A brief report in this document for assignment-2 contains problem approach for both the tasks, solutions & results and observations.

Elements of Assignment:

Data: Two sets of data are given based on the posts in stackexchange website. First dataset is about AI/machine learning which contains 3126 posts and second dataset is about Android which has size of 49784. Each post contains two fields title and body, title is describes short text about post and body has full description of problem user facing in html format. As these are taken from stackexchange website, we can see many code blocks in title body.

Tasks:

1) In first task, we're asked to do data analysis for both the datasets about posts and words in corpus.
2) In second task, problem is on finding similar posts for the input query post and for any post from corpus itself.

Problem approach, Results & Observations:

Task1:

In Task1, I've done data analysis for both datasets for frequency of tokens in entire corpus and posts lengths.

Token Analysis;

In the first part, I've made frequency distribution for the entire corpus for both datasets after removing special characters. However, it's observed in top of frequency table all are stop words, so I've have removed stop words and draw distribution again. To analyze, the less frequent tokens instead of going to the bottom of frequency distribution table, I wanted to see from where tokens became unique so that we can remove those for the next task (document similarity). For this, I've taken a head size which is length of frequency table to cut down from top and analyze most frequent and less frequent words in it.

Below are the frequency distribution tables for each dataset:

Dataset1				Dataset2			
with stop words (head size= 20000)		without stop words (head size= 15000)		with stop words (head size= 70000)		without stop words (head size= 60000)	
Total number of tokens: **472014** Number of unique tokens in data: **36838**		Total number of tokens: **277844** Number of unique tokens in data: **36700**		Total number of tokens: **5552409** Number of unique tokens in data: **179877**		Total number of tokens: 3241068 Number of unique tokens in data: **179731**	
Top 10 frequent tokens	10 last unique tokens	Top 10 frequent tokens	10 last unique tokens	Top 10 frequent tokens	10 last unique tokens	Top 10 frequent tokens	10 last unique tokens
'the', 22381	'poc', 1	'p', 2116	'alternativelybiased', 1	'the', 221112	'gids3002', 1	'phone', 45625	'pocketbag', 1
'to', 13581	'costfunction', 1	'pi', 1921	'confidentialityp', 1	'to', 185618	'xx3115953', 1	'android', 41256	'joggingwalking', 1
'a', 12981	'overexposure', 1	'would', 1837	'nonmathcs', 1	'i', 173353	'upgraderreceiver', 1	'pi', 40232	'bagp', 1
'of', 10976	'ebook', 1	'ai', 1827	'supplements', 1	'a', 114690	'cacheb\ lurversion\ 45608a956\ verizonenuszip', 1	'app', 25174	'skyp', 1
'and', 8751	'hrefhttpsgist githubcomhartgerv50ccdb999393eb1d2b50725e14cd0fa', 1	'learning', 1681	'metapost', 1	'and', 101861	'bootcommand', 1	'p', 23686	'wifigprsetc', 1
'is', 8487	'noreferrermost', 1	'neural', 1629	'satiate', 1	'it', 86040	'23103', 1	'device', 19952	'permsisions', 1
'in', 7008	'commmon', 1	'network', 1590	'curiositiesp', 1	'is', 82292	'disalbedm', 1	'google', 18950	'15gs', 1
'i', 6755	'tokensa', 1	'relnofollow', 1429	'nonmath', 1	'my', 81074	'1340', 1	'apps', 16791	'hrefhttpwwwsdcardorgconsumersformatter3', 1
'that', 5348	'professorsresearchers', 1	'like', 1275	'hrefhttpwwwalicebotorgarticleswallaceelizahtml', 1	'on', 64451	'shutdownthread', 1	'using', 16197	'relnofollowsdformattera', 1
'for', 4967	'codefcncode', 1	'one', 1247	'relnofollowfrom', 1	'in', 57777	'shutdownm', 1	'get', 14585	'pemnbem', 1

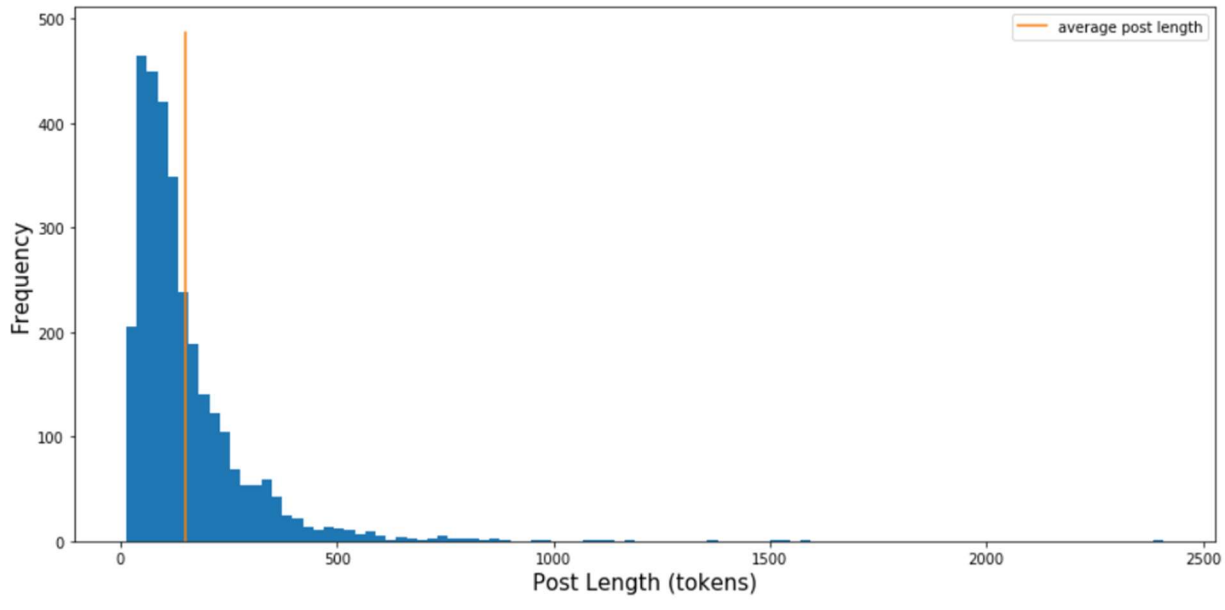
From the above table, we can observe following:

1) Number of tokens are reduced nearly to 50% after removing stop words in both datasets.
2) Number of unique tokens aren't changed much after removing stop words in both datasets.
3) Nearly half of the tokens in both datasets are appeared **only once** in entire corpus. For example, for dataset 1, for the head size = 20000, in the end we got unique tokens but the total frequency distribution size is 36838.
4) When we consider stop words, most frequent in both datasets are meaning less propositions and conjunctions etc. but when we remove we can see some meaning full words in top like 'ai', 'learning' , 'neural' etc. for dataset1 and 'android', 'phone' etc. in dataset2.

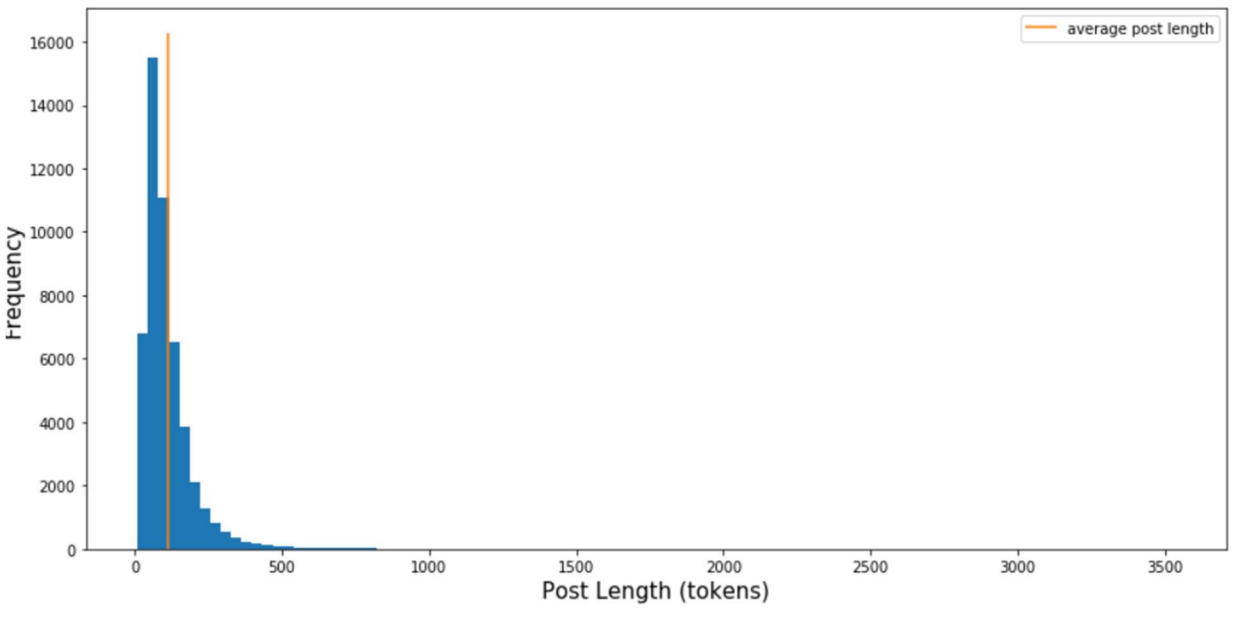
Post length analysis:

Along with frequency analysis for the tokens, I've analyzed posts lengths in corpus. For this, I've plotted histogram with post length (in tokens) in x-axis and frequency of the posts in a bin on y-axis with total bins=100 for both datasets.

Dataset1:
length of corpus: 3126
average post length 150.99616122840692
minimum post length 14
maximum post length 2409



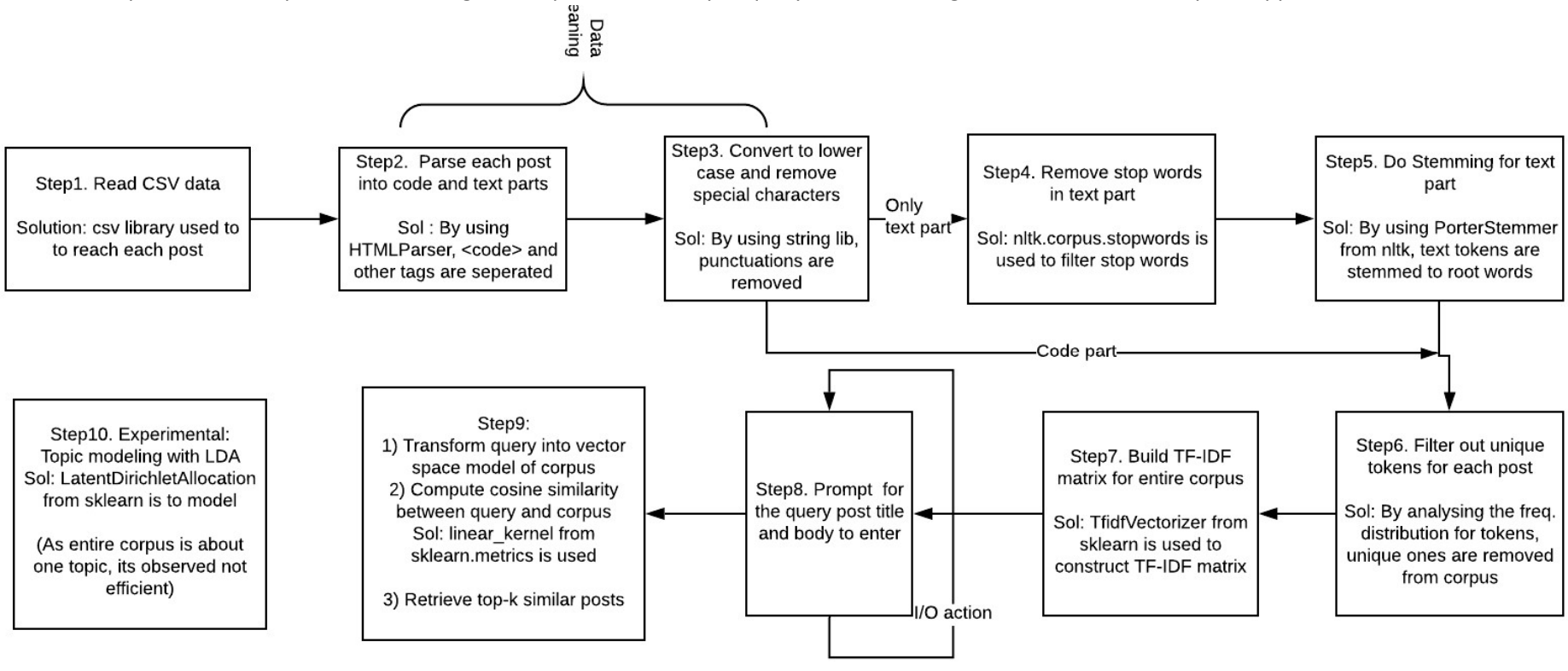
Dataset2:
length of corpus: 49784
average post length 111.52998955487706
minimum post length 10
maximum post length 3533



Observations:
1) In both datasets, very few posts have length more than 500.
2) Based on histogram bins, we can observe in which length range most number of posts are in:
   For dataset1, in second bin (14-37.95) we have the highest frequency of posts as 464.
   For dataset2 also in second bin, (10-45.23) we have highest number of posts as 15479

Task2:

In Task2, I've implemented the problem of finding similar posts for the input query based on the given datasets. The complete approach is delineated in block diagram:



Brief outline of approach:

Step1: Read the csv data to prepare corpus for both datasets

Step2: Instead of tokenizing the posts naively, we can utilize the different tags present in post body. As posts are taken from CQA platform users can post <code> blocks too to explain their issues. So, we can segregate code and simple text into two parts and preprocess them. For this, I've used HTMLParser to parse the post body.

Step3: Before tokenizing, I've done some data cleaning to remove special characters and to convert to lower case.

Step4: Based on the analysis in Task1, we can see that stop words will increase the corpus by almost double without adding any semantic meaning to the post. For this reason, I've removed stop words from only text part as tokens in code are already syntactical and provides very good semantic information. For example, "do" can be stop words in general English text but it's a keyword in code and has very syntactic meaning to it.

Step5: Like above, we do stemming only to text part but not code as for the same reasons like above.

Step6: After pre-processing is completed, I've merged both and analyzed the frequency distribution for whole corpus. Based on this analysis, we can remove tokens that are very unique to each post because algorithms that predict similarity based on co-occurrence of the tokens (like cosine) won't make any use of unique tokens.

Step7: For the filtered datasets, I've constructed TF-IDF matrix using sklean library with ngram_range=(1,3) so that it captures phase information of at most 3 words.

Step8: By now, everything is ready in the background to find similarity between query and existing posts. So, in the next step user is prompted to take query post details as title and body (in same form as corpus) and also other options such as dataset, k in top-k etc..

Step9: Once we have query, we can transform this into same vector space model as corpus, and then calculate cosine similarity between query and each post in corpus. Sklearn transforms both query and corpus TF-IDF into normalized scores, so we can directly apply linear_kernel to take dot product. At the end, based on input k, show only top-k similar posts.

Step10: I've tried to get similarity of the query with posts in corpus by doing topic modeling with 'Latent Dirichelet Allocation' method, but as the entire corpus is taken from single topic, its observed query and all the posts in corpus are mapped to same topic in output.

Results:

To test the problem I've taken a post from the input dataset1 and given it as query. As expected, in the top we have got the same post and remaining posts that are similar to it.

This is the prompt taken from Task2.ipynb notebook:

```
To match with existing questions already in corpus, enter dataset no.,  title and body .


Dataset 1 (or) 2 :
1

Enter the title to match :
How to program AI in Mindstorms

Enter the body to match :
<p>I have a LEGO Mindstorms EV3 and I'm wondering if there's any way I could start coding the bot in Python rather than the
default drag-and-drop system. Is a Mindstorm considered AI?</p>  <p>Is this possible?</p>  <hr>  <p>My goal is to write a basic
walking program in Python. The bot is the EV3RSTORM. I searched and found <a href=""http://bitsandbricks.no/2014/01/19/getting-
started-with-python-on-ev3/"" rel=""nofollow"">this</a>, but don't understand it. </p>

To show only matched posts titles, press 1
 or to see both title & body press 2:
1
Enter the number of top posts that matched to show :
5
Similar posts matching the query (best match at top) :


Title : How to program AI in Mindstorms
---------------------------
Title : Identify unnecessary inputs of NN
---------------------------
Title : Neural Network on EV3 Mindstorm without 3rd Party Software
---------------------------
Title : Are search engines considered AI?
---------------------------
Title : Implementing AI/ML for the card game "Cheat"
---------------------------
```

I've done the same for second dataset too and we can see posts similar to the query.

```
To match with existing questions already in corpus, enter dataset no.,  title and body .


Dataset 1 (or) 2 :
2

Enter the title to match :
Screen timeout versus secured lock time

Enter the body to match :
"<p>I just updated to Jelly Bean on my GS3 and am mostly impressed. </p>  <p>I am having trouble with the screen timeout
setting, however.  When I go into Display>Screen Timeout and select a time, say 30 seconds, I get a message that says ""Screen
will turned off when screen is locked.  Secured lock time is currently set to 5 seconds."" My question is where? I cannot find
this Secured Lock time setting in Display settings, Lock Screen settings or Security settings, which would seem to be the
intuitive places to look for it. Further, even though I get this message, my screen now does not turn off at after 5 seconds, or
ever as far as I can tell.  After walking away from the phone for 20 minutes, I returned to the screen  blazing in all its
AMOLED glory.</p> "

To show only matched posts titles, press 1
 or to see both title & body press 2:
1
Enter the number of top posts that matched to show :
```

```
5
Similar posts matching the query (best match at top) :


Title : Screen timeout versus secured lock time
---------------------------
Title : Change Screen Timeout to Never
---------------------------
Title : Setting the lock screen timeout
---------------------------
Title : Samsung S8 always on display timeout?
---------------------------
Title : Looking for app that provides different screen timeout and lock timeout
---------------------------
```

In next evaluation experiment I've taken a query from stackexchange CQA (https://ai.stackexchange.com/questions/11768/confused-about-group-convolution?rq=1) which is not in the given corpus but related to same topic, then I've observed that though the related posts shown in stackexchange are present in our corpus, I couldn't see them in my results. Below is the console output from the program:

```
To match with existing questions already in corpus, enter dataset no.,  title and body .


Dataset 1 (or) 2 :
1

Enter the title to match :
Confused about group convolution

Enter the body to match :
I think I don't understand group convolutions well. Say you have 2 groups. This means that the number of parameters would be
reduced in half. So assuming u have an image and 100 channels, with a filter size of 3x3, you would have 900 parameters (ignore
the bias for this example). If you separate this in 2 groups, if I understand it well, you would have 2 groups of 50 channels.
This can be made faster, by running the 2 groups in parallel, but how does the number of parameters gets halved? Isn't each
group having 50*9=450 parameters, so in total you still have 900 parameters. Do they mean that the number of parameters that the
backpropagation goes over (in each branch) gets halved? Because overall, I don't see how it can get reduced. Also, is there a
downside in using more groups (even going to 100 groups of 1 channel each)? Thank you!

To show only matched posts titles, press 1
 or to see both title & body press 2:
1
Enter the number of top posts that matched to show :
5
Similar posts matching the query (best match at top) :


Title : Regression with more than one output, neural network
---------------------------
Title : Reinforcement Learning to Grouped Scheduling Optimisation Problem
---------------------------
Title : Using unsupervised learning for classification problems
---------------------------
Title : Ensemble models - XGboost
---------------------------
Title : How to build my own dataset and model for an LSTM neural network
---------------------------
```

In second dataset too, I've observed mismatch between the stackexchange results and my implementation results. I've taken a query from https://android.stackexchange.com/questions/6422/what-is-hard-reset-and-what-is-factory-reset?rq=1 and used same to test my application, the related posts in stackexchange are different than what I've got here:

```
Dataset 1 (or) 2 :
2

Enter the title to match :
What can be saved before factory reset?

Enter the body to match :
"<p>My phone has a GPS problem and service says the need to wipe it completely.<br> They are probably just lazy and would rather
do it the easy way.<br> C'est la vie.</p>  <p>The phone is SGS unrooted 2.3.4 - Touch Wiz. I don't want to root it.</p>  <p>Now,
what I am going to do is this:<br> - Save contacts with Kies<br> - Save files</p>  <p>Is there anything else I can do or that I
should know?</p>  <p>For example, I believe that apps associated with the Google Account will be reinstalled after I re-enter my
account into the newly formatted phone, correct? However, app data like savegames won't be ported, I have to search for them in
the phone memory?</p>  <p>Can I backup SMS?</p>  <p>Can I back-up settings?</p>  <p>Will imported contacts keep all fields like
I have them now, i.e. work phone, home phone, work email, home email etc?</p>  <p>I know it's a ton of questions, sorry about
that. Thank you.</p> "

To show only matched posts titles, press 1
 or to see both title & body press 2:
1
Enter the number of top posts that matched to show :
5

Similar posts matching the query (best match at top) :


Title : What can be saved before factory reset?
---------------------------
Title : How to add new contacts to an outlook.com account in an Android device?
---------------------------
Title : Saved my contacts in google account but they don't appear on my new phone
---------------------------
Title : Android 7 - How to save new contacts to the phone
---------------------------
Title : Is it possible to backup settings and apps for SGS with Samsung Kies?
```

Observations:

1) Modeling in vector space has advantages to match with more relevant documents and sort as we have scoring mechanism with cosine similarity metric.
2) The naïve vector space model fails to take phrase information into the account, for example, for two posts ‘A is better than B’ and ‘B is better than A’ it can give high similarity score but I’ve taken phrase information of at most 3 tokens to mitigate this problem to lesser extent.
3) Although this model captured similarity using bag of words representation, but it has failed to capture semantic information like synonymy, polysemy and topic/context in the post. To fix that, we can prepare a synonym dictionary to replace them with its root word or language model the corpus to capture topic.
4) Finally, I’ve tried to do topic modeling with LDA to retrieve posts that are more similar based on over all topic rather just textual information, but it has failed to perform as the entire corpus is on single topic, all the posts are mapped to same topic in the results.
5) To improve the score, we can do language modelling to use semantic information for query-document similarity.