

Test Results on 4/20/2017

Feature: Search button (cloud)

```
Feature: Search button (cloud page)
  order to prove that the search button works
  the cloud page, we need to test that each of its
  features work as intended.

Scenario: Search button directs user to CloudPage # features/cloud/SearchButton.feature:6
  Given I search Halfond from cloud page          # FeatureContext::iSearchHalfondFromCloudPage()
    | Test passed = Still at same page
  Then I should be on cloud page                  # FeatureContext::iShouldBeOnCloudPage()
```

Feature: Search button (home)

```
Feature: Search button (home page)
  order to prove that the search button works
  the home page, we need to test that each of its
  features work as intended.

Scenario: If search name exists, navigate to cloud page # features/home/SearchButton.feature:6
  Given I search Halfond from home page                # FeatureContext::iSearchHalfondFromHomePage()
    | Test passed = Current url: http://127.0.0.1:8000/Halfond/8
  Then I should be on cloud page                      # FeatureContext::iShouldBeOnCloudPage()
```

Feature: Word cloud font sizes

```
Feature: Word cloud font sizes
  word cloud should have different font sizes according to the word frequencies

Scenario: Generate word cloud from home page # features/cloud/FontSize.feature:4
  Given I search Bengio from home page      # FeatureContext::iSearchBengioFromHomePage()
  Then the word cloud generated should contain different font sizes # FeatureContext::theWordCloudGeneratedShouldContainDifferentFontSizes()

Scenario: Generate word cloud from paper list # features/cloud/FontSize.feature:8
  Given I click on an author from list for focuses and Halfond # FeatureContext::iClickOnAnAuthorFromListForFocusesAndHalfond()
  Then the word cloud generated should contain different font sizes # FeatureContext::theWordCloudGeneratedShouldContainDifferentFontSizes()
```

Feature: Word cloud title

```
Feature: Word cloud title
  users should be able to
  see the searched text that
  is initialized to the word cloud

Scenario: The label's text lists the searched text of the Word Cloud Title is initialized for # features/cloud/WordCloudTitle.feature:3
  Given I am on the cloud page for Boehm # FeatureContext::iAmOnTheCloudPageForBoehm()
    | Test passed = h2 is Boehm
  Then the header should contain Boehm # FeatureContext::theHeaderShouldContainBoehm()
  Boehm()
```

Feature: Textbox

```
Feature: Textbox
  In order to prove that the textbox works
  On the home page, we need to test that each of its
  features work as intended.

Scenario: Click the search button when search bar is empty # features/home/Textbox.feature:6
  Given I am on the homepage and I search an empty form # FeatureContext::iAmOnTheHomepageAndISearchAnEmptyForm()
  | Test passed = Current url: http://127.0.0.1:8000/
  Then I should be on home page # FeatureContext::iShouldBeOnHomePage()

Scenario: Search button is clickable if and only if the Search Bar is filled # features/home/Textbox.feature:10
  Given I am on the homepage and I search for Halfond # FeatureContext::iAmOnTheHomepageAndISearchForHalfond()
  | Test passed = Current url: http://127.0.0.1:8000/Halfond/8
  Then I should be on cloud page # FeatureContext::iShouldBeOnCloudPage()
```

Feature: Abstract page

```
Feature: Abstract page
  Once clicking a paper title, the abstract page should appear

Scenario: Abstract page PDF link # features/papers/Abstract.feature:4
  Given I click the first title from the list for prone and Lupu # FeatureContext::iClickTheFirstTitleFromTheListForProneAndLupu()
  | method not completed
  Then clicking the PDF link opens a PDF # FeatureContext::clickingThePdfLinkOpensAPdf()
```

Feature: Author name

```
Feature: Author name
  Clicking on an author in the paper's authors list will do a new search

Scenario: If search name exists, navigate to cloud page # features/papers/AuthorName.feature:4
  Given I click on Sonal Mahajan from list for focuses and Halfond # FeatureContext::iClickOnSonalMahajanFromListForFocusesAndHalfond()
  | Test passed = Current url: http://localhost:8000/Mahajan/5
  Then I should be on the word cloud page for Sonal Mahajan # FeatureContext::iShouldBeOnTheWordCloudPageForSonalMahajan()
```

Feature: Paper list

```
Feature: Paper list
  Paper list page should contain the things needed

Scenario: Page should contain the following titles # features/papers/PaperTitle.feature:4
  Given I am on the list for focuses and Halfond8 # FeatureContext::iAmOnTheListForFocusesAndHalfond2()
  Then the page should contain the titles: # FeatureContext::thePageShouldContainTheTitles()
  | title |
  | Command-Form |
  | Detection and Localization |
  | Penetration Testing |
  | Estimating Android |
  | Detecting Display |

Scenario: Page should contain the following conference names # features/papers/PaperTitle.feature:14
  Given I am on the list for focuses and Halfond8 # FeatureContext::iAmOnTheListForFocusesAndHalfond2()
  Then the page should contain the conference names: # FeatureContext::thePageShouldContainTheConferenceNames()
  | conference |
  | 2009 International |
  | 2015 IEEE 8th |
  | 2013 28th |
  | 2012 First International |
  | 21st IEEE/ACM |

Scenario: Page should contain the following conference names # features/papers/PaperTitle.feature:24
  Given I am on the list for easily and Cote # FeatureContext::iAmOnTheListForEasilyAndCote()
  Then the page should be ordered by frequency # FeatureContext::thePageShouldBeOrderedByFrequency()

Scenario: Page should contain the following conference names # features/papers/PaperTitle.feature:28
  Given I am on the list for fields and Cote # FeatureContext::iAmOnTheListForFieldsAndCote()
  Then clicking authors will order to list based on authors # FeatureContext::clickingAuthorsWillOrderToListBasedOnAuthors()

Scenario: Page should contain the following conference names # features/papers/PaperTitle.feature:32
  Given I am on the list for system and Cote # FeatureContext::iAmOnTheListForSystemAndCote()
  Then clicking title will order to list based on titles # FeatureContext::clickingTitleWillOrderToListBasedOnTitles()
```

Feature: Conference word cloud

(there's an error; clicking on a word from a wordcloud based on a conference yields an empty paper list)

```
Feature: Conference word cloud
  Once clicking on a conference name, a new list is generated

  Scenario: Paper should contain the following titles # features/papers/ConferenceWordCloud.feature:4
    Given I click on the first conference from the list for term and Bengio # FeatureContext::iClickOnTheFirstConferenceFromTheListForTermAndBengio()
    | Test passed = At word cloud page for the specified conference
    Then I should be on the word cloud for IEEE Transactions # FeatureContext::iShouldBeOnTheWordCloudForIeeeTransactions()

  Scenario: Fixing bug with word cloud generated by conference # features/papers/ConferenceWordCloud.feature:8
    Given I click on the first conference from the list for across and Lee # FeatureContext::iClickOnTheFirstConferenceFromTheListForAcrossAndLee()
    word cloud not populated (Exception)
    And I click on the word second # FeatureContext::iClickOnTheWordSecond()
    Then I should be on a populated paper list page # FeatureContext::iShouldBeOnAPopulatedPaperListPage()
```

Feature: Paper list download

```
Feature: Download paper list
  Paper list page should be downloadable

  Scenario: Download paper list as PDF # features/papers/PaperListDownload.feature:4
    Given I click download as PDF in list for focuses and Halfond # FeatureContext::iClickDownloadAsPdfInListForFocusesAndHalfond()
    Then a download of PDF for paper list begins # FeatureContext::aDownloadOfPdfForPaperListBegins()

  Scenario: Download paper list as Plaintext # features/papers/PaperListDownload.feature:8
    Given I click download as plaintext in list for focuses and Halfond # FeatureContext::iClickDownloadAsPlaintextInListForFocusesAndHalfond()
    Then a download of plaintext for paper list begins # FeatureContext::aDownloadOfPlaintextForPaperListBegins()
```

Feature: Word cloud download

```
Feature: Word cloud download
  User can download the generated word cloud

  Scenario: Download begins after clicking download # features/cloud/WordCloudDownload.feature:4
    Given I click download on the cloud page for Bengio # FeatureContext::iClickDownloadOnTheCloudPageForBengio()
    Then a download for Bengio's wordcloud begins # FeatureContext::aDownloadForBengioSwordcloudBegins()
```

Unit testing

```
OK (7 tests, 7 assertions)
```

```
Code Coverage Report:  
2017-04-21 01:51:45
```

Summary:

```
Classes: 38.46% (5/13)  
Methods: 53.57% (15/28)  
Lines: 62.16% (69/111)
```

```
\App\Console::Kernel  
Methods: 50.00% ( 1/ 2) Lines: 33.33% ( 1/ 3)  
\App\Http\Controllers::PagesController  
Methods: 100.00% ( 6/ 6) Lines: 100.00% ( 48/ 48)  
\App\Providers::AppServiceProvider  
Methods: 100.00% ( 2/ 2) Lines: 100.00% ( 2/ 2)  
\App\Providers::AuthServiceProvider  
Methods: 100.00% ( 1/ 1) Lines: 100.00% ( 2/ 2)  
\App\Providers::EventServiceProvider  
Methods: 100.00% ( 1/ 1) Lines: 100.00% ( 2/ 2)  
\App\Providers::RouteServiceProvider  
Methods: 100.00% ( 4/ 4) Lines: 100.00% ( 14/ 14)
```

The undefined scenarios and steps are for the features that aren't implemented yet.

```
24 scenarios (18 passed, 1 failed, 5 undefined)  
52 steps (36 passed, 1 failed, 12 undefined, 3 skipped)  
2m57.99s (16.03Mb)
```