Second assignment

Problem: Software engineers use issue tracking systems to discuss different design issues. Currently, we have a classification of issues regarding the types of design decisions. This is however an abstract view for the design issues and does not give a guide about the topics discussed in issues. This prevents us to create tools that help software engineers search for certain design issues.

Goal: Explore the topics of design issues using LDA (a topic modeling algorithm) and explore their characteristics and co-occurrences with the decision types.

Each group will focus on a specific project with a specific list of issues (attached).
Each issue is classified based on three types of design decisions.

Group	**Project**
Group 1	Hadoop common
Group 5	Tajo
Group 6	HDFS
Group 7	Yarn
Group 8	Cassandra
Group 10	Map-Reduce

To achieve this goal, I propose these guiding steps in the following weeks:

1st week: **Download issue data and create vocabulary**.
In this week, you will prepare the data from issues and perform initial pre-processing of issue description. You will make the following steps:

1. *Use the Jira API to determine parent of an issue*: Please note that most issues do not have parents.
2. *Download issue data using jira API:* Specifically, you need to write a python program that download and store issue summary, description, all fields such as type and status, as well as number of comments. Please refer to the lecture for full data structure. Store this data in a good format as you will need it in the next steps.
3. *Perform tokenization and pre-processing for issue summary and description*:
   - Concatenate both issue summary and description.
   - Perform text cleaning and tokenization for the concatenation of issue summary and description. For this, you can use or (re-use) the Rust code in this repository. The Rust code can be called from python (e.g., this code).
   - Remove stop words.
   - Perform lemmatization and/or stemming.
4. *Create the vocabulary*: Based on the pre-processed issues, create a list of unique tokens in all issues, and count the number of occurrences of each token in all issues to create the vocabulary of the corpus.
   - What tokens have the most frequency?
   - Are there tokens candidate to be pre-processed, removed, or replaced with an ontology class?

Notes on running Rust code: To run Rust code from python, follow these steps:

1. Install Rust
2. clone the dl_manager repo

3. run `pip install nltk setuptools_rust`
4. run `python setup.py build_ext --inplace`

2nd week: **Run LDA on the issue description and determine topics.**
In this week, you will run the LDA algorithm on the list of tokens per issue. You need to perform the following steps:

1. *Create document token matrix for all issues after pre-processing*. This will be the input for the LDA.
2. *Run LDA with standard parameters*: It is very important to create useful topics. This can be done, if you adjust the tokenization and pre-processing to remove or replace tokens.
   Examples of useful topics:
   - Issues on security problems.
   - Issues on technology upgrades.
   - Issues on refactoring.
   - Issues on adding components.

   Examples of non-useful topics:
   - Issues on certain features of Cassandra.
   - Issues that focus on certain classes or methods.

   Try to adjust the pre-processing to get useful topics and prevent non-useful topics.
3. *Calculate and plot the perplexity* to evaluate the coherence of the topics.

Note: You can use the guide of this repository to write your code.

3rd week, **Fine tune pr-processing to create LDA topics, and analyze topics.**
In this week, you will fine tune the pre-processing step to create the final topics. Also, you will analyze the issues of each topic and their co-occurrences with the types of design decisions.

1. What topics emerge from LDA and what are common keywords per topic?

This is the main result of LDA, which contains the top list of keywords per topic, and a simple definition about the topic.

2. How many issues from the manual and automatic dataset discuss each topic?

A chart which shows the number of issues per topic.

3. What are the characteristics of the issues in each topic?

Here, you will analyze the fields that belong to each topic. For example, the number of comments and attachments for each topic, and the issue types for each topic. Also, the size of description of each topic. Charts like box plots and significance tests will be useful to apply.

4. Is there a significant co-occurrence between types of design decisions and the LDA topics?

Here, you will apply significance test on the co-occurrences between LDA topics and the three types of design decisions.