

**CS6113**

**Translation Technology Systems**

**ADRIAN HOGAN – 18159761**

**ARCHIT KHANNA – 18155723**

**DAVID BROMELL – 18055001**

**WALKTHROUGH**

* We utilised Scrapy as a web crawler in order to parse html files, this was an easy to use tool which was both fast and efficent.
* We then extracted the files into XLIFF 2.0 (we could not do 2.1 as the okapi xliff toolkit would not work for us) using okapi rainbow, this was an easy to use tool and it caused no issues throughout our extraction process.
* Once these files were in .xlf format, we used oxygen and lynx to preform a pre-translated mass validation, this was to ensure all the .xlfs were in a valid format so would not cause us trouble further into the process
* Next, we validated the Oasis Test Suite Files using both Lynx and Oxygen as a control in order to ensure the validators were working properly, this returned results we expected and caused no issues.
* The newly validated files were then translated using Memsource, this tool was quick and easy to use, overall, we preferred using Memsource to Trados. Once these files were translated correctly, it created .mxliff files which then had to be validated by Lynx and oxygen, these could not be validated as there is no support for mxliffs in the validator.
* Next, we once again machine translated the validated pre-translated files using SDL Trados. Although this tool was easy to use we did not favour it to Memsource for a number of reasons. Firstly, it created files which were xliff version 1.2, as a result these were not validatable by Lynx as it does not support backwards compatibility. Secondly it has SDL specific file formats, so it is not flexible. Lastly it does not allow the easy exporting of files, in general we found this tool to be much more of a hinderance than Memsource
* We used Github to manage the project, also created a traceability matrix in excel and released the project with the help of Github.

**Role David**

I contributed to this project by initiating the “Scrapy” framework, this is a web crawler which we used as a fast and efficient way to extract HTML content from various websites, and although this helped to speed up the HTML extraction process , we had to be careful about what websites we chose to use as we needed HTML code which was content rich. I also carried out the mass validation process using the Okapi Lynx command prompt, this aided in speeding up the validation process along with enabling us to quickly verify the content of a file. In addition to the above I also aided in the translation of files using SDL Trados. Lastly, I created and managed the GitHub for this project.

**Role Adrian**

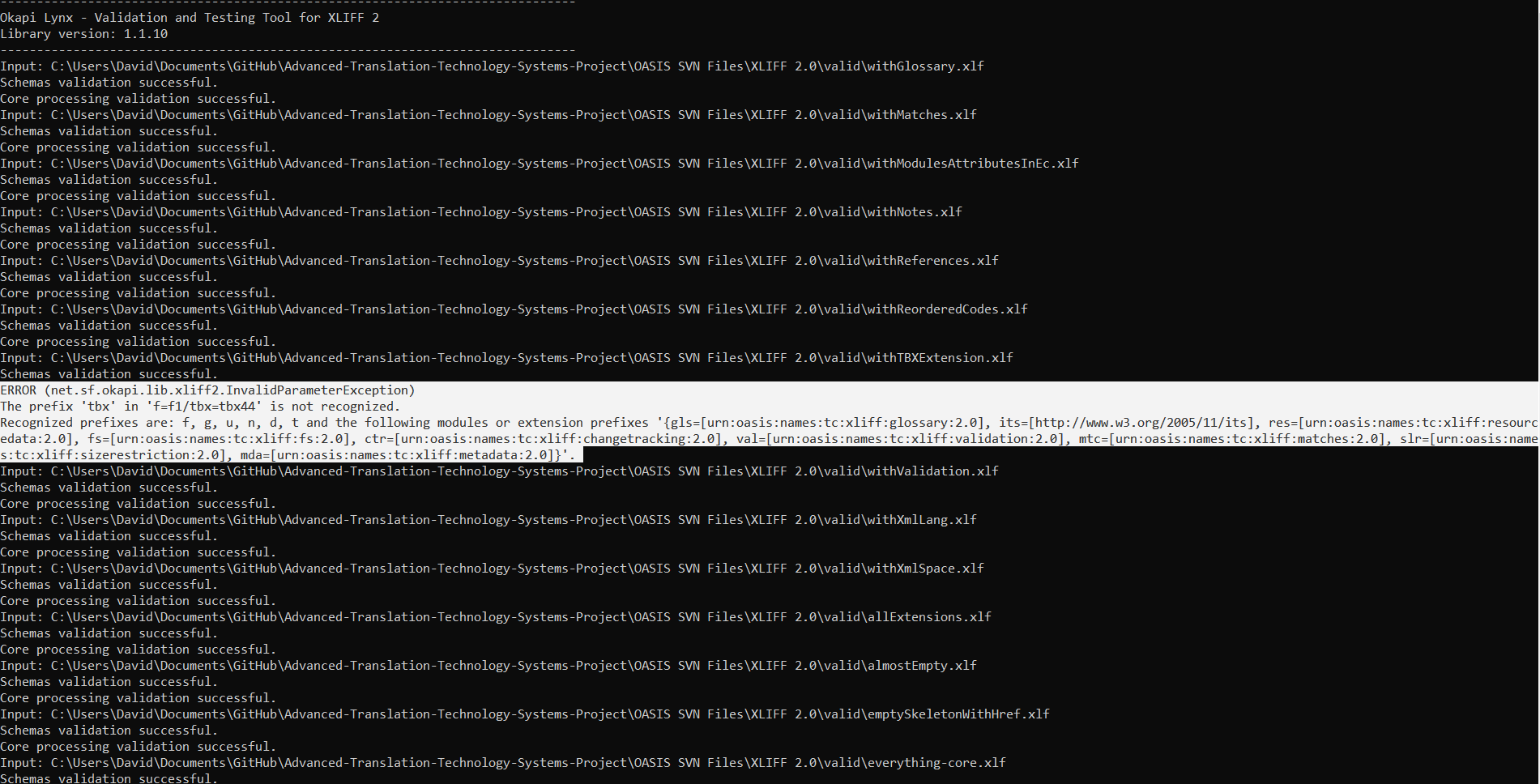
I oversaw the Memsource for the project, which I was the project manager of, I sourced translation memories for the project’s initial translation within Memsource and contributed by allocating the 7 different XLF files to different jobs for the team, which allowed for easier contributions. I also used the Okapi Rainbow framework in order to extract XLF 2.0 content from our HTML files which were extracted from the “Scrapy” framework. I contributed to the testing of the Oasis SVN test suite files in Lynx.

**Role Archit**

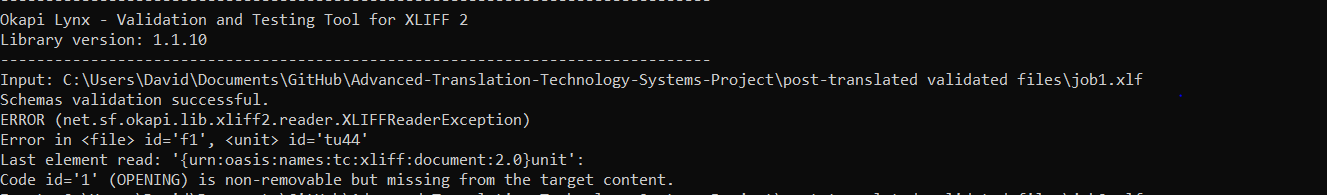
I contributed to the python code for Scrapy, finding appropriate websites to extract html from and assisted in carrying out the web crawling. I also worked on the translation of the xliff files in Memsource by carrying out the analysis, pre-translation using the sourced translation memories and then reviewed the translations using the Microsoft machine translation tool that is built into Memsource. After verifying the translations, I also confirmed the various segments in the files. I then exported these files as bilingual mxliff first and then only xliff. I also assisted in the translation of the files in SDL Trados and post-validating the files in Okapi Lynx.

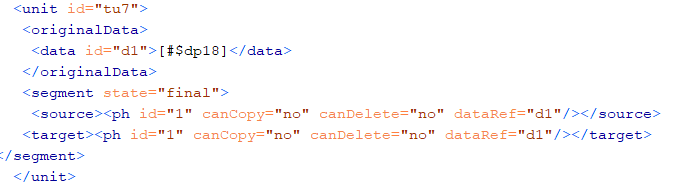
**Errors & Bugs**

1. TBX prefix not recognised in the valid Oasis test suite files. After examining the Lynx specification it appears it supports the TBX prefix, however when validating the files which were already pre-ordained to be valid, same was not prevalent, thus meaning there was an invalidity in the Oxygen Test Suite valid files. Please see the code snippet below as a reference. This issue will be logged on BitBucket



1. Here although the schema validation of the post translated files was successful and error was produced as “code id=1 (OPENING) is non-removeable but missing from the target content, however same was present in the file as is clear from the second screenshot below





<https://bitbucket.org/okapiframework/okapi/issues/889/>

https://bitbucket.org/okapiframework/xliff-toolkit/issues/34/