NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

BSc THESIS

Automatic Summarization of Video Game Reviews

George D. Panagiotopoulos

Supervisors: **Panagiotis Stamatopoulos,** Assistant Professor NKUA
George Giannakopoulos, NCSR Demokritos Research Fellow
Antonios Liapis, Lecturer University of Malta

ATHENS

October 2019

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αυτόματη Δημιουργία Περιλήψεων σε Κριτικές Ηλεκτρονικών Παιχνιδιών

Γεώργιος Δ. Παναγιωτόπουλος

Επιβλέποντες: **Παναγιώτης Σταματόπουλος,** Επίκουρος Καθηγητής ΕΚΠΑ
Γεώργιος Γιαννακόπουλος, Συνεργ/νος Ερευνητής ΕΚΕΦΕ ”Δημόκριτος”
Αντώνιος Λιάπης, Λέκτορας Πανεπιστήμιο Μάλτας

ΑΘΗΝΑ

Οκτώβριος 2019

BSc THESIS

Automatic Summarization of Video Game Reviews

George D. Panagiotopoulos
S.N.: 1115201400136

SUPERVISORS: **Panagiotis Stamatopoulos,** Assistant Professor NKUA
George Giannakopoulos, NCSR Demokritos Research Fellow
Antonios Liapis, Lecturer University of Malta

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αυτόματη Δημιουργία Περιλήψεων σε Κριτικές Ηλεκτρονικών Παιχνιδιών

Γεώργιος Δ. Παναγιωτόπουλος
Α.Μ.: 1115201400136

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Παναγιώτης Σταματόπουλος,** Επίκουρος Καθηγητής ΕΚΠΑ
Γεώργιος Γιαννακόπουλος, Συνεργ/νος Ερευνητής ΕΚΕΦΕ ”Δημόκριτος”
Αντώνιος Λιάπης, Λέκτορας Πανεπιστήμιο Μάλτας

ABSTRACT

Video game reviews have constituted a unique means of interaction between players and companies for many years. The dynamics appearing through online publishing have significantly grown the number of comments per game, giving rise to very interesting communities. The growth has, in turn, led to a difficulty in dealing with the volume and varying quality of the comments as a source of information.

This work studies whether and how game reviews can be summarized, based on the notions pre-existing in aspect-based summarization and sentiment analysis. We initially provide a formal definition of the problem in order to set the basis for our suggested approach. We then devise a baseline implementation, that attempts to tackle the individual subtasks that constitute the problem of video game review summarization. More precisely, given a set of reviews of a video game, we apply k-means clustering in order to identify groups of similar sentences. We then utilize word lists with the aim of mapping the produced clusters to predefined game aspects, like graphics and gameplay. Subsequently, we apply sentiment analysis using a rule-based method in order to extract the sentiments that pervade each cluster.

Additionally, we offer preliminary findings on whether aspects detected in a set of comments can be consistently evaluated by human users. The evaluation ascertains that review summarization subtasks are achievable and sets a method for the evaluation of performance of future systems.

SUBJECT AREA: Natural Language Processing, Automatic Summarization

KEYWORDS: Game Review Summarization, Clustering, Sentiment Analysis, Summary Evaluation, k-means Algorithm

ΠΕΡΙΛΗΨΗ

Τα ηλεκτρονικά παιχνίδια αποτελούν εδώ και πολλά χρόνια ένα μοναδικό μέσο αλληλεπίδρασης μεταξύ παιχτών και εταιρειών. Οι δυναμικές που εμφανίζονται μέσω των διαδικτυακών δημοσιεύσεων έχουν αυξήσει σημαντικά τον αριθμό των σχολίων ανά παιχνίδι, οδηγώντας στην ανάπτυξη ενδιαφερουσών κοινοτήτων. Αυτή η αύξηση έχει, με τη σειρά της, οδηγήσει στη δύσκολη αντιμετώπιση του τεράστιου όγκου και ποικίλης ποιότητας σχολίων σαν πηγή πληροφορίας.

Αυτή η δουλειά εξετάζει αν και πως οι κριτικές ηλ. παιχνιδιών μπορούν να συνοψιστούν, βάση των προυπάρχουσων εννοιών στην περίληψη βασισμένη σε χαρακτηριστικά και στην ανάλυση συναισθήματος. Παρέχουμε αρχικά ένα τυπικό ορισμό του προβλήματος με σκοπό να θέσουμε τη βάση για την προσέγγιση που προτείνουμε. Έπειτα, αναπτύσσουμε μία βασική υλοποίηση, που προσπαθεί να αντιμετωπίσει τις μεμονωμένες υποεργασίες που συντελούν το πρόβλημα της περιλήψης κριτικών ηλ. παιχνιδιών. Πιο συγκεκριμένα, δεδομένου ενός συνόλου από κριτικές για ένα παιχνίδι, εφαρμόζουμε συσταδοποίηση κ-μέσων για να αναγνωρίσουμε ομάδες όμοιων προτάσεων. Στη συνέχεια χρησιμοποιούμε λίστες λέξεων με σκοπό να αντιστοιχίσουμε τις παραγόμενες συστάδες σε προκαθορισμένα χαρακτηρικά των ηλ. παιχνιδιών, όπως τα γραφικά και το gameplay. Εν συνεχεία, εφαρμόζουμε ανάλυση συναισθήματος χρησιμοποιώντας μία μέθοδο βασισμένη σε κανόνες με σκοπό να εξάγουμε τα συναισθήματα που κυριαρχούν στη συστάδα.

Επιπροσθέτως, προσφέρουμε προκαταρκτικά ευρήματα για το κατά πόσο τα χαρακτηριστικά που εντοπίστηκαν σε ένα σύνολο από σχόλια μπορούν να αξιολογηθούν με συνέπεια από ανθρώπους. Αυτή η διαδικασία αξιολόγησης επιβεβαιώνει ότι οι υποεργασίες της περίληψης κριτικών είναι εφικτές και ορίζει μία μέθοδο για την αξιολόγηση της επίδοσης μελλοντικών συστημάτων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Επεξεργασία Φυσικής Γλώσσας, Αυτόματη Περίληψη

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Περίληψη Κριτικών Ηλ. Παιχνιδιών, Συσταδοποίηση, Ανάλυση Συναισθήματος, Αξιολόγηση Περίληψης, Αλγόριθμος κ-μέσων

ACKNOWLEDGEMENTS

I would like to wholeheartedly thank NCSR “Demokritos” researcher George Giannakopoulos for his guidance, patience and overall support during the writing of this thesis. His suggestions were of vital importance and the working methodology we followed equipped me with many important research skills and helped me structure a well-structured workflow.

I would also like to express my gratitude to prof. Panagiotis Stamatopoulos for suggesting the idea of collaborating with “Demokritos” and for always being willing to help me and support my efforts. His excellent way of teaching and personality has helped me acquire solid programming skills.

I would also like to thank Antonios Liapis for his valuable feedback and for his support by sharing his expertise on the subject of my thesis.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The rapid growth of video game industry with new products and technology has significantly increased the popularity of video games. As video games have now become one of the most profitable source of entertainment worldwide, the competition between development companies has increased notably.

Catering for gamers' needs is a demanding task that developers struggle to deal with. Thus, it is crucial for game companies to understand the overall consensus about their products. Additionally, what other people think of a game can also be an important piece of information for potential buyers. Video game reviews offer user-generated data that can be processed in order to identify both people's concerns and user-perceived quality of the game. A number of publishers (Steam[1], GoG[2], etc.) offer a wide range of games, spanning various genres. By visiting such a publisher's store, people are able to look through a game's description and its features, delve into the reviews of the game provided by other users and experts, but also contribute their own review. As some of the games can have millions of reviews, the large scale of information poses the need and challenge of automatic summarization.

This thesis researches whether and how game reviews can be summarized, in accordance with the notions pre-existing in aspect-based summarization and sentiment analysis. The primary aims of this thesis are: to examine if and how game review summarization can be different from other similar tasks (such as aspect-based summarization and sentiment analysis), to propose an initial approach on game review summarization and also to offer an evaluation process on the performance of the game review summarization task.

The remainder of this thesis is structured as follows:

Chapter 2 covers the underlying concepts of our study in order to assist the user with understanding the rest of this thesis. We review different types of automatic summarization process, followed by a review of sentiment and cluster analysis.

In Chapter 3 we overview the research endeavours related to this work, uniquely positioning it in the current research spectrum and discussing the unique setting of game review summarization.

Our main goal in Chapter 5 is to propose an approach to tackle the problem of game review summarization at hand. We establish a problem definition before elaborating on the individual processing steps that constitute our method.

In Chapter 6 we describe an empirical evaluation performed with four different human evaluators. The evaluation process attempts to understand whether the steps of the problem, as formulated in the Chapter 5, can be evaluated consistently .

Finally, Chapter 7 summarizes the work of this thesis and proposes future research steps towards the emerging and useful domain of game review summarization.

[1] https://store.steampowered.com/

[2] https://www.gog.com/games

2. THEORY

In this chapter we explain the concepts and techniques used in automatic summarization, sentiment analysis and clustering. More precisely, we firstly delve into the main approaches adopted in extractive and abstractive summarization. We then present the taxonomy of summary evaluation methods. Additionally, we provide the reader with a description of the existing approaches in the main categories of sentiment and document clustering.

2.1 Automatic Summarization

The ever-growing amount of text data available on the Web (news articles, books, tweets, etc.) has increased the need for processing and analyzing text with the aim of stripping superfluous information away. Since this process can be quite cumbersome for humans to deal with, the popularity of automatic summarization systems has escalated considerably. Automatic summarization can be described as the process of shortening a text document using software, in order to produce a summary with the salient points of the original document. Throughout this thesis we will use the term *summarization* to refer to the process of automatically summarizing text using software.

Summarization can be applied on two different scales, *single-document* and *multi-document*. The former attempts to summarize a standalone document, while the latter generates a summary that incorporates perspectives from across multiple documents, usually obtained through a query against a database or search engine[5]. It should be noted that a “document” could refer to different things depending on the use case. More precisely, there has been a wide variety of studies on summarization spanning different domains, like e-mails [6], blog posts [22] and even scientific papers [47]. Automatic summarization approaches are typically divided into two different categories based on the how the output summary is generated: *extraction-based* and *abstraction-based*.

2.1.1 Extractive Summarization

The vast majority of current literature focuses on sentence extraction [31][19][56][64][13]. These summarization systems identify the most important sentences in the input, which can be either a single document or a set of related documents, and join them together to form a summary. The decision about what content is important is mainly based on the input of the summarizer. In order to examine the different stages in the operation of extractive summarizers, we follow the comprehensive and well-structured survey written by Nenkova and McKeown [38]. The core of most extractive summarizers consists of three relatively independent tasks: constructing and intermediate representation of the input text, scoring the sentences based on the representation and selecting a summary consisting of several sentences.

There are currently two main approaches on document representation: topic representation and indicator representation. The former group of techniques transforms the input into an intermediate representation and interprets the topics discussed in the text, while the latter describes every sentence as a set of formal features (indicators) of importance such as sentence length, position in the document, having certain phrases, etc.

2.1.1.1 Topic Words

This technique attempts to identify words that describe the topic of the input document, i.e. topic signatures [31]. Topic signatures are words that occur often in the input but are rare in other texts, so their computation requires counts from a large collection of documents in addition to the input for summarization. These special words are identified by a log-likelihood ratio test [12]. Subsequently, the importance of a sentence can be determined by the proportion of topic signatures it contains.

2.1.1.2 TF-IDF Weighting

The tf-idf (term frequencny - inverse document frequency) weighting scheme provides a better alternative to the word probability approach, as it does not heavily rely on a stopword list to eliminate very common words. As its name suggests, tf-df method takes into account both the term's frequency in a document and its overall frequency in the whole corpus. The intuition behind it is that if a word occurs multiple times in a document, we should boost its relevance as it should be more meaningful than other words that appear fewer times (tf). On the contrary, if a word occurs many times in a document but also along many other documents, it is highly possible that this is a just a frequent word rather than a relevant or meaningful one (idf). Equation 2.1, gives the tf-idf weight for a given word w, a document d and a corpus D that contains N documents.

$$TF{-}IDF(w,d,D) = f_{t,d} * \log_{10} \frac{N}{\mid \{d \in D : t \in d\} \mid} \tag{2.1}$$

2.1.1.3 LSA

Latent Semantic Analysis (LSA) [9] is an algebraic-statistical method that extracts hidden semantic structures of words and sentences. It is an unsupervised approach that does not need any training or external knowledge. LSA uses the context of the input document and extracts information such as which words are used together and which common words are seen in different sentences. The LSA-based summarization systems usually perform three main steps:

- *Input matrix creation*: each input sentence needs to be converted into an appropriate representation. The most common technique to achieve this is to represent each sentence as a vector containing the tf-idf values of its terms. A term-sentence matrix is eventually formed by combining together these vectors.
- *Singular Value Decomposition (SVD)*: SVD is essentially the core component of every LSA method. It models relationships among words/phrases and sentences by decomposing the input matrix A (m words by n sentences) into three new matrices as follows:

  $$A = U * \Sigma * V^T \tag{2.2}$$

  Matrix U is a n by m matrix of real numbers. Each column can be interpreted as a topic or concept. Thus, this matrix illustrates the relationship between words and topics.

Matrix Σ is diagonal m by n matrix. The single non-zero entry in row i of the matrix corresponds to the weight of the "topic", that is the ith column of U. A very important feature of this technique is the ability to reduce the dimensionality of the original space, by truncating the last k rows U, the last k rows and columns of Σ and the last k rows of V^T.

Matrix V^T acts the same way as U but models the relationship between topics and sentences, rather than topics and terms.

Figure 2.1 provides an insightful illustration on how SVD works [54].

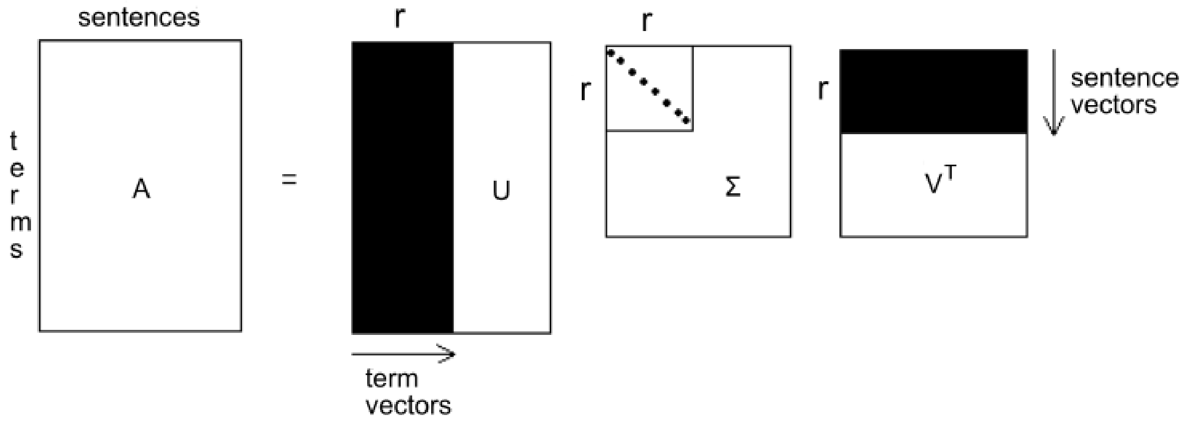


Figure 2.1: Singular Value Decomposition

- *Sentence Selection*: using the results of SVD, different algorithms are used to select important sentences. Most of them are based on Gong and Liu's [19] study on LSA-based text summarization. By performing dimensionality reduction, they keep only as many topics as the number of sentences they want to include in the summary. This number is given as a parameter. Following this, they select the sentence with the highest weight for each of the topics to be included in the output summary. The number of sentences to be selected is given as a parameter. A simple example of selecting four sentences based on their topic weights is demonstrated in Table 2.1.

Table 2.1: Sentence selection using the V^T matrix of the LSA method

V^T Matrix			
Topic ID	**Sent 1**	**Sent 2**	**Sent 3**
1	0.457	0.778	0.510
2	-0.242	0.991	0.123
3	0.421	0.311	-0.004
4	-0.441	0.123	0.331

2.1.1.4 Sentence Clustering

In this approach, clusters of similar sentences are treated as proxies for topics. Clusters consisting of many sentences are more likely to represent important topic themes in the input text. An extractive summary is generated by selecting representative sentences from each main cluster [20]. A study based on this approach is presented by Sauper et al.

[51]. They cluster similar section headings with the aim of identifying the topics discussed in each type of article. A more detailed description of document clustering algorithms is given in a later subsection.

2.1.1.5 Graph Methods

Graph methods are a common example of indicator representation methods. They are inspired by the PageRank algorithm [40], which produces a global “importance” ranking of every web page. These methods represent documents as a connected graph, where sentences form the vertices and edges between the sentences indicate how similar the two sentences are. Edges are assigned weights equal to the similarity between the two sentences. This similarity is measured with the help of cosine similarity with tf-idf weights for words. The vertices, i.e. sentences, are connected only if the similarity between them exceeds a predefined threshold. The intuition behind these methods, is that the more the links a sentence has with other sentences, the more meaningful this sentence is, and as a result it is included in the summary.

The most well-known study on graph-based summarization is the LexRank method [13]. The proposed algorithm measures the importance of a sentence in the generated graph (Figure 2.2) by considering its relative importance to its neighboring sentences, where a positive contribution will raise the importance of the sentence’s neighbor, while a negative contribution will lower the importance value of a sentence’s neighbor. This idea is basically the same with PageRank, but instead of ranking web pages it ranks sentences from various documents.

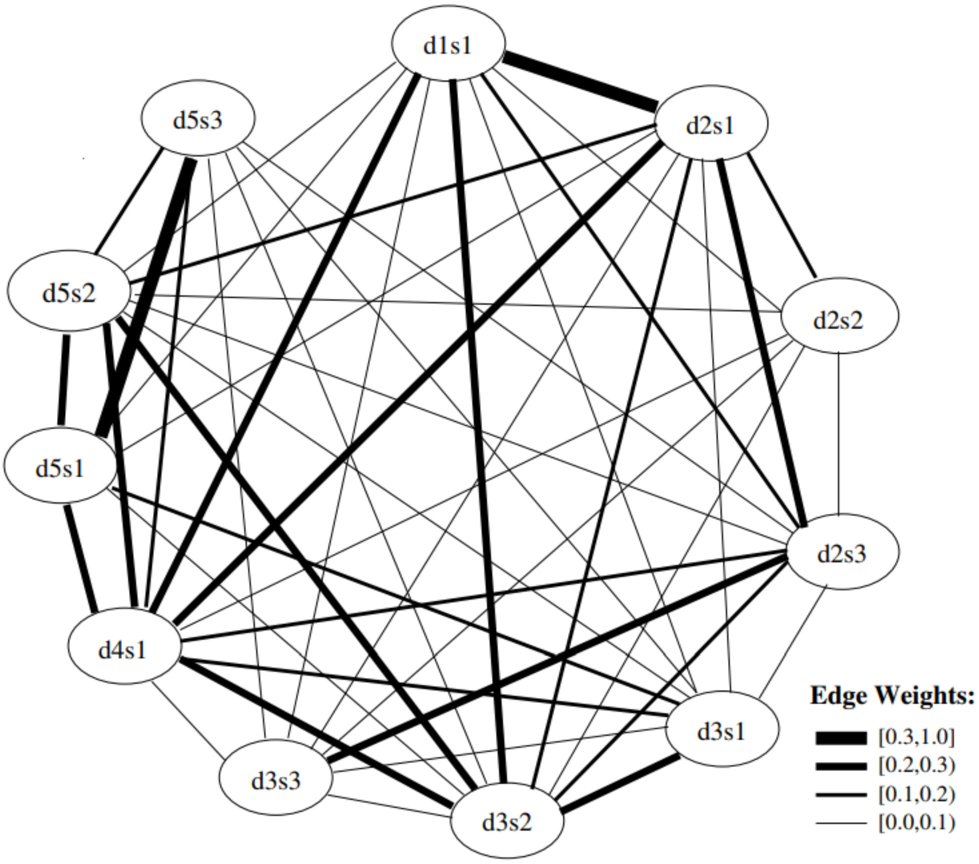


Figure 2.2: Weighted similarity graph example for a cluster of sentences presented in the original LexRank paper

2.1.1.6 Machine-Learning-based Methods

Over the past decade, machine learning approaches has been gaining much attention due to the abundance of text data available on the Web. In supervised methods for summarization, the task of selecting important sentences is represented as a binary classification problem, partitioning all sentences in the input into summary and non-summary sentences. While learning-based methods have proved to be considerably effective and successful, particularly in domain-specific summarization tasks [56] [64], they still have a major drawback: a set of labeled documents or sentences is needed in order for the classifier to train with. It is easily understandable that developing a large corpus of labeled documents can be a rather tedious task for human annotators.

2.1.2 Abstractive Summarization

Abstractive summarization techniques tend to mimic the process of “paraphrasing” from a text than just simply extracting sentences or phrases from it. Using an abstraction-based summarizer results in a more condensed and coherent summary. However, these techniques are much harder to implement than extractive summarization techniques, as they require use of natural language generation technology. It is worthwhile noting that, modern neural-network-based studies, that have been proposed recently, have led to increased popularity of abstractive summarization [44] [7].

2.1.3 Summary Evaluation

As stated earlier, the aim of automatic text summarization is to reduce the source text into a succinct version which will preserve contents and general meaning. In order to assess the performance of a summarization system we need to devise a strategy for evaluating the overall quality of the output summary. Thus, summary evaluation is a very important task. There are several serious challenges in evaluating summaries, which makes summary evaluation a very interesting problem [34]:

- There is always the possibility of a system producing a good summary that is quite different from any human summary used as an approximation to the correct output.
- Requiring human judges to assess the quality of a summary can be of great expense. An evaluation method based on a scoring algorithm instead of human judgments is preferable, since it is easily repeatable.
- The complexity of the evaluation process increases proportionally to the compression rate (ratio of summary length to source length) used during the summarization procedure.
- The domain as well as the group of users that the summary is intended for are two major factors that should be taken into account during the evaluation.

Summary evaluation methods can be classified in two main categories [29]: *intrinsic* evaluation and *extrinsic* evaluation methods.

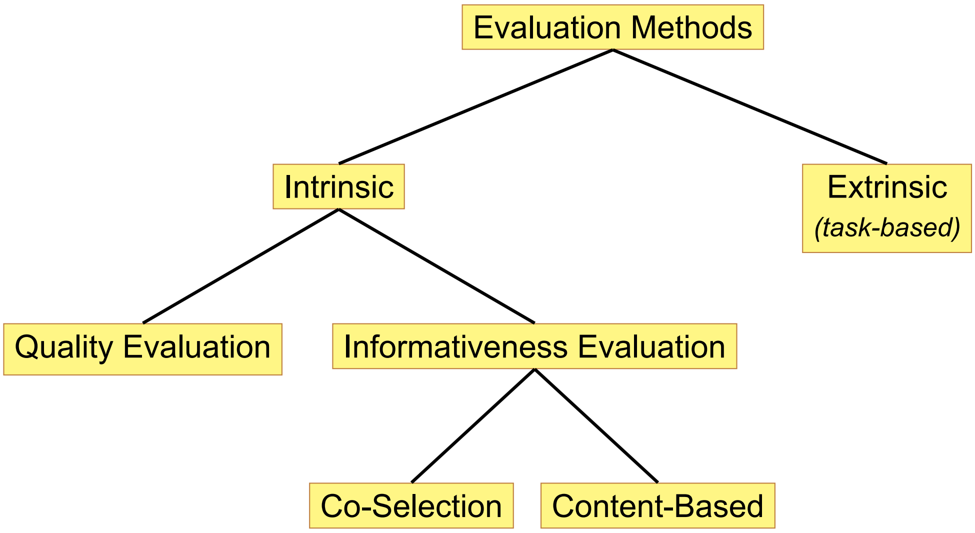


Figure 2.3: Taxonomy of summary evaluation methods

2.1.3.1 Intrinsic Evaluation

Intrinsic evaluation tests the summarization system in of itself. The summary is evaluated on the basis of two criteria: *quality* and *informativeness*.

In quality evaluation linguistic aspects of the summary are considered. Such aspects include redundancy, grammaticality, as well as coherence of the output summary. One important feature of the quality evaluation process is that it does not require the generated summary to be compared against a gold standard summary. An expert human evaluator or a linguist can evaluate the summary manually by assigning a score to the summary corresponding to five-point scale on the basis of its quality.

Informativeness evaluation aims at assessing the summary's information content. The informativeness of a summary is evaluated by comparing it with a human-made summary, i.e., reference/ideal summary. The wide range of existing studies on informativeness evaluation can be broadly divided into two separate categories: *co-selection* methods [2] and *content-based* methods [39][21][17][63][30] [54] .

The main evaluation metrics of co-selection are precision, recall and F-score . Precision is the number of sentences occurring in both system and ideal summaries divided by the number of sentences in the system summary. Recall is the number of sentences occurring in both system and ideal summaries divided by the number of sentences in the ideal summary. F-score is a composite measure that combines precision and recall. F-score is usually computed using with the following formula:

$$F = \frac{2 \cdot P \cdot R}{P + R} \tag{2.3}$$

where P is the precision and R is the recall.

The main drawback of evaluating a summary using these measures is that they count as a match only exactly the same sentences. Thus, they completely ignore the fact that two sentences can be semantically identical, i.e. contain the same information, even if they consist of totally different words.

In contrast to co-selection methods that are incapable of dealing with different sentences that have the same meaning, extrinsic content-based methods can deal with this kind of issues. We will now briefly describe two of the most widely used content-based methods:

- *Pyramid Method*: the pyramid method is a novel semi-automatic evaluation method [39]. Its basic idea is to identify summarization content units (SCUs) that are used for comparison of information in summaries. SCUs emerge from annotation of a corpus of summaries and are not bigger than a clause. In essence, the pyramid method addresses the problem by using multiple human summaries to create a gold-standard and by exploiting the frequency of information in the human summaries in order to assign importance to different facts.

- *ROUGE*: ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [30] is used as an automatic evaluation method. ROUGE is actually a set of metrics and a software package used for evaluating automatic summarization, which is based on the similarity of n-grams. Suppose a number of annotators created a set of reference summaries (RSS). The ROUGE-n score of a candidate summary is computed as follows:

  $$ROUGE\text{-}n = \frac{\sum_{C \in RSS} \sum_{gram_n \in C} Count_{match}(gram_n)}{\sum_{C \in RSS} \sum_{gram_n \in C} Count(gram_n)} \tag{2.4}$$

  where $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a reference summary and $Count(gram_n)$ is the number of n-grams in the reference summary.

It is worth mentioning that there many other variants of ROUGE scores. Among others, there is a longest common subsequence measure called ROUGE-L and a ROUGE-S measure that evaluates the amount of skip bigrams common between a particular summary and a collection of reference summaries. Skip bigram is a two ordered but not necessarily adjacent terms.

2.1.3.2 Extrinsic Evaluation

Extrinsic evaluation assesses the output summary based on how it influences the completion of some other task such as text classification[35], information retrieval [35][58], question answering [37] etc. Therefore a summary is considered as good if it is helpful to some other tasks. We will now discuss the extrinsic approaches used in two different studies to convey the idea behind this evaluation type.

TIPSTER SUMMAC [35] is an evaluation system that performs two different extrinsic evaluation tasks. The first task relates to the real-world activity of a U.S. Government information analyst working with an IR system to quickly determine the relevance of a retrieved document. Given a document (which could be a summary or a full-text source - the subject was not told which), and a topic description, the human subject was asked to determine whether the document was relevant to the topic. Thus, an indicative summary would be “accurate” if it accurately indicated the relevance or irrelevance of the corresponding source. In the second task, the human subject was given a document, which could be a generic summary or a full-text source. He then had to choose a single category out of five categories to which the document was relevant, or else choose ”none of the above”.

In [37] the authors picked four Graduate Management Admission Test (GMAT) reading comprehension exercises. The exercises were multiple choice, with a single answer to be selected from answers shown alongside each question. They measured how many of the questions the subjects answered correctly under three different conditions. Firstly, they were presented the original passages. Secondly, the were shown an automatically generated abstract and then a human generated one. The summary was evaluated by comparing the answers in the different conditions.

2.2 Sentiment Analysis

With the explosive growth of social media (i.e. reviews, forum discussions, blogs and social networks) on the Web, individuals and organizations are increasingly using public opinions in these media for their decision making. Consequently, it has become essential not only to distill text’s information but also to find a way to extract the underlying sentiment of a given text. This is why sentiment analysis proves its great usefulness.

Sentiment analysis or opinion mining is the computational study of people’s opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes. The most common task that sentiment analysis attempts to deal with is *sentiment classification*. Simply stated, sentiment classification is the task of processing a piece of text in order to identify whether the underlying sentiment is positive or negative. Thus, it can be modeled as binary classification problem. Throughout this section we will use the phrases “sentiment analysis” and “sentiment classification” interchangeably, although the former is considered to have a broader meaning. Existing approaches to sentiment analysis can be grouped into two main categories: *knowledge-based* and *statistical* or *learning-based*.

2.2.1 Knowledge-based Sentiment Analysis

A knowledge-based approach on sentiment analysis, sometimes called rule-based, is one which uses rule of thumb, i.e. heuristics, to determine sentiments. More precisely, the features of a given text are compared against words in a lexicon whose sentiment values are decided prior their use [14][36]. Some knowledge bases not only contain apparent affect words, but also assign arbitrary words a probable ”affinity” to particular emotions [55]. The appropriate handling of negation, syntax and POS tags plays a major role in the accuracy of these methods.

2.2.2 Learning-based Sentiment Analysis

As mentioned earlier, sentiment classification obviously can be formulated as a supervised learning problem with two or three classes, positive, negative and neutral. Training and testing data used in the existing research are mostly product reviews, which is not surprising due to the above assumption. Since each review already has a reviewer-assigned rating (e.g., 1 to 5 stars), training and testing data are readily available. For example, a review with 4 or 5 stars is considered a positive review, a review with 1 or 2 stars is considered a negative review and a review with 3 stars is considered a neutral review. Any existing supervised learning methods can be applied to sentiment classification, e.g., Naive Bayes classifier and Support Vector Machines (SVM). Pang et al. [42] took this approach to classify movie reviews into two classes, positive and negative. It was shown that using unigrams (a bag of individual words) as features in classification performed well with either naive Bayesian or SVM. More details on the existing literature on the domain of reviews will be given in the next chapter (3).

2.3 Clustering

Cluster analysis or *clustering* is the task of dividing a set of data points or observations into a number of groups, called *clusters*, such that data points belonging to the same group are more “*similar*” to each other than to those in other groups. The primary characteristic of clustering is that it is an unsupervised learning method. An unsupervised learning method is a method in which we draw references from datasets consisting of input data that are not coupled with labeled responses. Generally, it is used as a process to extract meaningful structure, generative features, and groupings inherent in a set of examples. Clustering is a very important task as it determines the intrinsic grouping among a set of unlabeled data. It has been utilized in a wide spectrum of fields, such as marketing [25], biology[4] and city planning [15].

2.3.1 K-Means Clustering

K-means clustering is one of the most popular clustering algorithms as it is widely used in many applications. Such applications include image segmentation, clustering genetic data, news articles clustering, e.t.c. The goal of this algorithm is to find groups in the data, with the number of groups determined by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided.

The most common algorithm uses an iterative refinement technique. Due to its ubiquity, it is often called “the k-means algorithm”. It is also referred to as Lloyd’s algorithm [32], particularly in the computer science community. Assuming we have inputs $x_1, x_2, x_3, ..., x_n$ and a value of K, the algorithm works as follows:

1. We randomly pick K cluster centers, called centroids. Let’s assume these are $c_1, c_2, c_3, ..., c_k$. C is the set of all centroids

$$C = \{c_1, c_2, c_3, ..., c_k\}$$

2. In this step we assign each input value to closest center. This is done by calculating Euclidean(L2) distance between the point and the each centroid.

$$\underset{c_i \in C}{\operatorname{argmin}} \, dist(c_i, x)^2$$

   where $dist(.)$ is the Euclidean distance.

3. In this step, we find the new centroid by taking the average of all the points assigned to that cluster.

$$c_i = \frac{1}{\mid S_i \mid} \sum_{x_i \in S_i} x_i$$

   where S_i is the set of all points assigned to the i^{th} cluster

4. In this step, we repeat step 2 and 3 until none of the cluster assignments change. That means until our clusters remain stable, we repeat the algorithm.

2.3.1.1 Optimal Number of Clusters

Unfortunately, there is no definite answer to the question of what is the optimal value of K. The optimal number of clusters is somehow subjective and depends on the method used for measuring similarities and the parameters used for partitioning. A simple and popular solution relies on inspecting the dendrogram produced using hierarchical clustering to see if it suggests a particular number of clusters. Unfortunately, this approach is rather subjective. Another simple solution to this problem is the *Elbow Method*. After running the algorithm for different values of K (say $K = 10$ to $K = 1$) and plot the K values against SSE(Sum of Squared Errors), we select the value of K for the elbow point as shown in the Figure 2.4. Because it is often ambiguous and not very reliable, other approaches for determining the number of clusters such as the Silhouette [49] method are preferable.

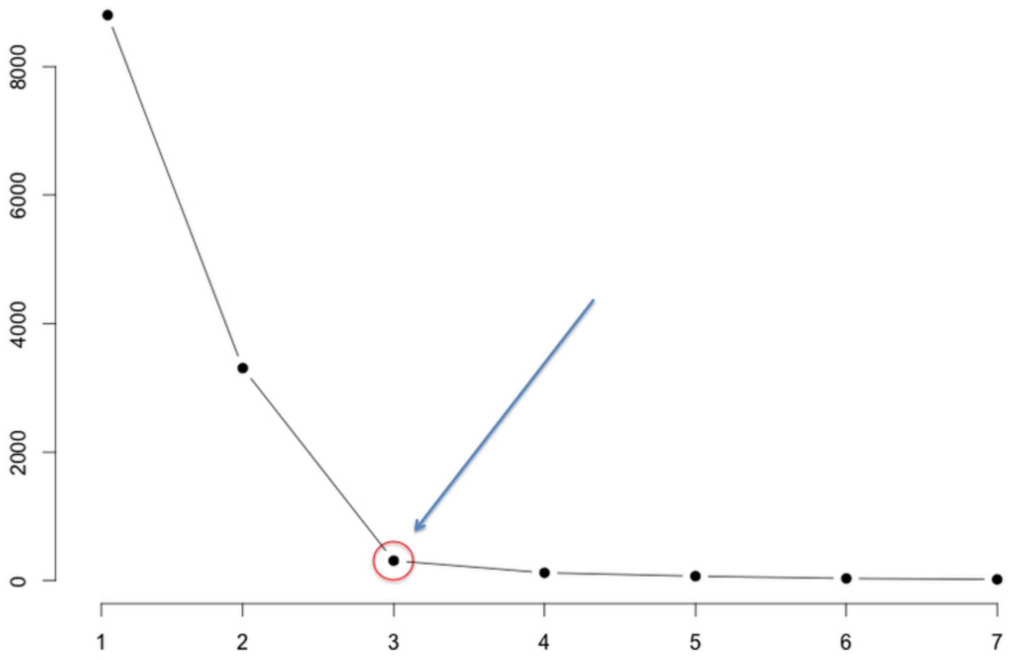


Figure 2.4: Elbow graph illustrating the optimal number of clusters K

2.3.2 Hierarchical Clustering

The hierarchical clustering technique(also called hierarchical cluster analysis or HCA) is a very popular clustering technique in the domain of NLP [53]. The fundamental idea behind it is that it seeks to build a *hierarchy* of clusters. This clustering technique is divided into two types, which :

- *Agglomerative*: This is a ”bottom-up” approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

  The basic steps algorithm of agglomerative clustering are the following:

  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. Merge the two closest clusters and update the proximity matrix
  4. Repeat step 3 until only a single cluster remains

- *Divisive*: This is a ”top-down” approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

In order to determine which clusters should be combined (for agglomerative), or where a cluster should be split (for divisive), a measure of dissimilarity between sets of observations is needed. In most methods of hierarchical clustering, this is accomplished by using an appropriate metric (a measure of distance between pairs of observations), and a linkage criterion which indicates the dissimilarity of sets as a function of the pairwise distances of observations in the sets.

Some commonly used metrics for hierarchical clustering are the Euclidean distance, the Manhattan distance, the Maximum distance and the Mahalanobis distance. Some commonly used linkage criteria between two sets of observations A and B are complete-linkage clustering and single-linkage clustering. In the former approach, the distance between clusters equals the distance between those two elements (one in each cluster) that are farthest away from each other, whereas in latter one the distance between two clusters is determined by a single element pair, namely those two elements (one in each cluster) that are closest to each other.

Another beneficial feature of hierarchical clustering is that the generated hierarchy can visualized using a dendrogram, as illustrated in Figure 2.5. A dendrogram is a tree-like diagram that records the sequences of merges or splits.

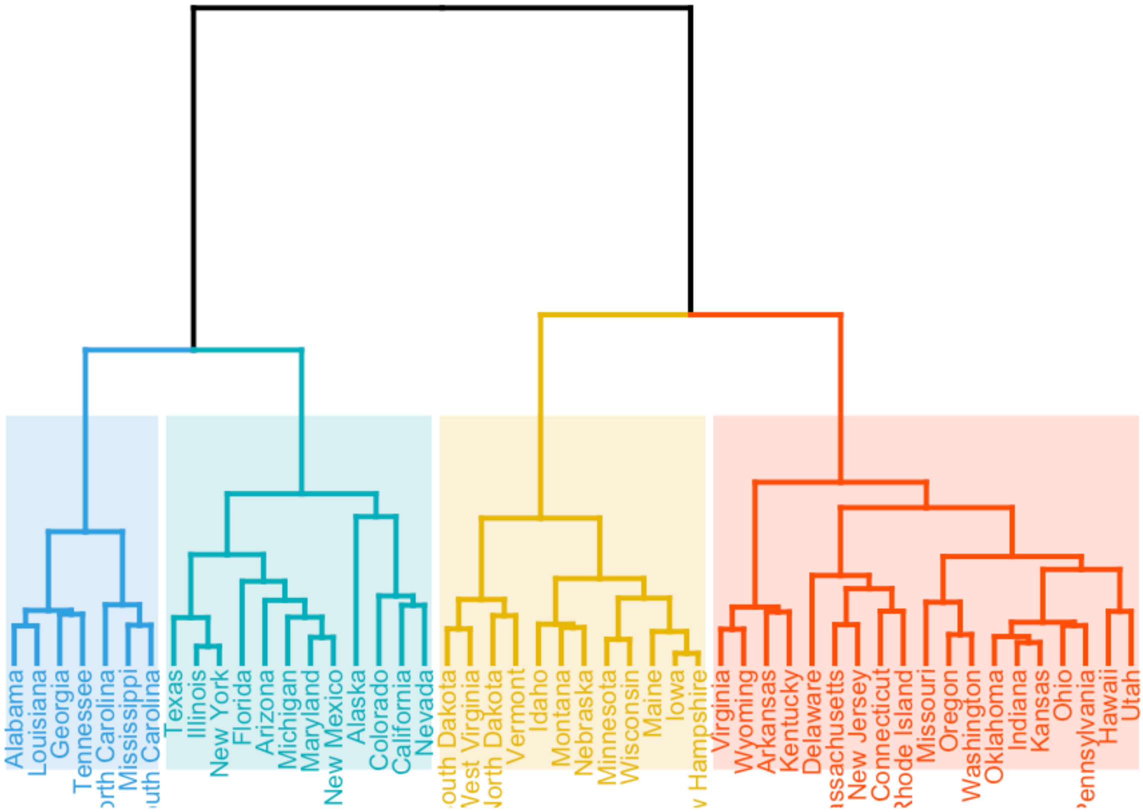


Figure 2.5: A dendrogram visualizing hierarchical clustering

3. RELATED WORK

The importance of analyzing user reviews has drawn a great deal of interest among researchers. There has been a plethora of studies presenting different approaches on sentiment analysis as well as summarization of user reviews from various domains, such as Amazon product reviews, IMDB movie reviews and hotel reviews. We will now present various studies that attempt to tackle the review summarization task based on the concepts and strategies described in the previous chapter.

Turney [59] suggests a PMI-based approach for classifying reviews as *recommended* or *not recommended*. His approach consists of three main steps: phrase extraction from a given review by applying POS tagging, orientation estimation for each phrase based on the PMI score between the phrase and the words *excellent* and *poor*, review labelling based on the average orientation of its phrases. In [23] [24] Hu and Liu present an approach for generating a feature-based opinion summary from a large number of reviews. They propose promising techniques for each stage of their method, which aims at classifying sentences rather than each review as a whole. They present, among others, an iterative algorithm for identifying the underlying sentiment of a word using a small set of seed adjectives combined with WordNet's synset relations [36].

Similarly, Zhuang et al. [65] propose their approach for producing feature-based summaries on the domain of movie reviews. They make use of regular expressions and WordNet for feature mining and opinion word identification respectively. POS-tag patterns are used in order to identify feature-opinion pairs. Their experiments produced lower precision and recall scores than the results obtained in the domain of product reviews [24], mainly because of the peculiarity of movie reviews. Instead of just producing an opinion summary, Jmal and Raiz [27] assess the opinion strength on a product and its features, while exploiting Twitter posts to highlight the most relevant features more effectively. A more recent work [48] identify aspect-based statements from product reviews through patterns extracted from dependency parse trees.

A number of studies have proposed supervised learning approaches by training sentiment classifiers. Pang and Lee [43] attempt to classify movie reviews using Naive Bayes, SVM and Max Entropy and multiple feature combinations. Their results indicate that ML techniques on sentiment classification can achieve high accuracy when feature presence instead of feature frequency is used. In [60] the authors attempt to recognize phrase-level contextual polarity by using a two-step process. They firstly classify expressions as polar or neutral and subsequently classify the polar ones as positive, negative or neutral.

A novel flexible summarization framework, called Opinosis, is proposed by Ganesan et al. in [16]. It is a graph-based approach that represents review text as a graph with unique properties and identifies various paths in it, each one acting as a candidate summary. The SMACk system [11] is an argumentation-based opinion mining framework which detects and extracts aspects coupled with polarities from documents by creating an argumentation graph.

Topic modeling has been widely used as a basis to perform extraction and grouping of aspects. Titov and McDonald [57] introduce a Multi-grain LDA model which models global topics and local topics that capture ratable aspects and properties of reviewed items respectively. Their method is particularly suited to aspect extraction from reviews as it does not only identify important terms but also clusters them into coherent groups. In [33] aspects in eBay's sellers' feedback comments are discovered using PLSA-based techniques. The authors try to group aspect terms that tend to co-occur in comments. Jo and Oh [28] proposed two generative models to discover aspects and sentiment in reviews. Sentence-level LDA (SLDA) constrains that all words in a single sentence be drawn from one aspect. Aspect and Sentiment Unification Model (ASUM) unifies aspects and sentiment and discovers pairs of aspect, sentiment, which we call senti-aspects.

Recent advances in computing hardware together with the increased availability of data have led to the ubiquitous use of neural networks as an effective tool for producing summaries and identifying sentiment in text.

In [10] the authors develop a deep convolutional neural network that exploits from character- to sentence-level information to perform sentiment analysis of short texts. Conversely, in [52] the authors construct a network with just a single convolutional layer and also presented a new model for initializing the weights of the network. A novel deep learning approach to aspect extraction is shown in [46] where a 7-layer CNN is combined with linguistic patterns. Using the dataset made available by Pontiki et al. [45], the authors in [50] propose a hierarchical LSTM-based approach for that task of aspect-based sentiment analysis whilst a Cascaded-CNN architecture is presented in [61].

Despite the widespread appeal of video games, there has been little discussion on the domain of game reviews. Yauris and Khodra [62] propose an aspect-based summarization system for Steam reviews. They employ a modified double propagation (DP) algorithm for extracting aspect-sentiment word pairs. Following this, they use a seed list and word similarity to categorize aspect terms into groups, thus producing an aspect-based summary. In [3] the authors developed a robust model using Gradient Boosting Machine algorithm to predict the Steam review helpfulness.

Most works so far have relied on supervised learning methods by utilizing annotated datasets [45]. As there is currently no existing dataset for aspect-based game review summarization, our work is designed with the aim to minimize the role of supervision. Furthermore, in our undertaking we take into account the following idiosyncrasies of the game setting:

- The folksonomy (dynamic) nature of the terms used in comments. Each genre and possibly game appear to be mapped to specific expectations by its users and, consequently, aspects that the users comment on. There appears that a fixed ontology or aspect set would not be sufficient to describe the aspects of all the game genres that get published over time. This is further accentuated by the fact that hybrid games, combining genres, become a common sight.

- The possible vagueness of aspects, based also on the above comment. We thus examine whether aspects identified through an unsupervised process can be consistently labeled by humans.

- The fact that it is important to hold not a single response of sentiment, but understand the full spectrum of sentiments of players. This means that a single ”positive”, ”negative” or ”neutral” answer to how people have commented for an aspect is only a secondary finding. The distribution of comments over these three labels is more interesting and useful, and may be the primary aim of a game review summarization process.

4. PROBLEM DEFINITION

Video game reviews are likely to discuss several aspects of the game, such as graphics, gameplay, community e.t.c. Expert/professional reviewers tend to follow specific patterns of summarizing reviews, utilizing the above established aspects. They also provide an overall recommendation and possibly grade, while oftentimes they highlight ”pros” and ”cons” of the reviewed game. These pros and cons essentially designate the specific, non-formalized, aspects of a game (and possibly of other games of its genre). On the other hand, we should note that the expert reviewers only summarize their own review, which forms a single-document setting. In our case, we examine an approach more suited for a multi-document summarization setting, where several texts (reviews) are to be summarized in a single summary.

To take into account the above “gold standard” human approach, while tackling the multi-document differentiation, we formulate the problem as follows:

Given a set of game reviews $R = \{r_1, r_2, ...\}$ for a game g, the game review summarization task tries to perform the following steps:

aspect identification identify the set A of aspects of the game, that the reviews R comment on.

aspect labeling map each aspect A to a label set $L_A = \{l_1, l_2, ...\}$, where each of l_i is a (possibly weighted) term.

sentiment extraction extract a sentiment distribution S_A which will be of the form $S_A = \{s_{\text{positive}}, s_{\text{neutral}}, s_{\text{negative}}\}$, describing the user sentiment over each aspect A.

highlight extraction extract the subset $P \in A$ of ”pros”, where $s_{\text{positive}} > s_{\text{negative}}$ and the subset $C \in A$ of ”cons”, where $s_{\text{positive}} < s_{\text{negative}}$.

review summary generate a single summary $\mathbb{S}$ containing all the above information.

Within this work we focus on the *aspect identification*, *aspect labeling* steps. We also touch the *sentiment extraction* and *highlight extraction*, providing baseline implementations. In the following paragraphs, we elaborate on the suggested methods that implement these steps, as demonstrated in Figure 4.1.

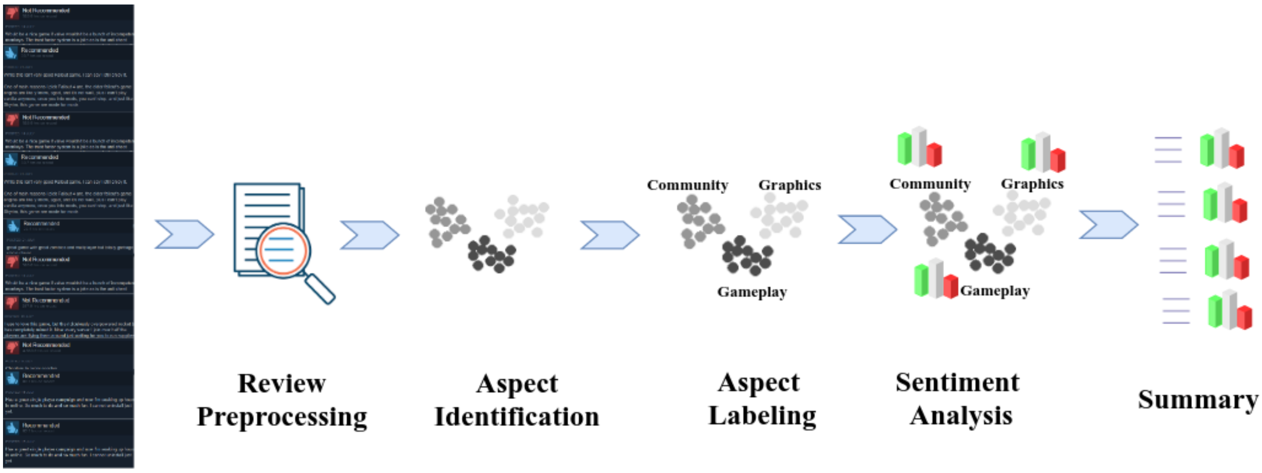


Figure 4.1: Pipeline steps

5. PROPOSED METHOD

In this chapter we demonstrate our processing pipeline that implements the individual steps as described in Chapter 4.

5.1 Method

As illustrated in Figure 4.1, our proposed implementation consists of five individual steps. In a nutshell, given a game, we first fetch a set of reviews, which are subsequently split into sentences. After having processed each sentence, we represent them using a bag-of-words (BOW) model. Sentences are then clustered followed by sentiment analysis on each cluster. A more detailed description of each step is given in the following subsections.

5.1.1 Review Preprocessing

It is common knowledge that words that appear in documents often have many structural variants. Thus, before any NLP task can be applied on the documents, data preprocessing techniques are to be utilized. Such preprocessing techniques are essential in order to convert our initial documents into a suitable form which will increase the effectiveness of any NLP system. In our approach, we apply sentence segmentation, tokenization, stopword removal and lemmatization on the text data (i.e. reviews). For this purpose we employ spaCy v2.0[1], an open-source software library for advanced NLP. Below is a detailed description of each one of these tasks.

Sentence Segmentation
Tokenization
Stopword Removal
Lemmatization

Figure 5.1: Text preprocessing steps

5.1.1.1 Sentence Segmentation

Although sentence segmentation is not a typical preprocessing step in every NLP task, it is essential for our study as we aim to cluster these sentences in the next step of our pipeline. Sentence segmentation is the process of determining the longer processing units consisting of one or more words. This task involves identifying sentence boundaries between words in different sentences. Since most written languages have punctuation marks which occur at sentence boundaries, sentence segmentation is frequently referred to as sentence boundary detection, sentence boundary disambiguation, or sentence boundary recognition [41]. spaCy performs this task by parsing the text and using a pretrained model based on which the sentences get splitted.

5.1.1.2 Tokenization

After having segmented each review into sentences tokenization is applied to each one of them. Tokenization is the process of splitting a sequence of text up into words, phrases,

[1] https://spacy.io/

symbols or other important elements. These elements are usually called *tokens*. The purpose of this task it to create a set of tokens on which further analysis can be carried out. Despite being considered a relatively easy task compared to other NLP tasks, it can be quite challenging, especially when dealing with Chinese or Thai text, where words are not separated by white space.

5.1.1.3 Stopword Removal

Stopwords constitute a division of natural language. These are words that are deemed irrelevant because they occur very frequently in textual data without contributing to the content or the context of the text. Thus, they are eliminated from input text, leading to lower dimensionality of term space. The most common stopwords found in text documents are articles, prepositions and pronouns, such as "*and*", "*with*", "*an*", e.t.c. Moreover, numbers and auxiliary verbs are also treated as stopwords.

Researchers typically use standard stopwords lists obtained from texts in many different domains. This is something of a pitfall because many stopwords are usually determined by the specific corpus of documents we are dealing with. In our case, words like "*game*" or "*play*", that appear very frequently, do not provide any meaningful information regarding what is being discussed by the reviewers. We thus enrich spaCy's default stopword list with the following words: "*game*", "*people*", "*thing*", "*play*", "*review*".

Additionally, given a set of reviews of a specific game, there are many game-specific terms that tend to appear very frequently among the reviews. Such terms can be the name of the game, the name of the development company, the game's genre, as well as abbreviations related to that specific game. We decided to treat these terms as stopwords because, although they appear frequently, they do not contribute to the identification of the game aspects being discussed. Table 5.1 shows two list of game-specific words for two different games.

Table 5.1: Game-specific stopwords

DiRT Rally	"dirt", "rally", "collin", "mcrae", "codemasters"
Age of Empires II HD	"age", "empire", "rts", "rt"

5.1.1.4 Lemmatization

Lemmatization is the process of finding the normalized form of a word. More precisely, the aim of this process it to group together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma. Lemma is the canonical or dictionary form of a set of words. For example, the words "*studied*", "*studies*" and "*studying*" all have the same base form, "*study*", as illustrated in Figure 5.2.

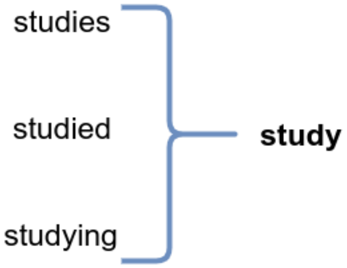


Figure 5.2: Lemma of the verb "to study"

Note that lemmatization and word stemming are very similar tasks. They both aim to reduce the inflectional forms of each word into a common base or root. However, they work differently. The latter works by truncating the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word. On the other hand, the former takes into account the intended part of speech (POS) as well as the context of a word in a sentence in order to correctly identify the word's base form. Table 5.2 highlights the different results produced by these two tasks. Note that lemmatization ensures that the reduced word is again a dictionary word.

Table 5.2: Lemmatization and Stemming

Word	Lemmatization	Stemming
was	be	wa
studies	study	studi
playing	play	play

Before we move on to the *Aspect Identification* stage, it is worthwhile noting that due to the syntactical peculiarities found in user-generated reviews, we also had to perform some extra preprocessing on the text data. This involved repeating phrases and multiple whitespace removal as well as filtering out terms consisting of non-ASCII characters. Table 5.3 demonstrates the result of the preprocessing stage after being applied on a review excerpt.

Table 5.3: Preprocessed review excerpt

Global offensive is not the key evolution point that we were hoping for
and the response from the community often reflects this view.
It is still however a glorious experience that sets a benchmark for all multiplayer shooters.

Sentence I	*"key", "evolution", "point", "hope", "response", "community", "reflect", "view"*
Sentence II	*"glorious", "experience", "set", "benchmark", "multiplayer", "shooter"*

5.1.2 Aspect Identification

In this step we attempt to extract the aspects of a game that are mainly discussed by the reviewers. This can be quite challenging as video game aspects can be either explicitly or implicitly mentioned in a review text. For example, the sentence "*Easily my favorite game with realistic graphics*" clearly expresses an opinion about the aspect "*graphics*". On the contrary, the sentence "*The grenade explosions are so fake*" does not mention the word "*graphics*" but it obviously refers to the graphics of the game, or possibly the physics engine.

5.1.2.1 Text Representation

Text representation plays a major role in the effectiveness and accuracy of clustering algorithms [1]. In our approach we represent each processed review as a set of sentences, which are in turn represented as a set of terms. Before a clustering algorithm can work on these sentences, it is essential to convert them into real-valued vectors. As mentioned in the *Theory* (Chapter 2), tf-idf representation is the most popular term-weighting scheme today. The primary reason for this is that tf-idf reduces the importance (i.e. weight) of common terms in the review corpus, ensuring that more descriminative words, namely, words with relatively low frequencies in the corpus, are assigned a greater weight. We thus decided to make use of this representation method.

We also examined whether a word embedding would provide better results. However, the BOW representation method appeared to give more coherent results in the clustering step. It is very likely that the short length of sentences combined with the large vocabulary size has led to this finding. Thus, capturing the context of each sentence via a sentence2vec method [8] can be challenging, probably requiring more specific training data.

5.1.2.2 Clustering

After having converted the sentences into tf-idf vectors, we are now ready for clustering them, with the aim of producing a cluster-wise summary. Note that we are referring to the sentences obtained from the reviews of a single game, rather than the reviews of all games. The intuition behind this approach is that the produced clusters will exhibit the most salient aspects appearing in the reviews of this game. It was decided that the best method for this study was to use the k-means algorithm. We opted for the this method on the basis of the considerably lower time complexity of k-means compared to a hierarchical clustering approach. In Chapter 6 we elaborate on our decision regarding the number of clusters (5). Table 5.4 lists the most frequent terms appearing in each cluster. As anticipated, the words are semantically close to each other and they seem to represent a specific game aspect. We choose these terms to label the aspect cluster.

Table 5.4: Most frequent words in each cluster

Cluster 1	story, character, mode, main, mission
Cluster 2	money, spend, earn, waste, real
Cluster 3	time, fun, long, loading, screen
Cluster 4	reason, ban, permanently, innocent, account
Cluster 5	support, great, bad, community, good

5.1.3 Aspect Labeling

Another way to label the clusters is to map them to a predefined set of aspect labels, based on gold-standard (i.e. professional) reviews. In Table 5.5 we show an indicative, human-provided mapping between terms and predefined aspect labels.

Based on the mapping illustrated in Table 5.5, each sentence can be classified into one of the aspects, by identifying the prevalent aspect of the sentence's words (i.e. terms). For instance, if the majority of the terms in a sentence belong to the "*community*" aspect, then the sentence is given this label. It should be noted that the term lists needs to be slightly modified based on the game's genre. The reason is that the terms, for example, that illustrate the "*gameplay*" aspect of a first-person shooter game differ notably from those of a puzzle or an adventure game. This fact highlights the intricacies of the game review task, where secondary (latent) variables alter the aspect descriptions.

Table 5.5: Selected terms for each aspect

Graphics	graphic, visual, look, aesthetic, animation, frame
Gameplay	mission, item, map, weapon, mode, multiplayer
Audio	audio, sound, music, soundtrack, melody
Community	community, support, toxic, friendly
Performance	server, bug, connection, lag, latency, ping, crash, glitch

By aggregating the aspect proportions exhibited by the sentences, we end up with the proportions that each cluster exhibits. Unfortunately, it was not possible to end up with an one-to-one relationship between the clusters and the predefined game aspects. This apparent lack of correlation can be attributed to the wide range of game characteristics being discussed by the reviewers coupled with the fact, that k-means clustering on tf-idf vectors is not a context-aware method. In Table 5.6 we provide a few indicative sentences from a specific aspect cluster. Then, in Figure 5.3 we show how the sentences of two different clusters led to two different distributions over the predefined aspects.

Table 5.6: Indicative sentences from cluster on "The Elder Scrolls V: Skyrim" game

"You get attached to so many characters and the world is amazing."
"Probably the best open-world rpg out there."
"The vast open world is absolutely stunning."
"If you're looking for a way to waste massive amounts of time just trolling around a world play this."

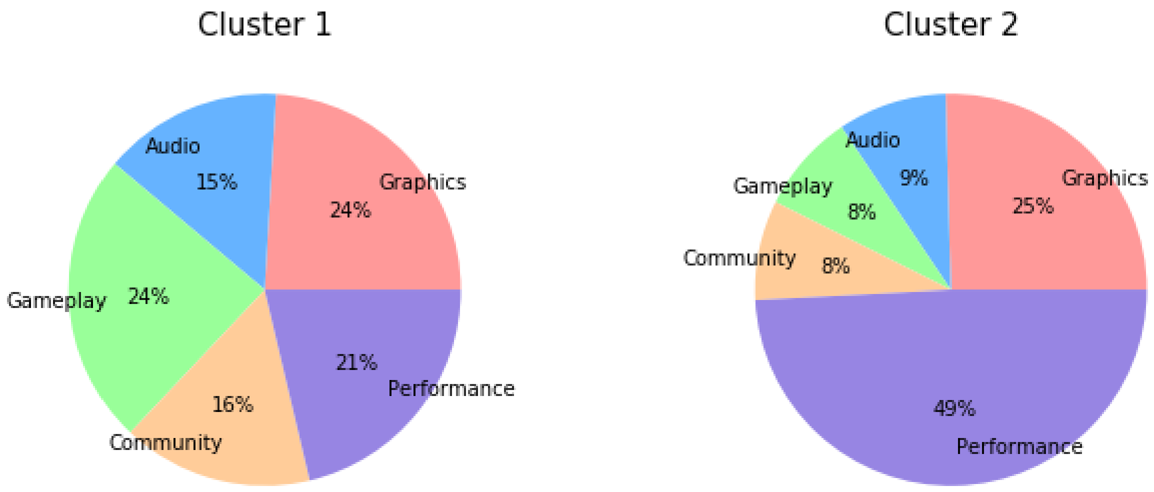


Figure 5.3: Aspect proportions exhibited in two clusters on “The Elder Scrolls V: Skyrim” game

5.1.4 Sentiment Analysis

The sentiment analysis step focuses on identifying the underlying sentiment that pervades each cluster. Since our clusters consist of sentences we perform sentence-level sentiment analysis.

As there is no sentiment analysis dataset specific to our domain, we decided to use VADER [26], a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. Interestingly, VADER can tell us how positive, negative and neutral a given sentence is, instead of just classifying the sentence in a single category. VADER combines a dictionary of lexical features to sentiment scores with a set of five heuristics (e.g. punctuation, degree modifiers, e.t.c). Consequently, by calculating the three sentiment scores for each sentence in a cluster and averaging, we can get the distribution of the reviewers’ sentiment for this cluster. Figure 5.4 presents the sentiment proportions of each cluster on “The Elder Scrolls V: Skyrim” game.

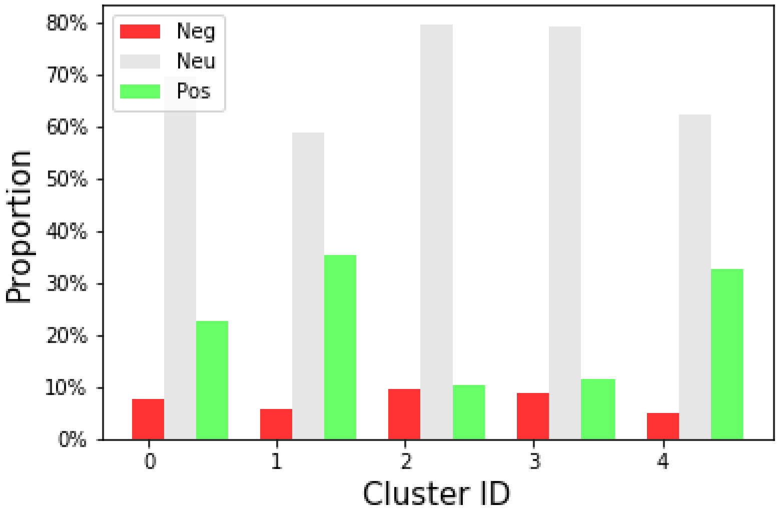


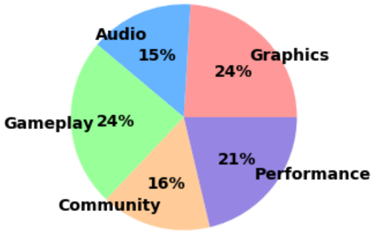
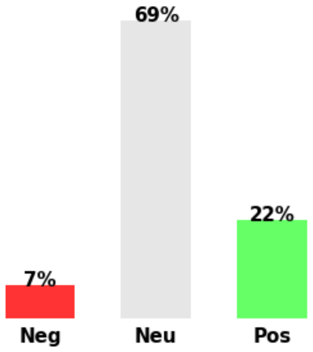
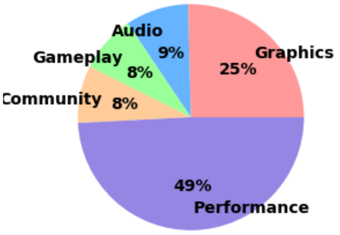
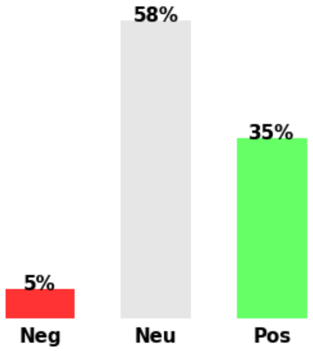
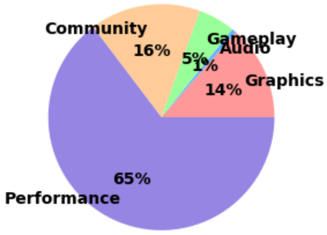
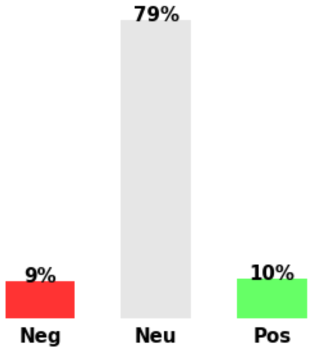
Figure 5.4: Sentiment proportions of the clusters on “The Elder Scrolls V: Skyrim” game

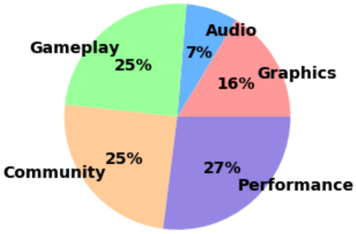
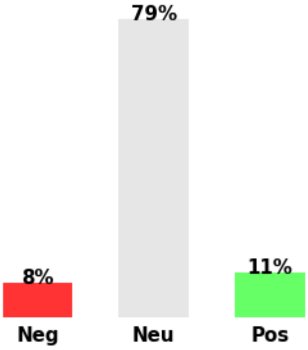
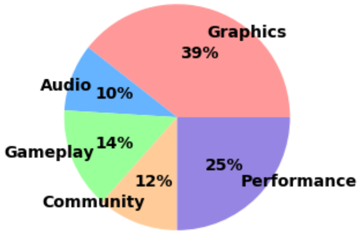
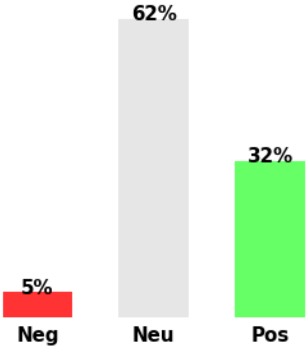
5.1.5 Final Output

The final output of the process is an aspect-based summary of a set of reviews of a specific game. This summary contains the following information for each produced cluster: Equidistant *indicative sentences*, starting from the one that is closest to the cluster’s center, i.e. the one that best describes the cluster’s theme, and moving towards the most distant one. *Aspect proportions* indicating how much each game aspect is exhibited in the cluster. *Sentiment proportions* illustrating the sentiment distribution of the sentences assigned to the cluster.

It is apparent from Table 5.7 that we have succeeded in providing the potential buyer with a well-structured summary. This kind of summary enables the reader to retrieve the most relevant information about the game according to his/her need very easily. On the contrary, the initial set of numerous unstructured reviews is considerably difficult to deal with, let alone the fact that it does not provide any information regarding the underlying sentiment or the game’s aspects.

Table 5.7: Output Summary

Sentences	Aspect Proportions	Sentiment Proportions
Valve takes 75% that is absolutely ridiculous in my opinion you are practically splitting the mod community with this policy. *There are so many little things to do even if you beat the main quest.* *Let's see if you want a story-rich and choices matter tagged game.* … *The skyrim game itself is a loved product by bethesda softworks.* *This is a sample sent to test the width of the cell.*		
I've only played skyrim for a few hours. *Tons of hours of gameplay.* *Awesome game hour's of fun and enjoyment.* … *You will spend hundreds of hours doing a bunch of quests excluding all of dlc..* *Spanning over 200 hours of gameplay there are quests dungeons enemies and vast beautiful landscapes for you to explore.*		
This game is the best game I have ever played. *Honestly this is probably the best game ever made with probably the best modding community in any game....ever.* *Graphics gameplay and soundtrack are all very impressive and it's one of the best games I have ever played.* … *The game provided a good use of textures for back in 2012 even now in 2015 the graphics still holds and is not too bad.* *Skyrim is already 4 years old but its one of the best games I have ever played + mods make everything more interesting.*		

Skyrim has always been game that I like. *Are 20 euros like come on.* *I liked that it was a 1 player game and that people couldnt beat a battle against you.* *…* *You may be sitting at your computer one day thinking that you don’t really feel like playing skyrim but as soon as you launch the game you get sucked in.* *Collisions are some times imminent even while using special tools like wrye bash.*		
Paid mods really valve. *It’s a great game and I like to see others whine about paid mods :D.* *And thats my experience with this awesome game that I have not played too much because mods make it crash way too often.* *…* *2017 update: still very immersive and provides endless fun and quality of life with the mods.* *Greenlight and mod shop are inventions of satan himself.*		

6. EXPERIMENTS

In the last chapter we presented a baseline implementation of a game review summarization system. We will now propose an evaluation procedure with the aim of ascertaining the attainability of the review summarization subtasks. We will begin by describing the dataset used in our experiments, then move to explaining the different experimental setups used to answer our research questions. We then also an interpretation of the evaluation results in order to relate them to the previously stated research questions.

6.1 Dataset

For our experiments we used the Steam review dataset gathered by Zuo [66]. It consists of more than 7 million reviews obtained via Steam's API. Each review text comes with a plethora of features concerning both the game being reviewed and the reviewer. For our experiments, we only utilized the game's ID, the review itself and the number of "helpful" votes the review has received by other community members. In our experiments, to speedup the clustering process we used only a sample of the reviews of each game consisting of the 10,000 most voted reviews.

6.2 Experimental Setup

As far as the clustering setup is concerned, it understandable that determining the appropriate number of clusters k can sometimes be one of trickier tasks. In order to deal with this issue, we attempted to use the elbow method and we also performed Silhouette analysis [49]. Nonetheless, no appreciably optimal k was found by the two methods. However, this is not particularly surprising, in light of the fact that the reviews address a wide range of themes. Thus, the more clusters we create, the higher the coherence will be. Considering, though, that we aim to produce a digestible aspect-based summary using these clusters, it would be irrational to produce too many of them. For this reason, we decided to work with 5 clusters.

In order to reach a sound conclusion we have performed an empirical evaluation with four different human evaluators. Before describing our evaluation process, we remind the reader of our main aim which is to provide an evaluation process for game review summarization. Given this requirement, we pose the following reasearch questions that we attempt to answer by interpreting the results of the evaluation procedure:

- We firstly attempt to find out whether a given cluster is coherent enough to be described by a representative subset of its sentences.
- Additionally, we investigate whether humans are able to consistently label a given aspect cluster
- Lastly, we examine the dynamic nature of aspects in our review setting.

This study allows us to find answers to the aforementioned research questions and also to understand whether steps of the problem, as formulated in Chapter 4, can be evaluated consistently and find answers to the aforementioned research questions. For the final output of the whole summarization pipeline we expect that standard summary evaluation methods, such as MeMoG [18] and ROUGE [30] will be useful.

We asked the help of 4 evaluators, who were fluent in the English language. The evaluators were given a set of 20 sentences fetched from each of the five clusters of three different games (for a total of 15 clusters). We also opted for different genres in order to examine the inter-genre differences with respect to the terms used for describing game aspects. They were then asked to read each set of sentences and complete the following tasks:

- Select up to n representative sentences from the aspect cluster to represent/summarize the cluster. The idea behind this task is to show whether the cluster was coherent enough to be described by a representative subset of its sentences. The lower the number of representative sentences one would need to use to represent the cluster, the higher the coherence of the cluster.
- Describe the theme of each set using 3 to 5 (possibly multi-word) terms. This task aims to see whether humans can consistently label a given aspect cluster. If so, then the agreed wording(s) can be considered gold-standard, similarly to a Pyramid evaluation [38].
- Select one or more predefined terms (gameplay, graphics, audio, community, performance, overall, other) that best describe the aspect, according to the opinion of the human evaluator. We also allowed the user to select “other” as an option, to examine whether a significant number of aspects go beyond the predefined ones. This would indeed indicate the dynamic nature of aspects in the game review summarization setting.

6.3 Results

Moving on to the results section, we are now ready to interpret the results obtained by the evaluation process with the aim of finding answers to our previously stated research questions.

In the “select representative sentences” task, we quantify how many sentences on average were selected by the evaluators to represent the cluster. We expect that the lower the number, the better the coherence of the cluster. In Table 6.1 we see, for each cluster, the average number of sentences selected as representative by the users, plus the standard error. We see that, given 20 sentences, the users selected on average from 4 to 9 representative sentences.

Table 6.1: Average representative sentences per aspect

ClusterID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Mean	9.75	7.00	9.00	7.50	5.25	9.25	5.00	5.75	7.00	4.25	8.50	4.25	5.00	9.00	8.00
+/- Std. Err	2.14	1.68	2.42	2.02	1.80	2.17	1.73	1.25	1.22	1.60	2.78	1.31	1.08	1.47	2.48

In the “describe the theme” task, we examine whether humans can assign consistent labels in an open terminology setting (i.e. without limiting the possible labels). To measure the agreement here we post-processed their terms, semi-automatically creating equivalence classes of terms (which could also have been determined based on an embedding or a linguistic resource). Indicative equivalence classes were:

- ban; ban possible; bans
- best game; best rally game; buy; buy game; buying recommendations; described as best game; ...
- bad community; community; community bad; community sucks; low rank player behaviour bad; toxic community

We then examined, for each cluster, the number of equivalent terms that were used across all evaluators to label the specific aspect cluster. If at least 2 of the 4 evaluators utilize equivalent terms, we consider that the labeling is possible and successful. In all the 15 clusters at least one equivalence class was used consistently. In Figure 6.1 we show the consistently used equivalence classes per cluster. [1]

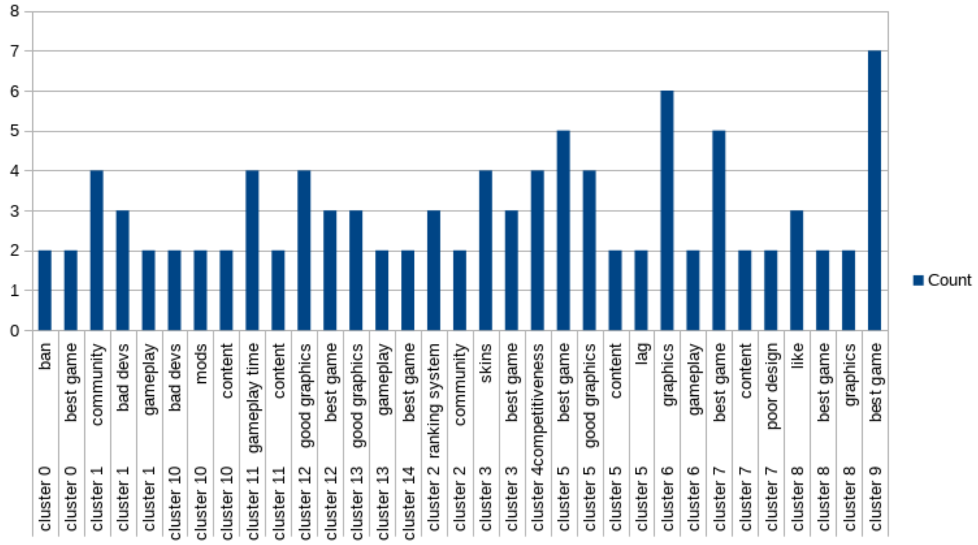


Figure 6.1: Count of usage for consistent equivalence classes per cluster

In the “select one or more predefined terms”, we examine whether human evaluators can consistently assign predefined labels to the clusters but more importantly whether the majority of the produced clusters exhibits aspects that are different from the predefined ones. Table 6.2 clearly highlights this issue, given that the “Other” option is selected by the evaluators for the majority of the clusters being examined. Interestingly, in nine clusters there were two or more people out of the four evaluators that selected the “Other” option, which stresses the fact that the aspects discussed by the reviewers go beyond the predefined ones. Another remarkable result emerging from the data is that in nine clusters there was at least one person that selected the “Overall” option. This was an expected finding given the fact that most clusters exhibit a mixture of game aspects rather than just a single one.

Table 6.2: Count of “overall” and “other” votes

ClusterID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
“Overall”	3	0	0	1	0	1	1	3	2	3	2	0	3	0	0
“Other”	3	1	3	4	3	1	0	1	1	2	2	2	1	3	3

[1] There are cases where a single evaluator used more than one term from an equivalence class, thus leading to counts over 4. However, our counting algorithm ascertains that at least 2 different evaluators will have used a term from the same equivalence class, before increasing the count.

7. CONCLUSION AND FUTURE WORK

In this thesis we discussed the domain of game review summarization. We highlighted main challenges of the domain, showing that a number of unique traits require different approaches from other summarization settings. We formally expressed a view of the task, consisting of individual steps. Based on this formal definition, we then devised a baseline procedure that receives as input a corpus of reviews of a game and outputs a well-rounded aspect-based summary for this game. Subsequently, we described a possible evaluation process, aiming to quantify the success of the aspect identification and labeling, taking into account coherence and consistent labeling from human evaluators.

This preliminary study of the game review setting opens a number of research questions that we can pursue in the future. First, how does the game genre affect the aspects of a game? Is there a causal relation that connects them? Can we perform automatic evaluation with or without human gold standard summaries? What is different from other summarization settings, concerning the evaluation?

In this work, we offer a first research step towards the emerging and useful domain of game review summarization. We understand that this first step simply highlights interesting points of focus, while providing some intuition on what is meaningful and doable from an evaluation perspective. We feel confident that this will help document and formulate a consistent setting and benchmarking process, helping related endeavors grow in the future.

ABBREVIATIONS - ACRONYMS

NLP	Natural Language Processing
SVM	Support Vector Machines
LDA	Latent Dirichlet Allocation
PLSA	Probabilistic Latent Semantic Analysis
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
ML	Machine Learning
POS	Part Of Speech
IMDB	Internet Movie Database

REFERENCES

[1] Charu C. Aggarwal and ChengXiang Zhai. A survey of text clustering algorithms. In *Mining Text Data*, 2012.

[2] Enrique Amigó, Julio Gonzalo, Anselmo Penas, and Felisa Verdejo. Qarla: a framework for the evaluation of text summarization systems. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 280–289. Association for Computational Linguistics, 2005.

[3] Mrinal Kanti Baowaly, Yi-Pei Tu, and Kuan-Ta Chen. Predicting the helpfulness of game reviews: A case study on the Steam store. *Journal of Intelligent & Fuzzy Systems*, 36(5):4731–4742, January 2019.

[4] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of computational biology*, 6(3-4):281–297, 1999.

[5] Adam Berger and Vibhu O Mittal. Query-relevant summarization using faqs. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 294–301. Association for Computational Linguistics, 2000.

[6] Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou. Summarizing email conversations with clue words. In *Proceedings of the 16th international conference on World Wide Web*, pages 91–100. ACM, 2007.

[7] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. Deep communicating agents for abstractive summarization. *arXiv preprint arXiv:1803.10357*, 2018.

[8] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[9] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[10] Cicero dos Santos and Maira Gatti. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[11] Mauro Dragoni, Celia da Costa Pereira, Andrea G. B. Tettamanzi, and Serena Villata. Combining argumentation and aspect-based opinion mining: The SMACk system. *AI Communications*, 31(1):75–95, January 2018.

[12] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1):61–74, 1993.

[13] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

[14] Andrea Esuli and Fabrizio Sebastiani. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. page 6, 2006.

[15] Vanessa Frias-Martinez and Enrique Frias-Martinez. Spectral clustering for sensing urban land use using twitter activity. *Engineering Applications of Artificial Intelligence*, 35:237–245, 2014.

[16] Kavita Ganesan, Chengxiang Zhai, and Jiawei Han. Opinosis: a graphbased approach to abstractive summarization of highly redudant opinions. In *In COLING*, 2010.

[17] George Giannakopoulos and Vangelis Karkaletsis. Autosummeng and memog in evaluating guided summaries. In *TAC*, 2011.

[18] George Giannakopoulos and Vangelis Karkaletsis. Summary evaluation: Together we stand npowered. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 436–450. Springer, 2013.

[19] Yihong Gong and Xin Liu. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25. ACM, 2001.

[20] Vasileios Hatzivassiloglou, Judith L Klavans, Melissa L Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen McKeown. Simfinder: A flexible clustering tool for summarization. 2001.

[21] Eduard H Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto. Automated summarization evaluation with basic elements. In *LREC*, volume 6, pages 604–611. Citeseer, 2006.

[22] Meishan Hu, Aixin Sun, Ee-Peng Lim, and Ee-Peng Lim. Comments-oriented blog summarization by sentence extraction. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 901–904. ACM, 2007.

[23] Minqing Hu and Bing Liu. *Mining and Summarizing Customer Reviews*. February 2004.

[24] Minqing Hu and Bing Liu. Mining Opinion Features in Customer Reviews. page 6, January 2004.

[25] Jih-Jeng Huang, Gwo-Hshiung Tzeng, and Chorng-Shyong Ong. Marketing segmentation using support vector clustering. *Expert systems with applications*, 32(2):313–317, 2007.

[26] Clayton J Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*, 2014.

[27] Jihene Jmal and Rim Faiz. Customer review summarization approach using Twitter and SentiWordNet. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics - WIMS '13*, page 1, Madrid, Spain, 2013. ACM Press.

[28] Yohan Jo and Alice H. Oh. Aspect and Sentiment Unification Model for Online Review Analysis. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 815–824, New York, NY, USA, 2011. ACM. event-place: Hong Kong, China.

[29] Karen Sparck Jones and Julia R Galliers. *Evaluating natural language processing systems: An analysis and review*, volume 1083. Springer Science & Business Media, 1995.

[30] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[31] Chin-Yew Lin and Eduard Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 495–501. Association for Computational Linguistics, 2000.

[32] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[33] Yue Lu, ChengXiang Zhai, and Neel Sundaresan. Rated Aspect Summarization of Short Comments. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 131–140, New York, NY, USA, 2009. ACM. event-place: Madrid, Spain.

[34] Inderjeet Mani. Summarization evaluation: An overview. 2001.

[35] Inderjeet Mani, David House, Gary Klein, Lynette Hirschman, Therese Firmin, and Beth Sundheim. The tipster summac text summarization evaluation. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, 1999.

[36] George Miller. *WordNet: An electronic lexical database.* MIT press, 1998.

[37] Andrew H Morris, George M Kasper, and Dennis A Adams. The effects and limitations of automated text condensing on reading comprehension performance. *Information Systems Research*, 3(1):17–35, 1992.

[38] Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the human language technology conference of the north american chapter of the association for computational linguistics: Hlt-naacl 2004*, pages 145–152, 2004.

[39] Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(2):4, 2007.

[40] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[41] David D Palmer. Tokenisation and sentence segmentation. *Handbook of natural language processing*, pages 11–35, 2000.

[42] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. page 94, 2008.

[43] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, 2002.

[44] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

[45] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California, June 2016. Association for Computational Linguistics.

[46] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49, September 2016.

[47] Vahed Qazvinian and Dragomir R Radev. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 689–696. Association for Computational Linguistics, 2008.

[48] Michael Rist, Ahmet Aker, and Norbert Fuhr. Towards Making Sense of Online Reviews Based on Statement Extraction. pages 01–12, January 2018.

[49] Peter Rousseeuw. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *J. Comput. Appl. Math.*, 20(1):53–65, November 1987.

[50] Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis. *arXiv:1609.02745 [cs]*, September 2016. arXiv: 1609.02745.

[51] Christina Sauper and Regina Barzilay. Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 208–216. Association for Computational Linguistics, 2009.

[52] Aliaksei Severyn and Alessandro Moschitti. Twitter Sentiment Analysis with Deep Convolutional Neural Networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 959–962, New York, NY, USA, 2015. ACM. event-place: Santiago, Chile.

[53] Michael Steinbach, George Karypis, Vipin Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000.

[54] Josef Steinberger and Karel Jezek. Evaluation measures for text summarization. *Computing and Informatics*, 28:251–275, 2009.

[55] Ryan A Stevenson, Joseph A Mikels, and Thomas W James. Characterization of the affective norms for english words by discrete emotional categories. *Behavior research methods*, 39(4):1020–1024, 2007.

[56] Simone Teufel and Marc Moens. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, 28(4):409–445, 2002.

[57] Ivan Titov and Ryan McDonald. Modeling Online Reviews with Multi-grain Topic Models. *arXiv:0801.1063 [cs]*, January 2008. arXiv: 0801.1063.

[58] Anastasios Tombros, Mark Sanderson, and Phil Gray. Advantages of query biased summaries in information retrieval. In *SIGIR*, volume 98, pages 2–10, 1998.

[59] Peter D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *arXiv:cs/0212032*, December 2002. arXiv: cs/0212032.

[60] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005.

[61] Haibing Wu, Yiwei Gu, Shangdi Sun, and Xiaodong Gu. Aspect-based Opinion Summarization with Convolutional Neural Networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3157–3163, Vancouver, BC, Canada, July 2016. IEEE.

[62] K. Yauris and M. L. Khodra. Aspect-based summarization for game review using double propagation. In *2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA)*, pages 1–6, August 2017.

[63] Liang Zhou, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. Paraeval: Using paraphrases to evaluate summaries automatically. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 447–454. Association for Computational Linguistics, 2006.

[64] Liang Zhou, Miruna Ticrea, and Eduard Hovy. Multi-document biography summarization. *arXiv preprint cs/0501078*, 2005.

[65] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie Review Mining and Summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 43–50, New York, NY, USA, 2006. ACM.

[66] Zhen Zuo. Sentiment Analysis of Steam Review Datasets using Naive Bayes and Decision Tree Classifier. 2018.