eSTEeM:  
Measuring qualification effects framework

Table of Contents

[Table of figures 2](#_Toc471918677)

[Summary 2](#_Toc471918678)

[Purpose 2](#_Toc471918679)

[Objective 3](#_Toc471918680)

[Supporting analysis of forum posts 4](#_Toc471918681)

[External links 5](#_Toc471918682)

[References 5](#_Toc471918683)

[Harvard reference 5](#_Toc471918684)

[Inline citation 5](#_Toc471918685)

[Focused analysis through filtering 5](#_Toc471918686)

[Classification 6](#_Toc471918687)

[Track posts during a course 7](#_Toc471918688)

[Visualising posts 7](#_Toc471918689)

[Prepare taxonomy workflow 8](#_Toc471918690)

[Suggest taxonomy keywords 10](#_Toc471918691)

[Filter *vs* Filterfalse 11](#_Toc471918692)

[Look at students as well as posts 12](#_Toc471918693)

[Student posts to a forum 12](#_Toc471918694)

[Students’ progress across courses 12](#_Toc471918695)

[UserIDs not students 13](#_Toc471918696)

[Using UserIDs to trace students 13](#_Toc471918697)

[Named Entity Recognition 14](#_Toc471918698)

[Future applications of NER 14](#_Toc471918699)

[Outputs 15](#_Toc471918700)

[Reports 16](#_Toc471918701)

[Data 18](#_Toc471918702)

[Future work 21](#_Toc471918703)

# Table of figures

[Figure 1: Forum screenshot 3](#_Toc471918704)

[Figure 2: Posts analysis workflow 4](#_Toc471918705)

[Figure 3: Bloom's Cognitive Taxonomy 6](#_Toc471918706)

[Figure 4: Prepare taxonomy workflow 8](#_Toc471918707)

[Figure 5: Suggest keywords workflow 10](#_Toc471918708)

[Figure 6: Filterfalse workflow 11](file:///C:\Users\David%20King\Documents\eSTEeM\docs\Report.docx#_Toc471918709)

[Figure 7: User analysis workflow 12](#_Toc471918710)

# Summary

As part of this eSTEeM project we have developed a framework to support our analysis and provide a model for similar future projects.

The primary function of the framework is to filter and consolidate forum posts for detailed review. In our available data, there are 6,813 forum posts for review. Filtering allows us to reduce this to 1,283 posts for review, about 19% of the original posts, and a far more manageable task. Further, the filtered posts are presented as consolidated reports, so removing the need to individually access every post in every forum in every course presentation that is part of the analysis. The framework delivers considerable time and effort savings in accessing relevant forum posts.

Among the forum’s secondary functions is classifying posts. The framework includes a working classifier, though its use in our analysis needs further refinement. As it stands, the classifier is a working model for other projects.

The framework also supports tracking forum usage during course presentation, enabling the analysts to easily relate posts to the course calendar. Similarly, students are tracked across courses, so providing information on who has studied which courses and in what order; which might be significant when reviewing their posts.

Examples of using the framework are provided in the Cookbook.

Accompanying the framework is a data file, containing the raw XML files for each VLE forum in our study, and available for others to analyse. This data is not readily available otherwise, ie it cannot be accessed directly from the VLE.

# Purpose

This report documents the project’s primary deliverable, a qualification-wide evaluation framework, so that the project may provide a model into the impact of changes in tuition/teaching strategies applicable to longitudinal studies of qualifications.

Though too few students have been through the new qualification so far for a statistically significant number of student case studies to be analysed, there are sufficient to support the development and testing of this evaluation framework, which can be applied as more qualification data become available.

# Objective

The primary objective of the framework is to support analysis of student forum posts, and thereby aid assessment of the impact of teaching interventions through the analysis of these posts.

The analysis workflow at the start of this project required direct review of each forum post in the forum. Hence, the academic analysing the forum posts had to log into each course, then each forum within the course, and select through each thread of posts. See Figure 1: Forum screenshot on page 3.

This workflow was neither manageable nor scalable as there can be many forums in a course, with each forum containing hundreds of posts, and each potential item of interest within the posts had to be identified manually. These are the initial problems addressed by the framework.

Through defining automatic scripts to make our analysis manageable, the framework helps ensure the rigour of our research. We had to define clearly, what we were doing, so that it could be automated, and with an iterative development cycle, we could refine the methodology to align with the research questions. The framework allows us to demonstrate what we’ve done, and makes it easier to show why what we’ve done is appropriate to the research questions

Through developing scripts to be scalable, we can also address the validity of our analysis through making it repeatable. Hence, the analysis can be run on other course presentations, making it easy to revisit the analysis as part of a longitudinal study, or cross-domain study.

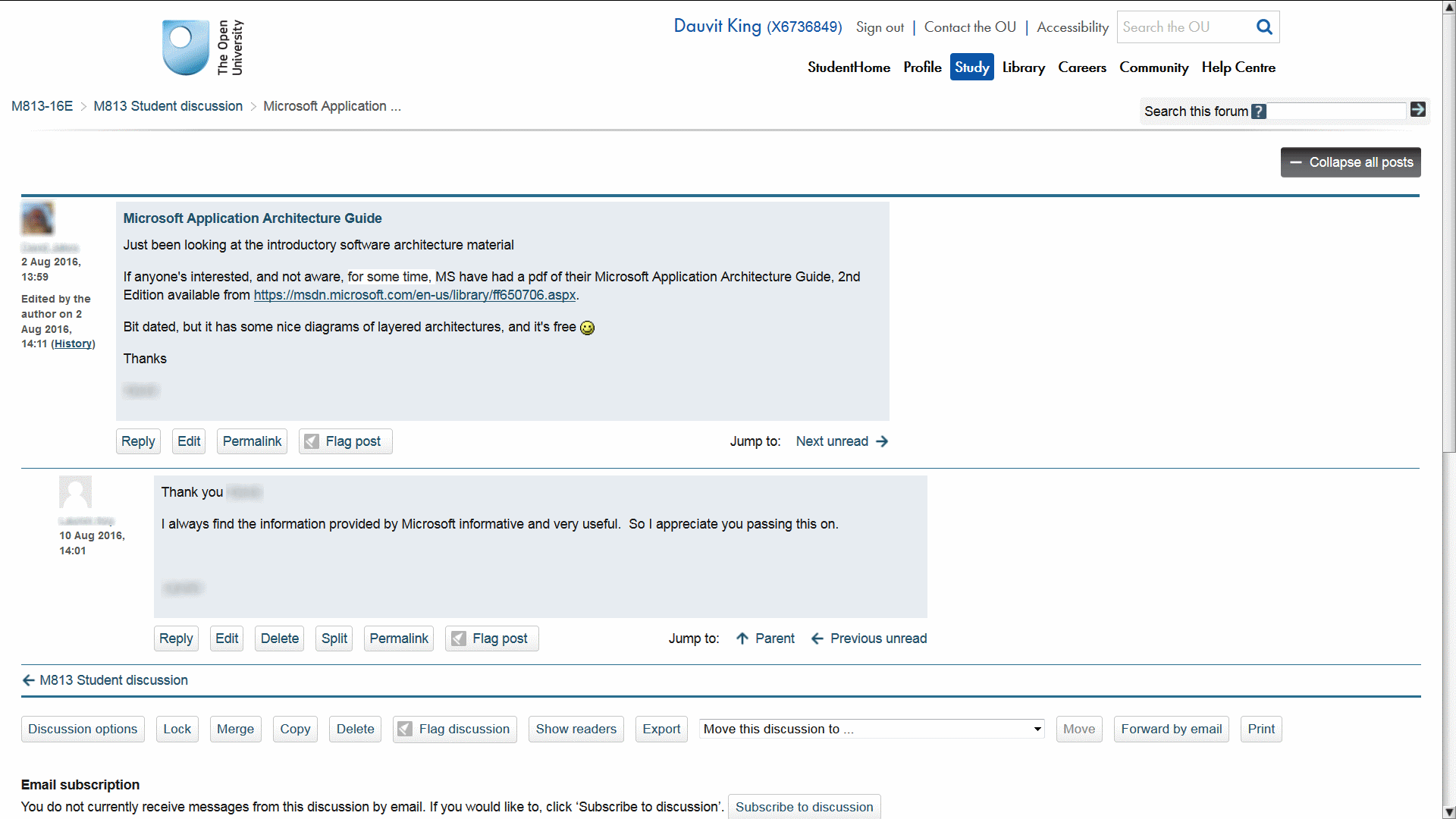


Figure 1: Forum screenshot

# Supporting analysis of forum posts

The initial focus in developing the framework was to support large-scale analysis of student posts. The framework workflow as ultimately developed can be seen[[1]](#footnote-1) in Figure 2: Posts analysis workflow on page 4.

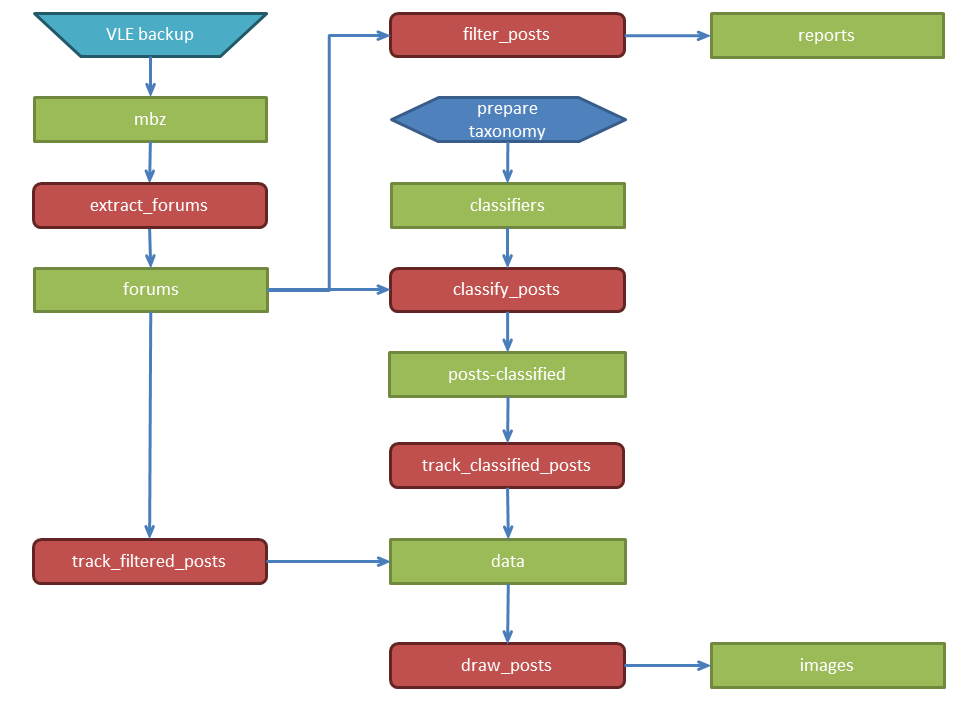


Figure 2: Posts analysis workflow

The workflow starts with acquiring a dump of the Moodle course: a VLE backup. This is done by request to LTS. The resulting mbz (Moodle Backup Zip) file is actually a tar file using gzip format for compression. The extract\_forums.py script extracts all forum xml files in an mbz file.

Once the forum XML files are stored in the forums folder, they are available for processing by framework scripts.

The filter\_posts.py script is the core of the framework. It filters the forum posts according to set criteria. Currently, four criteria are implemented and are described later in this report. (See External links on page 5, References on page 5, Harvard reference on page 5 and Inline citation on page 5.) The script produces reports that present a list of only those posts that meet the criteria. Hence, the close reading review of student posts is made manageable because only those posts that meet the criteria need to be reviewed. In addition, all of those eligible posts are presented in one consolidated report for that forum. This removes the need to access each post individually, considerably speeding up the review task.

To see how we use the framework to help refine our selection criteria, see Filter *vs* Filterfalse on page 11.

For a full description of all of the framework’s reports, see Reports on page 16.

For a full description of all of the framework’s data outputs, see Data on page 18.

The next four sections consider the currently applied filtering criteria.

## External links

The first criteria identified as being of interest was a student’s use of external references in their post. This is implemented by searching a post for the hypertext protocol identifiers ‘http’ and ‘https’ within an external link in each post.

This proved immediately effective in reducing the number of posts for detailed analysis to a manageable number. On reviewing the filtered posts, consideration was given to further analysis of the links to asses, perhaps via the link destination, the asses the quality of the link. However, this is a potentially large task to develop, and has not been progressed. Instead, three other filtering criteria were identified and implemented: references, Harvard reference and inline citation.

## References

Anecdotal review of the filtered posts suggested that students do put their references in a separate section at the end of their post, generally titled ‘References’. A filter selecting posts that use reference in this context is implemented.

## Harvard reference

Students are expected to format their references in the Harvard style. A filter for selecting posts with Harvard style year dates, ie full four digits enclosed within parentheses without spaces) is implemented.

## Inline citation

Feedback from Mark Slaymaker on students use of forums highlighted that often inline numeric references are used in posts, this being easy to type given the restrictions of the VLE text editor the students use to enter their posts into the forum. These inline references take the form of a number enclosed within square brackets without spaces. Therefore, a filter to select posts that have this style of inline reference is implemented.

## Focused analysis through filtering

The impact of reducing the number of posts that require review can be seen in the index\_to\_filtered\_posts.html report in the reports folder. In summary, using our existing data, after filtering we have 1,283 posts to review, out of the original 6,813 posts. This means that we have identified 5,530 that do not need review, representing a considerable reduction in the analysis workload.

For more details on the technical implementation of these filters, see the design.rst document on the docs folder.

## Classification

A second strand of the framework, alongside filtering posts, is classifying posts. This was one of the original inspirations for the framework and was inspired by Villarroel *et al*’s[[2]](#footnote-2) CLAP tool for categorising online software forum posts. The framework follows their process of tokenizing and then stemming words[[3]](#footnote-3) to aid finding significant terms. Currently, we have implemented a classifier based on Bloom’s Cognitive Taxonomy.

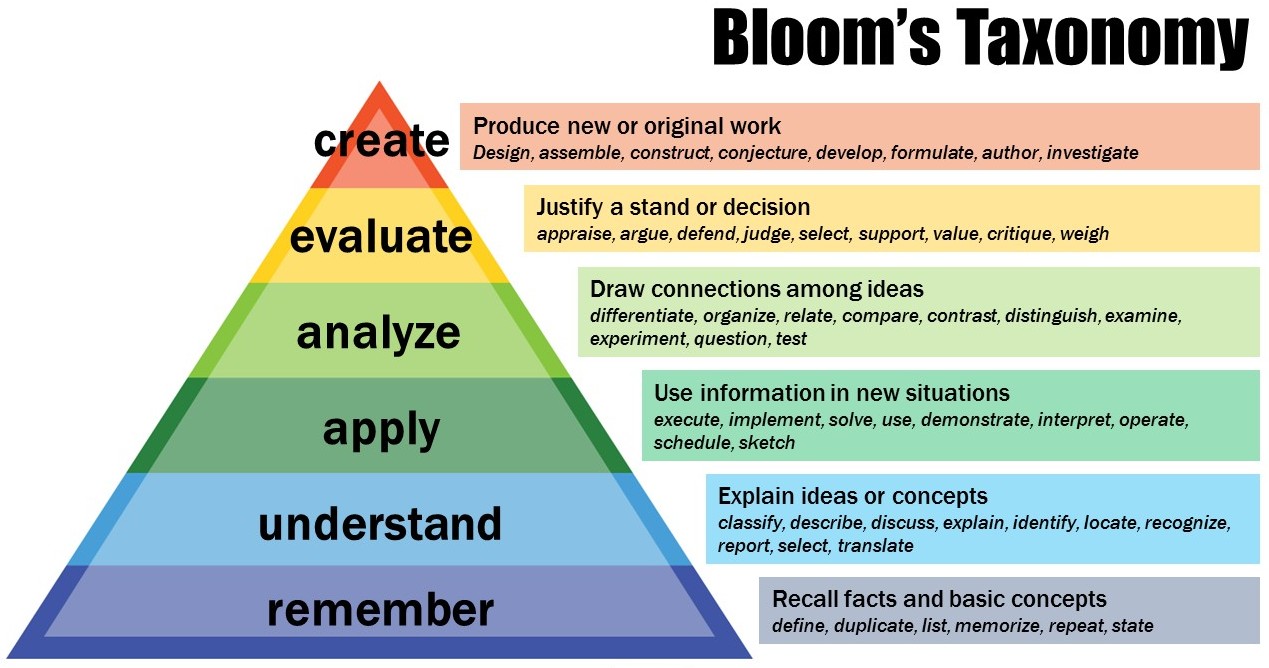


Figure 3: Bloom's Cognitive Taxonomy

To demonstrate the use of a classifier in our project, we selected Bloom’s Cognitive Taxonomy as revised by Krathwohl[[4]](#footnote-4) as an example classification schema. See Figure 3: Bloom's Cognitive Taxonomy on page 6, figure taken from <https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/>.

The taxonomy is intended to guide the authors of teaching and assessment materials as to the appropriate language to use when assessing different cognitive skills. As such, it clearly matches our needs, for we have terms to search for in forum posts that indicate the level of engagement with the teaching material by the student.

Note:

Posts can be classified in more than one category. Categories are not exclusive.

‘Relates’ is a keyword in both apply and create. Hence, any post that includes this term is classified in both categories.

The classification script, classify\_posts.py, applies the classifier to every forum in the source forum folder, producing a new forum file in the posts-classified folder containing the processed posts.

During the study this classification data was read in its raw state, to provide some insight into the nature of the posts. However, the full development of the workflow saw the creation of visualisations to make the insights more easily comprehended. The visualisations were incorporated into the work developing scripts to track the changing nature of posts during a course. See Track posts during a course on page 7,

## Track posts during a course

Two similar scripts, track\_filtered\_posts.py and track\_classified\_posts.py, process the filtered posts and classified posts to generate metrics on when during a course a post was made, who made, how many posts were made each day, and at what classification level. The scripts produce a variety of data files (for more details, see Data on page 18) that can be read in their own right, and also serve as the basis for visualising the posts in a forum.

## Visualising posts

During this study, several scripts to visualise data were written; however, only draw\_posts.py remains. It plots total number of posts against number of posting students through a course. Hence, one can easily see if a task engages many students or few, also one can easily see when posts are made, which can be usefully related to the course calendar. However, a common problem with all the scripts was scaling the graphs. There is a wide range of data values in the forums and relying on auto-scaling does not always produce pleasing, easy to read results. Therefore, in practice visualisation has been through Excel’s built-in chart functionality, generating line charts, stacked bar charts as appropriate and properly scaled ready for use in analysis and publications.

# Prepare taxonomy workflow

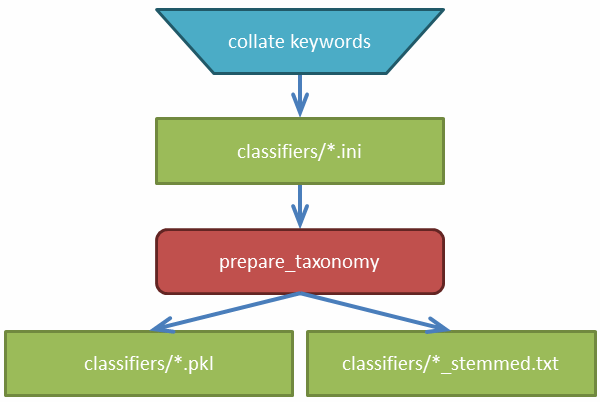


Figure 4: Prepare taxonomy workflow

The framework is supplied with a default taxonomy based on Bloom’s Cognitive taxonomy. See Figure 3: Bloom's Cognitive Taxonomy on page 6.

To analyse forum posts using a taxonomy, it is necessary to prepare the classification file used by the track\_classified\_posts.py script. See Figure 4: Prepare taxonomy workflow on page 8.

The first task is to manually collate keywords into an ini file. The format of this file is relatively straightforward, as can be seen in the example in the framework, see classifiers/bloom.ini. The format follows the rules for a Windows configuration file so that a classification level is defined by a word in square brackets and the list of words that follows it are the terms that indicate that classification level if found in a post. Comments in the ini file are indicated by starting a line with a semi-colon.

When collating words, duplicates are allowed, as are short phrases. Examples of both are included in classifiers/bloom.ini.

The raw collection of words is processed using prepare\_taxonomy.py to stem and de-duplicate the terms. The script produces two files: .pkl, a pickled file[[5]](#footnote-5) for use by the framework scripts; and .txt, a text file for a human readable view of the stemmed output.

We have not investigated the use of other taxonomies with the framework, though since completing the project a second classification scheme based on SOLO[[6]](#footnote-6) has been developed and tested. Another classification we would particularly like to explore in a follow on project is [BCS’s SFIA+ (British Computer Society’s IT Skills Framework)](http://www.bcs.org/category/17784).

One other development that did arise when reviewing this initial classification of forum posts using Bloom’s Cognitive Taxonomy was to consider creating a tailored classification scheme derived from the existing posts. This scheme would then be applied in a longitudinal study of subsequent presentations of the course in this study. This first step to achieving this goal was to prepare scripts within the framework help identify suitable keywords in our existing data.

# Suggest taxonomy keywords

Having established that we can apply a taxonomy to the forum posts, we wondered if a tailored taxonomy could be developed to provide more meaningful results than the Bloom Cognitive Taxonomy currently used. To that end, a new script was written to identify significant words in the existing forums. See Figure 5: Suggest keywords workflow on page 10.

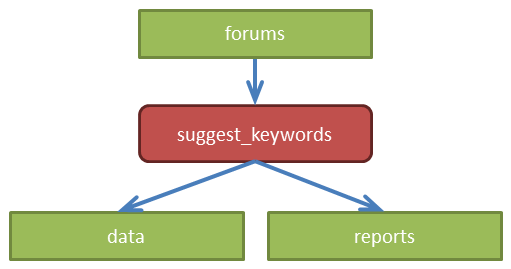


Figure 5: Suggest keywords workflow

The script suggest\_keywords.py produces a report \*\_significant\_words.html in the reports folder for each forum. The script uses tf:idf (term frequency–inverse document frequency) to identify significant words. This means that those words that appear more often in a particular forum (their term frequency) than they appear in all the forums considered together (their document frequency) is identified as significant. This approach produces some insight into each forum, but necessarily a lot of noise. For example, tutor names such as Ursula appear often in their own forums, but not at all in other forums, and so get highlighted as being significant in their forums.

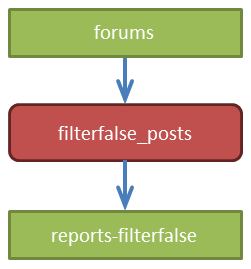
To help understand each forum further and see through this noise, the script has been extended to produce \*\_ word\_counts.csv in the data folder for each forum, and a summary \*\_ word\_counts.csv for the original source folder. These files have two columns:

* Word – the word itself
* Count – the number of times that word is used in the forum, or folder

The data is sorted into Count order, from highest to lowest, hence the most common word is first. Reviewing these tables is a crude, but helpful, mechanism to see what words are used in the posts, and may further suggest keywords on which to base future taxonomies.

The development of a tailored taxonomy could be the basis for future work

# Filter *vs* Filterfalse

A core use of the framework is to filter forum posts based on assorted selection criteria. To assist refining the criteria, the reports-filterfalse folder contains reports showing all the posts that did **not** meet the filtering criteria.

Each forum has an \*.html file, listing the excluded posts. For each listing, there is the id of the post, the userid of the post’s author, the post’s creation date and the post itself in its original html format.

The workflow to produce these reports matches that to produce the filtered reports: reading forum XML data current posts only are selected and then filtered. In this case, the filter criteria are reversed, so that posts not meeting the criteria are written to the filterfalse report.

Figure : Filterfalse workflow

The reports are produced by filterfalse\_posts.py. This script is derived from filter\_posts.py. The changes are: to move the data selection and writing statements to the other side of the else statement in filter\_forum(), the function being renamed filterfalse\_forum() in consequence; to drop the unnecessary \_filter epithet from the report file names; to direct the output to the reports-filterfalse folder.

These reports enable an academic wishing to refine the existing filter criteria, or to develop new filter criteria, to review only those posts that are excluded by the criteria. Hence, the academic is helped to engage in a focused review of these excluded posts only.

# Look at students as well as posts

## Student posts to a forum

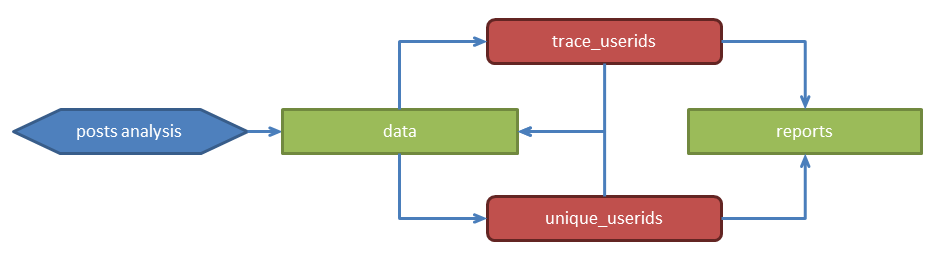


Figure 7: User analysis workflow

Having established the number and nature of posts made each day, we wanted to extend the framework’s functionality to look at students as well their posts.

To deliver this new requirement unique\_userids.py was developed to examine the forums, making use of the analysis already conducted of their posts thereby avoiding duplication, and track the userid of the authors. The output is a report \*\_usage.html in the reports folder for each forum, and a matching \*\_usage.csv file in the data folder. See Figure 7: User analysis workflow on page 12. The tables in both outputs share the same format and have four columns:

* Date – during the course presentation on which a post was made
* Posts – the total number of live posts made that day, ie not counting draft and edits, only final versions
* Users – the total number of users who made those posts that day
* UserIDs – a list of each unique user’s UserID, hence if a user makes more than one post their UserID will appear in this list only once

Looking at the report/table one can quickly gauge the number of active users against posts. The csv table can be enhanced by opening the csv file in Excel, the default on OU supplied Windows machines; select the Date, Posts and Users columns; then Insert; then 2-D Line chart; to provide a quick visualisation of the data.

## Students’ progress across courses

With our focus on students, we also wanted to trace their progress through the courses on offer. Gaining an understanding of this relates directly to the possible impact of the teaching elements in one course on the student’s performance in subsequent courses. This understanding is provided by trace\_userids.py, which analyses the forum posts and from that data produces a report, study-paths.html, and a table, study-paths.csv, for review.

A possible extension of the framework is to relate this understanding to results and progress as reported by the eTMA system. Work on this was begun in this project, but without time to carry it through to a conclusion. Various problems were encountered on the source data, and as they were not resolved within the scope of this study for incorporation into the framework, are not expanded upon in this report. See Future work on page 21.

## UserIDs not students

While the intention of this project, and subsequent projects, is to study the effect of teaching interventions in the exported Moodle forum XML data we have only the anonymised Moodle UserID. This means it is not possible to distinguish between tutors, moderators and students. A consequence of this can be seen in the study path results.

The study path analysis, produced by trace\_userids.py, is meant to show the path of study across courses taken by students. On reviewing the study-path results when testing the script, it was immediately apparent that several IDs, such as 016185, are present in both M816 14K and M816 15K, which seemed curious. Looking at the appropriate forum post web pages, it quickly became obvious that this userid is that of tutor who worked on both presentations. So, the reports and table include staff, that is tutors and moderators, as well as students. Hence the use of the term userid rather than student in the output. However, most staff enrolled on Moodle some years ago and hence have userids of a much lower number to their students, though that his not a reliable way to distinguish them. A consequence of this confusion is that staff posts are subject to the same filtering and classifying as student posts. Therefore, for future use, it is recommended to compile a list of staff Moodle userids so that they and their associated posts are excluded from processing.

## Using UserIDs to trace students

The two study-paths outputs produced by trace\_userids.py simply report if a userid, be it that of a student or a tutor, has posted to one of that presentation’s forums. Therefore, there is the possibility that a student will not be recorded; the userids of active participants only are captured.

In the study to date, all students are active participants of the forums and so are traced, The courses require students to contribute to a forum as part of an eTMA. Clearly, when applying the framework in future, this may not apply to other courses.

# Named Entity Recognition

One approach explored and rejected in developing this framework was the use of Named Entity Recognition (NER) techniques to filter forum posts. We chose to look for person names in messages as a possible filter as a possible indicator that students were making useful cross-references. An *ad hoc* script was developed using [nltk](http://www.nltk.org/) for NER[[7]](#footnote-7) to write a report containing those forum posts that included a person’s name. A sample post entry from the report is shown below:

Post: 11776320

UserID: 16185

Message: <p>Hi Ben</p><p>The materials are provided in Ebooks and Kindle formats for those wishing to read on the move.&nbsp; Better still, is the use of the OUAnywhere app - (Apple/Android).</p>…

As can be seen in this sample, the script has correctly identified this post as having a person’s name in it: Ben. However, the name is not the desired reference to an external authority, but a greeting to a student (This post was a tutor’s reply to that student’s post.)

This sample is taken from M816 15K’s Module discussion forum, which has a total of 280 posts. Of those posts, 202 are identified as containing named persons and 78 without. Hence, the technique is not sufficiently discriminating to aid analysis of the forum posts.

## Future applications of NER

NER as a technique has been applied successfully to filter forum posts. However, in the context of the available forums more consideration is required before it can be used to discriminate forum posts meaningfully. Other forums, serving different pedagogical directions, may benefit from the application of NER, selecting other entities such as organizations.

# Outputs

There are two outputs associated with the framework: the framework itself and the forum data used in its development.

The output files are esteem-mqe-framework and esteem-mqe-forums.

Each file is available in two formats, .zip for Windows and .tar.gz for Mac/\*nix.

# Reports

The framework produces .html report files. All are stored in the reports folder. For each forum, signified by \* in the report titles listed below, there are the following reports:

\*\_filtered\_posts.html

This report helps you assess engagement beyond immediate course material in a forum by consolidating in one place all those posts in which their author has made a reference to an external resource.

This report presents a listing of all posts in the forum that meet the filtering criteria. See External links on page 5 *et seq* for more details on the criteria used. Each listing contains the id of the post, the userid of the post’s author, the post’s creation date and the post itself in its original html format.

Produced by filter\_posts.py.

\*\_significant\_words.html

This report helps you devise a new, or review an existing, taxonomy. Presenting a list of the words in a forum ranked by their significance, that is how often they are used in this forum compared to the other forums, this helps you evaluate a word’s value in the taxonomy.

This report presents a table with two columns: Word and Score. For each word in the forum its associated tf:idf score is shown, sorted by most significance. See Suggest taxonomy keywords on page 10 for more details on tf:idf and its role in the framework.

Produced by suggest\_keywords.py.

\*\_usage.html

This report helps you assess forum activity during the course. Looking at both the number of posts made each day and the number of active users each day, you can assess engagement with the discussion not only in terms of overall quantity but whether all students are active or whether it is just a few more vociferous contributors.

This report presents a table with four columns: Date, Posts, Users and UserIDs. For each date on which one or more posts were made the total number of posts made that day, the number of users who made those posts, and the Moodle userids of the posts’ authors are shown.

Produced by unique\_userids.py.

index\_to\_filtered\_posts.html

This report lets you assess in one summary screen the activity in all forums, and the amount of that activity suggesting engagement beyond immediate course material. A link to each forum is provided to let you drill down to the relevant individual forum posts.

This report acts an index page to the individual forum \*\_filtered\_posts.html reports. In addition to providing a link to each forum’s report, simple metrics are shown: the number of posts containing external references, the number of posts without external referecnes, these two numbers are also presented as percentages, and the total number of posts in the forum.

Produced by filter\_posts.py.

study-paths.html

This report lets you see the courses in which each forum user has participated to help you assess the experience they might bring to their continuing learning.

This report presents a table with two columns: UserID and Courses. For each userid, the courses they have taken are listed.

This is not a forum specific file; it covers all users in all forums in the source forums folder.

Produced by trace\_userids.py.

# Data

The framework produces .csv data files. All are stored in the data folder. For each forum, signified by \* in the data filenames listed below, there are the following data files:

\*\_classified\_counts.csv

This data file helps you assess the content of posts made daily during a course, using criteria applied by the classifier.

This table contains thirteen columns: Date, Remember Posts, Understand Posts, Apply Posts, Analyze Posts, Evaluate Posts, Create Posts, Remember Use, Understand Use, Apply Use, Analyze Use, Evaluate Use, Create Use. For each day, the classification counts are listed. Posts refers to the number of posts made that day that include at least one term in the category. Use refers to the number of terms in posts made that day in the category.

Currently, the framework implements classification based on Bloom’ Cognitive Taxonomy. For more details, see Figure 3: Bloom's Cognitive Taxonomy on page 6. For each of the six levels of Bloom’s Cognitive Taxonomy two metrics are produced: posts, the number of posts that have an appropriate classification-level keyword; use, the number of uses of a term of that classification-level.

Should another classification be used then similarly there will be two metrics, posts and use, for each level defined by that classification.

Produced by track\_classified\_posts.py.

\*\_classified\_posts.csv

This data file lets you relate all posts and their authors, and on which day the post was made.

This table has nine columns: PostID, UserID, Date, Remember, Understand, Apply, Analyze, Evaluate and Create. For each post in the forum, the userid of the post’s author and the post’s creation date are listed, along with the count of terms used in the post in each of the six categories defined by the default classification scheme based on Bloom’s taxonomy.

Should another classification be used then similarly there one column for each level defined by that classification.

Produced by track\_classified\_posts.py.

Used by unique\_userids.py.

\*\_daily\_counts.csv

This data file lets you track the total number of posts made each day during a course.

This table contains two columns: Date and Count. For each day, the total number of posts made is listed.

Produced by track\_classified\_posts.py.

Used by daily\_counts.py.

\*\_filtered\_counts.csv

This data file lets you track the number of posts that contain an external reference made daily during a course.

This table has two columns: Date and Count. For each day, the total number of posts made that meet the filtering criteria is listed.

Produced by track\_filtered\_posts.py.

Used by daily\_counts.py.

\*\_filtered\_posts.csv

This data file lets you relate posts that contain an external reference and their authors, and on which day the post was made.

This table has three columns: PostID, UserID and Date. For each post that meets the filtering criteria, the userid of the post’s author and the post’s creation date are listed.

Produced by track\_filtered\_posts.py.

Used by trace\_userids.py.

\*\_usage.csv

This data file helps you assess forum activity during the course. Looking at both the number of posts made each day and the number of active users each day, you can assess engagement with the discussion not only in terms of overall quantity but whether all students are active or whether it is just a few more vociferous contributors.

This table has four columns: Date, Posts, Users and UserIDs. For each date on which one or more posts were made the total number of posts made that day, the number of users who made those posts, and the Moodle userids of the posts’ authors are shown.

Produced by unique\_userids.py.

\*\_word\_counts.csv

This data file helps you devise a new, or review an existing, taxonomy. Presenting a list of the words in a forum ranked by their significance, that is how often they are used in this forum compared to the other forums, this helps you evaluate a word’s value in the taxonomy.

This table contains two columns: Word and Count. For each word in the forum, the count of the number of times it is used is listed, sorted in descending number order.

In addition to the forum specific files, there another file of the same layout that aggregates the whole word count for the source forum folder.

Produced by suggest\_keywords.py.

study-paths.csv

This data file lets you see the courses in which each forum user has participated to help you assess the experience they might bring to their continuing learning.

This table with eight columns: UserID, M811 13K, M813 14E, M811 14K, M816 14K, M813 15E, M811 15K and M816 15K. For each userid, the courses that user has taken are shown in a matrix of all course presentations in the source forum folder. Currently, if a course is taken the cell is populated with the course code; if not taken, the cell is set to 0. These cell values can easily be changed in the generate\_matrix() function to something more appropriate if necessary.

Note, the column headings are dynamically generated by the script and list all forums which are found in the source forums folder.

This is not a forum specific file; it covers all users in all forums in the source forums folder.

Produced by trace\_userids.py.

# Future work

The primary future work is to apply this framework in a longitudinal study.

A significant enhancement would be to relate the results of the framework analysis to data in the eTMA systems, particularly student marks and consequent grades. This would provide a valuable, objective metric to assess the effectiveness of a teaching intervention.

|  |  |
| --- | --- |
| Combination | Count |
| M811 | 2 |
| M813 | 2 |
| M816 | 1 |
| M811 + M813 | 6 |
| M811 + M816 | 3 |
| M813 + M816 | 2 |
| M811 + M813 + M816 | 1 |

By way of an example of an aid to understanding the data we could produce, Excel scripts and formula were written to automatically produce the summary table shown to the right from the *ad hoc* eTMA data available within part of this study. This table shows a simple count of students against the variations of study pathways they took. To manually produce this information from the data would be very time consuming and error prone.

The framework can be extended in other ways, most noticeably through the development of one or more tailored classification schemas to assist in the qualitative analysis of student posts. This is facilitated by the suggest\_keywords component of the framework as described in Suggest taxonomy keywords on page 10.

Of particular relevance to the current project, is the development of a new taxonomic schema based on search terms derived from authentic learning intervention. In this case, the taxonomy is not derived from existing posts but from the intentions of the intervention, and the expected language that students will use. Once used in a course, the taxonomy can be assessed and refined by using the suggest\_keywords component to review those words that are significant (see reports/\*\_significant\_word.html), and to review the frequency of all words in all posts (see data/\*\_word\_counts.csv).

Having extracted the forums and made them accessible in a machine readable format, their posts are available for other analyses. A beneficial development would be the application of NLP techniques to distinguish simple 'I agree' style posts from posts that show contribution or engagement with the course material and discussion.

We could use similar techniques from NLP to highlight social posts. This can serve both to exclude a post from further analysis if appropriate, or if a course has an ice breaker activity in the forum (typically an introduce yourself and comment activity) then assess whether that is what the students have done or have they conflated the activity with more technical aspects of the course.

With the access to forum posts available through the framework, we can also analyse posts by time. This allows us to consider those students who post to a forum in the period between when the forum opens and the actual start of the course. In combination, with an analysis of whether this is a social post, or one more related to course content, we could explore the hypothesis that there a positive correlation between early engagement and final grade.

Similarly, we could identify those students who post at the end of a course. Again, it might be beneficial to discriminate between social posts and those that make a more considered contribution on the future, with suggestions on applying the course material for example. In either case, the intention is to see if there is a correlation between these late contributors and their final grades.

A further extension of this development of the framework could be to relate post data to retention; relating leavers to their contribution to forums, whether by quantity and/or quality of their posts.

The framework not only makes the forums more accessible, but extracts post and userid data so that they are available for other statistical analysis. One example, is to track not only the percentage change in the nature of forum posts during a course, (for example changing levels of cognitive thinking in our current study) but to combine that with an analysis of any changes in the quantity of posts made to provide a complete picture of any changes in student posts.

As noted in UserIDs not students on page 13, the framework processes all forum posts whether they are made by students or staff. For more accurate results of any intervention, we need exclude staff posts from the analysis. A list of staff VLE userids should be compiled and used in a newly developed filter that removes staff posts from the analysis algorithms.

For publications based on applying this framework, a more formal assessment of the framework is required. For example, we have used a formatted year date as a good enough proxy for a formal literature citation as experience suggests this is the case. In an academic publication, however, we should justify this design assumption. The expected metrics are the precision, recall and accuracy of our results, for which we will need to manually review all posts to provide a comparison against the posts identified by the framework’s algorithms.

There are technical aspects of the framework itself that can be enhanced. See scripts/TODO.md for more details.

1. All diagrams in this report follow the same scheme whereby a turquoise rhomboid signifies a manual process, a squared green rectangle signifies a folder, a rounded dark red rectangle signifies a Python script and a dark blue elongated hexagon signifies an external process. [↑](#footnote-ref-1)
2. Villarroel, L., Bavota, G., Russo, B., Oliveto, R., and Di Penta, M. (2016) *Release Planning of Mobile Apps Based on User Reviews*, ICSE ’16, May 14-22, 2016, Austin, TX, USA. [↑](#footnote-ref-2)
3. Tokenizing is breaking words out from the text into individual entities. Stemming is reducing the words to their roots. The latter is an useful technique in analysing texts. For example, in the classification the keyword ‘discriminates’ is stemmed to ‘discrim’. When we stem the forum post, that keyword will match ‘discriminates’ as well as ‘discriminating’, ‘discriminated’, and so on. Hence, the technique allows us to match against verb conjugations, noun declensions, etc. [↑](#footnote-ref-3)
4. Krathwohl, David R. (2002) *A Revision of Bloom's Taxonomy: An Overview*, Theory Into Practice, **41**:4, 212-218, DOI: 10.1207/s15430421tip4104\_2 [↑](#footnote-ref-4)
5. As all scripts in the framework are written in Python, then pickle is the most efficient method of storing the classifications for re-use. Pickling is a Python specific storage method. Other tools, written in other programming languages, can read the matching json file, so the classification is usable by these other tools too. [↑](#footnote-ref-5)
6. Structure of the Observed Learning Outcome (SOLO) is based on the study of the outcomes of academic teaching. The taxonomy names and distinguishes five different levels according to the cognitive processes required to obtain them. See: Biggs, J. B. & Collis, K. F. (1982). *Evaluating the Quality of Learning: The SOLO Taxonomy, Structure of the Observed Learning Outcome*. (London: Academic Press) [↑](#footnote-ref-6)
7. For more details about nltk and its other NER options, such as organization and location, see http://www.nltk.org/book/ch07.html#named\_entity\_recognition\_index\_term. [↑](#footnote-ref-7)