What can we recommend to game players?
-- Implementing a system of analyzing game reviews

Jueran Huang

University of Tampere
Faculty of Natural Sciences
Degree Programme in Computer Sciences
Software Development
M.Sc. thesis
Supervisor: Zheying Zhang
April 2018

University of Tampere

Faculty of Natural Sciences

Degree Programme in Computer Sciences

Software Development

Jueran Huang: What can we recommend to game players? – Implementing a system of analyzing game reviews

M.Sc. thesis, 63 pages and 4 index pages

April 2018

Abstract

With the rapid development of game industry, games take an increasing important role of entertainment in our daily life. With more and more new games coming to the market, it is difficult for players to choose suitable games. This thesis proposed a way of analyzing game user reviews to help players selecting games.

A large amount of game reviews is generated by game players in text format every day, evaluating thousands of reviews one by one becomes impossible. Topic mining techniques makes it possible to extract the topics(aspects) from a large amount of textual data and summarize the reviews in a general level. A lot of research has put efforts on developing and optimizing topic mining algorithms, including developing systems or applications based on these algorithms to analyze user review data from different sources. Theses source includes social media platforms such as Twitter, mobile application stores such as Google Play, movies websites such as Netflix, etc. However, there is a lack of an application especially focus on game reviews.

The aim of this thesis is to develop a system which covers all the necessary steps of topic mining of game reviews, including review data collecting, storage, processing and visualization. Topics generated by the system are supposed to reveal the content of the game and user's game experience. Testing and evaluation of the system are conducted after implementation. After a series of experiments, it is validated that the system runs smoothly in real-world situations and produces suitable topics that can help users to understand and select games.

Keywords and terms: text analysis, topic mining, Steam, game review, web crawler, topic mining, software development.

Contents

1. Introduction

With the development of game industry, games have taken a more and more important role of entertainment in our daily life. With more and more new games coming to the market, the game recommendation is needed for players to choose suitable games. Normally a game is recommended by some game media such as IGN [2018] and GameSpot [2018]. By reading reviews, users decide to try a game or not. However, with more and more games appearing on the market, there are not enough reviews to cover all kinds of games shortly after the games releases.

On the other hand, a large amount of game reviews is given by game players in text format on game websites such as Steam [2018] every day. These reviews contain abundant information about games and players' experience which could possibly be used in the game recommendation. Topic mining [Blei et.al., 2003] makes it possible to extract topics (keywords) from a large amount of text data and summarize the reviews at a general level. The results based on a large amount of data are more comprehensive and authorable, which are supposed to be recommended to users.

A lot of research has put efforts into developing and optimizing topic mining algorithms for analyzing user reviews. The research mainly focuses on improving existing algorithms such as Multi-grain topic models [Titov and McDonald, 2008], or developing a comprehensive system that addresses a specific situation such as SUR-Miner [Gu and Kim, 2015]. The source of the review includes social media platforms such as Twitter [2018], mobile application stores such as Google Play [2018], movies websites such as Netflix [2018] and so on. However, there is a lack of a system which focuses on analyzing game reviews. The aim of this thesis is to develop a system which covers all the necessary steps of topic mining of game reviews, including review data collecting, processing and results visualization.

This thesis aims to answer the following questions:

- What are the requirements of the system which could cover all the necessary steps of data mining (especially for Steam game reviews)?
- How to ensure the performance, usability and expandability of the system?
  For example, collecting thousands of user review data from web pages may take hours, but users of the system should not wait for that long time.
- How to evaluate topics and keyword generated from game reviews? For example, if the system generates five topics from the user reviews, how to ensure these topics are valid topics? How to judge if a certain topic is suitable to be

recommended to users.

The system implementation addresses the problem of fulfilling requirements and ensuring performance, usability and expandability. The system is implemented as a web application, and Spring framework is used to develop the system. Python beauty soup is used for crawling review data form Steam. MySQL database is used for review storage. Bootstrap is used for user interface design to support the usability of the system. LDA is used as the core mining algorithm. After implementation, testing and evaluation of the system and experiment results are conducted after implementation.

Key elements of the game [Sweetser and Wyeth, 2005] are used as a guideline for evaluating the result of topic mining. Topic validation, topic coverage test and keywords map test are conducted to evaluate the mining result. The performance test is conducted to ensure the system can run in real-world situation, particularly in dealing with a large amount of data, and the generated topics and keywords are accurate for recommending the main features of a game.

The thesis consists of eight chapters. Chapter 2 presents an overview of the game industry, the well-known game distribute platform, Steam [2018], the user reviews and proposes potential topics of Steam game review. Chapter 3 addresses topic mining, and introduces well know topic mining algorithms such as Probabilistic Latent Semantic Analysis(PLSA) and Latent Dirichlet Allocation(LDA). [Hofmann, 1999] [Blei, et al. 2003]. Chapter 4 and 5 demonstrate the design and implementation of the system, which includes the process of extracting requirements, system architecture and detailed techniques used for implementation. Chapter 6 illustrates the testing process of the system. The prformance test is conducted based on game review data from Steam. Topic related test is conducted by training the algorithm with different parameters and analysing the results manually. Chapter 7 analyses the results, highlights the contribution and discusses the limitation of the study and possible improvements. The last chapter summarizes the whole thesis.

2. Background and theory

2.1 Growth of the game industry and Steam

A Global Games Market Report from Newzoo [2016] shows that global games market jumped to $99.6 billion with an increase of 8.5% in 2016. The game industry has become one of the fastest growing businesses in the world. According to Egenfeldt-Nielsen et al. [2016], the global game sales were $64.9 billion in 2014, still lagging behind the film entertainment which had sales of $90 billion. However, game sales have already surpassed recorded music, which had only $20.97 billion of sales by the end of 2014. Additionally, the expected annual growth rate of games is 9.6 percent for 2013-2018, whereas films can only expect a 4.5 percent yearly growth rate during the same time. Researchers have predicted that global market scale of games will reach $107 billion in 2017. From these, we can see that the game industry is really promising.

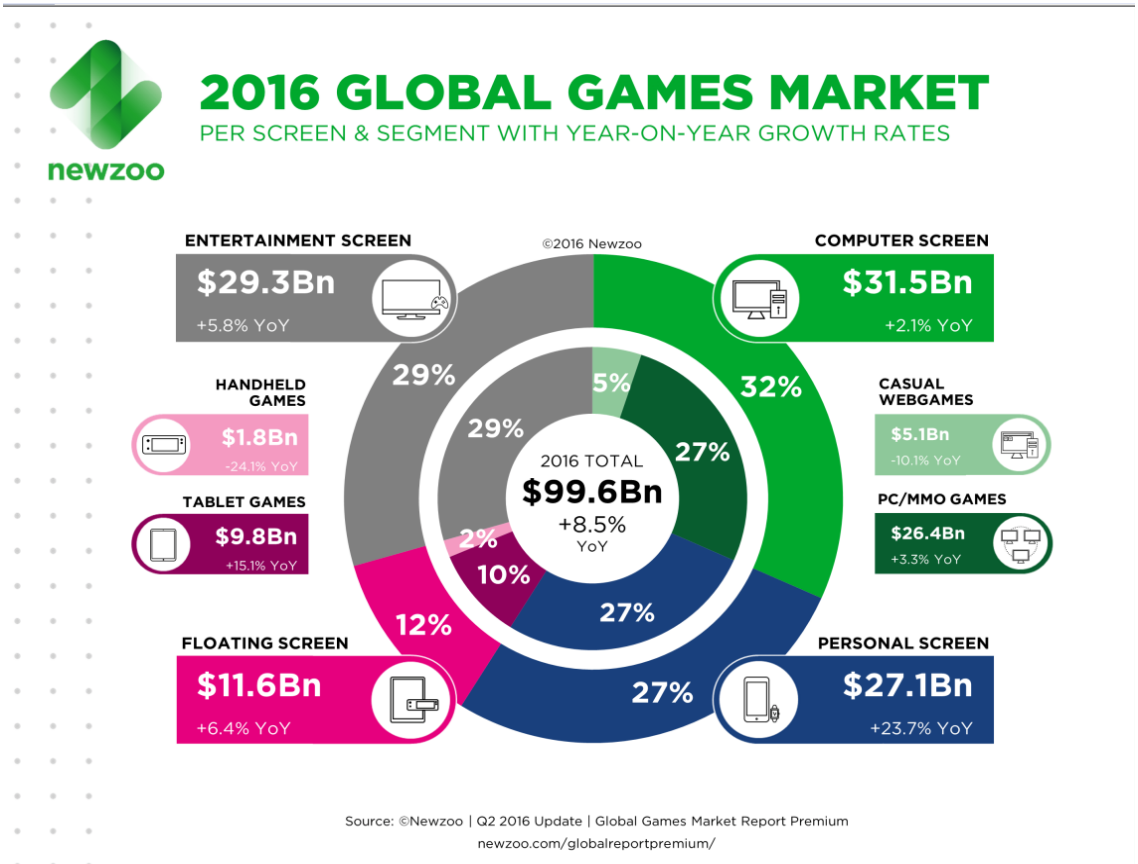

Figure 2.1 Global Games Market Report [Newzoo, 2016]

As shown in Figure 2.1, PC games have the biggest share within this rapidly growing game market, this draws the author's attention to a digital game distribution platform

called Steam [2018] which is considered to be the largest digital distribution platform for PC games

Steam offers digital rights management. It is quite similar to Google Play and App Store. Steam makes it possible for users to install and automatically update games on multiple computers. Besides, Steam provides users with community and social networking services features. For example, Steam provides cloud saving, in-game voice and chat functionality to support multiplayer gaming. The Steam platform occupied up to 75% of the whole market space in October 2013 [Edwards and Cliff, 2013]. In 2015, users purchasing games through Steam or through Steam keys from third-party vendors totalling around $3.5 billion, which accounts for 15% of the global PC game sales for the year, based on estimations made by the tracking website Steam Spy [Wawro and Alex, 2016] [Hall and Charlie, 2016]. By the end of 2015, Steam had over 125 million registered accounts, with 12.5 million concurrent users [Smith, 2015].

2.2 Recommendation and Steam tags

Recommendation systems produce a ranked list of items on which a user might be interested. In real life, the instance of items to be recommended to users can be movies, books, game, news, articles, etc. Two main approaches are used to build a recommendation system - content based and collaborative filtering [Melville et al., 2002].

Content based recommendation system provides recommendation directly based on similarity of items. The similarity is calculated based on the content attributes of items using appropriate algorithms [Debnath et al., 2008]. For example, the content of a game item can be the music, graphic, storyline, etc. By analysing these contents, the similarity between games can be calculated. The collaborative filtering recommendation system uses user preference data, it calculates the similarity between two users based on their activities. An item which is liked by a certain user is highly possible to be recommended to similar users. [Debnath et al., 2008].

In a real-world situation, researchers use an integration of content based and collaborative filtering to provide recommendations [Debnath et all., 2008]. Because content-based recommendation lacks diversity. It recommends similar items according to the item which the user is browsing currently but ignores other items that the user might be interested in. Collaborative filtering does not have the same high accuracy as content based, but it provides diverse items by analyzing similar users' activities. Additionally, some minor ways are used for the recommendation in special situations. Despite content

data and user preference data, location-based recommendation [Park et al., 2007] also uses location data. Location data is necessary when recommendation needs user location data. TripAdvisor [2018] and Airbnb [2018] recommend restaurants and hotels to users according to their locations or destinations, Uber [2018] sends taxis to users according to their locations as well. Cars, restaurants, and hotels which are close to a user will have a higher probability to be recommended.

2.2.1 YouTube Recommendation system

Personalized recommendation is a key method for information retrieval and content discovery in today's information-rich environment [Davidson, et al. 2010]. 'Personalize' means that appropriate recommendation is given especially to that particular user. Normally, Users access an entertainment website like YouTube for a wide variety of reasons: To watch a single video that they found elsewhere, to find specific videos around a topic, or to just be entertained by content that they find interesting [Davidson, et al. 2010]. Game players also have the same goals as YouTube users. Game players would come to Steam for some well-known games such as *Witcher3: Wild hunt* or search for some certain type of game like MMORPG (massive multi-player online role-playing game) or science-fiction. They could also just go to the game store to find any game that they find interesting. Personalized Recommendations help users to choose games when they do not have specific goals.

Nowadays the homepage of content providing website is all facilitated personalized recommendations. Figure 2.2 shows the recommendation given by YouTube,

Figure 2.2 YouTube recommendation [Davidson, et al. 2010]

It can be found from the figure that YouTube uses user activity data such as 'watching' and 'favoriting' in the figure. Actually, YouTube also uses the integration of content based and collaborative filtering for personalized recommendation [Davidson et al., 2010].

Content data includes raw video streams and video metadata such as the title, description, cover, etc. Collaborative filtering uses user activity data, which can further be divided into explicit and implicit categories. Explicit activities include rating a video, favoriting/liking a video, or subscribing to an uploader. Implicit activities are data generated from users' watching and interacting with videos, e.g., users started to watch a video and user watched a large portion of the video [Davidson et al., 2010].

2.2.2 Steam recommendation system and game tags

Steam also uses the integration of content based and collaborative filtering for recommendation. Two key inputs of the Steam recommendation system are the content of the game and user's activity of playing games. This thesis focuses on the first type of input: the game content, because the game content is quite different from video content. Normally a big game on Steam has more than 10GB of game files, it is impossible to analyse game files like analysing video streams to understand what a game is about. More importantly, game content can be changing all the time. For example, the well-known games like *Dota2* [2018] and *Counter strike: global offense* [2018] may have updates almost every month. The updates include new features, and enhancement of player experience. There should be a dynamic way to extract game contents.

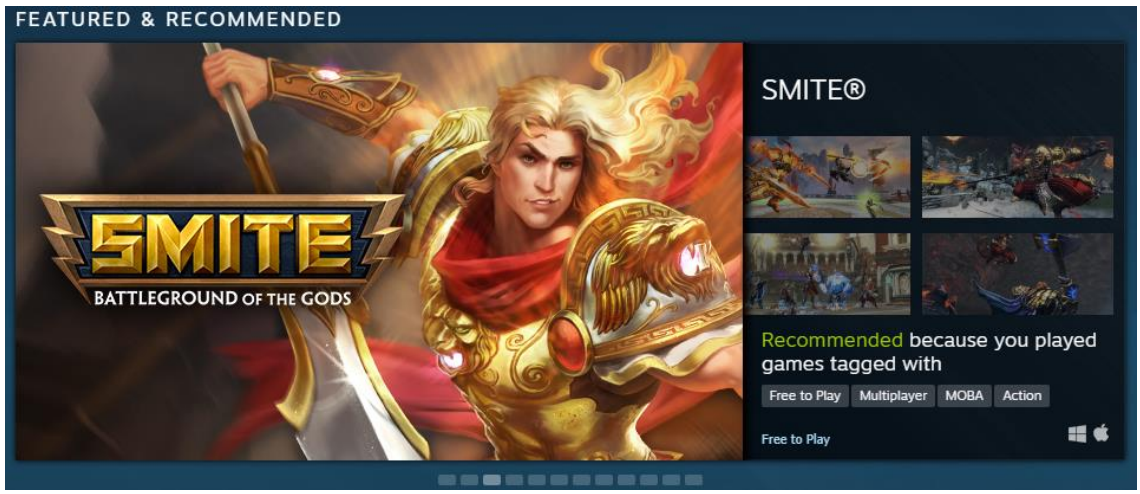

Figure 2.3 Steam recommendation module [A screenshot available at: http://store.steampowered.com/]

As shown in Figure 2.3, Steam use the so-called game tag to indicate game content in its recommendation. Tags are keywords that describing the contents of a game. A game tag is one of the key elements of Steam's recommendation system, which is used to provide the classification for games. Users are recommended with the games that have tags

similar to those that they have played before.

For a game on Steam, there is a list of tags that users add to describe the game, as shown in Figure 2.4. Tags are displayed according to their popularity from the highest to the lowest, and only top 20 tags are displayed on the game homepage. Users can add their own tags to the game or click the plus symbol to increase that tag's popularity. Users can also report any tags that are incorrectly applied to any games. The report influences the popularity of a tag. In this way, Steam ensures that tags are the opinion of the majority of users, which is relatively objective. And by adding frequency to tags, old and incorrect tags can be eliminated while new and hot features come up, which increase the accuracy of game tags after game updates.

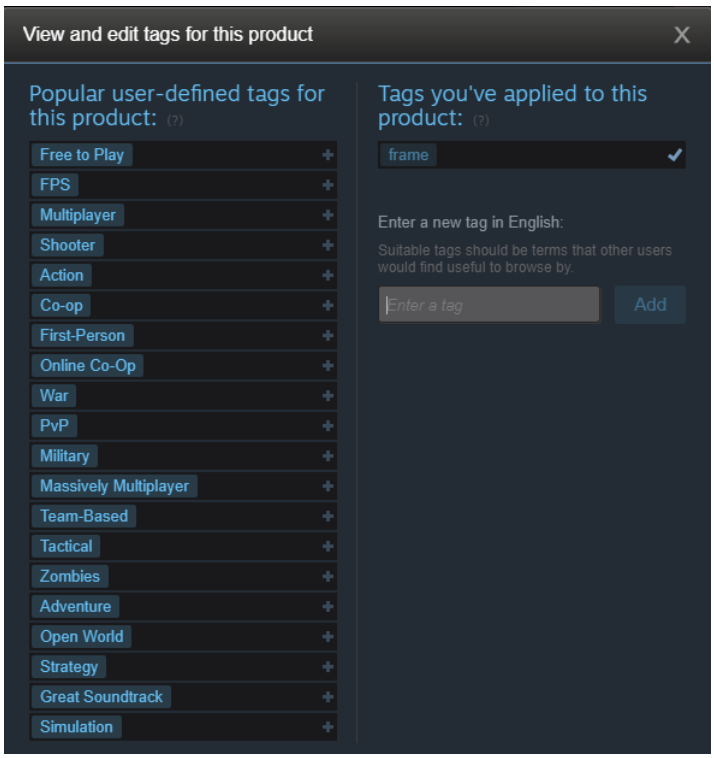

Figure 2.4 Steam game tags [A screenshot available at: http://store.steampowered.com/app/230410/Warframe/]

2.3 Game review and its features

Game reviews contain opinions and experience of players after they played a certain game. From game reviews, a lot of information could be found such as the content of a game, the quality of a game, experience of players, etc. A well-written game review can effectively attract new players. In general, game reviews could be divided into two types based on the type of reviewers: professional reviews and user reviews.

Professional reviews are provided by game media or entertainment-based websites, such as GameSpot [2018] and IGN [2018]. Figure 2.5 shows an example of such kind of reviews.

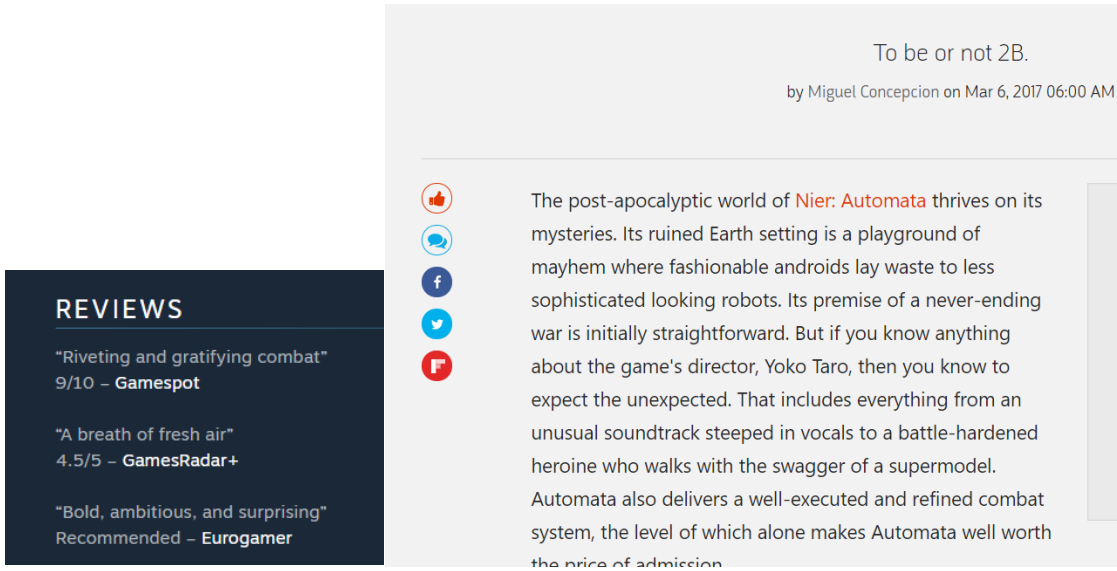

Figure 2.5 Professional reviews [A screenshot available at: https://store.steampowered.com/app/524220/]

This figure is retrieved from the homepage of the game: *Nier:Automata*. Steam provides links to three professional reviews, these reviews are provided by well-known entertainment-based websites such as GameSpot [2018]. Reviews from these websites are written by professional game reviewers, and the review can be pages long, covering all possible aspects of the games. The aim of these reviews is to introduce a certain game to players. In general, reviewers should be professional journalists with extensive knowledge and experience in game industry. They are required to have abundant experience of a game before writing a game review.

User reviews of the game are players' opinion about the game after playing. Most of the user reviews are short, partial and descriptive. Most players would only focus on one or two aspects of the game such as storyline, graphic, music, etc. They express their opinions but lack detailed analysis and explanations. Figure 2.6 shows an example of user review from Steam. In this review, the user only expressed his emotion about the ending of the game, which is subjective and incomprehensive.

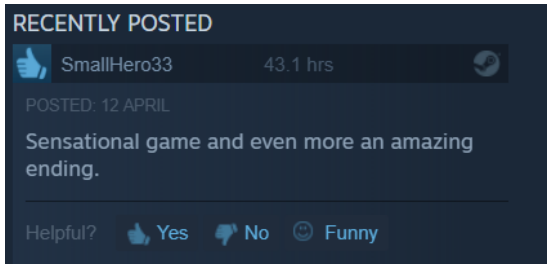

Figure 2.6 Steam user reviews [A screenshot available at: http://store.steampowered.com/]

In addition, the quality of user reviews is difficult to evaluate. A few reviews might even be inconsistent in content. For example, speaking highly of the game in some sentences of the review but complaint about the game in other sentences. Compared to professional reviews, the number of user reviews is enormous, and sources of user reviews are rich and varied. Professional reviews are comprehensive and authorable, but sometimes they can also be partial due to the limited experience of the writer.

2.4. Aspects in game user reviews

Professional reviews are more comprehensive compared to user reviews because they focus on multiple aspects while one user review only focuses on one or two aspects. Thus, retrieving aspects can be an effective method to get useful information from the user reviews. For example, Zhang [2017] tries to extract topics from different categories of tweets from Twitter. The categories include education, economy, military, sports, politics, etc, which can be regarded as topics in the tweets. Like the user review of games, one single tweet will focus on one topic such as economy, education or politics. Twitter provides developers with APIs that can get tweets according to a selected topic [Zhang, 2017]. Steam does not provide classifications for reviews of a certain game, so there is no pre-defined topics provided by Steam. However, game reviews still have same topics in common. According to the research of Bernhaupt et al [2007], two main aspects of evaluating games are usability and playability. Based on this, Sweetser and Wyeth [2005] further propose detail elements of the game, as shown in Figure 2.7. In this thesis the author use ‘aspects for evaluating games’ to refer to them.

Aspects	Flow
The Game	A task that can be completed
Concentration	Ability to concentrate on the task
Challenge Player Skills	Perceived skills should match challenges, and both must exceed a certain threshold
Control	Allowed to exercise a sense of control over actions
Clear goals	The task has clear goals
Feedback	The task provides immediate feedback
Tutorial	Instructions and guide to complete the task
Immersion	Deep but effortless involvement, reduced concern for self and sense of time
Social Interaction	N/A

Figure 2.7 key game elements [Sweetser and Wyeth, 2005]

1.The game itself

The fundamental elements about the game itself are the storyline, music, graphics, characters, etc. The phrase 'storyline' is frequently used in Steam reviews to refer to a game. It keeps the different parts of the game connected and makes players want to continue the game.

Characters are the avatar of players in the game, well-designed characters are the key feature to attract new platers

Nier: Automata is a Japanese RPG game, which is popular for its character. There are quite many reviews expressing love for the game character and the storyline. For example, here is a review from Steam which is related to the fundamental elements:

'I love the story in this game, great characters, great design. If you like open world RPG then you will definitely love this'

[Retrieved from https://store.steampowered.com/app/524220/]

2.Concentration

Games need to able to hold the concentration from players to be entertainable. The more concentration a task requires in terms of attention and effort, the more absorbing it will be. [Csikszentmihalyi 1990]. The time that a user spends on a game is used as the criteria to evaluate concentration.

'*From my almost 1k hours playing this game (I know, it's not a lot) I can definitely say that this is one amazing game*'

[Retrieved from https://store.steampowered.com/app/570/Dota_2/]

Listed above is a review from *Dota2* which talks about time spent on the game, players tend to express their concentration for games by discussing time spent on them.

3.Challenges

Games create enjoyment by challenging players [Lazzaro and Keeker ,2004]. Games should provide different levels of challenge so that it could appeal to players with different levels of skill. Meanwhile, new challenges should be provided as the player skills improve while the game progresses. Normally challenges are set in the process of completing a quest, and the player will get closer to the next stage of the game storyline. Figure 2.8 shows the achievement system on Steam. By completing the required challenges, players will be reward will different achievements, which can give them the feeling of accomplishment.

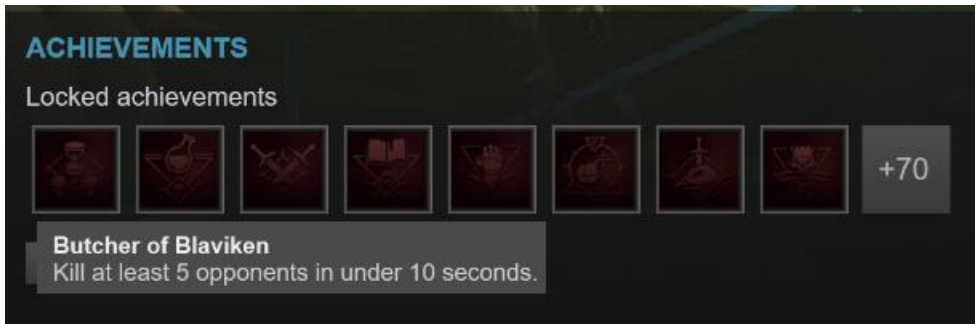

Figure 2.8 Achievements system on Steam [A screenshot available at: http://store.steampowered.com/]

4.Tutorials

Games should provide players with attractive and interesting tutorials [Federoff, 2002] that allow them to get involved in the game quickly and easily [Desurvire et al., 2004]. In-game tutorial feedback is vital in learning the basic mechanisms for playing [Pagulayan et al., 2003]. Nowadays most games provide compulsory new player tutorials. Only after finishing the tutorial, the player can start the real game. Some games would integrate the tutorials into game storylines, for example, there could be a game character giving instructions on how to play the game while interacting with players. Other ways of encouraging includes offering extra awards for players if they finish the tutorial quest.

5. Control

Control means players can feel a sense of control over their actions. The game should be able to translate players' intentions into in-game behaviour [Pagulayan et al., 2003] and feel in control of the actual movements of their character and the manner in which they explore their environment [Federoff, 2002].

There is a review from Steam about game control: '*I really enjoyed whole game, lovely gameplay, world and graphic! Keyboard/mouse control kinda suck and there has been some bugs as well. Anyway I may recommend to you this game, I'm sure you gonna like it. :)*' [Retrieved from https://store.steampowered.com/app/524220/NieRAutomata/]. By reading similar reviews, it is found that most game use mouse and keyboard to let player control their character, bugs in control will significantly decrease user experience.

6.Clear goals

Games should provide players with clear goals at appropriate times. Games must have an object or goal [Federoff, 2002]. Games should present players with a clear overriding goal early in the game [Federoff, 2002], which is often done through an introductory cinematic that establishes the background story [Pagulayan et al., 2003]. The goal should be conveyed to the player in a clear and straightforward way [Pagulayan et al., 2003]. Despite the general goal, each stage of the game should also include multiple small goals [Federoff, 2002], which is known as 'missions' or 'quests' that list the immediate goals of the current part of the game and suggest some of the obstacles that the players might face [Pagulayan et al., 2003]. From the author's point of view, the goal is quite similar to the storyline, since they are all flow of the game, but the difference is storyline is more like literature while the quest is concrete and clear. For example, 'the main character goes to a forest to save another important character', it can be part of the storyline and is abstract. And the actual quest to save the character can be: 'beat 30 enemies in a certain spot'.

7.Feedback

Players must receive appropriate feedback at appropriate times. During flow, concentration is possible because the task provides immediate feedback [Csikszentmihalyi, 1990]. Achievement can be seen as one kind of feedback (see figure 2.8), once a player reaches a certain achievement, some fancy animation may be played to inform the player, providing a strong sense of accomplishment.

8.Immersion

People play games to feel emotions that are not related to work, to calm down after a hard day or to escape from everyday worries [Lazzaro, 2004]. Games are often seen as a form

of escape from the real world or social norms, or as a way to do things that people otherwise lack the skills, resources, or social permission to do [Lazzaro, 2004].

Here is a review from the game *Nier:Automata* :

'I never really saw video games as an art. Video Games have defining art styles, music, atmosphere, themes, and so on...yet I never considered them as an artform, just something that is there to entertain me. NieR:Automata showed me the light. The atmosphere is truly something else. When it wants to, Nier makes you feel part of its universe, and then a moment later you are catching your breath and you feel something off. You then remember that you are playing a video game.'

[Retrieved from https://store.steampowered.com/app/524220/NieRAutomata/]

This review indicates that the player is completely absorbed in the game and gets a completely different experience from the real world.

Although immersion and concentration both refer to user's attention on the game, immersion differs from concentration in that it also highlights the different experience provided by the game from the real world.

9.Social interaction

To support social interaction, games should create opportunities for player competition, cooperation, and connection [Lazzaro, 2004; Pagulayan et al., 2003]. Game experiences should be structured to enhance player-to-player interaction and to create enjoyment in playing with others inside and outside the game [Lazzaro, 2004]. Take Dota2 for example, Dota2 is A Multiplayer online battle Arena(MOBA) game, in which one player co-operate with his four teammates to compete with another five players. In order to ensure good player experience, Steam established a report system (see Figure 2.9), players can report others for their bad behaviors, and those players get report frequently with be restrict from playing. Thus, social interaction is an important game element.

Figure 2.9 Dota2 report system [Screen shot from the game Dota2]

Based on the discussions above, and after manually reading around 1000 pieces of user reviews, the author founds reviews on Steam cover the following eight topics:

1)Storylines, 2) Graphics, 3) Soundtracks, 4) Characters, 5) Control, 6) Difficulty and challenges, 7) Interaction with other players, 8) Immersion.

Not all of the eight aspects for evaluating games are included when it comes to a concrete game. And because a lot reviews tend to describe 'the game' aspect, in order to distinguish more clearly, the author divides the first aspect into four sub-topics, which are storylines, graphics, soundtracks and characters. The number of topics within a certain period may vary, since each game has its own highlight. For example, review of Multiplayer online battle Arena(MOBA) games mainly focus on Interaction with other players. While Race games players care more about game control. And topics of adventure games are likely to be challenge related. Time is another feature that affects topics composition and numbers. Most games on Steam release updates frequently to add new elements, fix bugs or optimize performance so that the game can remain competitive with other similar products. With the change of game itself, game reviews will change consequently. In conclusion, the topic number is set to 10 at the beginning as a reference, further tests will be conducted in this thesis to figure out the accurate topic number for a certain game.

2.5 Make use of this large volume of game review data

This thesis proposes another way of generating game keywords(tags). Instead of letting users add tags arbitrarily (mentioned in section 2.2.2), tags will be generated from user reviews. Because based on a large amount of data, the results can more comprehensive and authorable. In this thesis, the author uses the word 'keyword' to refer to tags generated from reviews in order to distinguish from Steam official tags. Later in this thesis, there is a discussion on Steam official tags and keywords generated from the reviews.

Game reviews contain rich and detailed information about the players' personal experience and thought of a game. Analysing large quantity of user reviews can produce more comprehensive and convincing opinions compare to professional review, which will benefit game developers in designing new features and fixing bugs. Most importantly, understanding user reviews can also help to attract new players.

Most new players are attracted by the graphic, music, character or storyline of a game, and these elements are frequently mentioned in game reviews. Review help users to find the advantages of a game quickly without playing the game himself. For example, players would pay a lot of attention to the music of a game, as shown in Figure 2.10:

Figure 2.10 Review about game music [A screenshot available at: http://store.steampowered.com/]

Review contains user's emotion about the game as well, which are not subjective but can be persuasive in attracting new users. If the user can read enough number of reviews, he would be able to know all the advantages and disadvantages of the game without playing it. This why most application distribution platforms provide the function for users to review the products.

However, the quality of user reviews is difficult to verify, and the number of user reviews is enormous. For example, one popular on Steam platform called Dota2 has 52,922 pieces of reviews in the past 30 days and 655,328 pieces of reviews in total (shown in Steam game store web page in December 2016). Another problem is that the quality of user views varies widely, many reviews only contain user's emotion like 'awesome game' or 'I really like'. Some reviews might be too long for readers to find what the writer really wants to express. All these problems make it difficult to extract useful information for user review manually. In the next chapter, the author introduces the techniques which can be used to address the problems.

3. Topic mining

Text analysis, also known as text mining, is the process of extracting high-quality information from text. It has been a hot area in recent years with the development of Web 2.0. The traditional top-down way of content generation is gradually replaced by the bottom-up way [Cui, 2013], which means the more internet users, the more contents to be generated. With the help of text mining algorithms and techniques, much effort can be saved while making use of a large volume of data. Text analysis is quite a large area, including text information extraction, clustering, categorization, visualization, machine learning, data mining [Feldman and Sanger, 2007]. Topic mining, also called topic modelling, is a frequently used text mining technique for the discovery of hidden semantic structures in a text body.

Many studies have been done on mining social networking and mobile application reviews [Zhao and Zhang, 2012] [Hu and Liu, 2004], but only a few focuses on game products. Rather than using existing tool and techniques to analyse the user reviews in Steam, this thesis takes advantage of these studies to find the most suitable algorithms and solutions for mining topics from game reviews. Additionally, this thesis also focuses on the practical implementation rather than just a concept or a framework, since there is a lack of an application or system which provides a complete solution covering all steps of data collecting, data pre-processing and topic mining of game reviews

3.1 Keyword extraction

The keywords of a given document, also called as key terms, are a small group of representative words or phrases that can capture the primary topics of a document [Turney 2000]. The basic idea to extract keywords from a given document is the term frequency algorithm, which is to count the frequency of terms (words) that appear in the given document, words which appear with the highest frequency are the keywords. Steam uses the same frequency-based method to let users define games tags, in which game tags could also be considered as keywords of the game. However, simply counting the frequency may result in many noise words, for example, stop words like ‘and’ almost appear in every sentence, but will be meaningless if they are extracted as keywords.

Term Frequency Inverse Document frequency (TF-IDF) is an improved version of Term Frequency which solves the question mentioned above.TF-IDF works by calculating the

frequency of words in a specific document compared to the inverse frequency of that word appearing over the entire document corpus [Ramos, 2003]. The result of TF-IDF reveals how relevant a particular word is to a chosen document. According to the mechanism of TF-IDF, words that appear frequently in a single or a small set of documents tend to have higher TF-IDF results than those words that are common in the whole document set. This feature of TF-IDF almost solves the problem of meaningless keywords which mentioned at the beginning of this section.

The are some minor difference in implementing TF-IDF but the basic idea can be described as below:
Given a word *w,* a document set *D*, and an individual document $d \in D$, the equation of calculating the TF-IDF value of the frequency of word *w* over document *d* [Ramos, 2003]:

$$w_d = f_{w,d} * \log(|D|/f_{w,D}) \quad (2)$$

$f_{w,d}$ equals the number of times *w* appears in *d*, $|D|$ is the size of the document set, and $f_{w,D}$ stands for the number of documents which contains *w* in the document set *D* [Salton and Buckley, 1988] [Berger et al., 2000]. Given such a situation:

- The word 'DOTA2' appears 50 times in one piece of game review, this single piece of review is considered as a document), DOTA2 is the name of game on Steam.
- Out of total 1000 pieces of reviews for *Dota2*, 600 pieces of reviews contain the word 'DOTA2'

The TF value of the word 'DOTA2' in the chosen review is 50, and the corresponding TF-IDF value of 'DOTA2' is 50*log1000/600 (2) = 36.2, which is much smaller than 50. In TF-IDF, a word which appears throughout the whole document set has less probability to be the keyword. In the given situation, 'DOTA2' is the name of a game, which has no meaning in describing features of a game.

However, the disadvantage of TF-IDF is that: since TF-IDF is based on word frequency, which in other words is to count words, but it cannot deal with different words with the same meaning. Guzman and Maalej [2014] also proposes the same question in their research, they call it 'grouping of related features' [Guzman and Maalej, 2014]. Users

may use different words to describe one feature of a game. For example, the user may use UI, interface, user views and other words to describe the game interface, or use music, sound, soundtrack, etc to describe the game music. In this situation, TF-IDF may cause overlapping of keywords, and incorrect lower TF-IDF values. To solve this problem, a lot of previous research turned to topic modelling [Zhao and Zhang, 2012] [Gu and Kim ,2015].

3.2 Topic Mining

Topic mining, also known as topic modelling, is the process of applying statistical models for discovering the abstract 'topics' that appear in a large amount of text data. Topic mining of text has become a hot research topic in recent years. A lot of research has been done in the areas of application and improvement of existing algorithms. The most popular algorithms are Probabilistic Latent Semantic Analysis(PLSA) and Latent Dirichlet Allocation(LDA).

Probabilistic Latent Semantic Analysis (PLSA) is a technique from the category of topic models. PLSA was developed in 1999 by Thomas Hofmann [1999] and it was first used for text-based applications such as indexing, retrieval, clustering, etc. The LDA (Latent Dirichlet Allocation) is a variant of PLSA. It is an effective topic model for modelling huge document sets. PLSA is a frequentist statistical approach while LDA is based on the Bayesian approach. The key idea of both PLSA and LDA is to convert a text document to a vector containing words and frequencies, regardless of the order of words, and to represent the content of the vector as a mixture of topics [Zhao and Zhang, 2012]. Quite many recent approaches to mining document content are based upon the idea that the probability distribution over words in a document can be expressed as a mixture of topics, where each topic is a probability distribution over words [Blei et al., 2003] [Hofmann, 1999].

3.3 Related studies on user review analysis

Gu and Kim [2015] propose a framework called SUR-Miner that can be used to mine users' sentiments and opinions about different aspects of applications on Google Play. Instead of processing reviews as bags of words which is the method used by LDA, SUR-Miner makes full use of the monotonous structure and semantics of software user reviews, and directly parses aspect opinion pairs from review sentences based on pre-defined sentence patterns [Gu and Kim, 2015]. The SUR-Miner also analyses sentiments of each review sentence and associate sentiments with aspect-opinion pairs in the same sentence.

Finally, it summarizes aspects by clustering aspect-opinion pairs with the same aspects [Gu and Kim, 2015]. Gu and Kim [2015] aim at finding the parts of software which like by users such as the UI, so they propose a classification of app reviews, as shown in figure 3.1:

Category	Definition	Examples
Praise	Expressing emotions without specific reasons	Excellent! I love it! Amazing!
Aspect Evaluation	Expressing opinions for specific aspects	The UI is convenient. I like the prediction text.
Bug Report	Reporting bugs, glitches or problems	It always force closes when I click the ".com" button.
Feature Request	Suggestions or new feature requests	It would be better if I could give opinion on it. It's a pity it doesn't support Chinese. I wish there was a "deny" button.
Others	Other categories that are defined in [31]	I've been playing it for three years

Figure 3.1 Five categories for app user reviews [Gu and Kim, 2015]

The five categories are Praise, Aspect Evaluation, Bug report, Feature Request and the others. And the authors only focus on the Aspect Evaluation. In order to extract each feature(aspect) of the application from reviews, classification of reviews must be conducted first, so LDA is not suitable because it can only give general topics of all the review.

In the paper of Guzman and Maalej [2014], three steps are presented for review analysis: feature extraction, sentiment analysis and topic modelling, which are quite similar to Gu and Kim's. However, in step three, LDA is not used to extract topics but to classify keywords, they first use the collocation finding algorithm provided by the NLTK toolkit to extract each feature from the user reviews. Then continue to calculate sentiments of each feature. Lastly, instead of using processed reviews as the input of the LDA algorithm, which is normally used in LDA, they input the extracted features. In other words, they form topics by using the features they extract from the original user reviews, this does make the final topic more accurate on describing the reviews but also causes low efficiency because their feature extraction step overlaps with topic mining process (LDA) in selecting words which appear in high frequency.

Titov and McDonald [2008] analyse the shortage of LDA and PLSA in the situation of extracting fine-gained or multi-grained topics. Both LDA and PLSA methods use the bag-of-words representation of documents. The bag-of-words model simply represents a document which a bag full of words, ignoring the word's order and part of speech.

Therefore, LDA and PLSA can only extract topics in a general level of the document, but not good at extracting ratable topics [Titov and McDonald, 2008]. Ratable topics refer to topics that can be measured quantitatively. For example, in game reviews, ratable topics refer to keywords like soundtrack, storyline, character, etc. Users can say that the music of this game is better than another one. While other words like RPG, action, si-fi, etc are not ratable. User cannot say that RPG is good or bad, but can only express their emotions, e.g., 'I like RPG games'. Titov and McDonald [2008] aim to find topics that are related to ratable aspects, such as cleanliness and location for hotels, and they find that it is much more problematic with LDA or PLSA methods. However, in this thesis, in order to help the user to understand a game better, both ratable and unratable aspects need to be taken into consideration.

In conclusion, unnecessary steps such as classification of reviews and keywords are eliminated comparing to related studies. The main focus of this thesis is to implement a system, performance of the application needs to be taken into consideration. Thus, overlapping in the processing steps should be avoided, particularly in the situation that a large volume of review data need to be dealt with.

3.4 Latent Dirichlet Allocation

LDA (Latent Dirichlet Allocation) is a famous and efficient algorithm in the area of text topic mining, first introduced by Blei in 2003. LDA originates from Probabilistic Latent Semantic Analysis (PLSA), which is a technique from the category of topic models used for text-based applications such as indexing, retrieval, clustering, etc. Both LDA and PLSA use bag-of-words model but LDA presents lower perplexity and less overfitting [Blei et.al., 2003] [Minka and Lafferty, 2002]. According to previous research, PLSA has good performance on training documents that have been seen previously, while LDA is better at handling unfamiliar documents [Gimpel, 2006]. Furthermore, when evaluating mining results, the model of LDA has better consistency and will get results faster than PLSA [Kakkonen et.al., 2006]. Due to its high mining speed and relatively consistent results, LDA has been widely used in topic mining and text analysis. The LDA model is described in two parts. One is the process of generating documents using topics and words, or in other words, how the topic model represents documents. The other one is the actual problem that LDA aims to solve, i.e., Mining topics from a given set of documents.

3.4.1 Topic model of LDA

In the general topic model, a topic is represented by a set of words that are related to this topic. From a mathematical point of view, a topic is considered as a **probability distribution** over a set of words that can be used while talking about that topic. Same keywords may appear in multiple topics, but some words have a particularly high probability to appear in one topic than the others. These words are the keywords that can distinguish one topic from another. A topic is defined by the keyword that appears in it with high probability. Table 3.1 shows topics that are generated by LDA based on around 3,000 pieces of reviews from the game *Nier:Automata*.

Topic 1	Fps, Issues, Resolution, Settings, Port, Fullscreen, Gtx, Cutscenes, Mode, Graphics
Topic 2	Story, Gameplay, Experience, Characters, Combat, Soundtrack, Platinum, Music, Unique, Playing
Topic 3	Story, Combat, Lot, Hard, Bit, Quests, Enemies, Pretty, Character, Feel
Topic 4	Gods, Androids, Machines, Robots, Humans, Android, Earth, Human, Plot, Yorha

Table 3.1. Topics consist of words

The process of generating documents can be explained as below. [Blei, et.al. 2003]

- Pick one topic from a given probability distribution over possible topics. For example, if a student wants to write a dairy, he would think about what to write about first. If he decides to write about traveling, then ‘travelling’ can be considered as a topic.
- Pick one word from the word-distribution of the selected topic and add the word to the document. For example, travelling is an abstract concept. If the student wants to make his dairy concrete, he would use some concrete words like Paris, airplane, Seine river, Eiffel tower, etc)

It is obvious that one document can contain many topics, in this case, a document can be generated by repeating the process above. The probability that a document contains a certain word can be represented as conditional probability below:

$$\mathrm{p}(\,\mathrm{word}\mid \mathrm{document}\,) = \sum_{\mathrm{topic}_i}^{\mathrm{i=1\ to\ n}} \mathrm{p}(\,\mathrm{word}\mid \mathrm{topic}_i)\ \mathrm{p}(\mathrm{topic}_i \mid \mathrm{document}\,)$$

Formula 3.1[Blei et.al., 2003]

In this formula topic is the i:th topic in a set of n available topics. The formula also can be represented in matrix form as Figure 3.2.

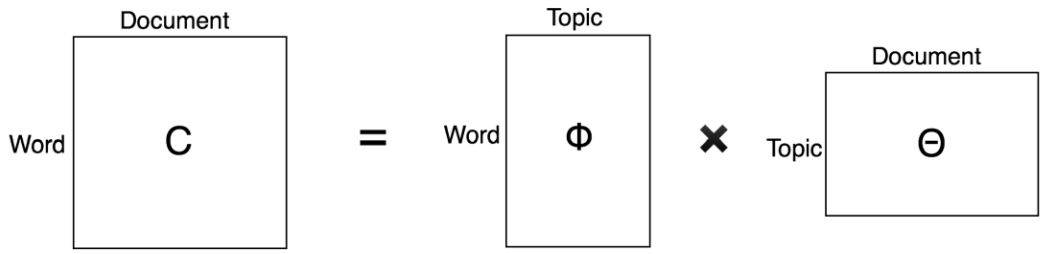

Figure 3.2. The matrix expression of topic model [Blei et.al. 2003]

The Document-Word matrix on the left side of the formula represents the probability of each word appearing in each document. The Topic-Word matrix represents the probability distribution of each word appearing in each topic. The Document-Topic matrix represents the probability distribution of each topic emerging in each document.

Given a set of documents, the Document-Word probability distribution on the left side of the formula can be calculated by word segmentation and counting frequency of each word. Based on this, topic model can calculate the Topic-Word probability distribution and the Document-Topic on the left side of the formula.

Blei et al. [2003] introduce the following process for generating each document **w** in a document set D:

- Choose random variable $\theta \sim$ Dirichlet distribution p is a topic vector $(\theta|\alpha)$; θ is a vector of topic probabilities for the document. α is a vector parameter for $p(\theta)$.

- For each word w_n in the document, where n ranges from 1 to N, N is the total number of words in the document:

a) Choose a topic z_n which subjects to Multinomial distribution $p(z | \theta)$; θ is the parameter of $p(z | \theta)$.

b) Choose a word w_n from $p(w_n | z_n, \beta)$, a multinomial probability conditional on the topic z_n. β is a matrix parameter for $p(w | z)$, which represents word probability distribution in every topic.

α is a parameter of the Dirichlet distribution (prior over topic proportions). β is another parameter of the multinomial word choice distribution.

Therefore, the joint probability of the whole process of LDA can be represented as below:

$$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta)\, p(w_n \mid z_n, \beta)$$

Formula 3.2 The joint probability of LDA [Blei et.al. 2003]

Figure 3.3 explains the formula more directly:

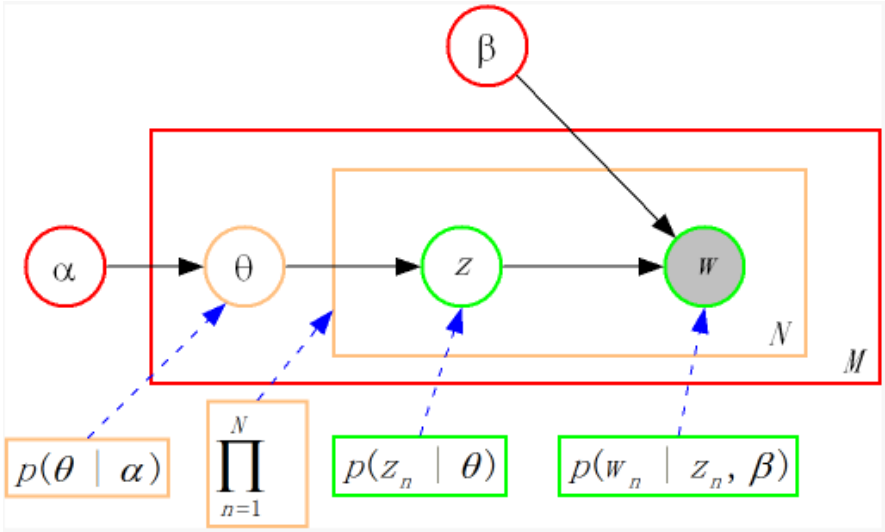

Figure 3.3 Comparison of LDA process graph and joint probability [Huagong, 2012]

The three layers of LDA are represented as above:

- Document set layer (surrounded by red block in figure 3.3): α and β are document set-level parameters which are set to constant values throughout the generating process.
- Document layer (orange block): θ is a document-level parameter. Each of document *w* has its θ, this is because the Document-Topic probability is different for each document
- Word layer (green block): *z* and *w* are word-level parameters. *z* is generated from θ. *w* is generated from the combination of *z* and β.

The topic model is illustrated in the example of using words and topics to generate documents. In next section, how LDA is used in mining topics are explained in detail

3.4.2 Mining topics from documents using LDA

The algorithm for detecting topics is the reverse process of generating documents. When

an individual would like to acquire useful opinions from a large amount of information, LDA can help to mine the topics in a large number of reviews. The input of LDA are documents. Other key inputs for LDA are α and β which mentioned in the last section, α is a vector parameter for p(θ) which generates topic vectors; β is a matrix parameter which represents word probability distribution in each topic.

In practice, the variational inference (E-M) algorithm [Blei, Ng and Jordan, 2003] and Gibbs sampling [Steyvers and Griffths, 2006] are common methods to figure out α and β by training the given document set. Griffiths and Steyvers [2004] suggest a value of 50/k for α and 0.1 for β, where K is the number of topics that contained in the given document set.

The outputs of LDA algorithm are the topics and related keywords generated from selected game reviews. The number of topics is decided by the input of the LDA algorithm. However, the number topics within a certain document is uncertain from the very beginning. In order to reveal the appropriate 'latent' topics hidden in the documents, one problem need to be solved. For example, if the author tries to mine 20 topics from a set of review documents that contains only 10 latent topics, the possible result may be that 10 out of the 20 mined topics are invalid, either meaningless or overlapping with the rest 10 topics. Thus, the number of latent topics needs to be figured out.

Zhang [2017] used LDA for mining twitter news, and part of his work focused on finding the number of latent topics within a given set of reviews. According to previous research, the topic number can also be decided based on some transcendental experiences. Zhang [2017] tries to extract topic from 20 categories of tweets from Twitter. The categories include education, economy, military, sports, politics, etc, which can be regarded as topics in the tweets. So in Zhang's research, 20 is used as the reference for topic number. Steam does not provide any pre-defined topics from which developers can collect reviews, so the author uses the eight topics proposed in Section 2.4 as a reference. In the test and evaluation chapter, the author will explain how to assign the topic number of Steam game review.

4. Design of the system

The design of the system is mainly based on the general process of data mining proposed by Indarto [2013], which includes six steps. The first step is data selection, normally data is retrieved from the database, buy different data sources can be used such as text file or data stream from the Internet. The second step is data cleaning and preprocessing, in which noise or inconsistent data is removed. Additionally, missing data fields are handled; time sequence information is handled. The third step is data transformation. In this step, data are transformed and consolidated into forms appropriate for mining. The next step in Data mining, which is the most important and difficult step. Intelligent methods are applied to extract data patterns in data mining step. The fifth step is pattern identification. Patterns that represents knowledge are identified. The last step is knowledge presentation in which the visualization of mined knowledge is represented to users [Indarto, 2013].

These six steps are universal for most of data mining processes. Different applications of data mining may contain all or part of the steps according to practical need. According to the situation of this thesis, the system is proposed to have four main functions:1) data collecting, 2) data storage 3) data processing 4) user interacting. The first and second function generate from the data selection step, the third function combines the data cleaning and preprocessing and data mining steps, and the four function implements knowledge presentation. In the next sections, the author will introduce each function in detail.

4.1 System functions

The **data collecting** function only focuses on collecting reviews. Steam provides social network function, which allows players to share their game experience under the homepage of each game. After a review is posted, other players can comment and write their own ideas as well. As mentioned before, a large number of game reviews are generated every day, in thesis , the author intends to collect around 10,0000 pieces of game reviews for experimental need. Additionally, since Steam doesn’t provide and API(Application programme interface) for developers to get game reviews directly, the only way to get games reviews is to crawl the homepage of each game and extract the element of interest .There are many elements on the game home page, but the author only aims at collecting the review and the date of posting review.

Data storage not only covers review storage but also contains storage of user input and system output. Crawling is a time-consuming process, because the crawler needs to go through the whole webpage to find the element of interest, and the game reviews are generated in quite a high speed, these require that data must be pre-collected and stored somewhere for future use. Meanwhile, original game reviews may contain a lot of noises which makes them cannot be used directly used for data mining. Thus, the pre-processed reviews also need to be stored. Moreover, the final topic mining results need to be stored both for result visualization and future analysis. In this case, data storage is involved throughout the whole system executing process, i.e., the data storage function will interact with all the other functional modules.

In order to improve the accuracy and efficiency of the **data processing**, reviews are pre-processed. The pre-processing of reviews includes two steps: word segmentation and stop-word removal.

Word segmentation in continuous texts is a fundamental problem in Natural Language Processing (NLP) [Chen, Xu and Chang, 2011]. In word segmentation, sentences are split into words because the expected system output are topics which consist of words

Stop-words are those words that appear extremely often in sentences but are of little value in data mining. An example of common English stop words is shown in figure 4.1. These words are normally used to connect sentences and indicate tense of the sentence.

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

Figure 4.1 An example list of 25 common English stop-words [Manning et.al., 2008]

In addition to stop words, other meaningless words also need to be removed. For example, the word ’game’ can appear almost in every review but contribute nothing to a concrete game.

The second step is to choose specific topic modelling algorithm and train the input from the previous step to get topics. These steps might need to be repeated several times to get optimized results. Accordingly, the system must have the ability to repeat the training process with different inputs and store the inputs and outputs in the database for future analysis.

The **user interface** should not only cover the result visualization but also smooth the whole process of system executing. The user interface should provide following functions:
1.Display existing review categories in the system database, a game is considered as a category. One execution of a system task is based on one certain game, so the system should be able to show users with the existing games in the system.
2.Guide users to operate the system and get user input properly.
3.Visualize the results. The output of data mining can be a dozen of topic contain hundreds of keywords, so suitable layout is needed to clearly present the result, e.g., charts and tables.
4.Track and display historical user operations. Users need to analyse historical data to optimize the algorithms, so user operation should also be recorded and can be displayed when necessary.

4.2 System architecture

This chapter illustrates the architectural design of the system. As shown in Figure 4.2, the system has five layers. The hardware layer provides the running environment of the system. The database layer implements the requirement for storing review data and the results of mining reviews. The data persistence layer serves as an interface for the business layer to access the database. Create, read, update, and delete (CRUD) of data are the four basic functions of the persistence layer[Heller,2007].The business layer is responsible for processing data according to user instructions and generating the results. Finally, the presentation layer gets instructions from users and passes them to the business layer. Meanwhile, it gets results from business layer and presents them to users.

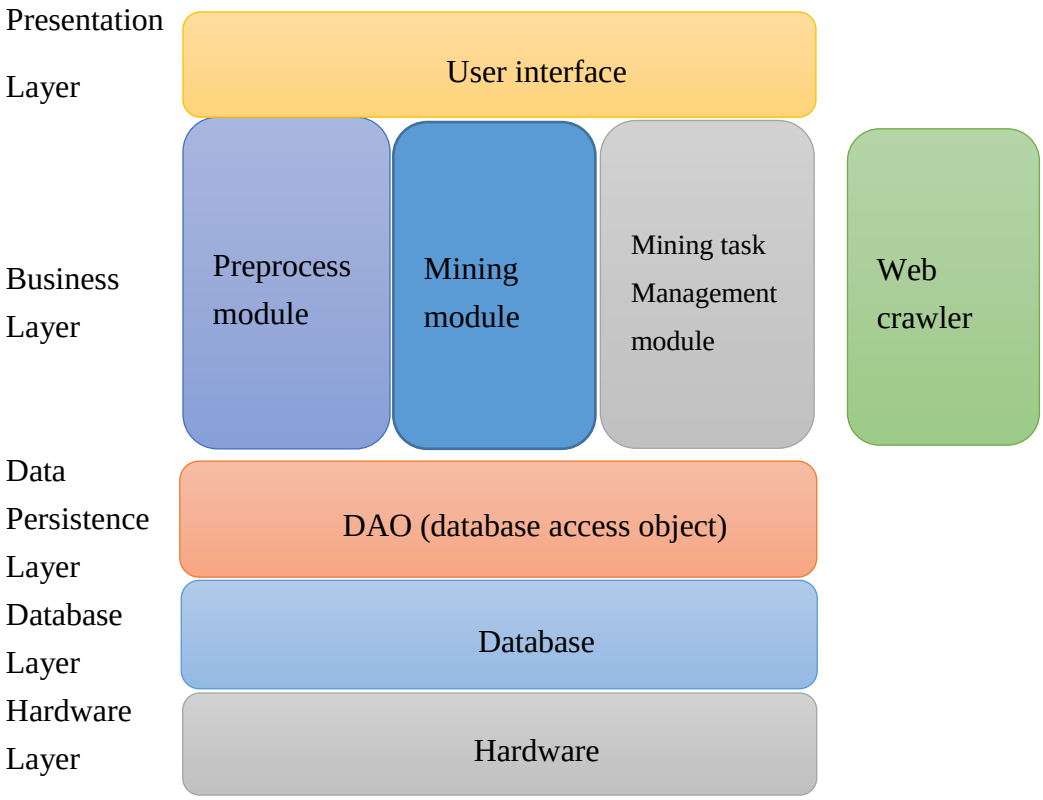

Figure 4.2 System modules and layers

In the business layer, there are four functional modules: web crawler, pre-processing module, topic mining module and the Mining task management module. The web crawler is implemented outside the system as a sub-system, because crawling data from web pages is time-consuming (mentioned in section 3.1), data need to be pre-collected and store into the database. The pre-processing module first gets user instructions and loads original reviews from the database according to these instructions, then processes the selected reviews and stores them into the database. The topic mining module gets pre-processed reviews from the database and instructions from the user to generated results. The mining task management module is responsible for creating, modifying and deleting mining tasks. User's operation of executing mining tasks are tracked as well. Because pre-processing original reviews is time-consuming, the mining task management module also supports the re-execution of mining tasks with different user inputs to make use of the pre-processed reviews.

4.3 The Flow of system execution

The flow of executing a mining task is shown in figure 4.3.

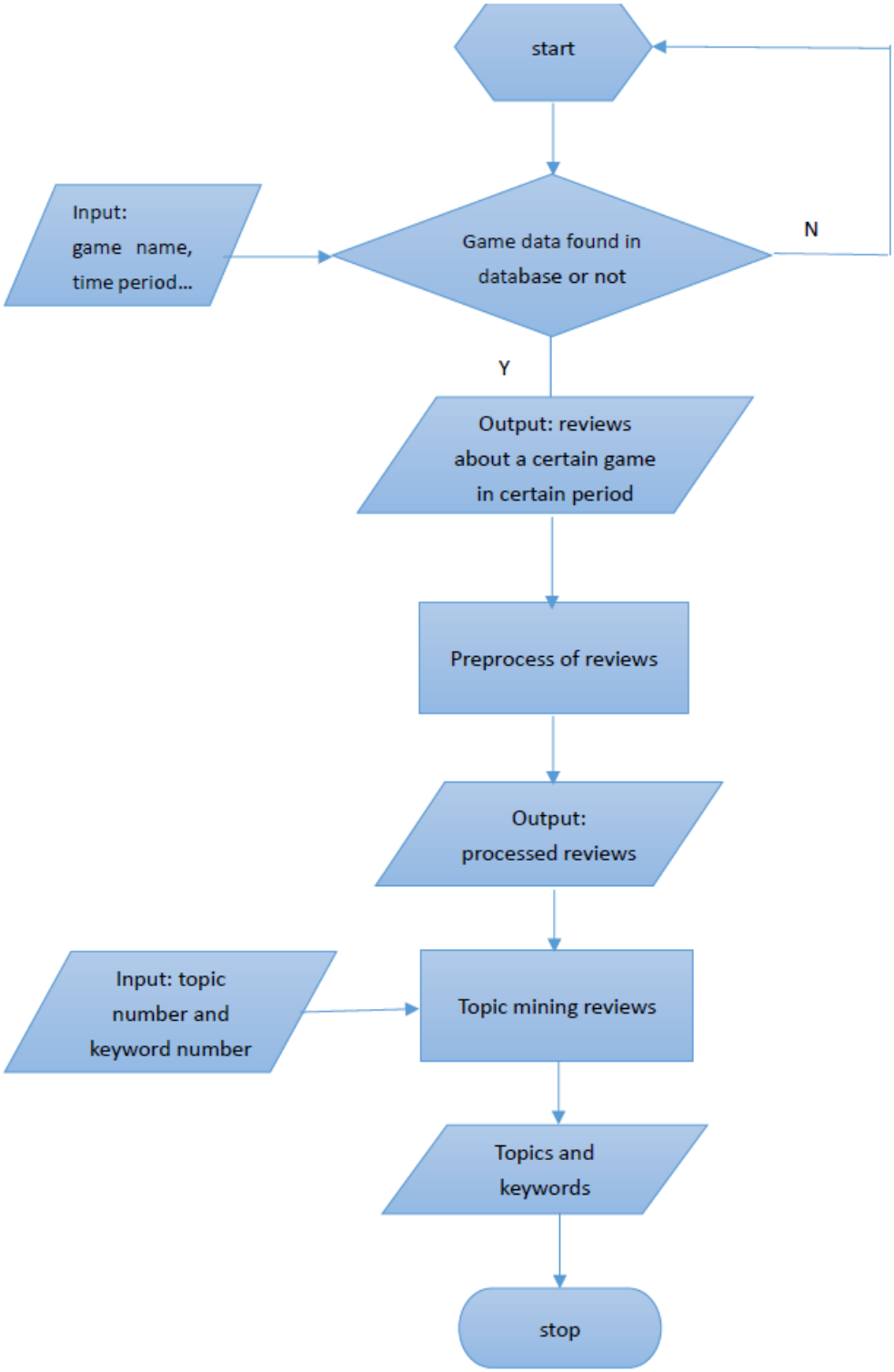

Figure 4.3 Flow chart of system execution

At first, each review is stored in the database with the date of posting the review and the game name of the review. When creating a mining task, the user needs to select a certain game and certain time period to make the mining task specific.

After the mining task creation is done, the pre-process module will load reviews from the database and execute the pre-process, if no reviews are found in the select time period and game type, the system will return to the task creation page to ask the user to re-select time period or game name.

After pre-processing is done, the system will turn to the page for the user to enter topic and keyword numbers. These two numbers together with the pre-processed reviews are the input of topic mining algorithms. If the topic number is i and the keyword number is j, the topic mining module will generate i topics, where each topic contains j keywords which can be used to describe the topic. For example, if the topic is about game control, then the probable keywords are Mouse, Keyboard, Camera, Control, Button, etc.

5. Implementation of the system

5.1 Spring framework

The system is implemented as a web application using the Spring framework [Spring, 2010]. Users interact with the system via web pages, data processing and storage are running on the server side.

The Spring Framework is an open source framework for build applications on Java platform. The framework's core features can be used by any Java application, but there are also extensions for building web applications on top of the Java EE(Enterprise Edition) platform [JavaEE, 2018]. Although the framework does not impose any specific programming model, it has become quite popular in the Java community.

The Spring Framework provides its own model–view–controller(MVC) web application framework, which is called Spring MVC. Before introducing Spring MVC, the author will first briefly introduce MVC.

MVC is a design pattern commonly used for developing web application. It divides the system into three independent but inner connected parts. Main components of the system are decoupled in the MVC design pattern that allows for efficient code reuse and parallel development. The three components are :1) Controller which handles navigation business logic, 2) Model which regulates the structure of data which transferred between Controller and View, 3) View which receives inputs from users and visualize output data [Gupta and Govil, 2010].

Spring MVC follows the principals of the MVC, which is the system should consist of three core collaborating components. Besides, Spring has been designed more for the desktop and internet-based applications [Gupta and Govil, 2010]. Spring MVC is a request-based framework. The framework defines business logic interfaces for all of the responsibilities that must be handled by a modern request-based framework. The goal of each interface is to be simple and clear so that it's easy for Spring MVC users to write their own implementations.

5.2 Web crawler

Python is one of the most popular programming languages in use, which provides numerous libraries for data mining, distributed computing and many other hot areas. Python also provides strong functions for web crawling. The package used in this thesis is called Beautiful Soup [2017], which is a Python library for pulling data out of HTML and XML files. A lot of different elements may appear in one web page but only one or two are the elements of interest.

Figure 5.1 HTML code of Steam web page

Figure 5.1 displays part of the elements on Steam web, the elements that the author needs to collect is inside the blue block, which are the review content and time of posting. Web page elements are normally organized in a tree structure, Beautiful Soup is good at searching and modifying this tree structure and locate the element of interest quickly. It commonly saves programmers hours or days of work.

After execution of the web crawler program, game reviews are stored in JSON format files. JavaScript Object Notation (JSON) is an open-standard file format that uses human-readable text to transmit data objects consisting of key-value pairs and array data types(or any other serializable value). As shown in Figure 5.2, there are two kinds of key-value pairs in the JSON file: review content pair and review date pair. The reason to choose is

that JSON is a language-independent data format. The is to say, even if the JSON file is generated by python, it can still be read and processed by other programming languages like Java or C++.


```
"steam": [
    {
        "review_contents": "1, This game is a masterpiece. Yes, it's imperfect, and has defa
        "review_date": "June 11, 2017"
    },
    {
        "review_contents": "8, I don't do reviews; probably going to be my only steam review
        "review_date": "March 28, 2017"
    },
    {
        "review_contents": "2, Where to even start with this game? I just finished 100%'ing
        "review_date": "March 22, 2017"
    },
    {
        "review_contents": "2, We need more Nier. Prequels and sequels. Anything related!Eve
        "review_date": "November 22, 2017"
    },
    {
        "review_contents": "1, You can dodge the bullets, but you can't dodge the tears.",
        "review_date": "March 21, 2017"
    },
```


Figure 5.2 example of game review in JSON files

Review content and the date of posting are crawled from the game home page, the data is important because most games will keep updating after their first release, analysing old reviews may result in inaccuracy of mining game content.

5.3 Data storage

Data storage is needed in all the modules of the system including crawling process, processing process, mining task management and visualization. According to previous research of this thesis, a JSON file which contains 10000 reviews reaches 8Mbs in size, which takes more than one minute for the system to parse. Reviews are frequently read and written during the whole topic mining progress.

MySQL is an open-source relational database management system which is used in applications for data storage. MySQL is also used in many high-profile, large-scale websites, including Google, Facebook, Twitter, YouTube, etc [Oracle,2012]. MySQL could complete query in million level within seconds. In this thesis, reading and writing of reviews will complete in less than 0.5 seconds, which significantly improves the system performance user experience.

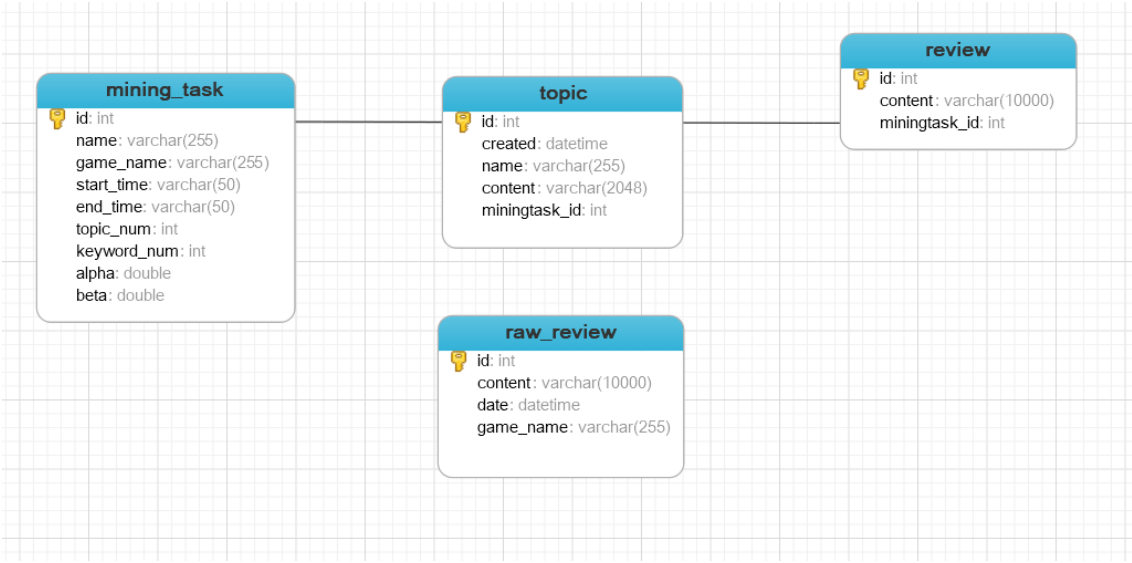

Figure 5.3 E-R diagram of the database

The entity-relationship(ER) diagram design of the MySQL database is shown in Figure 5.3. A block represents an entity. There are four entities, and they are the mining task entity, the topic entity, the review entity and the raw review entity. The entities are implemented with tables in the database.

The raw review table stores all the reviews collected from Steam by the python web crawler. Each review contains the ‘date’ and ‘game_name’ fields, these fields are used to filter the review of a certain game within a certain period.

The review table stores the pre-processed reviews. In the review table, each review is related to one mining task with ‘miningtask_id’. ‘Mining task’ table store the information about each mining task including the mining object(a certain game), time period, and parameters of the LDA algorithm.

The topic table stores topics created by mining tasks based on a certain set of game reviews. Each topic contains several key-values pairs, key refers to the keyword, and value reveals the frequency of the corresponding keyword.

Each table has its own Java Object implemented as shown in figure 5.4,

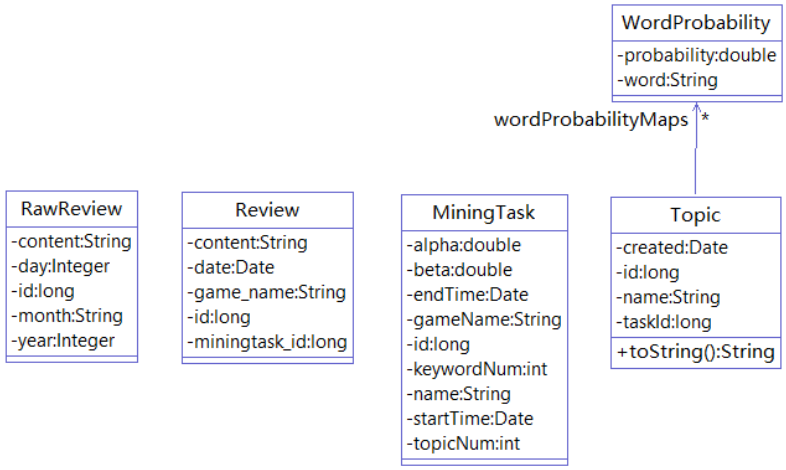

Figure 5.4 class diagram of models

After the data are read into the system from database, they are stored in different Java Objects. Each table has a corresponding class which shares the same attributes with it, so that important data will not be missing. Different modules communicate with each other via Java Objects

5.4 Mining task management

The mining task management module is responsible for creating, editing and displaying mining tasks.

The class diagram of this part is listed as below:

Figure 5.5 class diagram of mining task management module

The overall execution process of this module is that the Controller gets requests from the front end, resolves the request and invokes the corresponding Service. The Service will furtherly resolve the request and decide which DAO (data access object) to call. DAO will access the database, get and manipulate the data, and return the results to Controller. Lastly, the Controller will send a response containing data to the front end to visualize the result.

For example, if the Mining Task Controller received the request to display the detailed information of each mining task, then the function *findTaskById()a* will be invoked, it takes the *Id* of mining task as parameter and send a request to Mining Task Service , Mining Task Service will invoke its *findTaskById()* function ,it will send request to Mining Task Dao, after that, The function *findTaskById()* will access the database for the data of the task according to the *Id* parameter. As long as the data is received, Mining Task Dao will return it to the Controller, until now, the request to display task information is done.

5.5 Pre-processing

The pre-processing module mainly focuses on the review reading process.

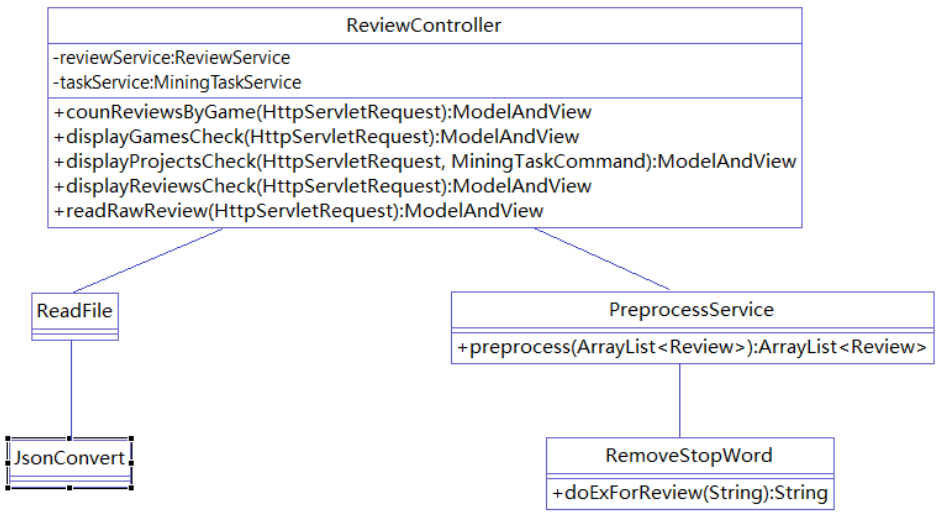

Figure 5.6 Pre-process module

As shown in figure 5.6, the Review Controller will first need to get reviews, which is done by JsonConvert and ReadFile classes. In this process, reviews contained in the

JSON file are transferred into the format which database can understand. Missing data fields are handled as well (as mentioned in Chapter 4).

After the reading process is done, the Review Controller will the invoke the Pre-process Service, then the real pre-process will be carried on. In the pre-processing, review sentences are split into single words, then the stop words will be removed. The author maintains a stop word table with contains most of the command English stop words. The stop word table is updated continuously according to the final mining results. Meaningless keywords in the final mining result will be added to the stop-word table, then the mining task will be re-executed to get more precise results. For example, words like Valve, Steam, Game, etc appear frequently in reviews though they are not common English stop words.

5.6 Topic Mining

The topic mining module is much more complicated compared to other modules, but the execution flow remains the same. In this module, after receiving a request from the front end, the controller needs send requests to three different services and wait for their responses.

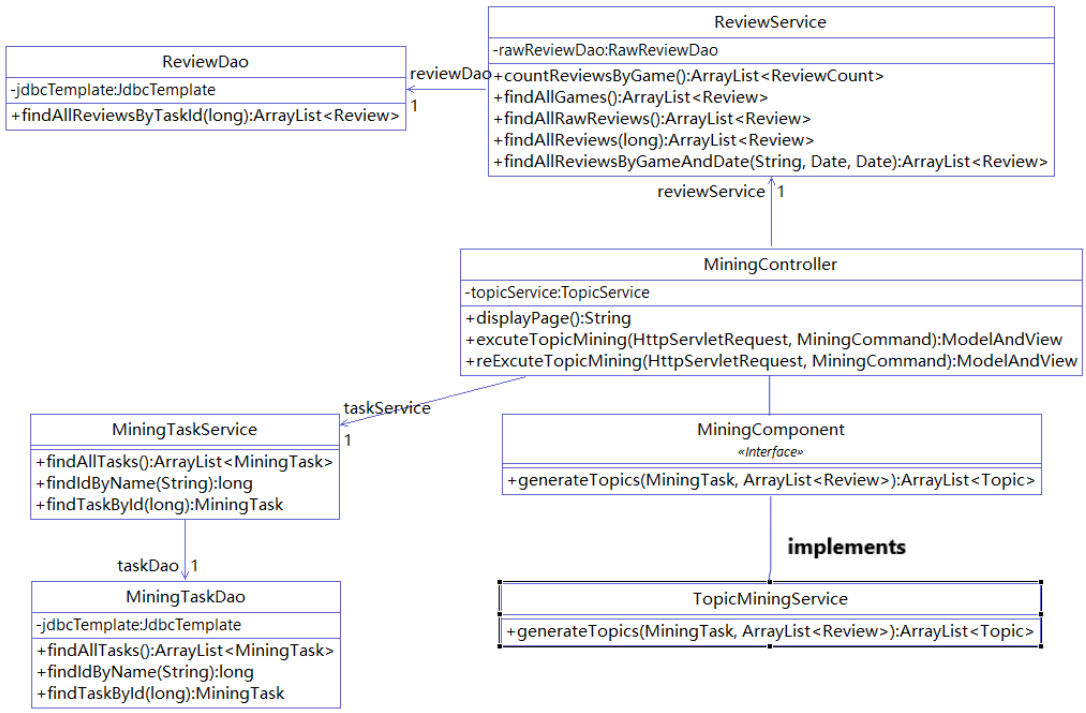

Figure 5.7 Class diagram of the mining module

The three services, i.e., the Mining Task Service, Review Service and Topic Mining Service are shown in Figure 5.7. The Mining Task service give instructions on how to execute the mining task, which include target game name, the time period for selecting reviews, parameters of LDA, etc. The Review service provides the pre-processed reviews according to instructions given by Mining Task Service, which are the fundamental data resource for topic mining. The topic Mining Service provides the final result of the whole system, i.e., topics and keywords.

Currently, only the LDA algorithm is integrated into the system for the topic mining process. However, with the continuous change of the system requirements and rapid progress in optimizing algorithms, the system should support the integration of new algorithms. This system uses interface design for the topic mining module in order to improve the expandability of the system.

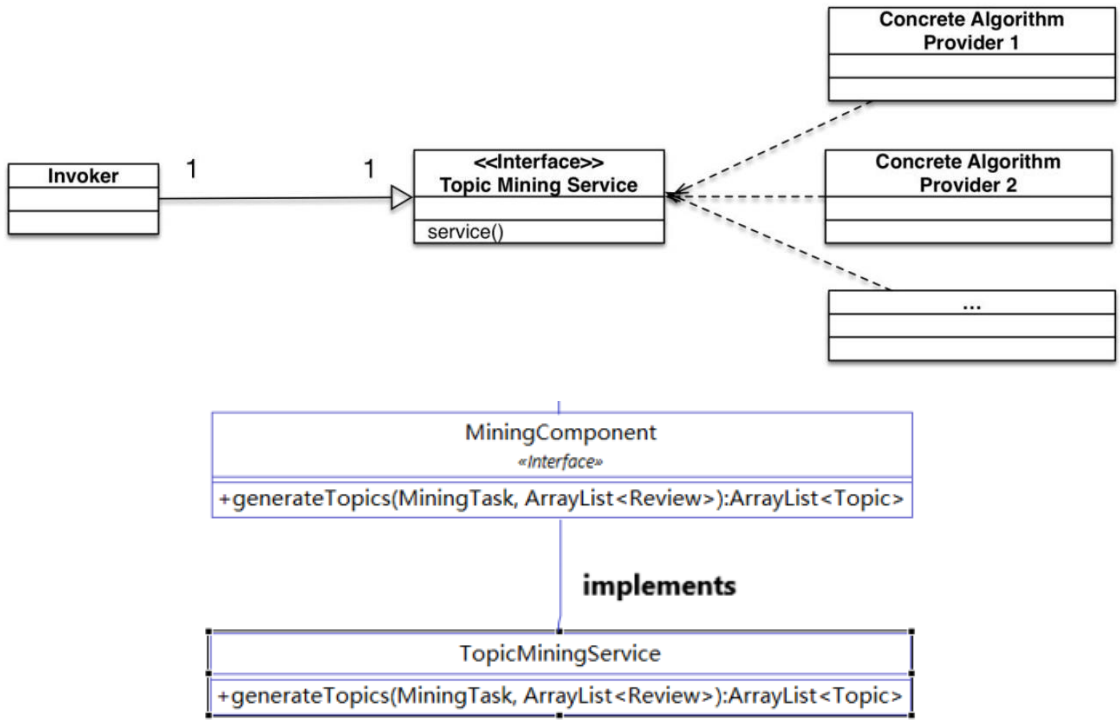

Figure 5.8 Interface design to support expandability of algorithms

As shown in Figure 5.8, since the Topic Mining Service needs to take reviews and parameters from Mining Tasks Service as input, and output the Topics, an interface called Mining Component is designed to take the same input and provide the same output. Then Topic mining service inherits the Mining Component and implements the detailed method of mining topics. Even if another algorithm is implemented for Topic mining

service, the Mining Component interface remains unchanged, thus other modules of the system will not need to change to adapt to the changes in Topic Mining Service, which decouples the Mining Task Service with the rest part of the system.

In this thesis, Mallet [2002] package is used to implement the LDA algorithm. Mallet is an open source Java-based package for natural language processing. It is developed by the University of Massachusetts. This package includes implementation of algorithms covering text classification, clustering, topic modeling, feature extraction, etc. The Mallet topic modeling also contains efficient implementations of LDA, Pachinko Allocation, and Hierarchical LDA [McCallum and Kachites, 2002]. Implementation of LDA in Mallet package takes processed reviews, α, β topic number and keyword number as inputs and output topics and keywords, each topic contains several keywords, each keyword has a corresponding number which indicates the probability of appearing in the topic.

5.7 User interface and Visualization

The system is first design to solve the questions proposed in this thesis work, however, during the experiment, training of model and algorithm is found to be quite a time-consuming process. As mentioned in section 4.4, the system provides an interface which supports the integration of new algorithms. In this case, a user-friendly interface is needed so that users can manipulate smooth event without and previous experience. The user interface is implemented using the bootstrap framework.

Bootstrap is a free and open-source front-end library for designing websites and web applications. It is the second most-starred project on GitHub, with more than 121,000 stars.［Search stars.*GitHub*. Retrieved February 13, 2018.］ Bootstrap contains many templates based on HTML and CSS for web front-end design. It this thesis, Bootstrap is mainly used for navigations, buttons, tables and charts.

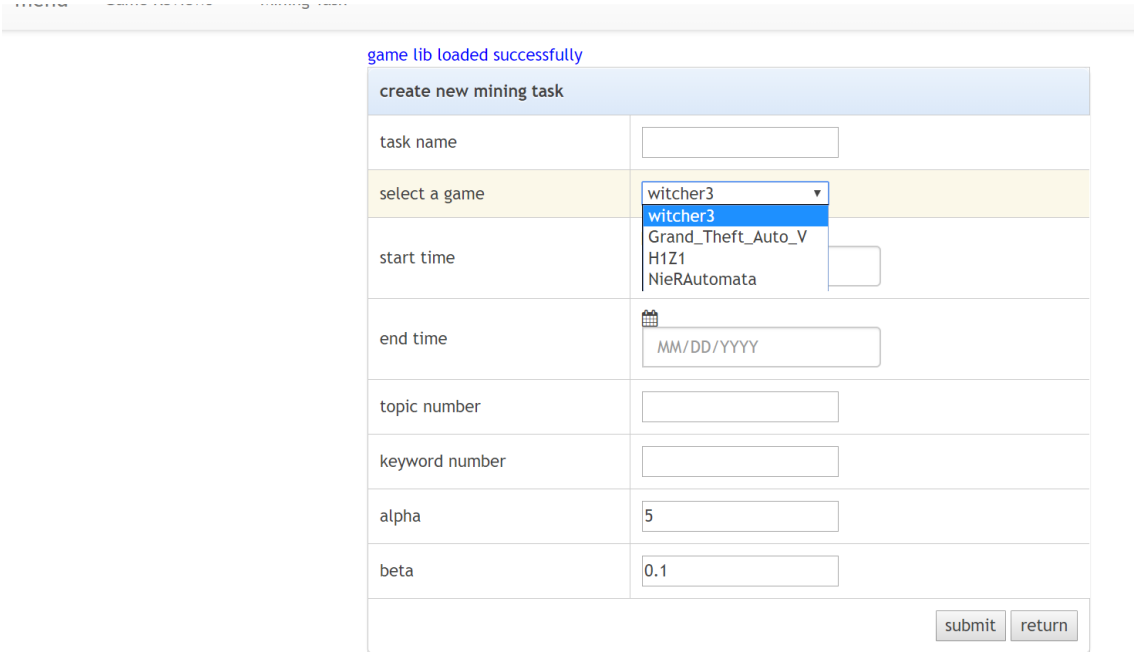

Figure 5.9 Mining page of the system

First of all, the author would like to introduce the mining task creating module. From the web page shown in Figure 5.9, users can start his mining work. Before loading the page, the system will check how many different game reviews are stored in data. Then users can generate the data set (chose a game and select a time period for the reviews) and set parameters for the algorithm. Afterwards, system will first start the pre-processing module to clean and format the original review data and then the real processing module. Lastly, topics containing keywords will be generated and displayed on the web page.

Another core part of the system is the Mining task management module, the user interface of this part is shown in Figure 5.10

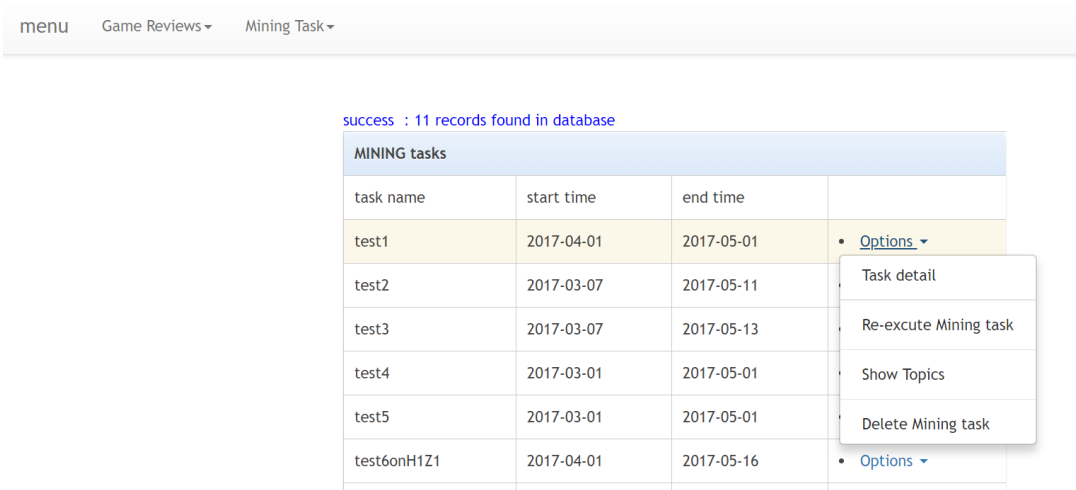

Figure 5.10 The Mining task management module

In the mining task management page, each task that executed is listed here, by clicking the option button of each task, a drop-down menu will be displayed:

By clicking the ‘Task detail’ button, details of the task will be displayed, which include the time of executing the task, game name, parameters of the LDA algorithm , etc.

The ‘Re-execute Mining task’ button allows users to change the parameters of LDA algorithm in order to train the selected data. By clicking the ‘show topics’ button, topics generated from this certain mining task will be displayed. Each topic contains several keywords and the probability value of the corresponding keywords to appear in a certain topic.

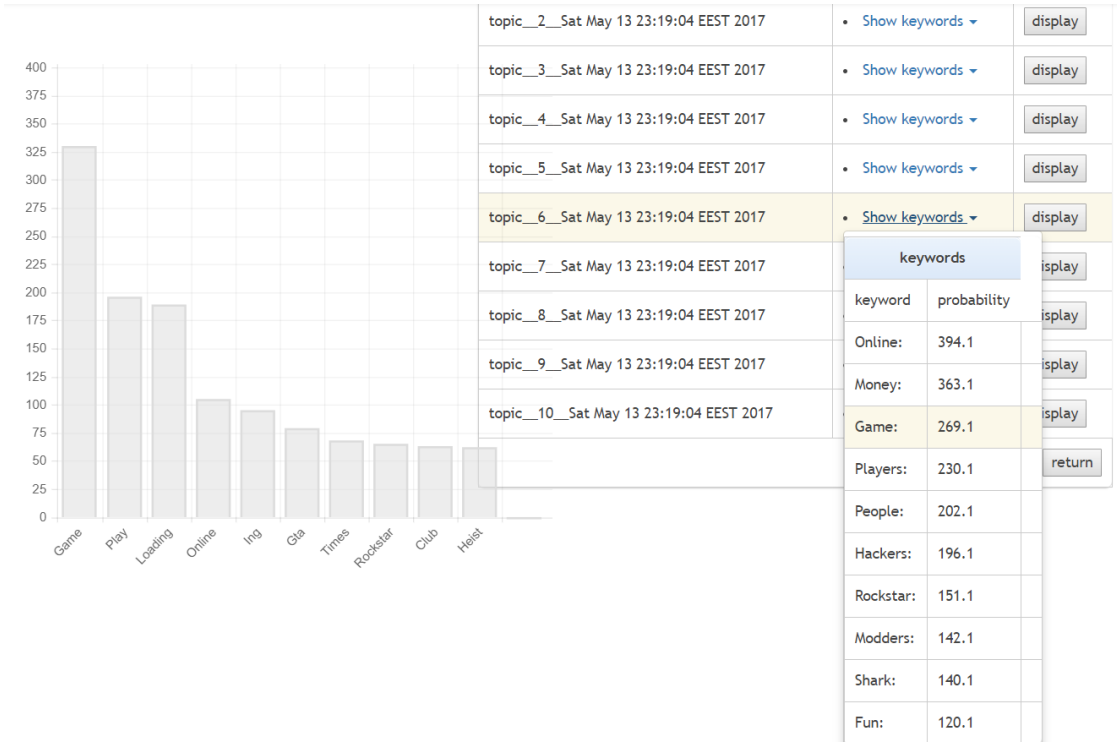

Figure 5.11 mining result visualization

So far, the keywords of a game are generating, and are simply classified so that similar words together generate topics. Meanwhile, each keyword has its corresponding probability, by eliminating keywords of low probability, it will be able to remove noise words and define the topics more precisely

6. Test and evaluation

This chapter describes the testing process and analyses the results. The test aims at calculating the running time, preciseness and accuracy of the system top prove that the system can run smoothly in real world situation and the LDA algorithm can mine valid topics from reviews. Two types of tests are included: performance tests of the system and topic related tests of LDA. Although the aim of the system is not much the same as the system implemented in related studies, common processes like data collecting, pre-processing, storage are included. Data collecting, storage, pre-processing and visualization are implemented with self-designed programmes based on existing technics, the data mining requirement is implemented by integrating an open source Java package called Mallet [2002], which provides an efficient implementation of topic mining algorithms. The integration of existing algorithm implement allows more effort to be spent on system level design and optimization.

Additionally, more focus is put on optimizing the parameters of the topic mining algorithm, which is the main purpose of topic related tests. Different existing algorithms for keywords extraction and topic mining like TF-IDF, PLSA, LDA, etc are evaluated to find out which fit the requirements of the system best. Eventually, LDA is chosen for topic mining because its high mining efficiency. Tests are carried out based on a large amount of review data. LDA is a probability model, so enough test data are needed to make the model relative adapted to the game review analysis.

The system is developed and tested on the author's personal laptop, the detailed hardware environment and software versions is listed as below:

CPU	Intel(R) Core(TM) m3-6Y30 CPU @0.90 GHz 1.50GHz
Memory	8 GB
Graphics Card	Intel(R) Graphics 515
Hard disk	256GB SSD

Table 6.1 Hardware environment of the system

Operating system	Window 10 Home 64-bit
Run-time environment	Java 1.8.0_121 Python 2.7.13 64-bit
Web server	Tomcat 8.5
Database	MySQL 5.7 Navicat for MySQL 11.0.9
Developing framework	Spring 3.0.5

Table 6.2 Software environment of the system

6.1 Performance test

Over 100,000 pieces of review from five different games (*NieR:Automata* [2018], *The Witcher3: Wild Hunt* [2018], *Grand Theft Auto V* [2018], *Player unknown's battle ground* [2018], *Dota2* [2018], *H1Z1* [2018]) are crawled form Steam community website, and stored in the database of the system and processed to mine latent topics and related keywords. All these five games are the most famous game within their types, i.e., RPG, racing, FPS (first-person shooting) and MOBA, which cover the main game types on Steam. And these games all have a lot of reviews written by users every day. For example, as shown in Figure 6.1, around 300 pieces of review are written every day in the game *Dota2*.

Figure 6.1 number of review written every day [A screenshot available at: https://store.steampowered.com/app/570/Dota_2/]

The performance of the system is evaluated by executing different scales of review data, monitoring the system run-time situation and recording the running time of the data collecting, reading, pre-processing and processing steps. Eight different review datasets

are used in the performance test. Each dataset contains from 1000 to 50,000 pieces of game reviews, and the review are all from the game *Nier:Automata* in the recent half year Table 6.3 shows the time used for crawling reviews from the Steam website.

Review scale	100	1000	2000	3000	5000	10000	20000	50000
Total time(sec)	65	420	631	783	1508	2824	5187	12658
Average time(sec)	0.65	0.42	0.315	0.26	0.31	0.282	0.259	0.253

Table 6.3 crawl time of different numbers of reviews

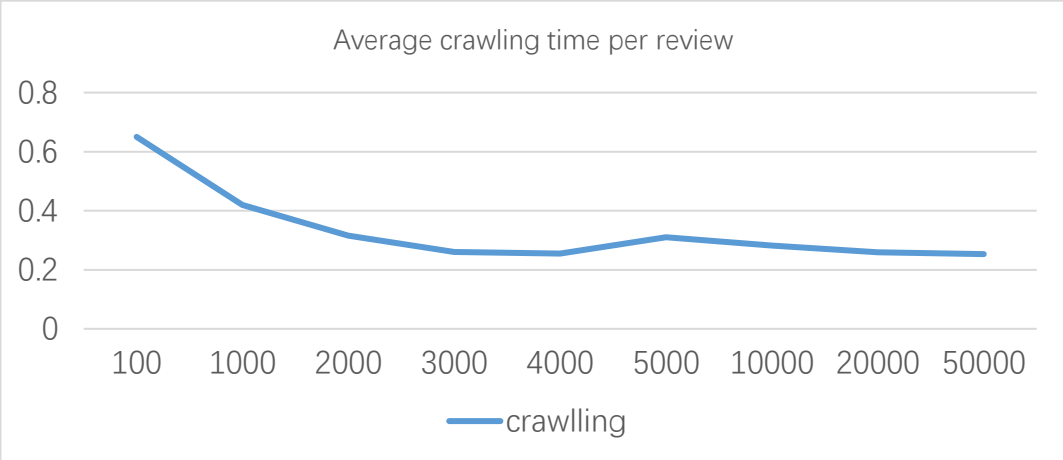

Figure 6.2 average crawling times

As shown in figure 6.2 the average crawling time per review decreases when the number of review grows, and finally stays around 0.23 seconds. This is because except from crawling each review from the website, the system also needs to establish a connection with the Steam website server, create JSON file, shut down the connection, etc. The time of establishing connections stays the same no matter how many reviews the system crawls. This results in the rise of average crawling time especially when crawling relative small number of reviews. And the other operations like creating JSON file and shutting down connection have the same effect on average crawling time.

Then the author further tests the performance of reading JSON files, pre-processing of reviews and topic mining of reviews on the same test data sets. Results are shown in Table 6.4:

Review scale	100	1000	2000	3000	5000	10000	20000	50000
Total reading time (sec)	5	42	62	60	105	190	398	1211
Average reading time (sec)	0.05	0.042	0.031	0.02	0.021	0.019	0.02	0.024
Total pre-processing time (sec)	-	9	16	23	36	69	146	331
Average pre-processing time (sec)	-	0.009	0.0082	0.0077	0.0072	0.0069	0.0073	0.0066
Total mining time (sec)	-	4.6	8.13	12.1	21.5	33	58	170
Average mining time (sec)	-	0.0046	0.0041	0.003	0.0043	0.0033	0.0029	0.0034

Table 6.4 processing time of different numbers of reviews

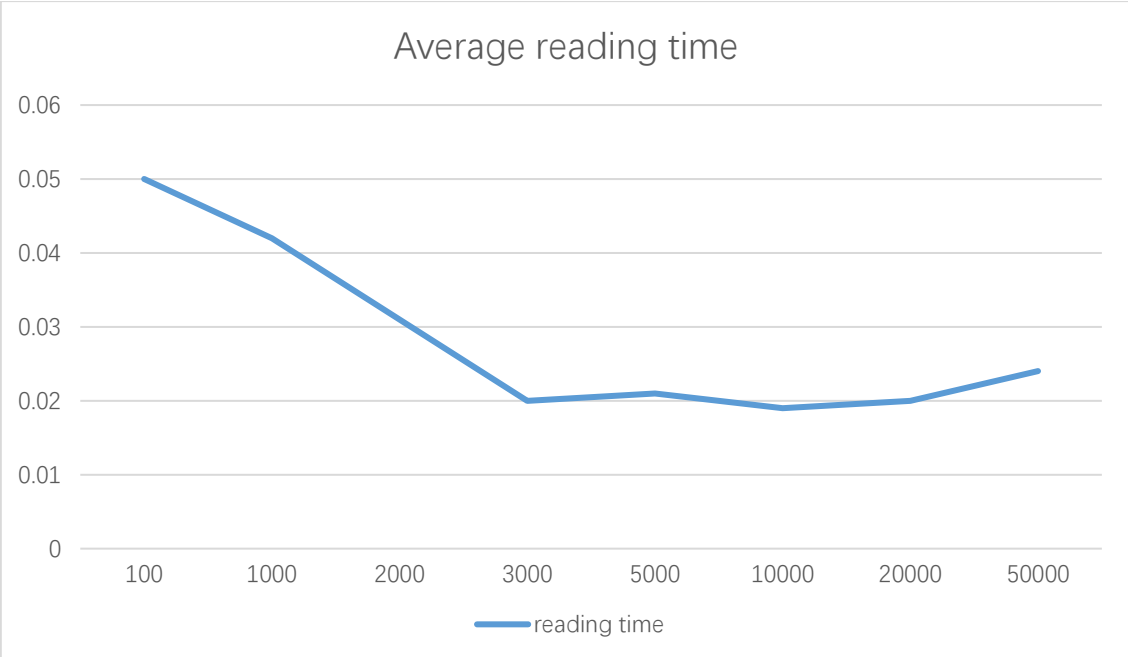

Figure 6.3 Average reading time

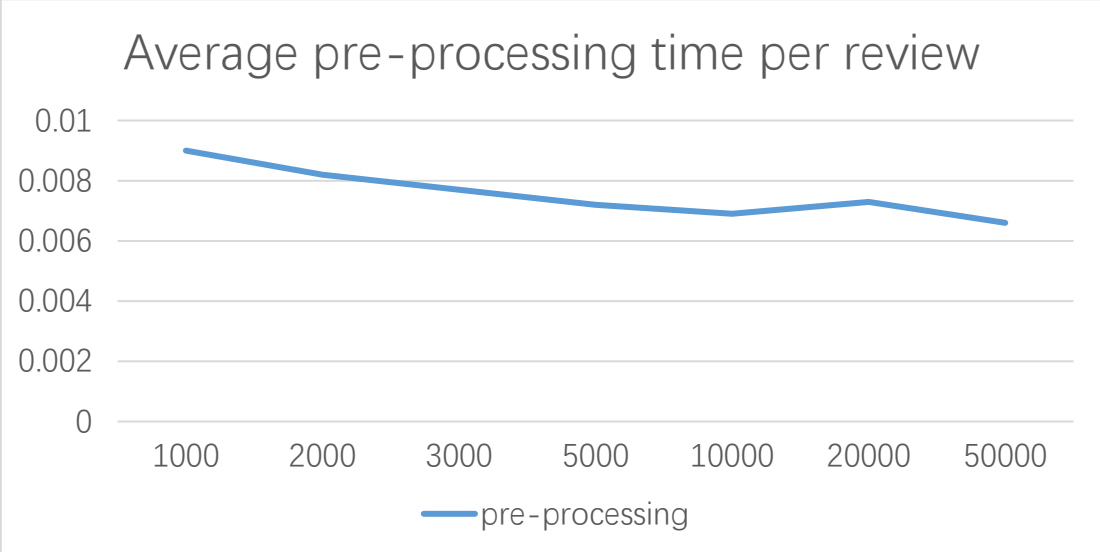

Figure 6.4 Average pre-processing time

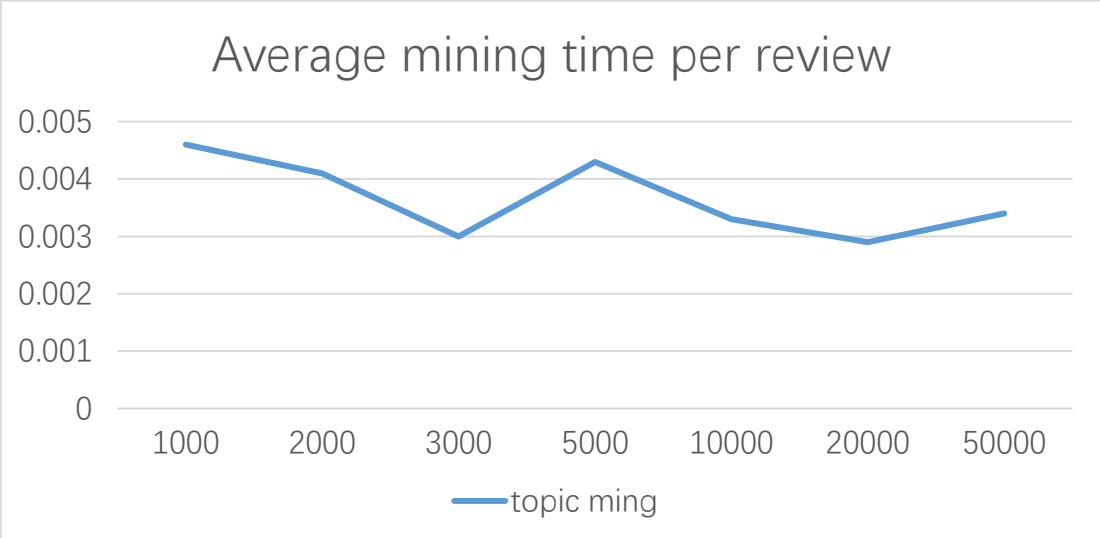

Figure 6.5 Average mining time

From the performance tests, it can be concluded that the order of execution time of different processes is: crawling > file reading > pro-processing > topic mining. Crawling is the most time-consuming process because it is actually data transfer from the website server to the local computer hard disk. The second time-consuming part is the file reading, which is reading data from outside, the system. The pre-processing and mining are processes within the system (considering the MySQL database as part of the system), which have the highest performance.

The system implemented in this thesis solved the problems which could appear in mining reviews from Steam. Functional requirements of the system are extracted inspired by the General process of data mining [Indarto, 2013], which are data collecting, data storage,

data pre-processing, data mining and data visualization. These requirements are proved to be necessary steps of data mining also by analysing related works.

From the test results, it is clear that when the scale of review used for mining growths, the result of topic mining gets more accurate. However, when the scale of reviews reaches a certain number, the accuracy does not increase significantly. Meanwhile, the time cost of for collecting and processing reviews growths linearly with the scale of reviews. Moreover, the number of reviews within a certain time period is limited, which means if more review data are needed, old reviews from several months ago would be collected. This could cause a decrease of the accuracy of mining results because most games update frequently, analysing old reviews may provide out of date topics, which cannot help users to understand the game.

6.2 Topic accuracy test

6.2.1 Topic validation test

Topic mining is the most important function of the system. In order to ensure the accuracy of mining results, a large number of reviews are collected and processed. The test first focuses on one game(*Nier:Automata*). The test is to find out the appropriate scale of reviews needed and the correct number topic and keywords. The reviews from other games will further be used as test sources to validate the results.

The evaluation is divided into two parts, topic validation test and topic coverage test.

Based on the previous discussions and referred to Zhang's work [2017] the evaluation method for topic validation is design as follows:

- Randomly pick 100 reviews for test person to read, and manually classify each review into one topic. After the process, if a topic has no reviews classified into, then the topic is invalid.
- If the test person finds that more than 50% of the keywords in a topic are the same as another topic, the topic is invalid.
- After reading the reviews, test person is asked to read the topics and their keywords, If the test person considers there are many noise words or stop-words occurring in a topic, the topic is invalid.
- If more than half of the topics are regards invalid topic, then the result of topic mining is also invalid.

The test is based on reviews from the game *Nier:Aoutoma* , around 20,000 pieces of reviews are collected. Five test data sets are used, which are generated by randomly pick from 1000 to 20,000 pieces of reviews. Additionally, different topic and keyword numbers are used to as test inputs. As mentioned in section 2.4, the author assumes that there are eight topics contained in Steam game reviews, so the number of topics is first set around 8. Due to experimental need, the topic number is set to 10 to benefit the decrease and increase. Other topic numbers used are 5, 15 and. The keyword number is set to a constant number 10 because keyword number has no influence on the result of LDA, it just decides how many keywords to be shown, and ten keywords is already enough to distinguish a topic from another.

Review scale	1000			2000			3000			5000		
Topic number	5	10	15	5	10	15	5	10	15	10	15	20
Keyword number	10	10	10	10	10	10	10	10	10	10	10	10
Topic validation rate (%)	20	20	30	20	30	33	10	30	47	50	66	55
Review scale	10000				15000				20000			
Topic number	5	10	15	20	5	10	15	20	5	10	15	20
Keyword number	10	10	10	10	10	10	10	10	10	10	10	10
Topic validation rate (%)	20	40	83	67	10	33	80	57	40	60	73	66

Table 6.5 topic validation test

As shown in table 6.5, when the number of reviews grows, topic validation rate rises accordingly (validation is count by calculating the valid topics among all the topics generated), when the number of reviews reach 5000, the validation rate reaches above 60% which is a relatively good result. When the scale of review exceeds 10,000, the maximum topic validation rate stays around 80%. However, the time needed for process review grows linearly when the number of review increases. In order to get the optimum result within tolerable time, the best scale of reviews should be set no more than 20,000. The author further test with different topic numbers, optimized validation rate reaches 80% when the topic number is set to 15.

6.2.2 Topic mapping test

After the topic validation test, an appropriate scale of review and suitable topic number are revealed. The coverage rate is calculated by comparing review-generated topics which the eight topics proposed in section 2.4: 1) Storylines, 2) Graphics, 3) Soundtracks, 4) Characters, 5) Control, 6) Difficulty and challenges, 7) Interaction with other players, 8) Immersion**.** Assume in a certain topic mining task, N topics from M valid topics can be classified into L topics from the proposed Steam review topic Model in section 3.5, then topic recognition rate is: M/N, and topic mapping rate is: L/8.
This test is carried base on reviews from different games and result is shown in table 6.6

Game name	Scale of review	Number of recognized topics (N)	Number of valid topics (M)	Topic recognition rate (N/M, %)	Topic coverage rate (%)
NieR:Automata	10,000	10	12	83	50
The Witcher3: Wild Hunt	20,000	15	11	73.3	37.5
Grand Theft Auto V	5,000	6	9	66.7	37.5
Player unknown's battle ground	25,000	5	7	71.4	25
Dota2	50,000	4	6	66.7	50
H1Z1	5,000	5	7	71.4	25

Table 6.6 topic mapping test

Test on each game has a relatively high topic recognize rate, which indicates the system and the LDA are able to mine topics for evaluating games. However not all tests have high topic coverage rate, this is because each game has its own emphasize. No game could cover all the game topics. Some games such as *Withcer3* focus on the storyline, while some others like *Dota2*. highlights the competition among players.

6.2.3 Keywords validation

Keywords are the actual content of topics, by comparing keywords with the tags provide by Steam for each game, the author furtherly analyzes the precise of the mining results.

Game name	Scale of review	Keyword map rate (%)
NieR:Automata	10,000	35%
The Witcher3: Wild Hunt	20,000	20%
Grand Theft Auto V	5,000	30%
Player unknown's battle ground	25,000	25%
Dota2	50,000	25%
H1Z1	5,000	20%
average	19,167	25.8%

Table 6.7 topic map rate

Steam provides 20 top tags for each game, the author compares these tags with keywords in the topics, assume K tags have relative keyword from mined topics, then the auhor calculate the Topic Map rate as K/20.

As shown in Table 6.7, the average keyword map rate is 25.8%, which indicates that keyword which users use in game reviews are quite different from what they use in define game. The Tag mainly focuses on the content of the game, like storyline, characters, type of the game, etc rather than topics about user experience. Keywords from reviews are more comprehensive. The same emphasis is given to user experience and game content in the reviews.

In conclusion, the topic-related tests prove that LDA is suitable for mining topics and related keywords from a large number of user reviews. In addition, a considerable amount of experiments is carried on to get the optimize parameter for LDA algorithm and appropriate scale of Reviews.

7. Discussions

7.1 Topic accuracy test

The author compares keywords from reviews with Steam tags, and identifies that only approximately 30% of the tags have corresponding keywords. Steam tags mainly focus on the game itself, like the storyline, soundtrack, while keywords from reviews also include user experience, usability and playability. Figure 7.1 shows the mapping from general game evaluation aspects to detail topics delivered by doing the topic mining task, and to the Steam Tags.

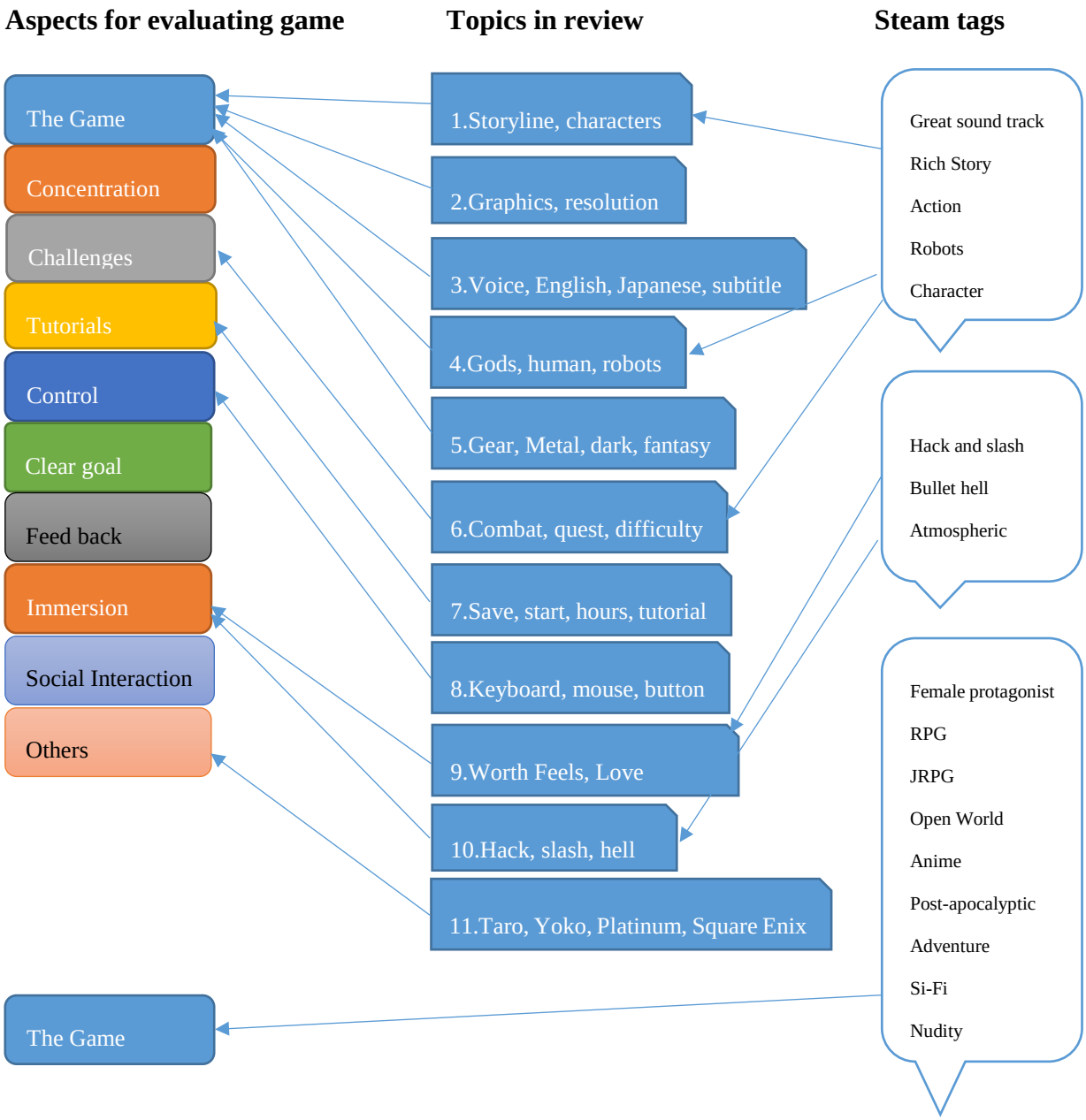

Figure 7.1 map from game topic to steam tags

This mapping is formed based on mining 15 topics from 10,000 pieces of review of the game *Nier:Automata*, 11 topics are identified as valid topics through topic validation test. The aspects for evaluating game from the research of Sweetser [Sweetser and Wyeth, 2005] are listed on the left side of the Figure. The topics generate by the system based on reviews from a certain game are given in the column in the middle (in blue). On the right side, there are the official Tags to describe the game provided by Steam. The arrows between aspects and topics indicate that the topic belongs to a certain aspect of evaluating a game. For example, topic 1 and 2 are two different topics, but they both describe the game itself. Topic 9 is another topic which talks about more than the game itself: user's experience while playing the game. The arrows between topics and tags indicate that the keywords in topics are the same (have the same meaning) as Tags provides by steam. Lastly, the arrow between tags and aspects indicates that tags don't have corresponding topics, but they can be classified into an aspect. This finding shows that even keywords from reviews are proved to be more comprehensive, tags still contain some missing information from them, which are also important for the game recommendation.

As shown in the figure, most topics are about the game itself, e.g., the first five topics. there are also some topics focus on control, challenges, tutorials and immersion, e.g., topic 6, 7 and 8. Steam tags only focus on the game itself, with a minority that is related to immersion. The high topic recognition rate in section 6.3.2 shows that almost each of the topics can be classified into one aspect for evaluation games, this indicates that the topics are helpful and convincing in help game users to know a game. Additionally, within the topics about the game itself, the keywords match well with the Steam Tags, which are officially used by Steam to recommend games to users. This indicates that the topics closely related to the game. Thus, in conclusion, LDA algorithm is suitable for mining topics from Steam game reviews, and the design and implementation of the system successfully support the process of optimizing the topic mining algorithm.

The main function of the system is discussed in this section, in the next section, the author discusses how well the system performs in the real-world situation and the system adapt to changes.

7.2 Performance

As shown in section 6.2, the average time for collecting and processing one single review is 0.28 second. Considering only when the scale of review reach 10,000, the mining result can be relatively effective, the time needed for one effective mining task execution is 46 minutes and 40 seconds. However, within the 0.28 second used for one review, 0.25 second is used for data collecting, that is to say, the time need for processing review in an effective mining task execution is only 5 minutes, which is a quite good result.

Performance of the system is ensured by limiting data transfer type as proposed in section 7.2, the order of execution time of different processes is: crawling > file reading > pro-processing > topic mining. Crawling process needs to establish a connection with the Steam website, data are transferred through the Internet, making it the most time-consuming process. JSON file reading transfer data through the system I/O, which is the second time-consuming. After the reading process, all reviews are stored in the database. MySQL supports queries in million level within seconds. Pre-process, mining, task management modules all transfer data with the database, which improves the overall system performance significantly.

7.3 Expandability

The Spring framework is used for the overall system design, it splits the system implementation into three parts, decouples system modules, which improves the system expandability.

In Spring MVC, Controller is the high level abstract for the business logic of the system. Each controller is responsible for controlling a series of inter-connected functions.
For example, in the mining task management module, there is a mining task Controller, The mining task controller deals with the logic for creating, modifying, deleting mining task. But the controller does not need to implement the detailed functions, it just decides which Service to call. If a new function needs to be added to the system, the existing business logic in the Controller can remain unchanged.

In the mining Controller, the main function is reading reviews from database and processing reviews. The Controller invokes the Mining Service to execute topic mining, the author used interface design especially for the Mining Service, and the interface

defines the input and output format of the topic mining process. If a new algorithm is integrated into the system, it will just need to extend the interface. In this case, other functions which could possibly interact with mining Service cold also remain unchanged. Decouple of the system and interface design keep the rest of the system unchanged while a new function is added to the system, which saves a considerable amount of effort in system development.

7.3 Limitation and potential improvements

The system presented in this thesis covers all the necessary steps for mining topics. As shown in the test and evaluation chapter, considering performance, the bottleneck of the system is data collecting, which accounts for 90% of the overall execution time.
This disadvantage is because Steam does not provide API for getting game reviews directed. Instead in this thesis web crawler is applied for data collecting, which is much slower compared to APIs. If Steam starts to provide API for getting reviews in the future, the collecting module of the system will need to be reimplemented. Another factor that restricts performance is the JSON file reading. As mentioned before, reading data from the database is the most effective way to deal with a large amount of data. A potential improvement is to put review data directly into the database while crawling from the web.

Considering the result of topic mining, the topics generated are proved to be effective to help users to understand the features of a game. However, by comparing with Steam's official game tags, the author finds that Steam tags mainly focus on the game content such as storylines, music or graphics, while keywords from reviews also focus on playability and usability. The review keywords are more comprehensive compared to Steam tags. However, in reviews, flaws of a game can be mention frequently, like 'the match system is bad', 'the tasks are unclear' and many other bad experiences. When helping users to understand a game, these flaws are useful, but are useless in game recommendation. If the keywords form reviews are used for recommendation, the sentimental analysis should be applied to filter aspects that are not liked by users.

8. Conclusion

This thesis presents a comprehensive system which aims at helping users to understand features of games. The system covers all the necessary steps for text analysis including data collecting, data restorage, pre-processing, and topic mining.

The data is collected from Steam official website. A self-design web crawler is implemented to get game review-related data. Besides data collecting, the system also provides functions for review storage, review pre-process, topic mining, and mining task management. The author uses Spring framework for system development, which significantly improves the development speed and expandability by decoupling different modules of the system and reuse of core codes. For topic mining, the LDA algorithm is used. LDA is a probability model which used to represent documents. The inputs of LDA are pre-processed review, Output are topics with the related keywords. Compared to TF-IDF algorithm, which only extracts keywords that appear in the document with high frequency, LDA provides aggregation for keywords, and similar keywords are aggregated into one topic.

Tests are conducted after the system implementation. Performance tests are conducted to evaluate if the system can run smoothly in the real-world situation, and the crawling process is identified as the bottleneck of the system performance. Topic-related tests are conducted to train the LDA model. Topic validation test is conducted to find the suitable scale of review and appropriate algorithm parameters for topic mining. It is found that 10,000 pieces of Steam reviews are enough to get satisfying results. The number of topics contained in Steam game review is identified as around 15. The Topic mapping test is conducted by comparing topics with aspects for evaluating games and it is found that most topics can be classified into an aspect. Keywords mapping test is conducted to evaluate if the mining results coordinate with the game itself. And the author finds that keywords do not match well with the official Steam Tags. However, the review keywords are more comprehensive in describing the game. Steam Tags focus on game content, in other words, the game itself while review keywords also cover user experience.

So far in this thesis, the system and topic mining results are evaluated by several people with knowledge in both topic mining and game evaluation. But when facing real users, e.g., Steam game players, there are some improvements of the system need to be done in

the future work. Different processes of mining need time which ranges from seconds to minutes. So far, the processing time is only displayed in the console of the integrated developing environment but not displayed in the system, only developers can see the time. Thus, the author is considering adding estimated time for processing before the user execute the mining task, and a progress bar while the task is being executed. In this way, the author believes that the user experience can be improved, and real users can evaluate the system performance more directly. Another improvement is about topic accuracy. The game tags from Steam should also be stored in the system beforehand, and can be displayed to the system users together with the mining result. There should also be a figure showing how many percentages of tags that overlap with keywords from the mining results so that the user can evaluate the results more quickly.

In conclusion, this thesis introduces a comprehensive system for mining topic from a large number of game reviews. Test results prove that the system can be useful in helping users to understand games. However, there is still room for improvements of the system performance and the user interface. Additional, the author finds that the mining result cannot be directly used for game recommendation, other necessary processes such as sentimental analysis will be needed.

References

[Airbnb, 2018] https://www.airbnb.fi/ Access on 1, May, 2018.

[Blei et.al., 2003] David M. Blei, Andrew Y. Ng and Michael I. Jordan, Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 2003, 993-1022.

[Beautiful Soup, 2017] https://www.crummy.com/software/BeautifulSoup/ Access on 1, April, 2018

[*Counter strike: global offense*, 2018]
https://store.steampowered.com/app/730/CounterStrike_Global_Offensive/
Access on 1, April, 2018

[Csikszentmihalyi, 1990] Csikszentmihalyi, M. 1990. Flow: The Psychology of Optimal Experience. Harper Perennial, New York.

[Desurvire et.al., 2004] Heather Desurvire, Martin Caplan, Jozsef A. Toth, Using heuristics to evaluate the playability of games, CHI 04 Extended Abstracts on Human Factors in Computing Systems, April 24-29, 2004, Vienna, Austria .

[Debnath et al., 2008] Debnath S, Ganguly N, Mitra P. Feature weighting in content based recommendation system using social network analysis. Proceedings of the 17th international conference on World Wide Web. ACM, 2008: 1041-1042.

[Dacidson et al, 2010] Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., ... & Sampath, D. (2010, September). The YouTube video recommendation system. In Proceedings of the fourth ACM conference on Recommender systems (pp. 293-296). ACM.

[Bernhaupt et al., 2007] Bernhaupt R, Eckschlager M, Tscheligi M. Methods for evaluating games: how to measure usability and user experience in games? Proceedings of the international conference on Advances in computer entertainment technology. ACM, 2007: 309-310.

[*Dota2*, 2018] https://store.steampowered.com/app/570/Dota_2/ Access on 1, April, 2018.

[Edwards, Cliff, 2013] Edwards, Cliff (November 4, 2013). Valve Lines Up Console Partners in Challenge to Microsoft, Sony. Bloomberg. Retrieved November 5,2013.

[Egenfeldt-Nielsen et al., 2016] Egenfeldt-Nielsen S, Smith J H, Tosca S P. Understanding video games: The essential introduction[M]. Routledge, 2016.

[Federoff, 2002] Federoff, M. 2002. Heuristics and usability guidelines for the creation and evaluation of fun in video games. Unpublished thesis, Indiana Univ., Bloomington. http://www.melissafederoff.com/thesis.html. Online Feb. 1, 2005.

[Gupta and Govil, 2010] Gupta, P., & Govil, M. C. (2010). Spring Web MVC Framework for rapid open source J2EE application development: a case study. International Journal of Engineering Science & Technology, 2010, 2(6).

[Guzman and Maalej, 2014] Emitza Guzman and Walid Maalej, How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews. equirements Engineering Conference. IEEE, 2014:153-162.

[Gu and Kim, 2015] Xiaodong Gu and Sunghun Kim. What parts of your apps are loved by users? Automated Software Engineering (ASE), 2015 30th IEEE/ACM International Conferencen.

[Gimpel, 2006] Kevin Gimpel. *Modeling Topics*. Inform. Retrieval 5. 2006. pp.1-13.

[GameSpot, 2018] https://www.gamespot.com/ Access on 1, April, 2018.

[Google Play, 2018] https://play.google.com/store Access on 1, April, 2018.

[Hall, Charlie, 2016] Hall, Charlie. January 4, 2016. Report: Paid Steam games market estimated at over $3.5 billion in 2015. Retrieved from https://www.polygon.com/.

[Hu and Liu, 2004] Hu M, Liu B. Mining and summarizing customer reviews Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004: 168-177.

[Hoffman, et.al., 2009] Matthew D. Hoffman, David M. Blei, and Perry R. Cook. Finding latent sources in recorded music with a shift-invariant hdp. In: Proc. of International Conference on Digital Audio Effects (DAFx), 2009.

[Hofmann, 1999] Thomas Hofmann, Probabilistic latent semantic indexing. In: Proc. of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, New York, 1999, 50–57.

[IGN, 2018] http://nordic.ign.com/ Access on 1, April, 2018.

[JavaEE, 2018] http://www.oracle.com/technetwork/java/javaee/overview/index.html Access on 1, April, 2018.

[Kakkonen et.al., 2006] Tuomo Kakkonen, Niko Myller, and Erkki Sutinen, Applying latent dirichlet allocation to automatic essay grading, In Proceedings of the 5th international conference on Advances in Natural Language Processing (FinTAL'06), Springer-Verlag, Berlin, Heidelberg, 110-120, 2006.

[Kucuktunc et.al., 2012] O. Kucuktunc, B. B. Cambazoglu, I. Weber, and H. Ferhatosmanoglu. A large-scale sentiment analysis for Yahoo! Answers. In Proc. of the International conference on Web search and data mining - WSDM ’12, pages 633–642, Feb. 2012.

[Lienhart et.al., 2009] Rainer Lienhart, Stefan Romberg and Eva Horster, Multilayer PLSA for multimodal image retrieval. In: Proc. of the ACM International Conference on Image and Video Retrieval, New York, 2009, 91-98.

[Lazzaro and Keeker, 2004] Nicole Lazzaro, Kevin Keeker, What's my method?: a game show on games, CHI '04 Extended Abstracts on Human Factors in Computing Systems, p.1093-1094, April 24-29, 2004, Vienna, Austria .

[Melville et al., 2002] P. Melville, R.J. Mooney, R. Nagarajan.Content-Boosted Collaborative Filtering for Improved Recommendations, Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-2002), July 2002, Edmonton, Canada.

[Monay and Gatica-Perez, 2004] Florent Monay and Daniel Gatica-Perez. PLSA-based image auto-annotation: constraining the latent space. In: Proc. of the 12th annual ACM international conference on Multimedia, New York, NY, USA, 2004, 348–351.

[Minka and Lafferty, 2002] Thomas Minka and John Lafferty, Expectation-propagation for the generative aspect model, In Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence (UAI'02), Adnan Darwiche and Nir Friedman (Eds.). Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 352-359, 2002.

[Makuch, 2015] Makuch, Eddie (November 1, 2015). Steam Reaches New Concurrent User Record. GameSpot. Retrieved November 2, 2015.

[Mallet, 2002] http://mallet.cs.umass.edu/ Access on 1, May, 2017

[Newzoo, 2016] The Global Games Market Reaches $99.6 Billion in 2016. Retrieved from https://newzoo.com/insights/articles/global-games-market-reaches-99-6-billion-2016-mobile-generating-37/

[Nallapati and Cohen, 2008] Nallapati R, Cohen W. Link-pLSA-LDA: A new unsupervised model for topics and influence of blogs, May 23, 2008.

[Netflix, 2018] https://www.netflix.com/ Access on 1, April, 2018

[Oracle, 2012] MySQL Customers. Oracle. Retrieved 17 September 2012.

[Pagano and Maalej, 2013] D. Pagano and W. Maalej. User feedback in the appstore: an empirical study. In Proc. of the International Conference on Requirements Engineering - RE ’13, pages 125–134, 2013.

[Park et al.,2007] Park M H, Hong J H, Cho S B. Location-based recommendation system using bayesian user's preference model in mobile devices. International Conference on Ubiquitous Intelligence and Computing. Springer, Berlin, Heidelberg, 2007: 1130-1139.

[Pagulayan et al. 2003] Randy J. Pagulayan, Kevin Keeker, Dennis Wixon, Ramon L. Romero, Thomas Fuller, User-centered design in games, The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications, L. Erlbaum Associates Inc., Hillsdale, NJ, 2002

[Ramos, 2003] Ramos J. Using tf-idf to determine word relevance in document queries. Proceedings of the first instructional conference on machine learning. 2003, 242: 133-142.

[Player unknown's battle grounds, 2018] https://store.steampowered.com/app/578080/PLAYERUNKNOWNS_BATTLEGROUNDS/ access on 1, May, 2018

[Sivic, Russell et.al., 2005] J. Sivic, B. C. Russell, A. A. Efros, A. Zis-serman and W. T. Freeman, Discovering object categories in image collections. In: Proc. of the International Conference on Computer Vision, 2005.

[Steam, 2018] Steam game platform: *https://store.steampowered.com/* Access on 1, April, 2018

[Sweetser and Wyeth, 2005] Sweetser P, Wyeth P. GameFlow: a model for evaluating player enjoyment in games. Computers in Entertainment (CIE), 2005, 3(3): 3-3.

[Smith, 2015] Smith Ryan. Valve to Showcase SteamVR Hardware, Steam Machines, & More at GDC 2015, February 24, 2015. Retrieved from: https://www.anandtech.com/show/9003/valve-to-showcase-steamvr-hardware-steam-machines-more-at-gdc-2015/

[Spring, 2010] https://spring.io/blog/2010/10/29/spring-3-0-5-is-now-available Access on 1, April , 2018

[Turney, 2000] Turney P D. Learning algorithms for keyphrase extraction. Information retrieval, 2000, 2(4): 303-336.

[Titov and McDonald, 2008] Ivan Titov and Ryan McDonald, Modeling Online Reviews with Multi-grain Topic Models WWW 2008 / Refereed Track: Data Mining - Modeling April 21-25, 2008 Beijing, China

[Twitter, 2018] https://twitter.com/ Access on 1, April, 2018

[TripAdvisor, 2018] https://www.tripadvisor.com/ Access on 1, April, 2018

[The wicher 3: Wild Hunt, 2018]
https://store.steampowered.com/app/292030/The_Witcher_3_Wild_Hunt/
Access on 1, May, 2018.

[Uber, 2018] https://www.uber.com/ Access on 1, April, 2018.

[Wang and Land, 2011] Han Wang and Bo Lang. Online N gram-enhanced Topic Model for Academic Retrieval. In: Proc. of Digital Information Management (ICDIM), Sixth International Conference on Date of Conference, 2011, 137-142.

[Alex, 2016] Wawro Alex , January 4, 2016. Steam Spy: Last year the paid Steam games market hit $3.5 billion. Gamasutra. Retrieved January 4, 2016. Available as http://www.gamasutra.com/view/news/263003/Steam_Spy_Last_year_the_paid_Steam_games_market_hit_35_billion.php

[Wei, 2006] X. Wei and W. B. Croft, LDA-based document models for ad hoc retrieval. In: Proc. of SIGIR 06 Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, 178-185.

[Zhao and Zhang, 2012] Xubin Zhao and Changkuan Zhang, Topic Community Mining in Blogsphere Based on LDA. Computer & Digital Engineering, 11, 2012.

[Zhang, 2017] Zhang C. A system of topic mining and dynamic tracking for social texts. Master thesis, School of information science, University of Tampere. 2017.