Use Cases for the Software Engineering project:
"Web-based visualization and corpus management for language modeling"

Developers:

Tiyash Basu, Bhavesh Bhansali, Supratim Das, Satabdi Ganguly, Clayton Greenberg, and Rohit Kumar Gupta

Tutor:

Rafaella Antonyan

Customer:

Prof. Dr. Dietrich Klakow, Spoken Language Systems Group, Saarland University

Index of Use Cases:

1. Login

Primary actor:	Researcher
Goal in context:	A researcher wants to access the features of the site
Precondition:	The researcher has permission to use the site
Trigger:	Researcher logs on to https://lm.lsv.uni-saarland.de

Scenario:-

1. The system prompts for a username and password.
2. The researcher enters his/her username.
3. The researcher enters his/her password.
4. The researcher selects login.
5. The system authenticates the logins credentials: first against the approved LDAP directory (LSV), then second against the local database.
6. If a user exists in the approved LDAP directory but does not have a local account, one is created.
7. The system displays function groups, namely, Corpus Management, Language Models, Run Experiments.

Exceptions:-

1. The researcher does not enter a username or password - the system displays the error message: “Please fill out this field” and remains on the login screen.
2. The username/password combination is not valid according to either the LDAP directory or local database - the system displays the error message “The username or password is incorrect” and remains on the login screen.

Priority:	High Priority
When Available:	Second Iteration
Frequency of use:	Frequent

2. Add Corpus

Primary actor:	Researcher
Goal in context:	Add new corpus for LM experimentation.
Precondition:	Appropriate login credentials must be obtained. A valid corpus file must exist on the researcher's current remote location OR within the GUI's accessible file system.
Trigger:	Researcher decides to upload his/her corpus.

Scenario:-

1. The researcher logs into the 'LSVLM' GUI website (see Use Case 1).
2. The researcher selects 'Corpus Management'
3. The system displays the corpora index page.
4. The researcher selects the 'Add new corpus' link.
5. The system expands the add corpus form.
6. The researcher enters metadata information into the fields of the add corpus form.
7. The researcher selects the 'Add Corpus' button.
8. The system displays the confirmation page of the corpus upload.
9. The researcher selects the 'Continue' button.
10. The file is uploaded to the server (if applicable) and the system displays the corpus metadata page.

Exceptions:-

1. Login credentials are incorrect or not recognized - see Use Case 1
2. The researcher selects Language Models - see Use Cases 6 - 8
3. The researcher selects Run Experiments - see Use Cases 9 - 10
4. From the corpora index, the researcher selects a corpus - see Use Cases 3 - 4
5. The researcher does not supply a valid corpus name - the system highlights the "name" field in red and displays the error message "This field is required."
6. The researcher supplies neither a description nor a metadata URL for the corpus - the system highlights the fields "Description" and "Metadata URL" and displays the error message "Enter either description or metadata_url"
7. The location supplied by the researcher does not correspond to a file on the server - the system highlights the "corpus file path" field in red and displays the error message "Invalid file path."

Priority:	High Priority
When Available:	Second Iteration
Frequency of use:	Frequent

3. Edit Corpus

Primary actor:	Researcher
Goal in context:	Edit the metadata for a corpus in use for LM experimentation.
Precondition:	Appropriate login credentials must be obtained. The corpus must be present in the site database. The corpus must belong to the researcher.
Trigger:	Researcher decides to edit the metadata for his/her corpus.

Scenario:-

1. The researcher logs into the ‘LSVLM’ GUI website (see Use Case 1).
2. The researcher selects ‘Corpus Management’.
3. The system displays the corpora index page.
4. The researcher selects a corpus.
5. The system displays the corpus metadata page.
6. The researcher selects the “Edit Corpus” button.
7. The system displays the edit corpus form.
8. The researcher makes his/her desired changes to the corpus metadata.
9. The researcher selects the “Save Corpus” button.
10. The system returns to the corpus metadata page.

Exceptions:-

1. Login credentials are incorrect or not recognized - see Use Case 1
2. The researcher selects Language Models - see Use Cases 6 - 8
3. The researcher selects Run Experiments - see Use Cases 9 - 10
4. The edits to the corpus metadata do not pass validation tests - see Use Case 2

Priority:	Moderate
When Available:	Second Iteration
Frequency of use:	Moderate

4. Delete Corpus

Primary actor:	Researcher
Goal in context:	Delete corpus because it will not be used for more experiments
Precondition:	Appropriate login credentials must be obtained. The corpus must be present in the site database. The corpus must belong to the researcher.
Trigger:	Researcher decides to delete his/her corpus.

Scenario:-

1. The researcher logs into the ‘LSVLM’ GUI website (see Use Case 1).
2. The researcher selects ‘Corpus Management’
3. The system displays the corpora index page.
4. The researcher selects a corpus.
5. The system displays the corpus metadata page.
6. The researcher selects either “Delete Corpus Record and Files” or “Delete Corpus Record Only”
7. The system displays the pop-up warning message “Are you sure you want to purge/delete NAME_OF_CORPUS?”.
8. The researcher selects “OK”
9. The system deletes the record of the corpus.
10. If the appropriate button was selected, the system deletes the corpus files.
11. The system returns to the corpora index page.

Exceptions:-

1. Login credentials are incorrect or not recognized - see Use Case 1
2. The researcher selects Language Models - see Use Cases 6 - 8
3. The researcher selects Run Experiments - see Use Cases 9 - 10
4. The researcher selects ‘Cancel’ on the warning message - the system does not delete anything and remains on the metadata page for the corpus.

Priority:	Moderate
When Available:	Second Iteration
Frequency of use:	Moderate

5. Search corpora

Primary actor:	Researcher
Goal in context:	Find the name of or view a desired corpus by searching for its label(s).
Precondition:	Appropriate login credentials must be obtained.
Trigger:	Researcher decides to search the corpora.

Scenario:-

1. The researcher logs into the ‘LSVLM’ GUI website (see Use Case 1).
2. The researcher selects ‘Corpus Management’.
3. The system displays the corpora index page.
4. The researcher enters some non-empty text into the search field.
5. The researcher selects “Find corpora with these labels”.
6. The system displays the corpora index page with only those corpora that contain all labels in the search query.

Exceptions:-

1. Login credentials are incorrect or not recognized - see Use Case 1
2. The researcher selects Language Models - see Use Cases 6 - 8
3. The researcher selects Run Experiments - see Use Cases 9 - 10
4. From the corpora index, the researcher selects a corpus - see Use Cases 3 - 4
5. The researcher clicks the search button without entering a query - trivially, all corpora contain every label that was entered, so the corpora index page remains unchanged.

Priority:	Moderate
When Available:	Second Iteration
Frequency of use:	Moderate

6. Add/Train Language Model

Primary actor:	Researcher
Goal in context:	Add a new language model configuration and train accordingly.
Precondition:	Appropriate login credentials must be obtained.
Trigger:	Researcher decides to make a language model.

Scenario:-

1. The researcher logs into the ‘LSVLM’ GUI website (see Use Case 1).
2. The researcher selects ‘Language Models’.
3. The system displays the language model index page.
4. The researcher selects the “Create New Language Model” button.
5. The system displays the LM design page.
6. The researcher enters metadata information into the fields of the LM design page.
7. The researcher sets language model parameters using options that dynamically appear.
8. The researcher selects the ‘Create LM’ button.
9. The system displays the metadata page for the new language model, including a button for downloading the machine-readable configuration file.
10. The researcher selects the “Train” button (available on the language model index page and the language model metadata page).
11. If training is complete, the metadata page will have a button for downloading the trained language model.

Exceptions:-

1. Login credentials are incorrect or not recognized - see Use Case 1
2. The researcher selects “Corpus Management” - see Use Cases 2 - 5
3. The researcher selects “Run Experiments” - see Use Cases 9 - 10
4. From the language model index page, the researcher selects a language model - see Use Cases 7 - 8
5. The researcher does not enter a name for the language model - the system displays the error message: “Please fill out this field.”
6. The researcher does not specify any parameters for the language model - the system configures the language model as a Unified Smoothing Trigram.
7. The researcher does not specify a required model parameter - the system displays the error message: “This field is required.”

Priority:	Moderate
When Available:	Third Iteration
Frequency of use:	Moderate

7. Edit an existing Language Model configuration

Primary actor:	Researcher
Goal in context:	Edit language model configuration and re-train accordingly.
Precondition:	Appropriate login credentials must be obtained. The language model must be present in the site database. The language model must belong to the researcher.
Trigger:	Researcher decides to edit a language model configuration

Scenario:-

1. The researcher logs into the ‘LSVLM’ GUI website (see Use Case 1).
2. The researcher selects ‘Language Models’.
3. The system displays the language model index page.
4. The researcher selects a language model.
5. The system displays the language model metadata page.
6. The researcher selects the “Edit LM” button.
7. The system displays the language model design page.
8. The researcher makes changes to the metadata or parameters of the language model (see Use Case 6).
9. The researcher selects the “Save LM” button.
10. The system displays the warning message: “Are you sure you want to save changes to LANGUAGE_MODEL_NAME? It will need to be retrained.”
11. The researcher selects “OK”
12. The system deletes the current language model or configuration files if they exist.
13. The system displays the metadata page for the new language model.
14. The researcher selects the “Train” button.
15. If training is complete, the metadata page will have a button for downloading the trained language model.

Exceptions:-

1. Login credentials are incorrect or not recognized - see Use Case 1.
2. The researcher selects “Corpus Management” - see Use Cases 2 - 5.
3. The researcher selects “Run Experiments” - see Use Cases 9 - 10.
4. From the language model index, the researcher selects the “Create New Language Model” button - see Use Case 6.
5. The desired language model is not present in the index - see Use Case 6
6. The edits to the language model configuration do not pass validation tests - see Use Case 6
7. The researcher selects ‘Cancel’ on the warning message - the system remains on the LM design page.

Priority:	Moderate
When Available:	Second Iteration
Frequency of use:	Moderate

8. Delete Language Model

Primary actor:	Researcher
Goal in context:	Delete language model and configuration because they will not be used for more experiments
Precondition:	Appropriate login credentials must be obtained. The language model must be present in the site database. The language model must belong to the researcher.
Trigger:	Researcher decides to delete a language model

Scenario -

1. The researcher logs into the 'LSVLM' GUI website (see Use Case 1).
2. The researcher selects 'Language Models'.
3. The system displays the language model index page.
4. The researcher selects a language model.
5. The system displays the language model metadata page.
6. The researcher selects "Delete LM Record and files" or "Delete LM Record Only"
7. The system displays the warning message: "Are you sure you want to purge/delete LANGUAGE_MODEL_NAME?"
8. The researcher selects "OK".
9. The system deletes the record of the language model.
10. If the appropriate button was selected, the system deletes the configuration file and the trained model if it exists.
11. The system displays the language model index page.

Exceptions -

1. Login credentials are incorrect or not recognized - see Use Case 1
2. The researcher selects "Corpus Management" - see Use Cases 2 - 5
3. The researcher selects "Run Experiments" - see Use Cases 9 - 10
4. From the language model index, the researcher selects the "Create New Language Model" button - see Use Case 6.
5. The desired language model is not present in the index - see Use Case 6
6. The researcher selects 'Cancel' on the warning message - the system does not delete anything and remains on the metadata page for the language model.

Priority:	Moderate
When available:	Second Iteration
Frequency of use:	Moderate

9. Input Text Experiment

Primary Actor:	Researcher
Goal in context:	Generate probabilities for words in a text using existing language models.
Preconditions:	Appropriate login credentials must be obtained. Desired language model must be trained.
Trigger:	The researcher wants to view surprisals for a non-corpus text.

Scenario -

1. The researcher logs into the 'LSVLM' GUI website (see Use Case 1).
2. The researcher selects "Run Experiments".
3. The system displays the experiment index page.
4. The researcher selects the 'Run Experiment' link.
5. The system expands the Run Experiment form.
6. The researcher selects one or more pre-trained language models from a list.
7. The researcher selects the "Input Text" radio button (by default this is selected).
8. The researcher types some text into the text box.
9. The researcher selects "Run".
10. The system displays the metadata page for the new experiment.
11. Once the experiment has finished running, this page shows delete buttons, perplexity results (available for download) and an interactive surprisal graph for the test text.

Exceptions -

1. Login credentials are incorrect or not recognized - see Use Case 1
2. Researcher selects "Corpus Management" - see Use Cases 2 - 5
3. Researcher selects "Language Models" - see Use Cases 6 - 8
4. From the experiment index page, the researcher selects an experiment - go to step 11.
5. In the Run Experiment form, the desired language model is not present in the list - see Use Case 6
6. On the Run Experiment form, the researcher selects the "Test Corpora" button - see Use Case 10
7. The researcher does not select at least one language model - the system highlights the list border in red and displays the error message: "Please select at least one trained LM".
8. The researcher does not enter an input text - the system highlights the list border in red and displays the error message: "Please enter a non-empty input text".

Priority:	Moderate
When available:	First Iteration
Frequency of use:	Frequent

10. Test Corpora Experiment

Primary Actor:	Researcher
Goal in context:	Generate probabilities for words in a text using existing language models.
Preconditions:	Appropriate login credentials must be obtained. Desired language model must be trained.
Trigger:	The researcher wants to view surprisals for an existing corpus.

Scenario -

1. The researcher logs into the LSVLM GUI website (see Use Case 1).
2. The researcher selects “Run Experiments”.
3. The system displays the experiment index page.
4. The researcher selects the ‘Run Experiment’ link.
5. The system expands the Run Experiment form.
6. The researcher selects one or more pre-trained language models from a list.
7. The researcher selects the “Test Corpora” radio button.
8. The researcher selects one or more existing corpora from a list.
9. The researcher selects “Run”.
10. The system displays the metadata page for the new experiment.
11. Once the experiment has finished running, this page shows delete buttons, perplexity results (available for download) and an interactive surprisal graph for the test text.

Exceptions -

1. Login credentials are incorrect or not recognized - see Use Case 1
2. Researcher selects “Corpus Management” - see Use Cases 2 - 5
3. Researcher selects “Language Models” - see Use Cases 6 - 8
4. From the experiment index page, the researcher selects an experiment - go to step 11.
5. In the Run Experiment form, the desired language model is not present in the list - see Use Case 6
6. On the Run Experiment form, the researcher does not select the “Test Corpora” radio button - see Use Case 9
7. The researcher does not select at least one language model - see Use Case 9
8. The researcher does not select at least one corpus - the system highlights the list border in red and displays the error message: “Please select at least one test corpus”.

Priority:	Moderate
When available:	Second Iteration
Frequency of use:	Frequent