Data Mining

Final Project Report

Hope English

E-Learning Website

105065421	Jorge André
105062466	Zater Zhou
105065426	Odilon Koutou
104065427	Phezulu Dlamini
X1050032-0	Pierre Michel Claisse
X1050014	Sami Laaroussi
103062710	Yamini Bitla
104065701	Lydia Chen

This final project report has been compiled and submitted as a requirement for the partial fulfilment of the Data Mining course at the National Tsing Hua University on the 3rd January 2016.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

We would like to express our deepest and profound gratitude to Professor Yi-Shin Chen who has continually displayed exceptional intellectual ability coupled with a hard-working ethic as she facilitated the learning process of the Data Mining course and ultimately offering us academic guidance throughout the project. It is also with our immerse gratitude that we acknowledge our Teaching Assistants, Chuck and John for their commitment and assistance offered to us during the course of the semester.

ABSTRACT

The ability to follow a good learning path is essential in boosting learners' skills. In this project, we propose an enhancement of the learning path based on users' saved vocabulary list. We generate association rules on saved word lists of previously watched videos to establish the next video to be watched and also suggest that users could guess the meaning of words to aid their learning process. This final project report contains the knowledge discovery process that followed to achieve the intended goal. First, the introduction contains the a summary of the Hope English e-learning website together with the goal and objectives. The data cleaning process follows. The next section illustrates the data exploration then the pre-processing stage. We also explain the processing, show the results and their discussion. Additionally, we have some findings, the conclusion and the challenges encountered.

INTRODUCTION

WEBSITE DESCRIPTION

Hope English is a course for non-English speakers to pick up English. The course uses what is called "planned learning", which is based on how much a person can absorb and remember each time he or she is taught a certain thing. The students are required to revise everything taught during the previous lesson at the start of each new lesson before teaching new things. Hope English has eight core principles :

1) There is no need to cram anything.
2) The learning method involves planning.
3) The students can have class at anytime of the day.
4) Each lesson costs less than 100 NTD.
5) The lessons are based on videos.
6) Hope English focuses on 4 skills : reading, listening, speaking and writing.
7) Lessons selected according to each student’s abilities.
8) The students get a quantified measure of what they have learnt.

Some students have experienced a rapid improvement in their scores. For example, some of them scored 355 additional points at TOEIC in 3 months. The students in the course generally think it helped them a lot, especially in terms of comprehension and listening. What is good about the course is also that you can follow it at home ; the timing is very flexible, so it is very good for adults with busy lives.

GOAL

Our goal is to enhance each user’s learning path based on the saved words.

OBJECTIVES

- To enhance the existing learning path using the student's saved words and adjusting the level
- For each student, suggest the next video to watch based on the words that he/she saved in the last watched video.
- To ask the student to guess the meaning of a word found using association rules.

In order to to improve the learning path, we analyzed the students' saved words in their first watched video such that we can to suggest the next video that each student should watch based on the association rules that we obtained using the common words between the student's vocabulary list and each unseen video.

An enhanced learning path is essential for each student in his learning process. We intend to enhance the existing learning path using the student's score, but not necessarily to replace it. Our assumption is that students save words on their vocabulary list for a particular purpose, thus we want to suggest the next video to watch based on their list of saved words. If the word used to suggest the next video appears in many videos, a video whose level is higher than the already watched one will be suggested considering the common word with the highest support from the association rule of the watched video and the words in the next video together with the student's score.

Furthermore, we suggest that a student be quizzed about the meaning of the word found in the association rule.

DATA CLEANING

Firstly, we performed data parsing as shown in the extract of the program in figure 1 below.

```
27                  userinfo newUser = new userinfo();
28                  String user = String.valueOf(iterator.next());
29                  JSONObject parse = (JSONObject) JSONObject.parse(user);
30                  Set<Map.Entry<String, Object>> infoset = parse.entrySet();
                    for (Map.Entry<String, Object> info : infoset) {
32                      //  System.out.println(info.getKey() + " " + info.getValue());
33                      if (info.getKey().equals("memberId")) {
34                          newUser.memberId = Integer.parseInt(info.getValue().toString());
                        };
36                      if (info.getKey().equals("chosenVideo")) {
37                          JSONArray parse1 = (JSONArray) JSONArray.parse(info.getValue().toString());
38                          for (int i = 0; i < parse1.size(); i++) {
39                              newUser.chosenvideo.add(parse1.getInteger(i));
40                          }
41
42                      }
43                      if (info.getKey().equals("listenScore")) {
44                          JSONArray listenscorearray = (JSONArray) JSONArray.parseArray(info.getValue().toString());
45                          for (int i = 0; i < listenscorearray.size(); i++) {
46                              int courseid = 0;
47                              int score = 0;
48                              JSONObject currentscore = listenscorearray.getJSONObject(i);
49                              for (Map.Entry<String, Object> entry : currentscore.entrySet()) {
```

Figure 1: An extract of the code for data parsing

Then, we defined the users. The definition of each user is shown in figure 2 “Serializable” means we can output each user as a file such that we do not need to rely on the json file. We used a lot of hashmap in this project. Hashmap means we can fetch each value using key in constant time O(1).

```
public class userinfo implements Serializable {

    public int memberId;
    public ArrayList<Integer> chosenvideo = new ArrayList<>();
    public HashMap<Integer, ArrayList<String>> wordlist = new HashMap<>();
    public HashMap<Integer, ArrayList<Integer>> score = new HashMap<>();

    public userinfo(int memberId, ArrayList<Integer> chosenvideo, HashMap<Integer, ArrayList<String>> wordlist, HashMap<Integer
        this.memberId = memberId;
        this.chosenvideo = chosenvideo;
        this.wordlist = wordlist;
        this.score = score;
    }
```

Figure 2: An extract of the code for user definition

Each user was converted to be an object as shown in figure 3 below. The total number of users is 6, 353. The user id is displayed as a filename and the object size. The more words saved by the user, the larger the size of the object.

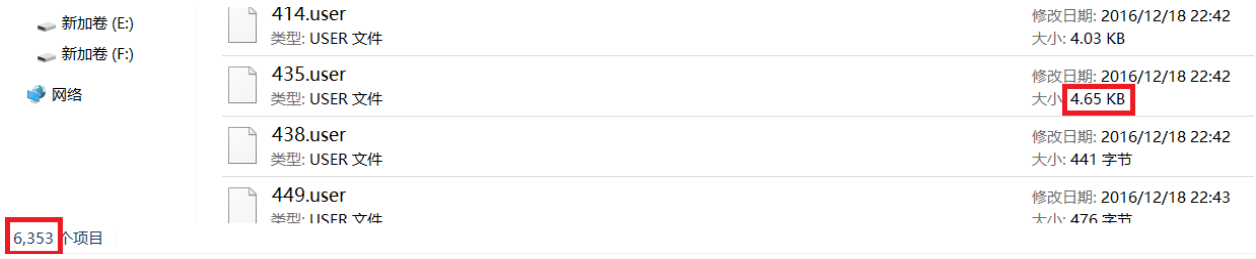

Figure 3: Showing users as objects

The original json file was transformed to a .csv file which enabled us proceed with the data cleaning stage. We observed that the data contained a number of unnecessary clutter. Thus, we had to perform some data cleaning in order to prepare the data for processing. The code in figure 4 filters all elements which are not English words. Each user may have watched several videos and saved some words. For each video, we use HashSet to let the vocabulary list to be unique. Furthermore, the code eliminates special characters such as double underscore, semicolon, spaces, commas, equal sign, exclamation mark and quotation mark. We also removed some words in the vocabulary list were in Chinese characters and split words like “Will Shakespeare”. The output is displayed in figure 5, an

extract of the csv file containing the member Id, number of videos chosen, number of saved words and average score.

```
33  private void dealuser(userinfo usr) {
34      for (Map.Entry<Integer, ArrayList<String>> t : usr.getWordlist().entrySet()) {
35          HashSet<String> arr = new HashSet();
36          for (int i = 0; i < t.getValue().size(); i++) {
37              String psp[] = t.getValue().get(i).split("[^a-zA-Z]{1,}");
38              for (int j = 0; j < psp.length; j++) {
39                  if (!"".equals(psp[j])) {
40                      arr.add(psp[j]);
41                  }
42              }
43          }
44          wordlistclean.put(t.getKey(), arr);
45      }
46  }
47
```

Figure 4: An extract of the code for data cleaning

	A	B	C	D	
1	memberId	numberVideosChosen	numberOfWordsSaved	averageScore	wordList
4	50679	8	230	90	devil;bunny;rabbit;bowl;throughout;goddess;harvest;moon;festival;giant;dress;awesome;
5	22808	11	498	71	planet;worse;totally;turns;out;sources;reliable;strikes;these;days;richest;exactly;superfic
7	6635	2	36	80	superficial;interactions;semester;fantasizing;barely;fairytale;reality;sorority;infatuated;br
8	50791	23	1394	76	though;probably;bowl;throughout;asia;stories;flowers;fall;full;bring;famous;goddess;mag
9	50779	5	55	8	hideaways;castles;bungalows;thousands;simple;decide;stranger;host;reviews;leave;belor
11	6991	9	91	92	goddess;strike;representative;distribute;accumulate;practically;colonialism;compensate;
12	50066	16	360	80	throughout;giant;goddess;awesome;scary;underneath;costume;rest;of;stuff;mention;sou
13	34011	4	178	94	come;on;paper;vaccinate;rabies;disease;certainly;causes;bullet;virus;damages;continent;
15	50906	5	117	80	factory;replenished;carbs;consumed;reserves;restrict;assesses;ultimate;allow;fitness;exp
16	46773	11	180	90	flew;harvest;dresses;awesome;scary;costume;strikes;way;worse;meanwhile;representat
17	50967	6	11	96	accomplice;inequality;versus;colonialism;perked;fastidious;interpreted;primarily;axis;visu
19	793	2	9	83	superficial;sorority;infatuated;cliche;embodies;matinee;movies;sneaking;nostalgic

Figure 5: Users' detail as transactions in a clean csv file

DATA EXPLORATION

After the data cleaning process, some graphs to aid in understand the data were generated to explore it. Using the .csv file, we generated a number of graphs in an attempt to understand the data particularly based on our goal. Table 1 and figure 6 shows the details of the dataset after we calculated the average numbers of users' saved words, watched videos and score.

	Whole Dataset	Old Dataset	New Dataset
Total number of users	6353	4000	3096
Average number of users' saved words	271	134.99	518
Average number of watched videos	10	5.51	17.75
Users' average score	62	49.47	85.44

Table 1: Details showing average number of users' saved words, watched videos and score

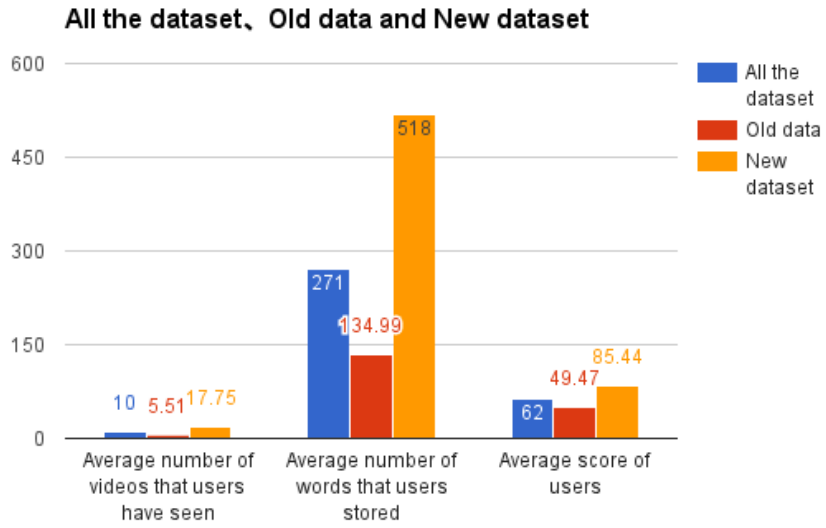

Figure 6: Average number of users' saved words, watched videos and score in the datasets

The transformed data enabled us to visualise the data as shown in figure 7 which displays the distribution of the member Id and the number of words.

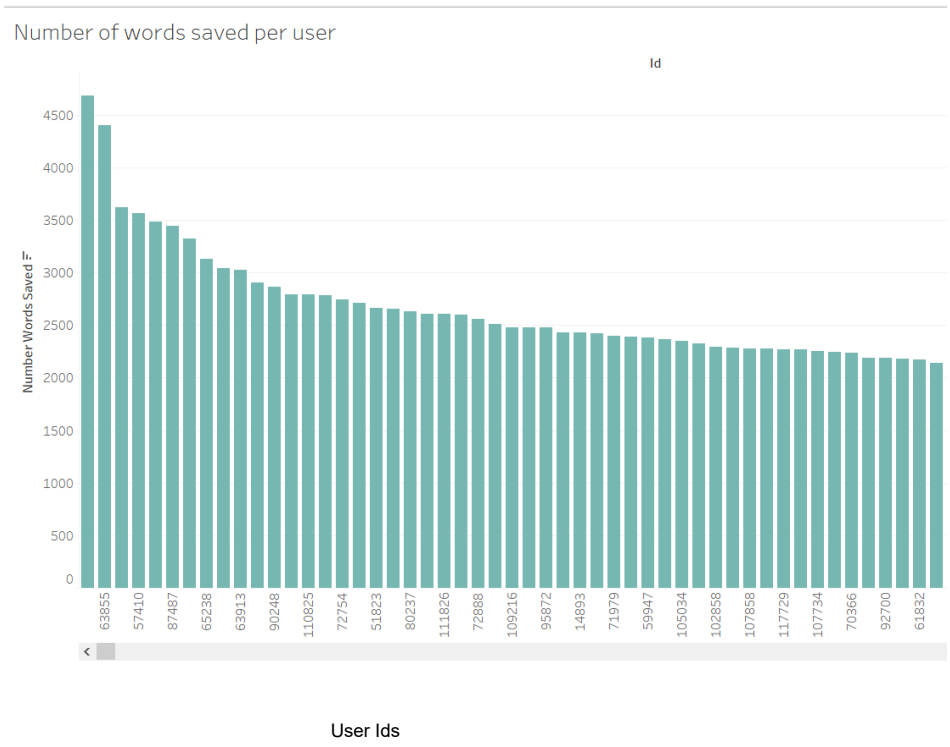

Figure 7: Distribution of the number of words saved by each student

To further observe the users' saved vocabulary list, we grouped users together with their saved words with a gap of 200 as shown in Figure 8. For example, 1486 users saved 0-200 words; likewise since 15 * 200 = 3000, it means only 3 users saved 3000-3200 words.

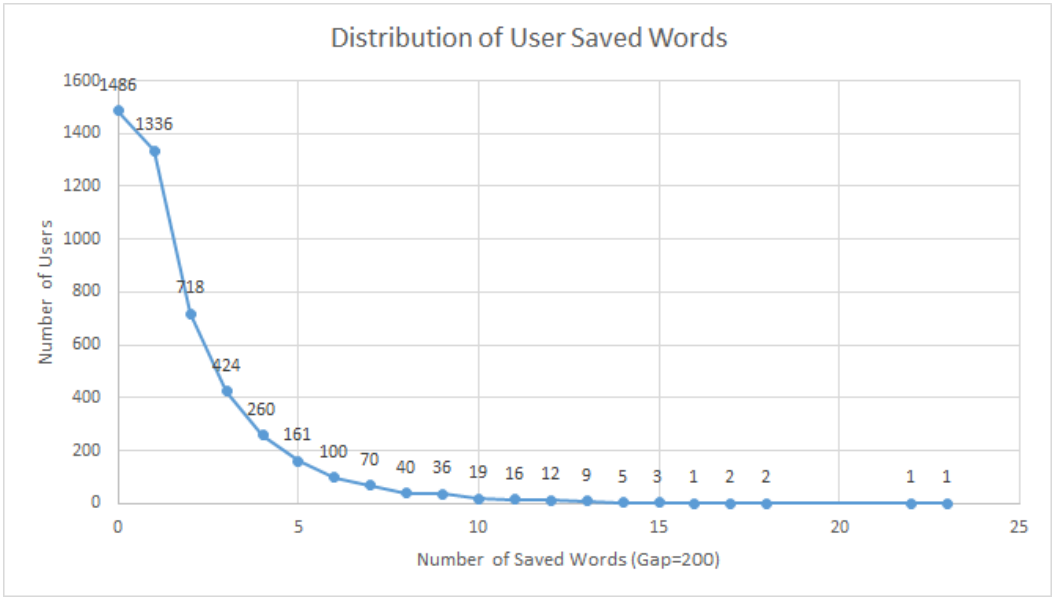

Figure 8: Grouped distribution of users' saved words with a gap of 200

Furthermore, we sorted the number of times each video has been watched and found that **video 3913** was the most watch video as illustrated in figure 9 which shows the distribution of the video id and the number of times a video was watched.

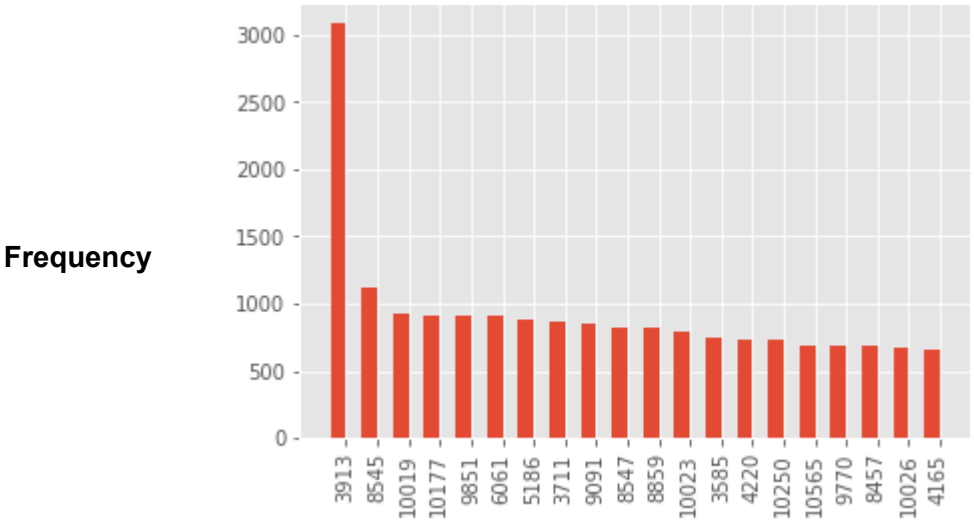

Figure 9: Distribution of watched videos with video id and frequency watched

We explored the data further and used performed clustering of users based on the average score and the average number of words saved as displayed in figure 10.

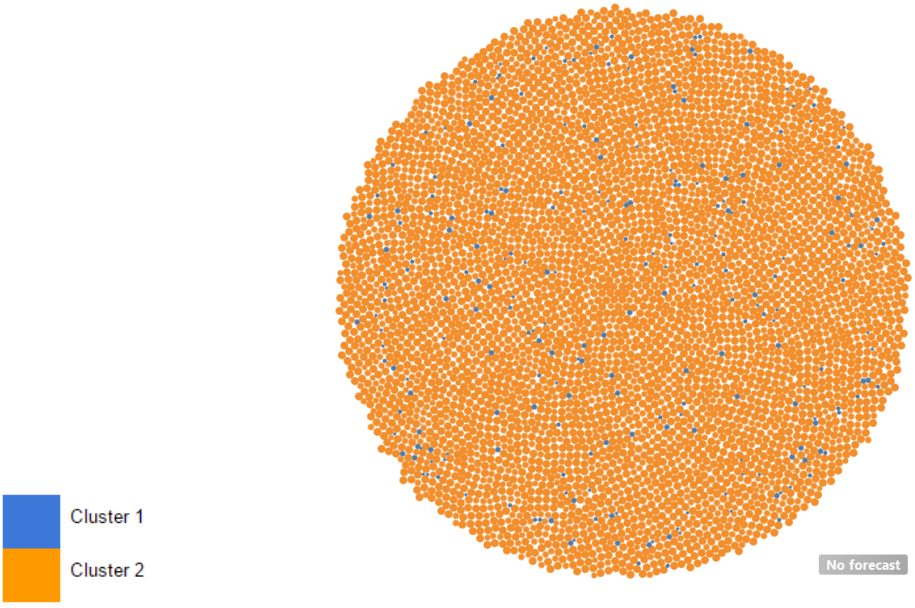

Figure 10: Cluster of users based on the average score and the average number of words saved

From this result, we deduced that students do save words for some reasons but not necessarily because they do not understand them. Table 2 illustrates details obtained from clustering the students.

Clusters	No. of Items	Average Score Sum	Saved Words Average
Cluster 1	1750	4.0611	32.55
Cluster 2	4601	85.115	390.91

Table 2: Details of the clustering results

PRE-PROCESSING

Since we intend to perform association rule mining on the users' saved word list, we extracted all users who did not watch video 3913. Furthermore, we removed users who saved at least one word in video 3913 and at least one word in any other video which totalled to 4637 users, including 102 users were found to have watched no videos and also removed duplicate users which resulted in a dataset with 6353 users.

After the data cleaning stage, we developed an algorithm using Java to generate a table for each video and the data was transformed in preparation for association rule mining. Each line in the figure below, represents one user who watched the video and saved some words separated by square brackets [ ] as indicated in figure 11. While analysing the existing learning path, we focused on users saved vocabulary list for video 3913 and the next watched video. The found 234 combinations.

```
1  [[pattern, articulate, articulation], [varied, proclamation, annually, feast]]
2  [[standard, diminishing, notes, held, sound, imitate, North, exaggerate, package, musical, Especially, patterns, articula
3  [[tries, diminishing, rid, imitate, blows, beat, if, exaggerate, train, musical, in, kind, articulate, come, develop, eve
4  [[rhythms, diminishing, clients, tongue, held, crucial, articulation, pattern, drum, imitate, hold, shortcut, client, jud
5  [[melody, organs, diminishing, shortcut, articulation, mastered, exaggerate, imitate], [stuffing, violent, Europeans, obs
6  [[no, in, ear, beat, driven, articulate, time, develop, an], [stuffing, a, revolves, in, parades, conquest, around, usher
7  [[through, standard, diminishing, practice, told, held, whole, rid, developed, imitate, these, blows, beat, twisters, Nor
8  [[organ, Especially, articulation, pattern, articulate, drum, rhythm, accent, diminish, imitate, phrase, shell, tons, sea
9  [[Twisters, clients, phrase, articulate, beats, exaggerate, imitate], [celebratory, observance, ritual, subscribe, evolve
10 [[diminishing, clients, tongue, crucial, articulation, pattern, units, rid, out, shortcut, if, exaggerate, all, Twisters,
```

Figure 11: Vocabulary list of watched videos 3913 and 8717

PROCESSING

The vocabulary list was used to run a PrefixSpan algorithm using Spark to find the association rules based on the sequence of users' watched videos as illustrated in figure 12 below. For example, we found the rule {exaggerate} => {crucible} which implies that users that saved the word "exaggerate" in video 3913 also saved "crucible" in a certain video X they watched next.

```
SparkConf sparkConf = new SparkConf().setAppName("ps").setMaster("local[2]");
JavaSparkContext sc = new JavaSparkContext(sparkConf);
Iterator<Integer> it = hm.keySet().iterator();

while (it.hasNext()) {
    Integer id = it.next();
    JavaRDD<List<List<String>>> sequences = sc.parallelize(hm.get(id));
    PrefixSpan prefixSpan = new PrefixSpan()
            .setMinSupport(0.90)
            .setMaxPatternLength(5);
    PrefixSpanModel<String> model = prefixSpan.run(sequences);
    List<PrefixSpan.FreqSequence<String>> collect = model.freqSequences().toJavaRDD().filter(new Function<PrefixSpan.FreqSequence<String>, Boolean>() {
        @Override
        public Boolean call(PrefixSpan.FreqSequence<String> v1) throws Exception {
            List<List<String>> javaSequence = v1.javaSequence();
            if (javaSequence.size() == 1) {
                return false;
            }
            for (int i = 0; i < javaSequence.size(); i++) {
                if (javaSequence.get(i).size() == 0) {
                    return false;
                }
            }
            return true;
        }

    }).collect();
    out.put(id, collect);
```

create a prefixspan object and set the minsupport and maxpatternlength

use prefixspan function to generate the model

to force the rule to have at least one word

Figure 12: Extract of code using PrefixSpan algorithm to generate association rules in Spark

We explored with different minimum supports, however, Spark does not provide confidence. Thus, we coded an algorithm to calculate the confidence and the following formula was used:

confidence (A implies B) = P (B/A), which is equal to P(A, B) / P(A)

RESULTS

We started with a lower minimum support of 70% and obtained 286757 rules. Then we used 80% minimum support and the computer generated 17299 rules as indicated in figure 13. The algorithm was executed in 46 seconds.

	A	B	C	D	E	F
1	video_id	premisse	conclusion	frequency	min_Support	confidence
2	8545	['imitate']	['crucible']	1033	80.83%	30.63%
3	8545	['articulate', 'exaggerate']	['crucible']	1121	87.72%	30.12%
4	8545	['articulate', 'rhythm']	['crucible']	1036	81.06%	29.98%
5	8545	['exaggerate']	['crucible']	1166	91.24%	29.65%
6	8545	['articulation']	['crucible']	1053	82.39%	29.49%
7	8545	['rhythm']	['crucible']	1065	83.33%	29.39%
8	8545	['articulate', 'exaggerate']	['conscientiousness']	1081	84.59%	29.04%
9	8545	['articulate', 'exaggerate']	['crucible', 'conscientiousness']	1063	83.18%	28.56%

Figure 13: Association rules for the minimum support of 80%

Ultimately, we generated the frequent itemset with a minimum support of **90%** to obtain **11239 rules** displayed in figure 14 and the computer took **45 seconds** as shown in figure 15.

	A	B	C	D	E	F
1	video_id	premisse	conclusion	frequency	min_Support	confidence
2	8545	['exaggerate']	['crucible']	1166	91.24%	29.65%
3	8545	['articulate']	['crucible']	1197	93.66%	27.78%
4	8545	['articulate']	['conscientiousness']	1158	90.61%	26.87%
5	3711	['articulate']	['sorority']	1129	90.18%	26.20%
6	10019	['articulate', 'exaggerate']	['treacherous']	872	90.64%	23.43%
7	10019	['exaggerate']	['treacherous']	902	93.76%	22.93%

Figure 14: Generated rules with a mininum support of 90% in a csv file

Figure 15: Execution time for generating all frequent itemset with 90.0% of support

RESULTS DISCUSSION

In this project we explored the idea of using association rules generated from user's saved words from the first watched video and the next watched video to suggest a better learning path in two ways. The more frequently a pair of words in the first two videos appears, the more support that a rule has. First, we can say that in the rule like "**articulate => exaggerate**" we can suggest the videos containing the word "**exaggerate**" as the next video to be watched. In the event the word "exaggerate" occurs in more than one videos, the user's score in the previous video such as video 3913 is taken into account to determine the most suitable level of the next video. The suggestion of videos using the user's vocabulary list can be beneficial to augment (not necessarily replace) the existing learning path that considers user's score.

Secondly, we can ask the student to try and guess the meaning of the word **"exaggerate"** in the following video. Since users can have as many reasons to saved words as the number of users, we assumed that that those words are very important in that video. If the user can guess these words in context correctly, they can know this video much more accurately. We think it is very important skill in reading and important for the students learning process or future usage.

ADDITIONAL FINDINGS

“Domains of exploration" based on user’s saved words

Our group really focused on analysing patterns in the user’s vocabulary list. And while trying to change our perspective, we have agreed that the saved vocabulary is something very personal and unique for each user. Thus, this information should be used to provide more user specific contents.

Independently of the reasons, the words saved have captured user’s attention in some way. Therefore we agreed on a different term to define them as “domains of exploration” since we could not provide enough evidence to support the idea of finding domains of interest.

Domains of exploration are the domains of knowledge extracted from user’s vocabulary.

They can represent:

- **Domains where the user shows interest in**
- **Domains the user has difficulties in**

On this task we have used Wordnet along with Wordnet Domains.

WordNet® is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations.

WordNet Domains is a lexical resource created in a semi-automatic way by augmenting WordNet with domain labels. WordNet Synsets have been annotated with at least one semantic domain label, selected from a set of about two hundred labels structured according the WordNet Domain Hierarchy.

For each user, we only took into account the five most frequent domains in their vocabulary list. The drawback is the difficulty to map words based on the context. A lot more exploration could be done is this aspect.

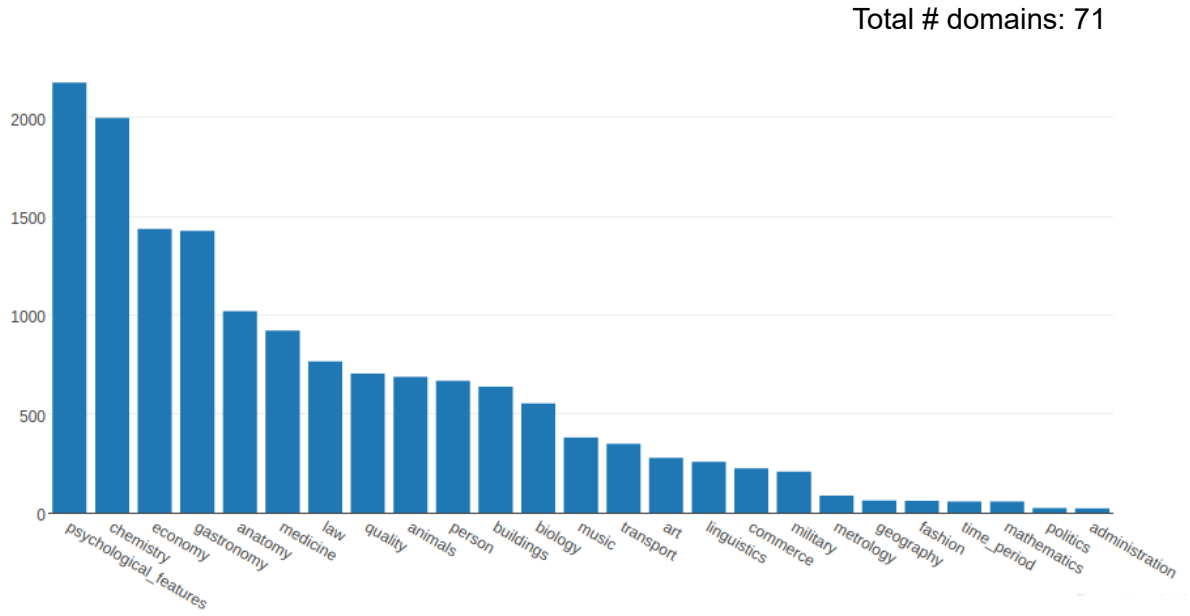

Figure 16: 25 most common domains

CHALLENGES

Five main challenges were encountered by the group which are lack of prior experience, time lost to idea generation, time wasted in failed attempts and coordination. Firstly, none of the group members had prior experience with practical data mining thus more time was dedicated to learning the implementation of the knowledge discovery process. Although some members had considerable coding ability, each member used different tools and techniques. For instance, some used Python while others used Java and Spark.

IDEAS

Secondly, a considerable amount of time was also spent in idea generation. Initially, the group identified the following ideas:

Idea 1: Predict the new words to save in a video

When the user watches a video, he may save some words in his vocabulary list for later use. Our idea is to suggest to the user new words to save based on the words that other users saved. We can use the vocabulary list the user saved as transaction data. For example, a user who saved the word “articulate” is most likely to save the word “learn” based on the rule {articulate} -> {learn} that we would have found in the data.

Idea 2 : Finding related videos to propose a learning path

We noticed that in the dataset, that the videos attributes such as **wordLevel**, **sectionLength**, **videoSpeed** and **subtitleLengthRatio** tend to have the same pattern.

First, to create cluster of videos based on their level of difficulty (hard, medium and easy). We noticed that on the website, we don't have any grouping of the videos by their level of difficulty. For example, I may have a medium level in English and I would like to watch the videos that will help me to improve my English. Second, once the member chose a level of difficulty, we want to show him the next video that other members viewed based on association rules. For example, we could find rules like {video 3193} -> {video 3003}. This will be the learning path for the user. We could either suggest this learning path to the user or try to predict the next video that the user will chose after seeing a certain video.

Idea 3: Predict user video score

The user's score depend on many dimensions. The idea is to use the level of difficulty of the video (**wordLevel**, **sectionLength**, **videoSpeed** and **subtitleLengthRatio**) and user's previous score to predict the user's score on a video he has never seen.

Idea 4 : Student performance and learning path mining

Taking into consideration each student's learning path. We first want to find out common patterns among students with lowest average score and highest average score.

Also for students that have used the platform more than once. Access if there is any relationship between student's chosen learning path and their grade. After a student's first experience with the platform, we can see if their first video score influenced or not how they chose their next video (our opinion: after getting a good grade, students might tend to try a similar video, trying to replicate their last experience, or try something harder). To determine how hard are the videos compared to each other we will use **wordLevel**, **sectionLength**, **videoSpeed** to compare.

The goal is to find out how their chosen path have influenced their overall average grade, because this might explain why most students are underperforming on the platform.

After presenting the ideas, we were advised to pursue one of the three ideas preferably the first idea but the group was to explore the idea further. According to the advice, idea two could not be verified.

ATTEMPTS

In an attempt to implement idea one, the group was advised to generate a huge matrix and further generate some clusters of the data. Actually, it was necessary to compare all the users to determine how many words they have in common in their vocabulary list. For example, we compared user 1 and user 2 to ascertain how many words they have in common. The huge matrix was to done to establish its sparsity which was essential to ensure that the dataset is sufficient to make meaningful association rules. However, video 3913 needed to be removed.

To generate the matrix, the group used Java and Spark. An extract of the code is shown in the function in figure 17 which gets two vocabulary lists. The vocabulary list (each word is unique) of each user. If user 1 vocabulary list is greater than user 2, we take user 2's list then we compare user 2's words with user 1's.

```
85      private static int calc(UserDeal get, UserDeal get0) {
86          HashSet<String> small = get.vlist.size() > get0.vlist.size() ? get0.vlist : get.vlist;
87          HashSet<String> huge = get.vlist.size() > get0.vlist.size() ? get.vlist : get0.vlist;
88          int i = 0;
89          Iterator<String> iterator = small.iterator();
90          while (iterator.hasNext()) {
91              String next = iterator.next();
92              if (huge.contains(next)) {
93                  i++;
94              }
95          }
96          return i;
97      }
```

Figure 17: Function to obtain two vocabulary lists

From the common words obtained above, we developed the following matrix showing the number of common between two users as shown in figure 18.

	100015	100016	100031	100058	100098	100108	100133	100165	100171	100175	100176	100182	100227	100234	100245	100250	100251	100274	100280	100284	100296	100298	100310
100015	345	34	1	51	7	1	102	129	9	34	3	13	0	3	25	16	1	4	6	73	2	2	6
100016	34	410	2	18	4	1	93	63	7	8	10	3	6	4	65	8	1	0	5	30	2	3	26
100031	1	2	17	0	0	0	5	0	0	0	0	0	0	0	2	0	0	0	0	2	0	0	0
100058	51	18	0	473	25	2	328	314	45	26	6	32	2	8	32	31	0	2	5	192	1	1	25
100098	7	4	0	25	106	19	8	5	17	1	4	16	0	0	5	1	0	0	4	5	0	0	22
100108	1	1	0	2	19	52	9	3	0	0	0	0	1	0	0	2	0	0	1	5	1	0	2
100133	102	93	5	328	8	9	2083	702	31	52	30	28	2	22	112	39	1	14	27	502	12	10	41
100165	129	63	0	314	5	3	702	1174	44	139	16	24	1	17	74	39	0	5	15	261	12	5	27
100171	9	7	0	45	17	0	31	44	173	12	1	22	3	0	6	10	0	4	4	51	2	1	17
100175	34	8	0	26	1	0	52	139	12	211	26	5	3	2	32	2	0	1	6	33	1	0	6
100176	3	10	0	6	4	0	30	16	1	26	151	2	1	5	13	13	0	0	4	8	1	1	5
100182	13	3	0	32	16	0	28	24	22	5	2	149	0	1	20	27	1	0	2	21	0	13	14

Figure 18: An extract of a huge matrix for common words between users

The size of the matrix was 4700x4700 = 22.090.000 cells. We found that the percentage of zeros in the matrix was 14.57%, the sum of the common words for each user and the average of the common words for each user. Table 3 displays the probability of the common words for each user i.e. the probability of each user equal to the number of common words for both users divided by the number of words which the user stored and the average of the average number of words was 29.8217. For example:

	sum	**average**	**probability**
User (100015)	121889	25.9393	0.55%

Table 3: Probability of common words for a single user

Additionally, coordination was also a challenge in this group. It was difficult to establish consensus regarding the division of work hence some group members attempted to do whatever they could in an ad hoc manner which led to the duplication of some of the work. Most deliverables were rendered unusable to due changing project goals.

Lastly, due to the deadline we were unable to choose an efficient way to determine the rules interestingness.

CONCLUSION

Association rule mining which is one of the data mining techniques has been utilised to suggest the next video to be watched as an improvement of the learning path using the students' saved vocabulary list. Additionally, the results of the rules can be used to ask the student to guess the meaning of the word. We think that using the users' saved words is possible to provide a better learning path to users. We learnt from the project that data mining is an important tool to achieve a goal, the vision defined by humans. The goal has to be clear before any data mining task is carried through. It is important to know what data we have and the message in the data.

ASSESSMENT OF GROUP MEMBERS

Group Name: Group 1		
Member Name	**Student Number**	**Percentage of Contribution**
105065426	Odilon Koutou	15%
105065421	Jorge Andre	15 %
105062466	Zater Zhou	15%
104065427	Phezulu Dlamini	12%
104065701	Lydia Chen	11%
103062710	Yamini Bitla	10%
X1050032	Pierre Claisse	11%
X1050014	Sami Laaroussi	10%

The group held 14 meetings since the inception of the project. All group members attended meetings unless there was a valid reason for an apology. All group members made an effort to contribute and explore ideas.

Odilon, Jorge and Zater were the main developers who coded many of the group's attempts and ultimately the final idea using Python, Java and Spark. They were also responsible for presenting the ideas, attempt and the final product in class.

Phezulu, Lydia, Yamini, Pierre and Sami were mainly responsible for the compilation of the report and presentations as well as arranging meetings however, they had minimal contributions in the coding process and generation of some visualisations.

Odilon Koutou	Jorge André	Zater Zhou	Phezulu Dlamini
Lydia Chen	Yamini Bitla	Pierre Claisse	Sami Laaroussi