<https://colab.research.google.com/drive/1xtpKgzoyB1VcwruLjW-7B-hlNYVE2RL2#scrollTo=Ec4GrBJ2FMls>

22 july

* Next steps:
* Coursename SIS and coursename warehouse--- use warehosue if it exists, otherwise use SIS.
* Create combined name field which uses warehouse if it exists, if not-- SIS.
* Vectorize combined course name into n grams of 1,2,3…
* For each n-gram length we will have count of distinct course\_ids and a list of those courses.
* Then we get matrix of course name as column names and the counts as values (like a crosstab matrix).
* THEN lets see descriptions.
* remove stopwords
* Vectorize descriptions---
* Then n grams of 1,2,3 and do the same as above.
* Can append table and have a col for gram length.
* Every course can be assigned one of the topics
* How broad should topics be?
* Maybe courses can have multiple tags.

August 4, 2025

* Don’t do lda yet—
* Coursenames:
  + Remove stopwords from course\_names
  + split into unigrams, bigrams and trigrams-- and count distinct course\_ids that have those unigrams, bigrams and trigrams.
  + Go through list of unigram, bigram and trigram. -- add list to googledoc and mark as use or not use.----
  + For LDA part:
    - One row per student for all course they have taken and count of the unigram,bigrams,trigrams from all the courses they have taken.---
* Descriptions-- run alongside the coursenames.
  + Stopwords-- remove from descriptions.

Observations in process:

* Manual Tokenizing::: may be ignore as tfidf may be better!
  + Output: course names uni-bi-trigrams.csv, unique\_subset.csv🡪 added this to Gsheet and manually cleaned and was manually coding to keep or not when tf-idf idea struck:
* Tfidf:
  + While doing tfidf, I noticed some coursenames are not clean—they have numbers (e.g., ‘From Feast Famine & Back Again: Credit Markets & LBOs 2003 2013’, ‘

|  |
| --- |
| * Food Entrepreneurship Half Term 1.5 Credits |
| * Food Entrepreneurship Full-Term 3 Credits |
| * Consulting 101: Driving Strategic Impact |

* + May need to clean for this?... or if we are going the tfidf route—can manually remove these while cleaning out lower scoring tf-idf words.
  + Output matrix: "tfidf\_unigrams\_bigrams\_trigrams.csv”

Note:

* **TF-IDF Score:** The product of TF and IDF. A higher TF-IDF score indicates that a unigram, bigram, or trigram is both frequent within a document and relatively rare across the entire collection of documents, suggesting its importance in distinguishing that document.

Interpretation:

**[Higher TF-IDF scores for unigrams, bigrams, or trigrams](https://www.google.com/search?sca_esv=ea267502e9e23089&rlz=1C1GCEA_enUS1166US1166&cs=1&q=Higher+TF-IDF+scores+for+unigrams%2C+bigrams%2C+or+trigrams&sa=X&ved=2ahUKEwjJ7fehlfKOAxU-FlkFHaJwJ6gQxccNegQILxAD&mstk=AUtExfD8-Z1DHAtAMgm8CZhAUtz2_Hit74CmDUI5XV67vTzCEWRXszxZ-kqfkkID-c-9VMiQ8HbsegZ_i-uc_-SQR3aZrdYy6cy5xuD1v00vpcXy9CWqq3xaUgNpyxEmWmb-Q02qasCs9710lEseJI99B06KExADBKwdbFd7UX4wUVhLWbqnkxRbplbUosK-L8eJ_ijs&csui=3" \t "_blank)**

[.](https://www.google.com/search?sca_esv=ea267502e9e23089&rlz=1C1GCEA_enUS1166US1166&cs=1&q=Higher+TF-IDF+scores+for+unigrams%2C+bigrams%2C+or+trigrams&sa=X&ved=2ahUKEwjJ7fehlfKOAxU-FlkFHaJwJ6gQxccNegQILxAD&mstk=AUtExfD8-Z1DHAtAMgm8CZhAUtz2_Hit74CmDUI5XV67vTzCEWRXszxZ-kqfkkID-c-9VMiQ8HbsegZ_i-uc_-SQR3aZrdYy6cy5xuD1v00vpcXy9CWqq3xaUgNpyxEmWmb-Q02qasCs9710lEseJI99B06KExADBKwdbFd7UX4wUVhLWbqnkxRbplbUosK-L8eJ_ijs&csui=3" \t "_blank)in a document suggest that these specific terms or phrases are particularly relevant and characteristic of that document.

Maybe idf makes more sense---as in tf-idf if a word doesn’t occur commonly, it will have a low score—but we actually want these uncommon words as they can give extra context into what the course is!!.

5 August 2025:

I used idf score (along with document frequency numbers) and manually marked if terms should be kept. (I also ran terms through chatgpt initially for it to mark it based on idf score and terms for an LDA—I THEN manually checked everything—and re-marked). Based on observations most of the terms that shouldbe kept are bigrams, and unigrams are mostly to be tossed. There are some informative trigrams, but it is nothing that the bigrams will miss.

If going with bigrams, remove: the, in, to, for, and….but before or after? And do we really need to if it is going through the full list of bigrams anyway?

Or should we just go with all 3, and remove the ones that were manually marked 0?

Did a pivot in GSheet: idf\_scores\_ngrams: manual+cg.xlsx – to see which ngrams to retain.

11 August

Kiersten wanted each term to have a count of unique course IDs that contained that term.

The df counts: I think this is counting each ngram list as a document, and NOT the clean course name.

<https://colab.research.google.com/drive/1LPlEk7YN_iTTnMW6GzeevrKcJylsdiH_#scrollTo=x_PfMdxi8rrl&line=1&uniqifier=1>

I added this step under the IDF vec calculation on colab and did this by taking the ~~GSheet that already had clean\_names linked to uni,bi,and trigrams and~~ I merged them into a list. Problem with this was it didn’t use Count vectorizer like I did for the idf part—so I re-made a it using vectorizer but got a similar output with clean names and list of unigram, bigram and trigram instead of reading in Gsheet. Now all names should be matched and no count should be 0. <https://colab.research.google.com/drive/1LPlEk7YN_iTTnMW6GzeevrKcJylsdiH_#scrollTo=7scNtKTxN77R&line=9&uniqifier=1> wrote to a Gsheet just for future use: ‘CV\_unique uni-bi-trigram\_subset.csv’

I then counted the number of time each term from the vectorization (idf\_Df) occurred in the ngrams list for EACH coursename (note that the ~~gSheet~~ subset containing this is already a unique list of coursenames) element created from the ~~GSheet~~ subset, and merged them on term.

So this would give the unique course name which has exact match of each term.   
<https://colab.research.google.com/drive/1LPlEk7YN_iTTnMW6GzeevrKcJylsdiH_#scrollTo=2LOOsZ4IFUZJ&line=6&uniqifier=1>

I then wrote it to “[idf\_ngram\_COUNT](https://docs.google.com/spreadsheets/d/1Q7VBjcisrghEop0AwEmeefnGbFoPd0mHJJJcvTyRvAo/edit?gid=851516708#gid=851516708).csv” (converted to GSheet) in the ‘Student cluster Analysis GDrive’ folder on GDrive.

I copied ~~the count column~~ all columns and pasted it to the GSheet where I had marked keep for ngrams already (So I didn’t have to re-do it), then used xlookup to match keep cols—just to ensure something doesn’t go amiss by just pasting: [idf\_scores\_ngrams keep: manual+cg](https://docs.google.com/spreadsheets/d/17HYOnIe59SkAQbNaAQizI1C9Ks2CjyFRkIOE5VCA1BY/edit?gid=766887787#gid=766887787) GSheet! (‘idf\_scores\_ngrams’, ‘manual+cg\_coded’ subsheets)

This would have automatically updated the pivot, which would have updated my plots on the GSlide. Just need to change the plots which have the desc and asc ordered on count on the side.

There were still blanks for some counts—just added 1 as they seemed way too unique

Playing around with key word extractors (RAKE and YAKE) to see if process can be made quicker—RAKE and YAKE are good for short text, so I thought this might be a good idea for course names. This will give us an idea about how many clusters they might be?

Questions: So, should we use keywords for LDA, atleast for the coursenames LDA as RAKE and YAKE is good for short text. Ngrams can be used for description.

Then For coursename and descriptions, we can do the LDA itself.