

Software Design

Specification

**Terminology Creation**

*Thao Thai, Bikram Pokhrel, and Mary Hogan*

*Revision 1.0*

Table of Contents Go Here

**1. Introduction**

**1.1 Purpose**

There is a lack of translation for technical terminologies in the World Wide Web. Not all the technical terms in English are being translated to different languages. Some languages do not even have the translation for these technical terms. Therefore, the goal of Terminology Creation is to provide a web interface translator for uncommon technical terminologies. Users will also be able to input their own translation of technical terms in the languages that they are fluent in. In addition, users can provide the back translation of the foreign terms to English. This web service will be interactive for both developers and users.

**1.2 Design Map**

The algorithm will be implemented in Python. It will pull the open source translation files for all and parse it using Python’s libraries for .po files, polib.

This project will use the Pointwise Mutual Information (PMI) algorithm to calculate the probability of a pair being the translation of each other in a set of sentences. The probability then will be ordered from the highest value to lowest value. The cut off PMI for a pair will be [insert a value].

A database for all the possible pairings will be created using SQL and PHP. This data will interact with the web user interface and retrieve the translation that correspond to the terms.

**1.3 Definitions**

**Back translation** – a procedure in which a translator or team of translators interpreted a previously translated document from one language back to its original language.

**Collocations** – an expression of two or more words that often go together

**Pointwise Mutual Information** – theory that measure the relationship and probability between two words and information that can be gained from them. Approach to find collocations in the sentences.

**2. Product Description**

This section gives background information about specific requirements of the web based software product to be developed in brief. This software product will be deployed to web site and all users of the product will access by use of the website. Website will be main user interface where users can operate all the provided functionality. To use this product, users can just view it publicly without registering unless users want to add or update the data in database.

**2.1 User characteristics**

We have identified three potential classifications of users of our system:

- Software developers: They generate the model of the software and implement it in code.

- System Administrator: Some one needs to be responsible for security and maintenance of

the system. This is System Administrators role.

- End Users: They are the intended audience who will be using this web based software after

its development. More often than not, software is designed for end users who are

client of this software.

**2.2 Assumptions**

**2.3 Constraints**

General design and implementation constraints include:

• The software shall be developed using python language.

• The web-browser shall be developed using PHP language and database using sql.

• The user interface for the software shall be compatible to any web-browser.

• This web-based software shall use the HTTP protocol for communication over the Internet.

**2.4 Dependencies**

**3. Requirements**

**3.1 Functional Requirements**

*Algorithm:*

The Pointwise Mutual Information algorithm is expected to run in O(n2) on each .PO files of the English sentences and the translated sentences.

It will parse the given .PO files and check for the probability of a pair. The top probability values will be selected to create a database of the terminologies for different languages

*Database:*

The Terminology entity will store over 1000+ words across 200 languages. The attribute will be “languages” with value being the translation of the technical terms.

There will be entities for users’ registered information and hash table of users’ passwords. The attributes of Users entity will consist of Name, Email, DOB, Security Answers, etc.

This information will be private to the public, only accessible to the developers.

*Web User Interface:*

There will be a sign-up/sign-in screen that allows user to create an account. This will keep track of legitimate users, preventing spam and inappropriate back translation.

The web user interface has an option of choosing the embedded languages in “Translate from” and “Translated to” menu. In addition, there will be a text box where the user can input the terms to be translated.

Once the user finish inputting the data, there will be a “Translate” button, which will execute the algorithm that interact with the database and find the translation of the inserted terms.

The back end of the user interface will generate a hash function for user’s passwords, this will create a layer of security for the users.

There will also be a public database that the users can insert their own back-translation of the undocumented terminologies.

**3.2 Manageability**

The terminologies will be updated in the database with users’ insertion of back-translations and translations for the non-existed languages in the database.

The Password and Registration entities will be constantly updated with new users sign-up and created account.

**3.3 Security**

Users will need to register to add or update any data in to database.

• The password will be encrypted using PHP hash function, this will add a layer of security to

the users’ accounts.

• The system shall automatically log out all users after a period of inactivity.

• The system shall confirm all transactions with the users web browser.

• The system shall not leave any cookies on the users on the user computer containing the

users password or other important informations.

• Users will have restriction on accessing database to only update an add data.

• The system's back-end server shall only be accessible to authenticated administrators.

**4. User Scenarios**