Authors:

Magd Bayoumi (MyCourseIndex) <mb2363@cornell.edu>
Jenna Kressin (MyCourseIndex) <jek343@cornell.edu>
Souleiman Benhida (MyCourseIndex) <sb2342@cornell.edu>
Sheetal Athrey (MyCourseIndex) <spa42@cornell.edu>
Oscar So (MyCourseIndex) <ons4@cornell.edu>

GitHub Link: https://github.com/oscarso2000/Piazza
Piazza Link: https://piazza.com/class/k5h3t0fy1gm5vv?cid=418 (@418)
URL: https://www.mycourseindex.com/
Spreadsheet:
https://docs.google.com/spreadsheets/d/10Nn4V-toUWH4aQ1GjhL7ttG7Cf-Hk7eqWCykyOIFyII/edit?usp=sharing
Color Coding: -Deletions -Insertions -Reason for modifications

P01 -- Updated P02

High level goal/Use case:

This tool aims to help students relevant course materials to their coursework questions.
Use case: User writes query i.e. “Fast cosine similarity on Kardashians transcripts”
=> Results: Piazza @26 @112 (for example)
=> Link to worksheet from website on this
Inputs are free text and outputs are links/blobs of text

Data Sources:
The primary data source we would be using are existing Piazza questions and answers as a corpus to the query form, this will help us with our aim to improve the way Piazza search queries work. Other data sources that we plan to use are course notes, powerpoint presentations, assignments as well as if available the course video lectures (and transcripts), and the course textbook (if free). Another source we plan to possibly use is the User’s own notes that they would upload. These additional sources will be what we would use to build the tool that would use keyword queries and give relevant excerpts from piazza answers as well as the course site.

Input:
The user would input a single free form keyword query that could be a concept, course material question, or technique. In addition, the user would specify the course that we should search within.

Output:
A user would receive an output in the form of a list of one or multiple relevant Piazza questions and answers, or an excerpt or page from the respective, course site.

Social Component:

We are using a question-answer forum in Piazza as our human generated data. This offers a rich environment with questions and answers that we plan to use in our information retrieval component as a way to answer their information needs. Piazza also offers information about whether a post has been answered and is considered a “good” post. To further encourage social engagement, we can prompt users to ask their question on Piazza if no good matches are found in order to enhance the corpus. Additionally, we can gather pseudo relevance feedback by noting which Piazza posts and resources are clicked on in order to improve results.

Information Retrieval Component:
Hopefully, we will have access to a bot inside a Piazza course which will allow us to web scrape and gain information on any posts. Thus, we will then be able to preprocess the data and compare user searches with TF-IDF, Cosine Similarity, etc. Thus, this will allow us to get more relevant and similar piazza posts to answer any queries that the user has. In addition, we would have other course documents (worksheets, slides, etc) in PDF form that would be loaded into our database and processed similar to the piazza posts. Previously, we had this as a data source but did not leverage it in our IR component. Since course documents also have answers that may be relevant to the user and not just Piazza, we believe this source will add more value to the user.

Machine Learning Component:
~~Another critical feature for our users is finding related concepts in their queries. We plan to train a BERT model to tokenize keywords or concepts and map them to a higher order concept (similar to an NER model) on a per class basis~~.
We plan on using a BERT model (or something like word2vec), as a method of encoding our strings into vectors. We lean towards BERT as it is a contextual model. This would hopefully give us a representation that more closely matches the word and its relation to other words. We would use this vector in our search methods by using it as our representative vector. We would also have to use this to process our documents. We believe using a model that has been pre trained then fine tuning on a smaller dataset focused on education will perform well. We made this transition due to reading a paper (linked in the references), about concept mapping for medical concepts. This led us to consider using this unsupervised approach for concept mapping and using some form of a model to give our vectors more meaning since the indices are arbitrary. We will further discuss the opportunities between our website and ML in future milestones.

Other Critical Components:
Security: We have added security protocols to the website, which includes Microsoft Active Directory Federation Services. Currently this is in the initial phase and there are only two scopes however, we are expanding the scopes as we reach out to professors for access to data. We added this feature to alleviate a lot of the security, copyright, and privacy concerns that had been brought up by the course staff.

Front End: We will also be using React.js with CSS for the UI of the website.

P02

Part 1: Statistics + Exploration of Dataset:

For now, we scraped the 100 most recent piazza posts from CS 4300 and stored them in one json file to use as our sample data. This includes the full history of updates to the question, student and instructor answers, feedback, relevant folders, and post number used in the url among other information. A sample of this json file, only post number 505, can be seen in the appendix section because it is extremely long.

Given that this is just a first pass, we only used the “most important information” when we loaded the json file into a dictionary with the structure detailed below:

```
{nr: {“folders”: [],
      “question”: {“word”: frequency},
      “s_answer”: {“word”: frequency},
      “i_answer”: {“word”: frequency}}}
```

We used the nr as the key since it is a unique integer identifier, the post number, used in the url and in the upper left corner of a post. The value in the outermost dictionary is another dictionary that contains keys that refer to the post’s folders and most updated question, student answer (s_answer), and instructor answer (i_answer). The value of the key “folders” is simply a list of strings of the folders that a user filed this post under, for example, “hw1” and “logistics”. Here is an example of part of the dataset, after it is put into this data structure:

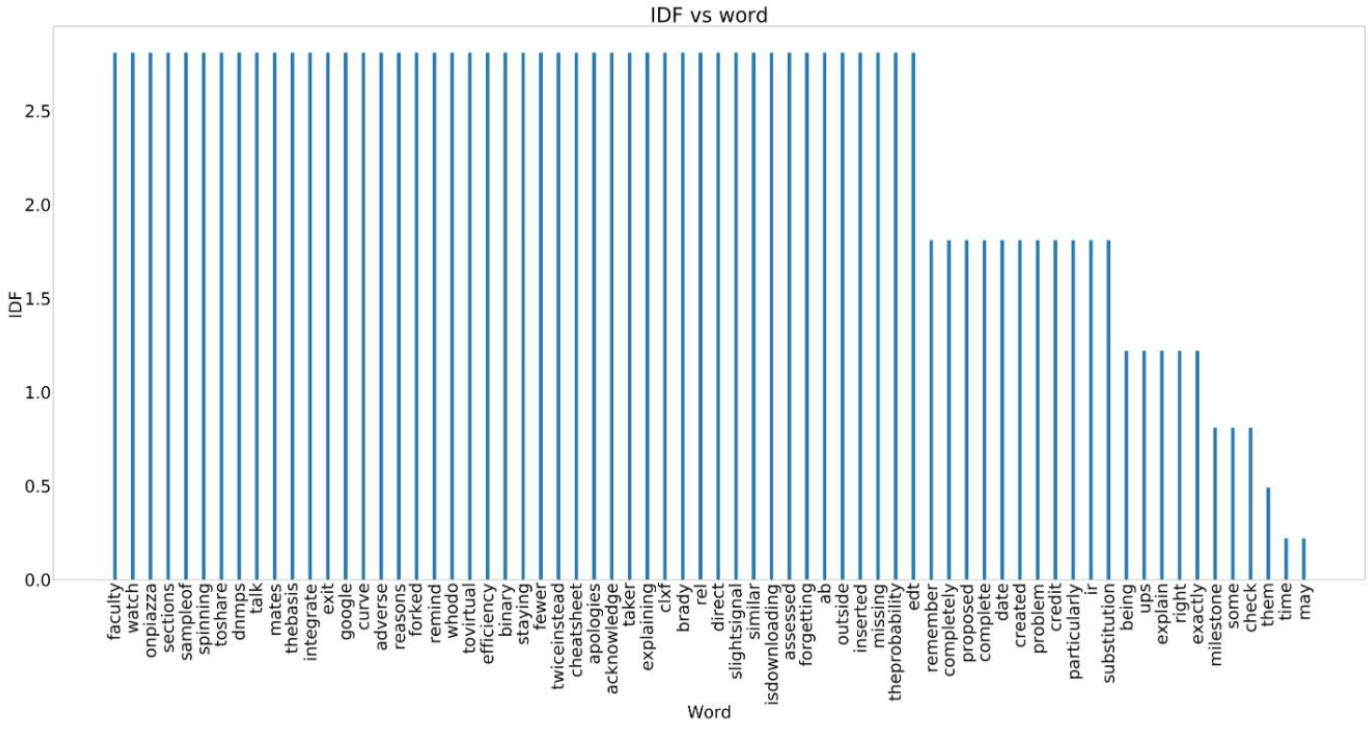
```
505: {'folders': ['exam'],
  'i_answer': '',
  'question': 'What are words associated with the update Rocchio query?',
  's_answer': 'bank bank financial financial financial'}
```

Within the “question”, “s_answer”, and “i_answer” fields, we changed the sentence into a word frequency dictionary. This lists the word as the key and the number of occurrences of that word within the respective key as the value. In order to obtain this word frequency dictionary, we first defined what we considered to be a “word” as alphabetical letters using regular expressions for ease of implementation. We also removed web addresses, punctuation, and any html characters such as new lines and new paragraphs. We were left with only what is colloquially referred to as a word.

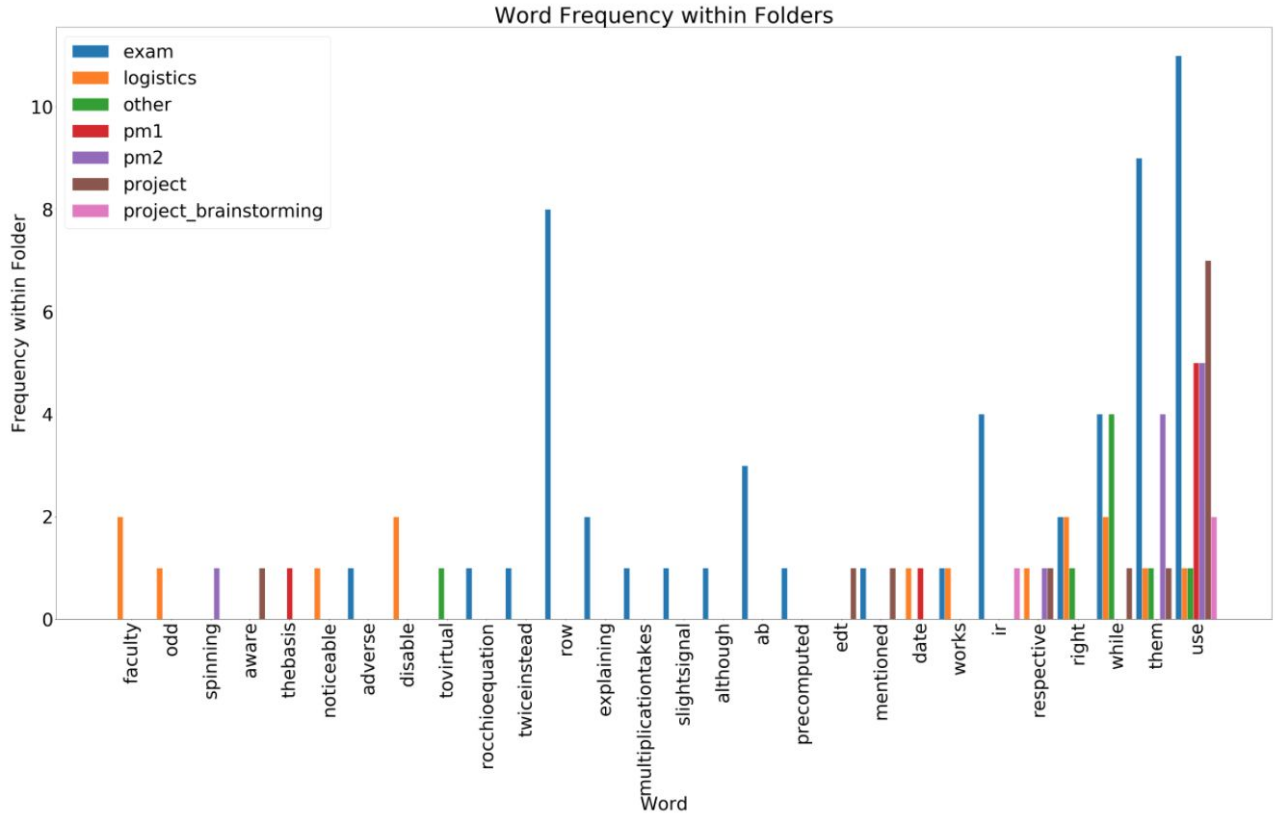
```
505: {'folders': ['exam'],
 'i_answer': Counter(),
 'question': Counter({'are': 1,
          'associated': 1,
```

```
            'query': 1,
            'rocchio': 1,
            'the': 1,
            'update': 1,
            'what': 1,
            'with': 1,
            'words': 1}),
 's_answer': Counter({'bank': 2, 'financial': 3})}
```

Using this data structure, we began making the charts seen below. The purpose of the first chart is to understand which words add real value to the post and can help during a search, and which are just filler words or not used enough to be relevant. To begin, we created an inverted index where the words are the keys and the value is a list of tuples that contain two values, the folder (a string), and the number of occurrences of the respective word in the respective folder. This is the same format as the inverted index used in assignment 4 earlier in the semester. After obtaining the inverted index, we calculated the IDF of each word and stored these in a dictionary with the word as the key and the respective IDF as the value. The process of calculating the IDF was nearly the same, with one exception - we did not need to add 1 as a smoothing factor in the denominator because we knew from the way we obtained the data that there would never be a word that we have in the dictionary with 0 occurrences. For these reasons, the code used is very similar to the code that we wrote in assignment 4. From the IDFs, we created a skewed histogram to show the tapering off of the IDF words.



In this graph, we chose to display the 68 words (by sorting in the decreasing order of IDF and systematically sampling every 40th word). For the second chart, we took 10 words (by sorting in the decreasing order of IDF and systematically sampling every 275th word) and produced a bar graph with each of the frequencies of words stratified by the folder label. The goal is to understand which words are most relevant to which folders, rather than just which words are most relevant overall.



As of now, we are not planning on using **supervised** ML - we will be using unsupervised.

In the future, we will use all the piazza posts for CS 4300, and hopefully piazza posts and course documents for other classes, which we will ask professors for their permission for. We hope to use posts, and possibly documents, for about 4 classes. Assuming that these are large classes with an average of 1000 posts each at the end of the semester, we hope to have about 4000 posts to use. Please note that this number is a very rough estimate, especially since we rely on frequency piazza usage for the course. Since we are all remote, we are hoping that there will be a large amount of posts that we can use as data. Piazza posts will eventually be scraped on a relatively frequent basis and course documents will be scraped on a weekly basis since professors usually assign homework with the previous week's content.

We are also planning on incorporating more information from the existing json files, such as weighting the results by the amount of “thanks” an answer received. We would return results with more “thanks” as more relevant to those that are of similar content but less “thanks” were received.

Part 2: Sketched Examples and Descriptions:

Our system will allow users to enter free-text queries and find matching piazza posts. We will download the Piazza corpus for several classes (including posts and comments) and convert them to tf-idf vectors, and continuously update this internal database of vectors. We will maintain an inverted index of term-document listings to enable fast lookup. We will take user queries, compute their word vector, and then find the closest matching documents using cosine similarity. We will also take into account votes, instructor endorsements, and response length to influence the ranking of relevant documents. We will take into account both question text, response, follow up comments as parts of the same question document.

Example 1:

Query Input: “Benefits of cumulative gain metric”

System output (in order):

Piazza #450

Excerpt: Cumulative gain and discounted cumulative gain are both measures of how "good" an information retrieval system is. Discounted cumulative gain takes into account the order in which documents are returned. Intuitively if two systems return 10 documents, 2 of which are relevant, but the first system returns the two relevant documents first and the other returns them last, the first system is better. If we looked at just the cumulative gain, both systems would appear to perform equally well, but discounted cumulative gain would tell us that the first system performed better.

Stanford IR Book: Evaluation of ranked retrieval results

Excerpt: A final approach that has seen increasing adoption, especially when employed with machine learning approaches to ranking svm-ranking is measures of cumulative gain , and in particular normalized discounted cumulative gain (NDCG). NDCG is designed for situations of non-binary notions of relevance (cf. Section 8.5.1).

Piazza #480

Excerpt: Retrieved Results and Rankings

1.A

2.B

3.C

4.D

relevant: C,D

Simple Cumulative Gain: 0 + 0 + 1 +1 (two 0s at the beginning since AB are irrelevant)

Discounted = 0 + 0 + 1/3 + 1/4 (C takes position 3 and D takes position 4)

Piazza #412

Except: From what I understand from cumulative gain in my notes, it seems to be the sum of the scores of each document. The scoring for each document may be different, but the example in the notes scores it by assigning 1 for relevant documents and 0 for not relevant ones, so if there were 10 documents and 5 were relevant then the cumulative gain is 5.

Analysis: Our system weighs the first one higher as it has more content, has instructor answers, and more “thanks!” response.

Example 2:

Query Input: “What is high precision with low recall good for”
System output (in order):
Piazza #532

Excerpt: “Typically, when precision-recall curves have higher levels of precision at lower levels of recall, that means the system prefers precision over recall (such as Google since they want the first few results to be relevant because most users just click on the first link). When a precision-recall curve has higher levels of precision at higher levels of recall, that means that the system prefers recall over precision (such as a system a researcher would want to use to get as many documents as possible that might be related to their topic of interest) “.

Stanford IR Book: Evaluation of ranked retrieval results

Excerpt: An alternative, which alleviates this problem, is R-precision . It requires having a set of known relevant documents Rel, from which we calculate the precision of the top Rel documents returned. (The set Rel may be incomplete, such as when Rel is formed by creating relevance judgments for the pooled top k results of particular systems in a set of experiments.) R-precision adjusts for the size of the set of relevant documents: A perfect system could score 1 on this metric for each query, whereas, even a perfect system could only achieve a precision at 20 of 0.4 if there were only 8 documents in the collection relevant to an information need. Averaging this measure across queries thus makes more sense.

Piazza #523

Excerpt: Yes, the break-even point is just when precision is equal to recall. R-precision measures how many of the top R documents are relevant.

Piazza #555

Excerpt: The F1 score is a measure of weighted harmonic mean of precision and recall. It’s a way to tell a balance of precision and recall. When the F1 score reaches the best value(=1), it means perfect precision and recall. When it is 0, it means lowest precision and lowest recall.

Analysis: The top response is the best as it directly addresses the user's query, and is ranked highly because it features keywords such as 'higher' and 'lower' that match the query. The other results are related to the topic of precision and recall but are looser fits and are ranked lower.

Part 3: Updated P01 is above.

Comparison to other systems:

Some of the uses of our app are covered by existing systems. Piazza has a search function, users can search public course websites with Google, and users can search PDFs using built in tools. With regards to Piazza and PDF search tools, these are very primitive engines that generally are only good at exact matches - things like misspellings or indirect phrasing can prevent users from finding matches. Google is more effective on these fronts, but is not specially fine-tuned to course content like our search engine. Our search engine will have custom databases for individual courses and use techniques such as TF-IDF and Levenshtein distance to find the best matches for course content despite incorrect spelling or indirect wording.

Part 4

Experiences:

Oscar: Data Analytics, Data Preprocessing, Machine Learning, Authentication

Magd: Python, Flask, Docker, React, Machine Learning, Deployment, Authentication

Jenna: Data Analytics, Data Preprocessing, Machine Learning

Sheetal: Python, React/HTML/JS/CSS, Machine learning, idk...

Souleiman: HTML/JS/React, Flask, Python

Tentative division of labor:

Oscar: UI + WebDev + Deployment + Security

Magd: Security + Database + PDF viewer + API

Jenna: IR algorithms + Data Preprocessing

Sheetal: IR algorithms + Data Preprocessing

Souleiman: Web Scraping + IR algorithms + Database

Schedule: (Including Weekly Wednesday 6-8pm Meetings)

Apr 6: Data Collection + Data Preprocessing + Data Analysis + Deployment + WebDev

Apr 13: Generate IR algorithms + Database + WebDev + Finish Piazza related queries

April 20: Testing + First live prototype, a brief presentation and demo in class with a few live examples + add on PDF viewers + analysis
April 27: Front-end Development + add on class notes and other features
May 4: Final app goes live

References:

1. http://medir2016.imag.fr/data/MEDIR_2016_paper_16.pdf

Appendix:

A.


```
{
   "folders": [
       "exam"
   ],
   "nr": 505,
   "data": {
       "embed_links": []
   },
   "created": "2020-03-09T02:57:45Z",
   "bucket_order": 3,
   "no_answer_followup": 0,
   "change_log": [
       {
           "anon": "stud",
           "data": "k7jvl1flyrr4v7",
           "type": "create",
           "when": "2020-03-09T02:57:45Z",
           "uid_a": "a_0"
       },
       {
           "anon": "stud",
           "data": "k7jvmu7c9gb242",
           "to": "k7jvl1fjm6v4v6",
           "type": "s_answer",
           "when": "2020-03-09T02:59:09Z",
           "uid_a": "a_1"
       },
       {
           "anon": "no",
           "uid": "ismc78opglp2z4",
           "to": "k7jvl1fjm6v4v6",
           "type": "followup",
```

```
            "when": "2020-03-09T03:54:24Z"
        },
        {
            "anon": "no",
            "uid": "ismc78opglp2z4",
            "to": "k7jvl1fjm6v4v6",
            "type": "feedback",
            "when": "2020-03-09T03:56:38Z"
        },
        {
            "anon": "no",
            "uid": "is67tr8ajy022u",
            "to": "k7jvl1fjm6v4v6",
            "type": "feedback",
            "when": "2020-03-09T04:09:16Z"
        },
        {
            "anon": "stud",
            "to": "k7jvl1fjm6v4v6",
            "uid_a": "a_2",
            "type": "feedback",
            "when": "2020-03-10T16:28:21Z"
        }
    ],
    "bucket_name": "Today",
    "history": [
        {
            "anon": "stud",
            "uid_a": "a_0",
            "subject": "Midterm Question 2: What is a possible string for qa?",
            "created": "2020-03-09T02:57:45Z",
            "content": "What are words associated with the update Rocchio query?"
        }
    ],
    "type": "question",
    "tags": [
```

```
        "exam",
        "student"
    ],
    "tag_good": [],
    "unique_views": 108,
    "children": [
        {
            "folders": [],
            "data": {
                "embed_links": []
            },
            "children": [],
            "created": "2020-03-09T02:59:09Z",
            "bucket_order": 3,
            "tag_endorse": [
                {
                    "role": "ta",
                    "name": "Caroline Chang",
                    "endorser": {},
                    "admin": true,
                    "photo": "1504735424_35.png",
                    "id": "is67tr8ajy022u",
                    "photo_url":
"https://d1b10bmlvqabco.cloudfront.net/photos/is67tr8ajy022u/1504735424_35.png",
                    "published": true,
                    "us": false,
                    "facebook_id": null
                },
                {
                    "role": "student",
                    "name": "Andre Lee",
                    "endorser": {},
                    "admin": false,
                    "photo": null,
                    "id": "is67u38ddyn29b",
                    "photo_url": null,
```

```
                "published": true,
                "us": false,
                "facebook_id": null
            }
        ],
        "bucket_name": "Today",
        "id": "k7jvmu79hri240",
        "history": [
            {
                "anon": "stud",
                "uid_a": "a_1",
                "subject": "",
                "created": "2020-03-09T02:59:09Z",
                "content": "<p>bank bank financial financial financial</p>"
            }
        ],
        "type": "s_answer",
        "tag_endorse_arr": [
            "is67tr8ajy022u",
            "is67u38ddyn29b"
        ],
        "config": {},
        "is_tag_endorse": false
    },
    {
        "anon": "no",
        "folders": [],
        "data": {
            "embed_links": null
        },
        "no_upvotes": 0,
        "subject": "<p>Could it also be &#34;bank financial financial&#34; and
would the vector point towards the midpoint between the two &#34;X&#34;s?</p>",
        "created": "2020-03-09T03:54:24Z",
        "bucket_order": 5,
        "bucket_name": "This week",
```

```
            "type": "followup",
            "tag_good": [],
            "uid": "ismc78opglp2z4",
            "children": [
                {
                    "anon": "no",
                    "folders": [],
                    "data": {
                        "embed_links": null
                    },
                    "subject": "<p>If not, could you explain what/how it should
be?</p>",
                    "created": "2020-03-09T03:56:38Z",
                    "bucket_order": 5,
                    "bucket_name": "This week",
                    "type": "feedback",
                    "tag_good": [],
                    "uid": "ismc78opglp2z4",
                    "children": [],
                    "tag_good_arr": [],
                    "id": "k7jxoryq34qyy",
                    "updated": "2020-03-09T03:56:38Z",
                    "config": {}
                },
                {
                    "anon": "no",
                    "folders": [],
                    "data": {
                        "embed_links": null
                    },
                    "subject": "<p>The updated query would not be &#34;bank financial
financial&#34; because that corresponding vector &lt;1,2&gt; is a completely separate
vector from the correctly calculated vector &lt;2,3&gt; (&#34;bank bank financial
financial financial&#34;) and the former is not generated using the Rocchio algorithm
whereas the latter, &lt;2,3&gt; is. When generating the result &lt;2,3&gt;, Rocchio
```

```
accounted for the relevant documents vector as well as the irrelevant documents
vector--both pieces of information that the problem gives.\u00a0\u00a0</p>",
                    "created": "2020-03-09T04:09:16Z",
                    "bucket_order": 5,
                    "bucket_name": "This week",
                    "type": "feedback",
                    "tag_good": [],
                    "uid": "is67tr8ajy022u",
                    "children": [],
                    "tag_good_arr": [],
                    "id": "k7jy50p6jyq3fr",
                    "updated": "2020-03-09T04:09:16Z",
                    "config": {}
                },
                {
                    "anon": "stud",
                    "folders": [],
                    "data": {
                        "embed_links": null
                    },
                    "subject": "How do we get &lt;2,3&gt; as the resulting vector ?",
                    "created": "2020-03-10T16:28:21Z",
                    "bucket_order": 3,
                    "bucket_name": "Today",
                    "type": "feedback",
                    "tag_good": [],
                    "uid_a": "a_2",
                    "children": [],
                    "tag_good_arr": [],
                    "id": "k7m3zcd72oz3de",
                    "updated": "2020-03-10T16:28:21Z",
                    "config": {}
                }
            ],
            "tag_good_arr": [],
            "no_answer": 0,
```

```
            "id": "k7jxlw561e41m2",
            "updated": "2020-03-09T03:54:24Z",
            "config": {}
        }
    ],
    "tag_good_arr": [],
    "no_answer": 0,
    "id": "k7jvl1fjm6v4v6",
    "config": {
        "seen": {
            "484": 4,
            "134": 9,
            "411": 2,
            "335": 0,
            "337": 1,
            "403": 7,
            "405": 5,
            "408": 6,
            "351": 8,
            "362": 3
        }
    },
    "status": "active",
    "request_instructor": 0,
    "request_instructor_me": false,
    "bookmarked": 4,
    "num_favorites": 1,
    "my_favorite": false,
    "is_bookmarked": false,
    "is_tag_good": false,
    "q_edits": [],
    "i_edits": [],
    "s_edits": [],
    "t": 1586199390386,
    "default_anonymity": "no"
}
```