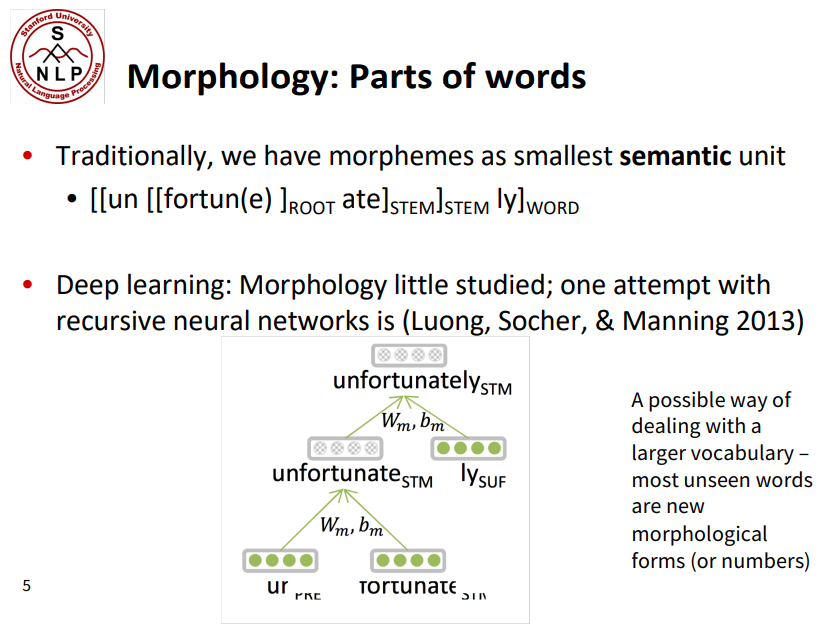
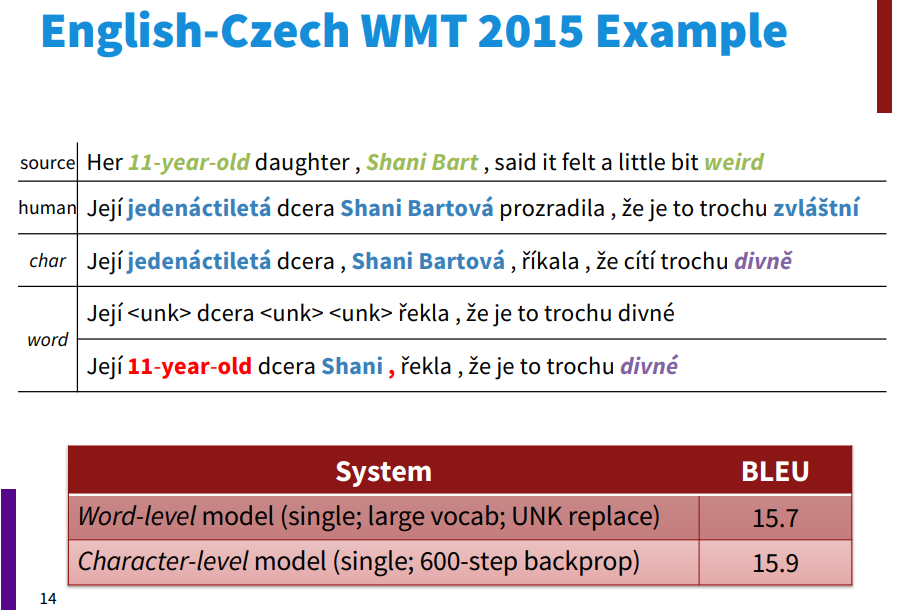
Linguistics의 first는 phonetics 이다. (인간 음성의 학문, 소리의 이해?)



이렇게 한 단어 안에서도 여러 부분이 semantic 하게 작동할 수 있다. (즉 morpheme 단위로 단어를 쪼갤 수 있고 이 부분을 이용할 수는 없을까? = subword model의 idea)

Word-Level: Large Open Vocab 필요, 단어 단위이다 보니까 모델 크기는 Character 단위보다 작음, OOV 문제 생길 수 있음, 언어의 종류마다 segmentation 등 단어의 특성이 다르기 때문에 word 단위로 나누기 애매할 때가 있음.

Character-Level: unknown words에 대해 embedding 가능, 유사한 스펠링들은 유사한 임베딩을 가짐, OOV 문제 해결, 다양한 언어의 특성을 어느정도 일반화하여 segmentation의 문제에 애매하지 않을 가능성이 큼



NMT 작업을 character vs word level 모델로 학습시켜 평가해 보았다.

Sub-Word Models: two trends

1. Word-level model 과 same architecture(word pieces 단위 사용(character 위 word 아래 정도의 단위))
2. Hybrid architecture: main으로 words 이용, 서브로 characters 이용

<https://wikidocs.net/22592>

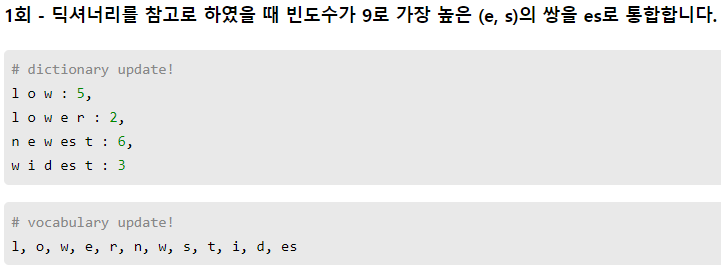
BPE(Byte Pair Encoding), Sennrich(2016)

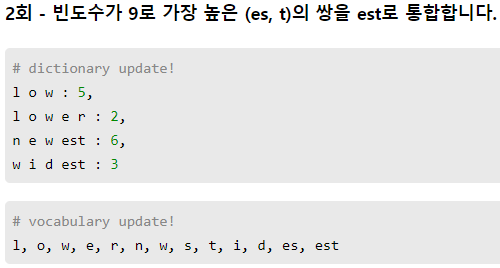
기계가 아무리 많은 단어를 학습해도 세상의 모든 단어를 알 수는 없다. 만약 입력 문장에 모르는 단어가 발생할 시 OOV, UNK 라고 표현하며 문제 푸는데 방해가 됩니다.

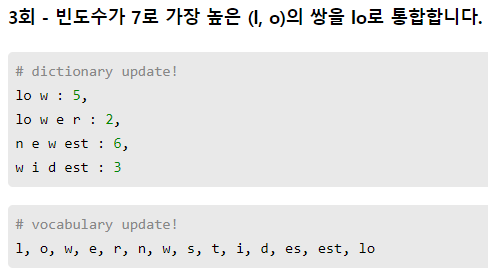
따라서 word를 여러 서브 word로 분리하여 인코딩 및 임베딩하면 OOV, 희귀 단어, 신조어 문제를 완화시킬 수 있습니다.

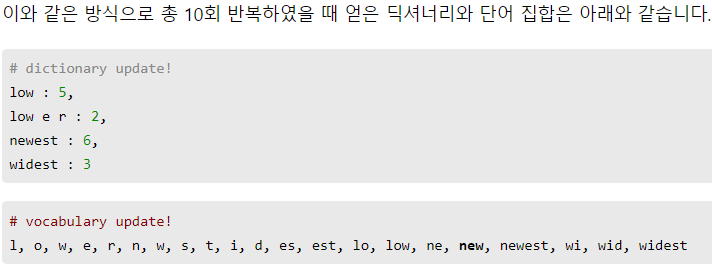
BPE는 subword segmentation 알고리즘이다.

입력 단어를 byte(글자) 단위로 바꿔 알고리즘을 통해 인코딩하는 작업.





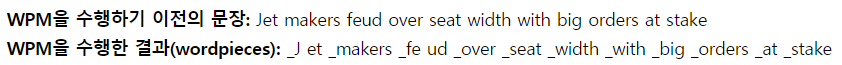




만약 lowest를 인코딩하길 원한다면 기존 dict에서는 OOV 였지만 low와 est(low와 est라는 새로운 byte)를 통해 OOV가 아니게 된다. 이렇게 두 단어를 따로 byte 단위로 encoding 하게 된다. (10회는 하이퍼 파라미터)

왜 그냥 character 단위로 사용하지 않느냐? Rare-word에는 character 단위가 되기 때문에 유리할지 몰라도 의미성이 떨어질 수가 있다. 따라서 자주 만나는 character들은 하나의 byte unit으로 묶어줘서 OOV를 해결하면서 의미성을 가져오는 dict를 만들어 encoding 하자라는 것이다.(BERT와 GPT-2에서 사용했음.)

WPM(WordPiece Model) 도 역시 BPE를 이용했지만 빈도수가 아닌 병합되었을 때 코퍼스의 우도를 가장 높이는 쌍을 병합하여 vocab에 추가하도록 한다.



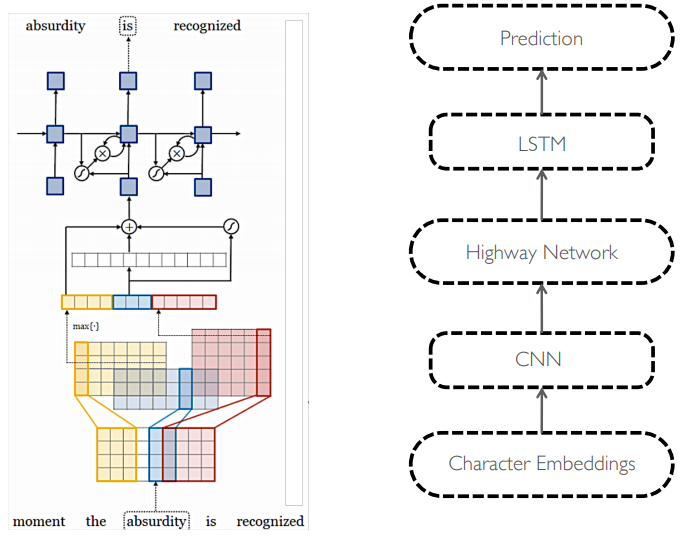
\_는 WPM 수행 시 모든 단어 앞에 붙여 띄어쓰기 분리, 문장 복원 장치이다.

이렇게 단어를 subword 단어로 나누고, vocab도 새로 구성한 뒤 입력으로 들어가게 된다.

Sentencepiece model 도 비슷한 것 같다..?

Subword model 중에 convolution을 character에 시키는 모델들도 있고, character based LSTM을 적용한 모델, HighWay Net 등등이 있다.

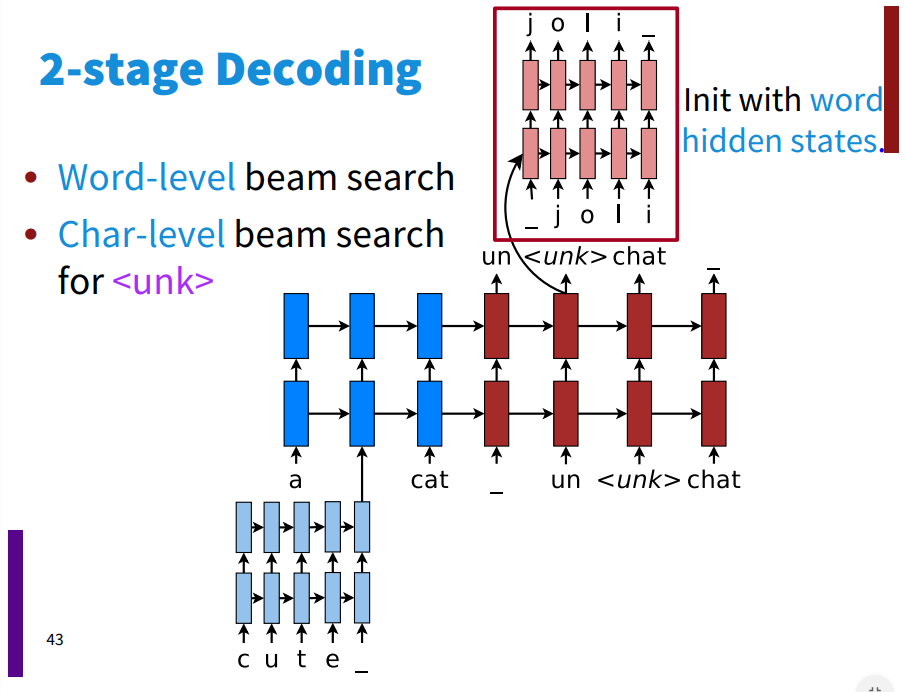
그 중 Character-Aware Neural Language Models가 있는데



Word embedding의 필요에 의문을 가졌고, CNN + Highway Net은 character 단위로 semantic and structural info를 가져왔고, building block을 사용했다.

Hybrid NMT

Word level로 번역 하지만 필요할 때 character level로



와 같이 cute로 모르는 단어가 있을 때는 char-level 인코딩으로! 입력이 들어가고, 디코딩 시 <unk>가 나온다면 <unk>를 subnet에 넣어 char-level 디코딩을 한다.

nmt에서는 beam search를 디코딩할 때 사용했는데 이때 word, character 두 단위에서 모두 사용하도록 한다.

(Cao 2016)에서도 word2vec + morphology의 임베딩을 하려 했다.

<https://ratsgo.github.io/from%20frequency%20to%20semantics/2017/07/06/fasttext/>

FastText Embeddings(Bojanowski,2016): word2vec을 기본으로 하되 역시 subword 임베딩 기법을 포함한다.