# Requirements for the Software Engineering project: “Web-based visualization and corpus management for language modeling”

## Project Description:

The high level goal of this project is to create a functional, web-based graphical user interface (GUI) for the language modelling toolkit developed by the Spoken Language Systems Group, Saarland University (LSVLM). When possible, the design of this GUI must be close to that of the existing non-functional GUI in order to provide a consistent user experience for existing users. The GUI must extend the functionality of the LSVLM toolkit and existing GUI in the areas of corpus management and visualization of provided outputs (Experimentation). The intended users of this GUI are expected to know the basics of language modelling, but not command-line tools.

## 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

## Must-have features:

| Requirement | Priority | Functional (F) / Non-Functional (NF) |
|---|---|---|
| The system must authenticate usernames and passwords, preferably using the University authentication system. | 1 | F |
| The system must NOT allow users who are not in the authorized user database to login. | 1 | NF |
| The login page must be accessible to anyone from any location. | 1 | NF |
| The web interface must use HTTPS. | 1 | NF |
| A user must be able to add new corpora using the GUI. | 1 | F |
| The GUI must insist on a minimum amount of metadata for each corpus. | 1 | NF |
| A user must be able to edit metadata for his/her own corpora. | 1 | F |
| A user must be able to view others’ corpus metadata. | 1 | F |
| A user must NOT be able to edit and delete others’ corpora/metadata. | 1 | NF |
| The system must calculate word counts and file size when adding a corpus. | 1 | F |
| The GUI must display an intuitive structure for corpora, preferably a hierarchical file/folder system. | 1 | NF |

| | | |
|---|---|---|
| A user must be able to perform experiments (perplexity and probability in context visualizations) | 1 | F |
| A user must be able to create (graphically) and save LM configurations. | 1 | F |
| A user must be able to edit his/her own saved LM configurations. | 1 | F |
| A user must be able to delete his/her own saved LM configurations. | 1 | F |
| A user must be able to download his/her machine readable LM configurations. | 1 | F |
| A user must be able to view others' LM configurations. | 1 | F |
| A user must NOT be able to edit others' LM configurations. | 1 | NF |
| A user must NOT be able to delete others' LM configurations. | 1 | NF |
| A user must be able to train LMs using his/her own configurations and corpora. | 1 | F |
| A user must be able to download his/her own ngram LMs in ARPA format. | 1 | F |
| A user must be able to delete his/her own trained LMs. | 1 | F |
| A user must NOT be able to delete others' trained LMs. | 1 | NF |
| The system must obtain a confirmation from the user before deleting data. | 1 | NF |
| The user must be able to train: class LMs, linearly interpolated LMs, Zero LMs, log-linearly interpolated LMs, map (skip) LMs, and [n-gram LMs with absolute discounting, Dirichlet, and Kneser-Ney smoothing] | 1 | F |
| A user must be able to select an appropriate experiment from a menu. | 1 | NF |
| A user must be able to load one or more previously trained LMs and some form of test corpus into a selected experiment. The test corpus must either already exist in the corpora database, or be provided directly at run time. | 1 | NF |
| A user must be able to obtain the perplexity of the test corpus according to the selected LMs. | 1 | F |
| A user must be able to obtain conditional probabilities of each word in the test corpus according to the selected LMs. | 1 | F |
| A user must be able to view the word vs. surprisal graph for the test corpus according to the selected LMs. | 1 | F |
| A user must be able to search for and view small context concordances for the test corpus according to the selected LMs. | 1 | F |
| A user must be able to view the corresponding word vs. surprisal graph for a selected entry in the concordance list. | 1 | F |
| A user must be able to view summary statistics (mean, min, max, std dev) over the occurrences of a given word in the test corpus. | 1 | F |

**May-have features:**

| Requirement | Priority | Functional (F) / Non-Functional (NF) |
|---|---|---|
| A user may be able to delete his/her own corpora. | 2 | F |
| A user may be able to do a lossy conversion of a non-ngram model to an ngram model in ARPA format. | 2 | NF |
| A user may be able to upload an existing LM configuration file such that the system can attempt to parse it and display it in the GUI. | 2 | F |
| A user may be able to view surprisals of occurrences in a histogram. | 2 | F |
| If user attempts to upload a LM configuration that is not one of the supported types, the system should provide a user-friendly error message. | 2 | NF |
| A user may be able to download others' trained ngram LMs (and converted LMs) in ARPA format. | 3 | F |
| A user may be able to obtain most likely completions (cloze and internal gaps) using trained LMs for user provided strings. | 3 | F |
| The admin may be able to change what metadata is required. | 3 | F |
| A user may be able to generate word clouds. | 4 | F |
| A user may be able to view summary statistics over the occurrences of a given string of words in the test corpus. | 4 | F |
| The LM configurations may be version controlled. | 4 | NF |
| The system may suggest metadata completions for corpora. | 4 | F |
| A user may be able to connect the trained LM and the test corpus to an appropriate acoustic model and generate word error rates. | 5 | F |
| A user may be able to create neural network architectures.  External dependency on how quickly NN capabilities are implemented outside of this project. | 5 | F |
| A user may be able to sample from a trained LM an artificial corpus of a given size and analyze the results. | 5 | F |
| A user may be able to visualize the trained model (e.g. neural network architecture). *(External dependency on how quickly NN capabilities are implemented outside of this project.)* | 5 | F |
| The system may be able to automatically collect and display parameters (required and optional) according to the definition in LSVLM | 5 | NF |

*Note: Features with priority 1 have the highest priority and items with priority 5 have the lowest priority.*

**Must-not-have features:**

| **Requirement** | **Priority** | **Functional (F) / Non-Functional (NF)** |
|---|---|---|
| Existing user accounts and data are migrated to the new system. | 6 | NF |
| The GUI extends the functionality of language model (LM) training, encoding, or architecture. | 6 | F |
| The Experimentation features require additional data than that which can be obtained from existing LSVLM functions. | 6 | NF |
| The GUI supports languages other than English for non-corpora related functions. | 6 | NF |
| The user can preprocess corpora within the GUI. | 6 | F |
| The user can edit trained language models within the GUI. | 6 | F |
| The user can perform standard NLP tasks within the GUI, such as part-of-speech tagging. | 6 | F |
| The system calculates custom notions of entropy, cross entropy, surprisal, or perplexity other than those defined in language modeling literature. | 6 | F |
| The system automatically detects presence or absence of sentence division markers and handles them in a custom way. | 6 | NF |
| The GUI has a mobile version. | 6 | NF |

As of 27 November 2015, the developers, tutor, and customer agree to the above enumeration, classification, and prioritization of features concerning the deliverable product associated with this project. Adjustments to these features must be addressed in a subsequent engineering iteration.

Clayton Greenberg, representative for the Developers

Rafaella Antonyan, Tutor

Dietrich Klakow, Customer [SIGNED 25.11.2015]