Assignment 7: fastText

[Objective]

Your model should generate word vectors.

[Requirements]

1. Implement fastText model with Gensim.
2. You should experiment with settings stated in the evaluation report, and report the result of each settings.
4. You should report the experimental results.

[Code structure]

- fasttext_train.py

Pre-trained fastText

- https://fasttext.cc/docs/en/english-vectors.html
- https://dl.fbaipublicfiles.com/fasttext/vectors-english/wiki-news-300d-1M-subword.bin.zip

Download pre-trained word vectors

Pre-trained word vectors learned on different sources can be downloaded below:

1. wiki-news-300d-1M.vec.zip: 1 million word vectors trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset (16B tokens).
2. wiki-news-300d-1M-subword.vec.zip: 1 million word vectors trained with subword infomation on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset (16B tokens).
3. crawl-300d-2M.vec.zip: 2 million word vectors trained on Common Crawl (600B tokens).
4. crawl-300d-2M-subword.zip: 2 million word vectors trained with subword information on Common Crawl (600B tokens).

fastText training

```
from gensim.models import FastText
from gensim.models.word2vec import PathLineSentences

sentences = PathLineSentences("./data/1billion/")
model = FastText(sentences=sentences, size=100, window=5, min_count=5,
workers=4, sg=0, hs=0, negative=5, ns_exponent=0.75, alpha=0.01,
min_alpha=0.0001, iter=1, word_ngrams=1, min_n=3, max_n=6)
model.save("fastText.model")
print(len(model.wv.vocab))
score, predictions = model.wv.evaluate_word_analogies('./data/questions-words.txt')
print(score)
```

fastText training

- Hyperparameters

  - size: 단어 벡터의 차원
  - window size: context 단어 수 / 2
  - min_count: 최소 빈도수 기준, 단어사전에 포함 여부 결정
  - wokers: 스레드 수
  - sg: 1이면 skip-gram 사용
  - hs: 1이면 hierarchical soft, 0이면 negative sampling 사용
  - negative: negative sample의 개수
  - ns_exponent: unigram distribution에 적용될 지수 값
  - cbow_mean: 1이면 context 단어의 평균을 사용, 0이면 합을 사용
  - alpha: learning rate
  - min_alpha: learning rate decay 시에 최소 learning rate
  - max_vocab_size: 단어 사전의 최대 크기
  - iter: epoch 수
  - sorted_vocab: 1이면 사전의 단어들을 빈도수 기준 내림차순 정렬
  - batch_words: batch size
  - **word_ngrams**: subword 정보 사용여부, 1이면 사용, 0이면 그냥 word2vec
  - **min_n**: character n-gram의 최소 길이
  - **max_n**: character n-gram의 최대 길이

Experiments: Effect of the size of n-grams

- Taking a large range such as 3 − 6 provides a reasonable amount of subword information

	2	3	4	5	6
2	57	64	67	69	69
3		65	68	70	70
4			70	70	**71**
5				69	**71**
6					70

(a) DE-GUR350

	2	3	4	5	6
2	59	55	56	59	60
3		60	58	60	62
4			62	62	63
5				64	64
6					**65**

(b) DE Semantic

	2	3	4	5	6
2	45	50	53	54	55
3		51	55	55	**56**
4			54	**56**	**56**
5				**56**	**56**
6					54

(c) DE Syntactic

	2	3	4	5	6
2	41	42	46	47	**48**
3		44	46	**48**	**48**
4			47	**48**	**48**
5				**48**	**48**
6					**48**

(d) EN-RW

	2	3	4	5	6
2	78	76	75	76	76
3		78	77	78	77
4			79	79	79
5				**80**	79
6					**80**

(e) EN Semantic

	2	3	4	5	6
2	70	71	73	74	73
3		72	74	**75**	74
4			74	**75**	**75**
5				74	74
6					72

(f) EN Syntactic

Assignment 7: fastText

[Evaluation report]

fastText Evaluation Report

	Model	n-gram	Negative Sampling	# of negative samples	Learning rate	Learning rate decay(O/X)	dimension	iteration	training time	Accuracy	OOV word	most_similar
setting #1	SG	2,3	O	15	0.01	O	100	5				
setting #2	SG	3,4,5,6	O	15	0.01	O	100	5				
setting #3	SG	2,3	O	15	0.01	O	300	5				
setting #4	SG	3,4,5,6	O	15	0.01	O	300	5				

[결과 정리]

```
model = FastText.load("fastText.model")
score, predictions = model.wv.evaluate_word_analogies('./data/questions-words.txt')
print(score)
print(model.wv.most_similar("thank____you", topn=20))
```

Assignment 7: fastText

- **Evaluation Criteria**

Simplicity	How concisely did you write the code? - 배점 2점
Performance	How well did the results of the code perform? - 배점 4점 - acc 55%이상 달성: 3점 - OOV word 생성 및 유사단어 확인: 1점
Brevity and Clarity	How concisely and clearly did you explain the results? - 배점 4점

Assignment 7: fastText

- Due to : **~ 11.1 (Sun)**

- Submission : Online submission on blackboard

- Your submission should contain
  1) The whole code of your implementation
  2) The evaluation report

- You must implement the components yourself!
- File name : StudentID_Name.zip