Use Latent Dirichlet Allocation for topic learning with gensim

Latent Dirichlet Allocation is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions.

Gensim is a popular NLP library with python support. It specializes in unsupervised topic modeling and is designed to handle large text collections using data streaming and incremental online algorithms.

In this technology review, I will use a dataset that contains thousands of news articles and try to extract the main topics from these articles.

First thing we need to do is preprocess the raw text. This includes

- Tokenization: Split chunks of texts into sentences and the sentences into words. Lowercase the texts and remove punctuation.
- Remove stopwords
- Stemming and Lemmatization to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

```
def lemmatize_stemming(text):
    return stemmer.stem(WordNetLemmatizer().lemmatize(text, pos='v'))

# Tokenize and lemmatize
def preprocess(text):
    result=[]
    for token in gensim.utils.simple_preprocess(text) :
        if token not in gensim.parsing.preprocessing.STOPWORDS and
len(token) > 3:
            result.append(lemmatize_stemming(token))
    return result
```

Next thing we need to do prior to topic modelling is converting the text into bags of words. More specifically we want to get a dictionary where the key is the word and value is the number of times that word occurs in the entire corpus. This can be done by gensim:

```
dictionary = gensim.corpora.Dictionary(processed_docs)
```

Then, we want to use the dictionary we built for the entire corpus and get a dictionary on document level and convert individual documents into bags of words.

```
bow_corpus = [dictionary.doc2bow(doc) for doc in processed_docs]
```

After the preparation from all steps above, we’re ready to apply LDA and kick off the modelling process. This can be done easily with a gensim LDA model with one function call. Below we’re specifying we want to get eight unique topics with 10 training passes of the documents.

```
lda_model =  gensim.models.LdaMulticore(bow_corpus,
                                   num_topics = 8,
                                   id2word = dictionary,
                                   passes = 10,
                                   workers = 2)
```

The output from the model is 8 topics each categorized by a series of words. The LDA model doesn’t give a topic name to those words and it is for us humans to interpret them.

```
Topic 1: Possibly Graphics Cards
Words: "drive" , "sale" , "driver" , *"wire" , "card" , "graphic" ,
"price" , "appl" ,"softwar", "monitor"

Topic 2: Possibly Space
Words: "space","nasa" , "drive" , "scsi" , "orbit" , "launch" ,"data"
,"control" , "earth" ,"moon"

Topic 3: Possibly Sports
Words: "game" , "team" , "play" , "player" , "hockey" , season" ,
"pitt" , "score" , "leagu" , "pittsburgh"

Topic 4: Possibly Politics
Words: "armenian" , "public" , "govern" , "turkish", "columbia" ,
"nation", "presid" , "turk" , "american", "group"

Topic 5: Possibly Gun Violence
Words: "kill" , "bike", "live" , "leav" , "weapon" , "happen" ,
*"gun", "crime" , "car" , "hand"
```

Conclusion:
The LDA model performs very well extracting topics out of the corpus we provide. The entire program runs less than 10 minutes with thousands of pieces of texts. The model is suitable for datasets that have a certain set of topics, it will be less efficient given random texts that has different types of content.