pa - build word2vec on splits

June 2, 2019

In [1]:
```
# Start logging process at root level
import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.
logging.root.setLevel(level=logging.INFO)
```

In [2]:
```
import multiprocessing
import os, os.path

chunk_size = 99999
chunk_start_at = 5
chunks_basepath = "datasets/chunks/enwiki-chunks-"+str(chunk_size)
model_path = "models/enwiki-full-dict-"+str(chunk_size)+".model"
dictionary_path = "dictionaries/enwiki-english-lemmatized.txt.bz2"
chunks_count = len([name for name in os.listdir(chunks_basepath) if os.path.isfile(os.p
#chunks_count = 4

lemmatization = True
w2v_size = 300
w2v_window = 5
#w2v_cores = multiprocessing.cpu_count()-10
w2v_cores = 20
w2v_min_count = 1
w2v_epochs = 5
```

In [3]:
```
# Load dictionary from file
from gensim.corpora import Dictionary
dictionary = Dictionary.load_from_text(dictionary_path)
```

```
2019-04-03 00:21:45,633 : INFO : 'pattern' package found; tag filters are available for Englis
```

In [4]:
```
# Initialize simple sentence iterator required for the Word2Vec model / memory savior
class SentencesIterator:
    def __init__(self, wiki):
        self.wiki = wiki

    def __iter__(self):
        for sentence in self.wiki.get_texts():
            yield list(map(lambda x: x.decode('utf-8'), sentence)) #set encode('utf-8'
```

```
In [ ]: import os
        from gensim.models import Word2Vec
        from gensim.corpora import WikiCorpus

        for e in range(chunk_start_at,chunks_count+1):
            chunk_name = chunks_basepath+"/"+"enwiki-chunk-"+str(chunk_size)+"-"+str(e)+".xml.
            model_backup_basepath = "models/enwiki-full-dict-parts-"+str(chunk_size)
            model_backup_path = model_backup_basepath+"/enwiki-full-dict-"+str(chunk_size)+"-pa

            if not os.path.exists(model_backup_basepath):
                print("Mkdir the model base path")
                os.mkdir(model_backup_basepath)

            # Build WikiCorpus based on the dictionary
            wiki = WikiCorpus(chunk_name, dictionary=dictionary,lemmatize=lemmatization)

            # Set generator
            sentences = SentencesIterator(wiki)

            print("Running with: ",str(w2v_cores)," cores")
            print("Selected model:",model_path)
            if not os.path.exists(model_path):
                print("Building model on:",chunk_name)
                # Build model on first part
                model = Word2Vec(sentences=sentences,
                                 size=w2v_size,
                                 min_count=w2v_min_count,
                                 window=w2v_window,
                                 workers=w2v_cores,
                                 iter=w2v_epochs
                                )
                model.save(model_path)
                model.save(model_backup_path)
                print("Model saved")
            else:
                # Train model on anthor part
                print("Training model on:",chunk_name)
                model = Word2Vec.load(model_path)
                model.train(sentences=sentences, total_examples=model.corpus_count, epochs=mode
                model.save(model_path)
                model.save(model_backup_path)
                print("Model updated")

            print("Backup model saved:",model_backup_path)

            del wiki
            del model
            del sentences
```

```
        del dictionary
        print("Done.")
```

In [ ]: