Winter 2019

NASA Discovery 2:

An NLP Driven Mission for Topic Generation from NASA Datasets

Wells Fargo Campus Analytics Challenge: Natural Language Processing

Submitted: Dec 02, 2019

Contents

I) Introduction

Topic models, methods that extract themes from unstructured text data, often provide a first layer of insights. Two main requirements regarding model output present a challenge to traditional topic models:

1. Correlations between topics present in many of the corpora violate the assumption of topic independence made by many topic models, including the popular latent Dirichlet allocation ("LDA").
2. Hierarchical models, which provide insights on multiple levels along the spectrum of broad to detailed, are more useful than a single, coarse, high-level segmentation of documents into large thematic bins.

The Campus Analytics Challenge 2019 puts students in the role of a natural language processing (NLP) data scientist and calls them to develop a topic model beyond a common approach (latent Dirichlet allocation, LDA). The consists of descriptions of open datasets published by NASA. These data also exhibit a high degree of overlap between high-level topics.

II) Goals

The primary goal of this project was to categorize each "item" into a topic then a sub-topic, while preserving the ID number. Ideally there should be a confidence estimate for the assignment.

There were also several secondary goals such as including a sentiment model and creating a *unique* model. While it is technically a secondary goal, it is also crucial to provide insights about the corpus based on both the initial LDA analysis and the novel method.

These were stated in the challenge statement as follows:

> Therefore, the two main objectives in this Challenge are for your Solution to appropriately handle correlated topics and to generate subtopics (in addition to topics).

III) Data Processing

Raw data formatting

Every data project starts with a quick look at the data and it's formatting. The file in this case ("a391d853147b-NASA_DataSets_Scrub.tsv") is a tab separated file with the fields of the following:

- title (text): the title of the dataset
- issued (date): the date when the dataset was first made public
- modified (date): the latest date when the dataset was modified
- description (text): the description of the dataset
- id (string): an identifier for the dataset

Therefore, the file can line by line, dumping each of the corresponding fields into an array for processing. Since the goal of this project focuses primarily on linguistic analysis, the date fields can be disregarded for now.

Cleaning the data

From the initial look at the data, the corpus contains several aspects we want to remove.

Table 1: Removal of certain data components

	Aspect removed	Rationale
Title field	"Phase n"	NASA uses the "Phase n" where n is a number (I, II, etc.) to describe different parts in the mission timeline.
	"V n.n"	NASA distinguishes version numbers of data sets, but this is not part of the analysis
Description field	"N/A" or "not applicable"	This description is not relevant to analysis
	"xxxx"	Several parts of the description have been edited with numerous x's. Any word with 2+ consecutive x's has been removed since it does not contribute to the analysis
	"[" or "]"	This are often used to indicate a reference
Both fields	Misc. punctuation	Punctation is not relevant to topic determination/ generation
	"n.n"	This is another way of indicating a version number

This process was accomplished through the usage of regular expressions. Each aspect was removed separately, allowing for further modification and tuning in the future.

Furthermore, the corpus is encoded as UTF-8: this must be specified in `csv.open` to avoid an error.

Initial insights

Based on the parsing of the raw data, I ran a frequency analysis of the words in the titles and the descriptions. This produced the tables below.

Table 2: Most common 10-word stems in data set titles

Word Stem	Occurrences
Data	2920
System	1969
Ge	1221
Disc	1220
High	1178
Global	1135
Model	1004
A	813
Degre	810
Space	785

Table 3: Most common 10-word stems in data set descriptions

Word Stem	Occurrences
The	39175
Data	30914
Use	24479
Developt	16380
Thi	13817
Propos	11772
Provid	11615
Product	10987
Design	10968
Mission	10152

While these insights are explored further in “Insights from the topics & sub-topics” several aspects are already apparent. For example, the titles of the datasets seem to focus on using data at a systems level approach to generate models. This is actually in line with what was discussed in my aerospace class “Design and System Engineering Methods” where it was stressed that NASA uses a “systems of systems” approach where possible. Another insight is that there is a clear distinction between Earth and space, shown by “global” and “space” respectively.

Turning now to Table 3, the words “the” and “data” show up a great deal. It is unclear if these are actually words or the stemmed version- in either case, it would be beneficial to add them to the stop words list.

IV) Creating a model

Initial LDA

As one of the requirements stated that we should compare our model to an LDA model, the first step is to create that LDA model. The topics for this are generated by `LDA_initial.py` and the sub-topics are generated by `LDA_initial2.py` files. The output of this initial LDA model is called initial_LDA_out.csv with the topics and subtopics labeled in initial_LDA_out_topicLabels.csv and initial_LDA_out_subtopicLabels.csv respectively.

The number of topics was selected as eight. I choose these numbers based off of how NASA categorizes their data on their Open Data Portal [4].

Novel model

The novel model I have created uses an LDA model to generate the topic of a data set, but pivots to creating sub-topics based on NASA’s data portal tagging system [1]. This system is in turn a derivative of my intended approach using the NASA Glenn Research Center archive of public information.

Rationale

NASA, as an organization, follows very strict documentation procedures. I am currently studying Aerospace Engineering, so I have had the advantage (though possible displeasure depending on your point of view) of having to implement many of their standardizations.

Knowing this, I intended to use NASA Glenn Research Center's documentation on their work to classify the data sets in the corpus. Essentially, NASA GRC provides information about most of NASA's public facing work in an easy to read web format. It is structured in such a way that topic level fields could be pulled as topics, while their children could be considered sub-topics (see below).

- Propulsion (*topic*)
    - Air Breathing (*sub-topic*)
        - Article 1 (*information*)
        - Article 2
        - …
    - Rockets (*sub-topic*)
        - …
- Atmospheric observation (*topic*)
    - …

By querying NASA GRC's texts on each topic, one could match the contents in the titles & descriptions of the corpus. If a word commonly appears in the NASA GRC article "Performance of Jet engines", it most likely is [topic, sub-topic] = [propulsion, air-breathing]. Therefore, if the word in question was "turbine", text from the corpus containing "turbine" would most likely fall under the same topic/sub-topic classification.

Even though this was my preferred method, I took a shortcut due to time constraints. I stripped the titles and formatted them according to NASA's Open Data API specifications, then passed them as a search query. From the returned JSON, I could then directly pull the keywords. These keywords are equivalent to sub-topics.

Using the aforementioned approach in combination with an initial LDA analysis allowed me to then create both topics *and* sub-topics for each data set in the corpus.

Assumptions & justifications

The initial corpus was scrubbed to prevent LDA on the descriptions of the data set, *not* the titles. Therefore, it is possible to create an LDA model from the titles, though any subsequent sub-topic generation is of questionable accuracy.

Note that the results file was generated manually due to the time constraints imposed by the web queries. This was done by first outputting "initial_LDA_out.csv" to retrieve the topics and confidence. Then the main program was run on subsets of the data, producing sub-topics. These subtopics were then combined with the initial file to produce the results file.

V) Novel Model explored

Strengths & limitations

The strength of this model is that it directly pulls from how NASA classifies their datasets into topics. In other words, it is directly generated from the objective classification. The benefit of this is that if one were to search the topic/sub-topics for a data set, they could easily find related data sets.

That is not to say that this novel model is without limitations. Since the model is pulling from the NASA data portal for each query, it is a bit slow- clocking in around 3 hours on my home WiFi to complete the searches. Furthermore, the queries are sensitive to punctuation. I had quite a few errors when parsing the data and was unable to remove instances with 3 or more "-" in sequence. This could be easily remedied with better RegEx's but unfortunately, I do not have enough practice with them.

Insights from the topics & sub-topics

Based on the topic "equations" in Table 5 it is clear that NASA focuses on a variety of topics.

Table 4: Generalized topic interpretation

Topic	Interpreted contents
0	Rossetta program (orbital)
1	Cassini program (extraterrestrial)
2	Spacecraft control
3	Asteroids
4	Orbital data (EDR, RDR)
5	Thermal analysis
6	Code implementations (extraterrestrial)
7	Code implementations (orbital)

However, I will say that these descriptions are interpreted. They make sense to a human (assuming they have some knowledge of NASA programs). The actual breakdown in Table 5 tells a more complete story.

With regards to the sub-topics, there were too many to produce in a single table, so I refer the reader to **Deliverable 1: Results**. In any case, the two most common sub-topics appeared to be "active" and "completed", indicating that NASA not only tags datasets based on their contents, but also their status. Generally though, the sub-topics usually consisted of the follow: mission name, mission purpose, location, process related words, and who carried out the data work.

Of note, I actually tested most of my program on a smaller subset of the data, labeled "a391d853147b-NASA_DataSets_Scrub2.tsv" or some variation since my laptop is rather slow. The final run was done on all the data (i.e. **utilizing a holdout set**) and the topics were essentially the same.

Sentiment

A sentiment analysis is not currently integrated into the novel model. This was due to time constraints on the project. Even so, I have designed my approach in a modular format, allowing for future modifications. There are several Python libraries that support sentiment analysis- the text of `raw_titles` and `raw_descriptions` is readily accessible within the program to allow for the deployment of said libraries.

Comparison to baseline LDA

The novel method can be compared against the baseline LDA model by checking if the LDA topic constituents are contained in the novel method topics.

As an example, let us say the following:

- LDA model produces Topic 0: data*0.1 + text*0.05 + char*0.01
- Novel method produces Topic 0: [data, mars, text]

We can see that the LDA model consists of [data, text, char] while the novel method consists of [data, mars, text]. Two of the LDA terms are found in the 3 novel terms, so the “accuracy” can be calculated as 0.6667 approximately. Therefore, the comparison between the models can be represented by the following equation:

$$accuracy = \frac{\sum(LDA\ term)\ in\ (novel\ topic)}{\#\ of\ novel\ topic\ terms}$$

VI) Deliverables

Deliverable 1: Results

The results are located in the folder “results”.

The first results file tilted “final_1_out” is formatted as follows:

topic	sub_topic	dataset_id	topic_confidence
	['completed', 'jet-propulsion-labo...']	12289	
	['mars', 'phoenix']	12290	
…	…	…	…

Descriptions of the topics can be found in “results_topicLabels.csv”. They are also listed below. Sub-topics vary much more so they can be found in the original results file discussed above.

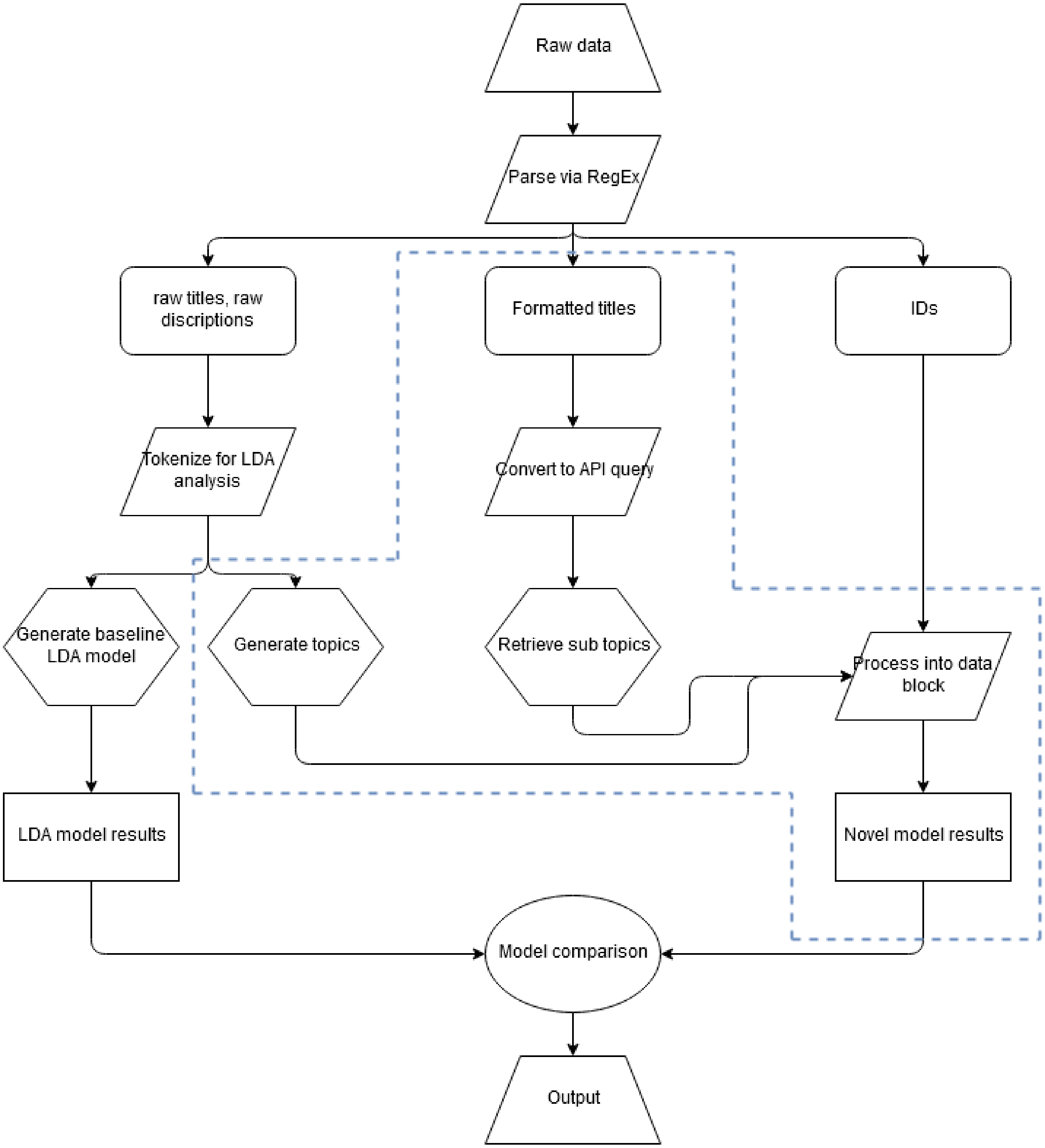
Table 5: Topic results

Topic	Contents
0	0.014*"data" + 0.012*"L2" + 0.010*"scienc" + 0.010*"new" + 0.010*"lidar" + 0.009*"horizon" + 0.009*"rosetta-orbit" + 0.009*"cloud" + 0.008*"calipso" + 0.008*"cruis"
1	0.017*"cassini" + 0.014*"-" + 0.013*"raw" + 0.011*"set" + 0.010*"rss" + 0.009*"data" + 0.008*"nasa" + 0.008*"grid" + 0.008*"budget" + 0.007*"orbit"
2	0.015*"system" + 0.010*"control" + 0.008*"space" + 0.008*"high" + 0.008*"spacecraft" + 0.007*"design" + 0.007*"manag" + 0.007*"A" + 0.007*"model" + 0.006*"solar"
3	0.025*"asteroid" + 0.016*"nomenclatur" + 0.016*"gazett" + 0.009*"radar" + 0.007*"deriv" + 0.007*"data" + 0.007*"launch" + 0.007*"halley" + 0.007*"resampl" + 0.007*"lightcurv"
4	0.016*"rdr" + 0.015*"mar" + 0.014*"comet" + 0.012*"rosetta-orbit" + 0.011*"op" + 0.011*"edr" + 0.010*"data" + 0.010*"multifunct" + 0.010*"record" + 0.010*"mer"
5	0.012*"high" + 0.010*"engin" + 0.009*"fiber" + 0.009*"applic" + 0.009*"space" + 0.009*"system" + 0.009*"laser" + 0.008*"academi" + 0.008*"thermal" + 0.007*"sensor"
6	0.010*"code" + 0.009*"near" + 0.008*"global" + 0.008*"data" + 0.007*"aerosol" + 0.007*"soil" + 0.006*"sin" + 0.006*"object" + 0.006*"track" + 0.006*"discoveri"
7	0.015*"degre" + 0.015*"ge" + 0.015*"disc" + 0.014*"ground" + 0.012*"x" + 0.011*"gpm" + 0.008*"monthli" + 0.008*"case" + 0.006*"observ" + 0.006*"multipl"

Deliverable 2: Method description

The method description is detailed in **Novel model**. Further descriptions and explanations can be found in **Novel Model explored**. (these are hyperlinks to those sections).

The path of data through my work pipeline is described below with the novel method boxed.



Deliverable 3: Code

The format of this report is not conducive to pasting code, nor is it good scientific practice to include code in a report. As such, I have included my code in the folder labeled “code”.

main.py

This file is where the entire process runs. The line commented “speed test” where used to run the data in chunks due to the performance limitations of my laptop.

config.py

Dependencies and libraries are specified in this file. Called in other files via `from config import *`

csvRead.py

The raw data is read using this file. It also tokenizes the data

LDA_initial.py

Carries out initial LDA analysis on the tokenized titles

LDA_initial2.py

Carries out initial LDA analysis on the tokenized descriptions

LDA_initial_out.py

Formats and writes output for initial LDA analysis.

LDA_novel_out.py

Formats and writes output for initial LDA analysis with the addition of the novel approach.

method_subTopicReq.py

Makes requests to the NASA data portal as part of the novel approach.

user_comm.py

This is where the end user communication tools are generated.

ex_aaa.py

These files are scratchpad notes where I tested small code snippets.

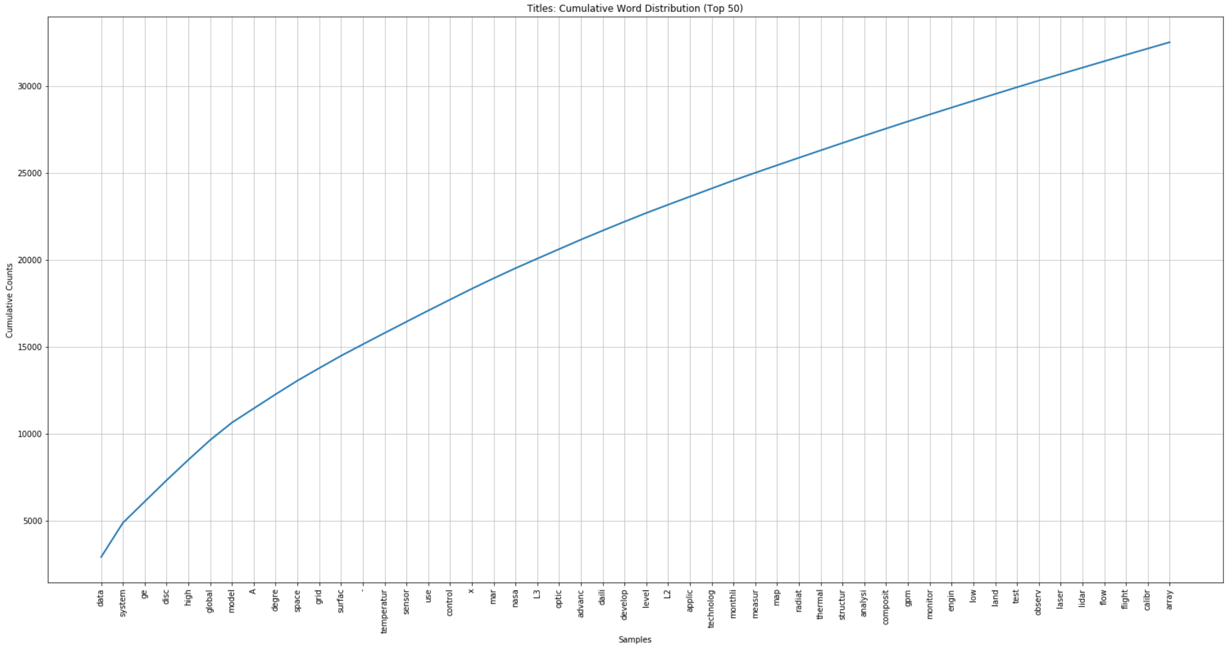
test_aaa.py

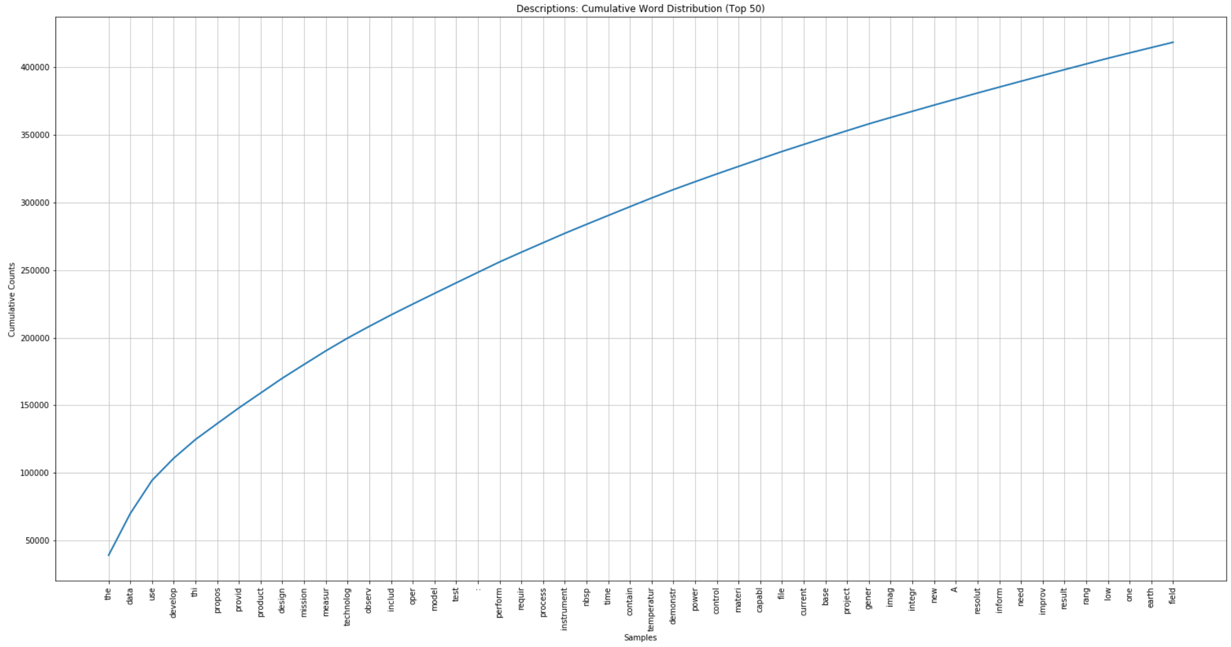
These files are scratchpad tests for speed due to the performance limits of my laptop.

Deliverable 4: End user communication

End user communication is delivered by the function `user_comm` in *user_comm.py*

In its current state it shows the cumulative distribution of the top 50 occurring words for both the titles and descriptions of the data sets. The top 10 most common words and the number of their occurrences are output to the console. These files are also located in the results folder as .png files.





VII) Future work and Remarks

Regrettably, I did not get to explore additional methods in the sub-fields of NLP such as summarization or sentence simplification. The vast majority of my time was spent on cleaning the data for processing.

Furthermore, the NASA API has a rather low rate limit of 1000 requests per hour, so I resorted to making rather slow GET requests instead. This meant that in order to run the whole corpus, it would take an upwards of three hours on my more powerful desktop. I actually tried to run that process on my laptop and the estimated time was well over nine hours. Clearly there is some space for optimization of my code, though the method makes it inherently slow.

VIII) References

1. All Topics A-Z. (2019). Retrieved 2 December 2019, from https://www.nasa.gov/tags/
2. Barber, J. (2019). Latent Dirichlet Allocation (LDA) with Python. Retrieved 2 December 2019, from https://rstudio-pubs-static.s3.amazonaws.com/79360_850b2a69980c4488b1db95987a24867a.html
3. Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. Sebastopol: O'Reilly Media, Inc.
4. NASA Open Data Portal. (2019). Retrieved 2 December 2019, from https://data.nasa.gov/browse#
5. Topic Modeling and Latent Dirichlet Allocation (LDA) in Python. (2019). Retrieved 2 December 2019, from https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24
6. Why do search results change depending on your language setting?. (2019). Retrieved 2 December 2019, from https://support.springer.com/en/support/solutions/articles/6000081845-why-do-search-results-change-depending-on-your-language-setting-