냈어']
개막을 해냈어 -> ['개막-' + '-을' + '해-' + '-냈어']
```

이 유닛들은 '개막공연' 이라는 복합명사도 분해하는데 이용될 수 있습니다. '개막공연'을 독립된 유닛으로 만들 필요가 없습니다.

```
개막공연을 끝냈어 -> ['개막-' + '공연-' + '-을' + '끝-' + '-냈어']
```

그런데 문제는 위처럼 토크나이징을 하려면 해당 언어의 언어학적 지식과 학습데이터가 필요합니다. 그러나 언어가 다르고, 도메인이 다르면 이를 준비하는 것은 어렵습니다.

https://lovit.github.io/nlp/2018/04/02/wpm/

A2-3 (강필성)

- Word Piece Model

"나는 오늘 아침밥을 먹었다." =>

- idx : [2, 5951, 6105, 7623, 3482, 3216, 1472, 5595, 130, 3]
- tokens: ['[CLS]', '나는', '오늘', '아침', '##밥', '##을', '먹', '##었다', '.', '[SEP]']
- offset: [(0, 0), (0, 2), (3, 5), (6, 8), (8, 9), (9, 10), (11, 12), (12, 14), (14, 15), (0, 0)]

"교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정" =>

- idx : [2, 25910, 5691, 12664, 130, 130, 5821, 6481, 5659, 130, 130, 5577, 13610, 3]
- tokens: ['[CLS]', '교도소', '이야기', '##구먼', '.', '.', '솔직히', '재미는', '없다', '.', '.', '평점', '조정', '[SEP]']
- offset: [(0, 0), (0, 3), (4, 7), (7, 9), (10, 11), (11, 12), (12, 15), (16, 19), (20, 22), (22, 23), (23, 24), (24, 26), (27, 29), (0, 0)]

https://lovit.github.io/nlp/https://keep-steady.tistory.com/37/wpm/

Q3 (천주영)

- Introduction to Text Analytics part2 강의에서 Text Transformation과 관련된 질문 드립니다.
  단어를 distributed representation 으로 표현했을 때, 비슷한 관계를 가진 단어쌍은 유사한 형태의 벡터를 가져 semantic relationship이 유지되는 것으로 이해했습니다.
  (대한민국->서울 과 일본->도쿄의 경우 유사한 형태의 벡터를 보임)
  이때 만약 단어의 의미가 동일하다면 언어와 관계 없이 유사한 형태의 벡터가 되는지 궁금합니다.
  예를 들어, 'cat -> dog'벡터와 '고양이 -> 개'가 서로 비슷한 벡터로 표현이 되는지, 혹은 언어가 다르다면 전혀 다른 형태로 표현이 되는지 궁금합니다.

A3-1 (윤훈상)

- Distributed Representation을 Word2Vec으로 예시를 들면, 하나의 Embedding을 구성함에 있어 주위 단어와의 관계를 반영하여 계산하는 양상을 보입니다. 따라서, 해당 Representation를 두 개의 다른 언어에 대하여 적용했을 때, 똑같은 대상에 대한 표현이 다르더라도 문장 구조가 유사하면 비슷한 Representation을 나타낼 것이라고 생각합니다.그렇다면, 영어-한국어와 같이 주어와 술어의 위치가 다른 언어 간에는 해당 Representation들의 관계가 보존이 될 지 궁금합니다. 처음에는 잘 보존되지 않을 것이라고 생각했으나,

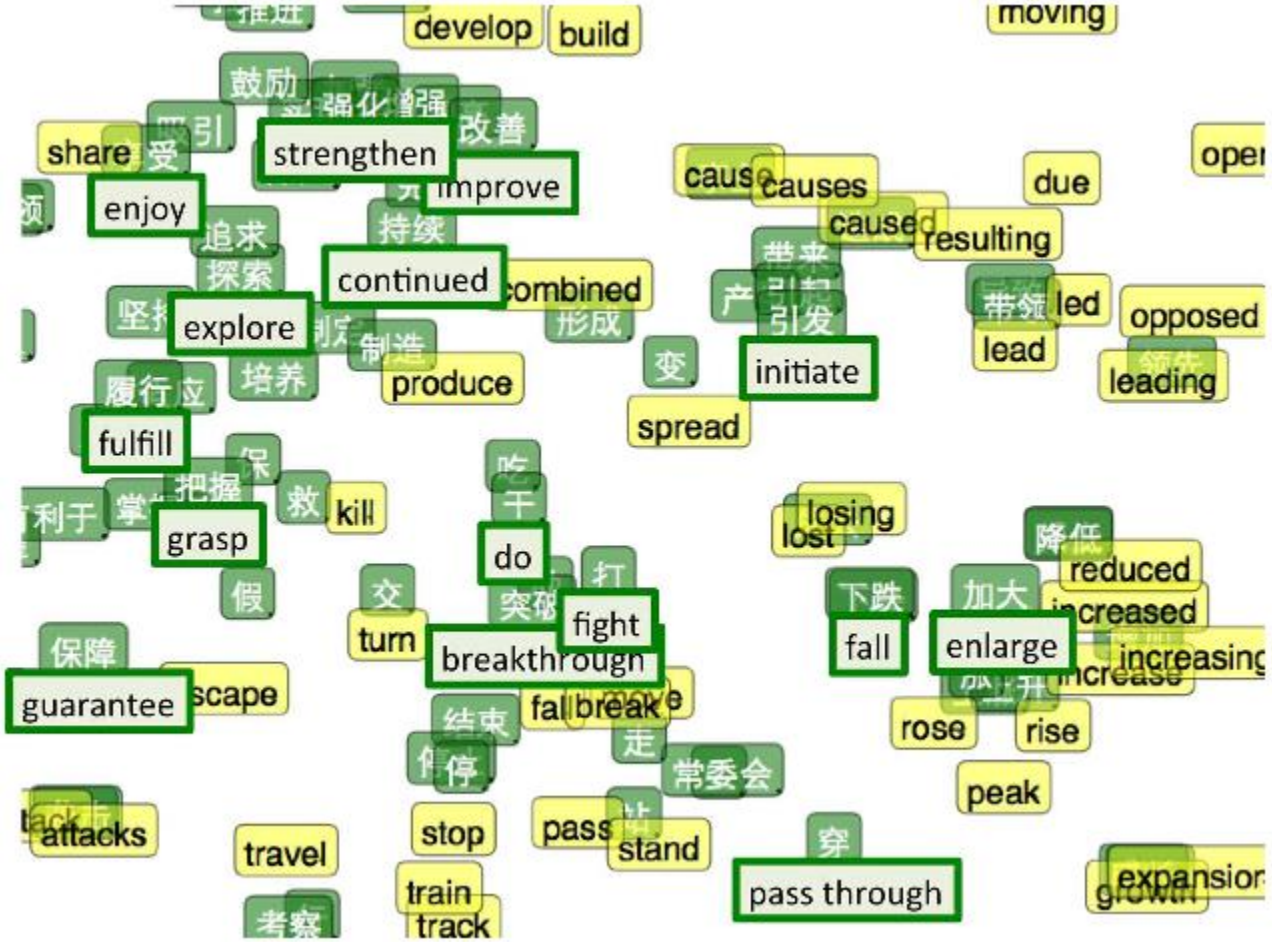
  Exploiting Similarities among Languages for Machine Translation (Mikolov et al, 2013)

  라는 Word2Vec 저자가 직접 작성한 Paper를 살펴보면 언어 구조가 달라도 큰 차이가 없다고 표현하고 있습니다.

A3-1 (윤훈상)

- 간단하게 Paper의 내용을 말씀드리자면, 두 언어 별로 Word Embedding을 구축한 뒤, 문장 별로 단어들의 Embedding이 유사한 단어들로 대체하면 Translation이 될 것이라고 주장하고 있습니다. 그리고 이 때, 영어와 스페인어 / 체코어 / 베트남어와의 단어 Translation을 진행하는데 모든 Test 언어에 대하여 좋은 성능을 나타낸다고 하며, 언어 간의 차이가 없음을 주장하고 있습니다. (하지만 실험 결과를 살펴보면, 저자는 큰 차이가 없다고 하지만 Precision 수치들을 보면 Spanish, Czech, Vietnamese 순으로 낮아지기에, 어느 정도 성능 차이는 있다고 생각합니다.)

A3-1 (윤훈상)


develop
build
moving
share
strengthen
improve
enjoy
continued
explore
produce
fulfill
grasp
kill
cause
causes
caused
resulting
due
led
lead
opposed
leading
initiate
spread
losing
lost
reduced
increased
fall
enlarge
rose
rise
peak
do
fight
breakthrough
break
turn
guarantee
attacks
travel
stop
train
track
pass
stand
pass through
expansion

A3-1 (윤훈상)

- 결론적으로 말씀드리자면, 문장 구조가 유사한 언어들끼리는 Word-Embedding이 유사하게 나타날 것이나, 구조의 차이가 큰 경우에는 Discrepancy가 발생할 수 있다고 말씀드릴 수 있을 것 같습니다.또한 언어별로 상이할 수 있는 Embedding들에 대하여, 같은 의미를 갖는 단어라면 유사한 Embedding을 갖도록 하는 연구하는 분야가 있는데 이를 'Cross Lingual Word Embedding Models' 라고 합니다. 아래의 블로그에서 자세히 설명하니 관심이 있으실 경우 살펴보시는 것이 좋을 것 같습니다. https://ruder.io/cross-lingual-embeddings/index.html#fn2

A3-2 (안시후)

- 우선 질문자님께서 옳은 방향으로 이해하고 계신 것이 맞습니다.
  이 분야에서 가장 유명한 예시로 다음과 같은 연산이 있습니다.
  "King - Man + Woman = Queen"
  위 연산으로 성별을 이용하여 왕에서 여왕으로 변환해주는 예제입니다.
  때문에 질문자님께서 예시로 들어주신 Cat to Dog Vector의 경우 Word2Vec 기반으로 연산을 해보면,
  "Cat - English + Korean = 고양이"와 "Dog - English + Korean = 개" 두 방식으로 변환을 할 수 있습니다.
  따라서 고양이 to 개 Vector로 표현이 될 것이며
  한국어 데이터와 영어 데이터의 양이 충분한 Word2Vec과 같은 모델을 만든다면 Cat to Dog Vector와 고양이 to 개 Vector가 상당히 유사한 Vector로 표현될 것으로 추정됩니다.Word2Vec 관련 논문들을 소개해준 글을 참고하여 작성하였습니다.
  직관적으로 설명을 해주어 이해하기 좋으니 한 번 읽어보시면 좋을 듯합니다.
  https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/

A3-3 (강필성)

- Language-agnostic BERT Sentence Embedding

A3-3 (강필성)

- Language-agnostic BERT Sentence Embedding

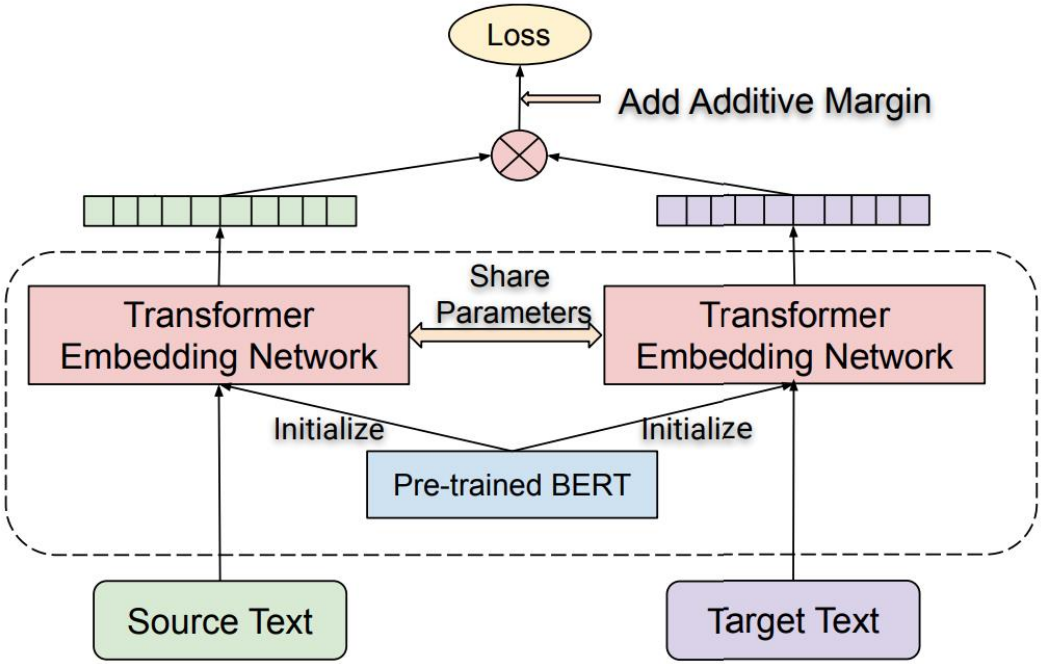


Figure 1: Dual encoder model with BERT based encoding modules.

https://ai.googleblog.com/2020/08/language-agnostic-bert-sentence.html