Mining Steam Games: A Content-Based Recommender System and Review Analysis of "The Elder Scrolls V: Skyrim"

Evangelos Tsogkas
Data Science and Machine Learning
National Technical University of Athens
Athens, Greece
Email: evaggelostsogkas@mail.ntua.gr

***Abstract*—The gaming market is growing very fast in recent years and as a result, gaming has become an extremely popular form of entertainment. Digital retail platforms make it even easier to buy and review games and Steam, the largest digital PC game distribution platform makes thousands of games available to consumers. Recommendation systems have been used widely in many industries such as movies, and now they are also an important tool for the gaming distribution platforms as it can help consumers to find games they may like and at the same time help distributors and developers to increase their sales. Additionally, due to the scale of the computer game industry, developing a successful game is challenging and user reviews can have a great impact on whether a game succeeds or not. Therefore, by studying the reviews developers can understand what their audience wants and improve the game to fit their expectations. In this work, we propose a content-based recommender system for Steam games using information such as the developers, game tags, description and number of positive or negative ratings. Finally, we analyze the reviews of a popular game with the title "The Elder Scrolls V: Skyrim" by leveraging techniques such as topic-modeling and sentiment analysis, with the goal to understand what users like and discover possible issues.**

***Index Terms*—ECE NTUA, data-mining, Steam games, recommender system, topic modeling, sentiment analysis**

I. Introduction

Video games have grown considerably popular and according to the European Mobile Game Market, in 2016 over 2.5 billion people spent part of their time playing them. For many decades, games were distributed on a physical storage device which required customers to go to stores, but with the increasing accessibility of the Web more and more people purchase digital content as we can now find multiple platforms dedicated to games with enormous collections from different genres. In 2003 the Valve Corporation released Steam[1], an online digital distribution platform for PC games, which became the dominant such platform.

As there is a large variety of games and it is easier than ever before to obtain them, it is an important task to satisfy customers by helping them choose. Many industries have used recommender systems to suggest products to users, such as Netflix recommending movies [1] and Amazon recommending retail products [2]. As a result, there are a few works that focus on building recommender systems for Steam games. For example, Dylan et al. [3] use Steam user data in conjunction with a Deep Auto-encoder learning model to generate potential recommendations, while Cheuque et al. [4] experiment with mixing Factorization Machines with Deep Neural Networks. By building an effective recommender system not only can costumers be satisfied, but as an immediate result the sales of distributors and developers can increase.

Another advantage of digital retail platforms, such as Steam, is that they offer an easy way to review games or evaluate them based on other user reviews. This feature is extremely important to developers as well, as reviews are crucial to understand how well received their game is, and if not, what needs to be improved. There are a few works which generally study the reviews of games in order to gain useful insights. Lin et al. [5] conduct an empirical study on Steam reviews and explore whether game reviews are similar to mobile app reviews as a starting point for further research. Other works focus on investigating or predicting the helpfulness of video game reviews [6], [7], since Steam allows for users to label them as helpful or unhelpful and then sort them by helpfulness.

Acknowledging the importance of recommender systems, in this work, we build a content-based recommender by using textual features of games meta-data. We use the names of developers, popular game tags voted by users and keywords extracted from the descriptions, in order to build a similarity matrix. We also filter the similarity based recommendations by removing from the results games with many negative ratings relative to positive ratings. Additionally, we conduct an analysis of the reviews for the game *The Elder Scrolls V: Skyrim*. We use topic-modeling in order to understand the general topics that users write about in their reviews and then apply sentiment analysis per topic to get an insight of what they like about the game or not.

The rest of the paper is organized as follows. In section II we

[1] store.steampowered.com

describe the datasets, pre-processing steps and methods used to build the recommender system and analyze the reviews. In Section III we present the results of our recommender system, as well as the topic-modeling and sentiment analysis of *Skyrim* reviews. Finally, in Section IV we conclude and mention possible future extensions of our work.

II. Experimental Methodology

In this section, we describe the datasets, pre-processing steps and methods used to build our recommender system and to analyze *Skyrim* reviews. The main tools we used in our experiments are scikit-learn [8] and NLTK [9].

A. Recommender System

The Dataset used to build our content-based recommender system is provided in Kaggle[2] and contains various meta-data of about 27,000 games available at the Steam Store. The data of these games were gathered at May 2019. This dataset contains many tables and features, but we only use a small subset of them in our work:

1) *appid*: The unique app ID of the game. We use it for indexing and merging tables.
2) *name*: The title of the game. It's simply used to get a game's recommendations, not as a feature in the model.
3) *developer*: The name of the group/company (or companies) that developed the game.
4) *steamspy_tags*: Steam provides some general tags about a game's genre, which are few and very basic. SteamSpy tags, on the other hand, are a variety of tags that describe better the type of a game and they are voted by users. We prefer this feature over the genre, as it is more informative.
5) *short_description*: The description of the game as provided in the Steam store.
6) *positive_ratings*: The number of users that Recommend the game.
7) *negative_ratings*: The number of users that don't Recommend the game.

The features we use to build the model are the developer, tags and description of the game. The reason we chose these features is because users may like games of the same type, or developed by the same company and the description may provide additional useful information to determine how similar some games are.

The developers and tags can be multiple entities and each entity can consist of multiple words. We tackle this issue by joining sub-words into one. For example, the developer "Bethesda Game Studios" becomes "Bethesda_Game_Studios" and the tag "Open World" becomes "Open_World". In order to get better quality results we do not use the whole description as it is, but we extract the top keywords from each description using the rake_nltk, a tool which implements the Rapid Automatic Keyword Extraction Algorithm (RAKE) [10]. RAKE is an algorithm which extracts key phrases by analyzing the frequency of word appearance and its co-occurance with other words in the text. We then combine these features in a single text and we create simple Bag of Words representations.

After the described pre-processing steps and feature extraction, we use the Bag of Words model to build a similarity matrix based on the cosine similarity. Then, by providing a game's name we get its top recommendations based on the highest cosine similarity to other games. We exclude the top similar game from the results as it is the game of the query itself. Additionally, we filter the results by taking into consideration the number of positive and negative ratings so that we do not recommend unpopular or low-rated games.

B. Review Analysis of Skyrim

We chose to analyze Skyrim's reviews as it is a very popular game with a lot of reviews, so it is not easy to just read them all. The game's information from the Steam store is shown in Figure 1. The reviews of Skyrim are provided in Kaggle[3] along with many other game reviews. The .csv file contains about 95,000 reviews for Skyrim.

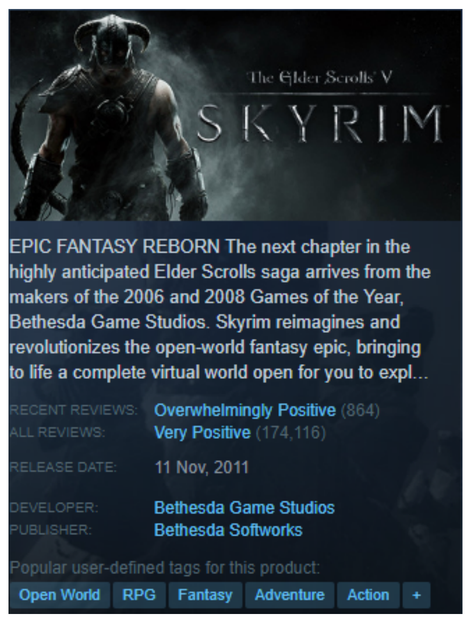


Fig. 1: Information of "The Elder Scrolls V: Skyrim". Screenshot taken from the Steam store.

1) Topic Modeling: In order to model the topics of the reviews we first extensively clean them to obtain quality results. The text pre-processing includes lowercasing, removing URLs, fixing contractions, removing numbers and special characters, POS tagging and keeping words tagged as nouns, lemmatization with the help of WordNet and stopword removal. In

[2]www.kaggle.com/nikdavis/steam-store-games

[3]www.kaggle.com/smeeeow/steam-game-reviews

the stopword list we added some more domain specific words like 'game', 'steam', 'play' and the game's name. We then tokenized the reviews and discarded those with less than three tokens. Finally, we created TF-IDF vectors.

For the topic modeling we used the Latent Dirichlet Allocation (LDA) [11] and assigned to each review the most dominant topic. In LDA, each document can be viewed as a mixture of various topics, where each topic has probabilities of generating various words. Our goal is interpretability, so we choose the number of topics such as their most relevant words do not overlap too much.

2) Sentiment Analysis: In order to analyze the sentiment of the reviews we use VADER (Valence Aware Dictionary and Sentiment Reasoner) [12], which is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. We use the compound score determined by VADER and we characterize a review as positive if the score is greater than 0.1, negative if it is lower than - 0.1, else neutral. We apply VADER on the original text of reviews, without cleaning them, as the tool can handle that.

III. Results

In this section, we present the results of our work. First, we show some examples using our recommender system and then we analyze the results of the data mining techniques applied on *Skyrim* reviews, including topic modeling and sentiment analysis.

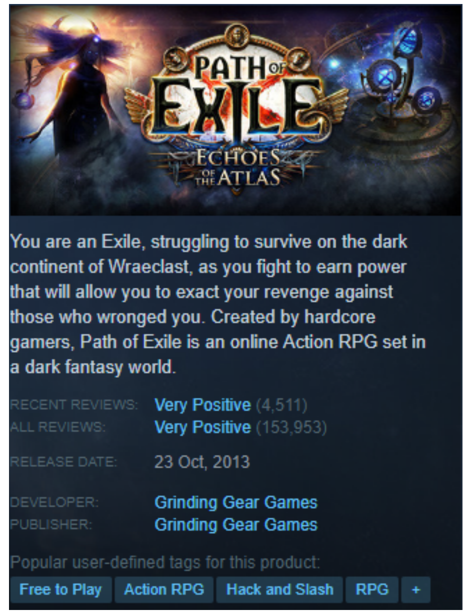
A. Recommender System

Content-based recommender systems like the one we created have no obvious evaluation criteria. In order to test it we can try manually evaluating the recommendations of some example instances based on our knowledge. Table I shows the top 10 recommendations our system chose based on cosine similarity to "The Elder Scrolls V: Skyrim" (Figure 1).

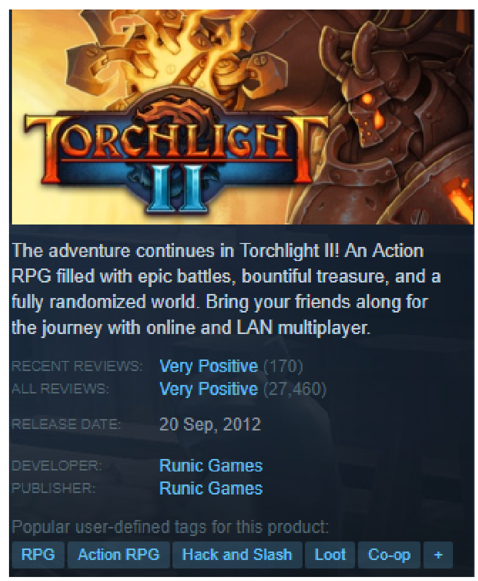
TABLE I
RECOMMENDED GAMES BASED ON SIMILARITY TO "THE ELDER SCROLLS V: SKYRIM"

Recommended	Score
The Elder Scrolls® Online	**0.371**
The Elder Scrolls IV: Oblivion®	0.357
The Elder Scrolls V: Skyrim VR	0.357
The Elder Scrolls III: Morrowind®	0.297
Fallout 4	0.297
Planet Explorers	0.297
Risen	0.286
Dragon's Dogma: Dark Arisen	0.286
ELEX	0.286
Divinity II: Developer's Cut	0.242

We can see that "The Elder Scrolls® Online" got the highest similarity score and additionally the next three most similar games are also part of the *Elder Scrolls* series. That is a positive indication that our system works as expected, since all these games are developed by the same company, i.e. Bethesda Game Studios and they are part of the same series.



(a) Path of Exile



(b) Torchlight II

Fig. 2: Information of "Path of Exile" and its most similar game "Torchlight II". Screenshots taken from the Steam store.

Table II shows the top 10 most similar games to "Path of Exile", a classic top-down action RPG. The top recommended game is "Torchlight II" and both games information are shown in Figure 2. "Torchlight II" is also a top-down action RPG and that makes it a good recommendation for anyone who likes "Path of Exile". The rest of the top recommended games are also of similar genres and often appear in community-based recommended lists.

TABLE II
RECOMMENDED GAMES BASED ON SIMILARITY TO "PATH OF EXILE"

Recommended	Score
Torchlight II	**0.371**
Grim Dawn	0.357
Warframe	0.357
Vindictus	0.297
Exanima	0.297
Dungeons & Dragons Online®	0.297
CrossCode	0.297
Shakes and Fidget	0.297
Torchlight	0.297
Dragon's Dogma: Dark Arisen	0.286

B. Review Analysis of Skyrim

In this subsection, we present all the information extracted from *Skyrim* reviews. First, we explore the topics mentioned by users and then we analyze the sentiment of the reviews based on each topic in order to see what general aspects of the game users like (or dislike) and more importantly, in order to find possible major problems based on negative feedback.

1) Topic Modeling: Latent Dirichlet Allocation (LDA) is an unsupervised algorithm and it's not easy to evaluate the results without labeled data. Since our main purpose is interpretability, we choose the number of topics such as they make sense and don't overlap too much. Table III shows the top 15 most relevant terms in each topic. We chose to classify the content of the reviews in four topics named based on their most relevant terms. These topics are:

- *Content & Authenticity*: Reviews that write generally about the content of the game, such as its story and gameplay.
- *World Details*: Reviews that mention specific objects, areas or quests of the game. For example the words arrow and knee are part of the famous in-game line "I used to be an adventurer like you, then I took an arrow in the knee".
- *Company & Modding Community*: Reviews that refer to the developers, publishers and the modding community. Mods are downloadable modifications of the game made by the users themselves and their creation is officially supported.
- *Playthrough Experience*: Reviews that talk about the campaign, different playthroughs and Steam achievements for the game.

Figure 3 shows an example visualization of the most relevant terms in Topic 1: *Content & Authenticity* using pyLDAvis [13]. The word "mod" is quite popular in this topic, which is of no surprise since Skyrim is famous for its variety of mods and ease to install them.

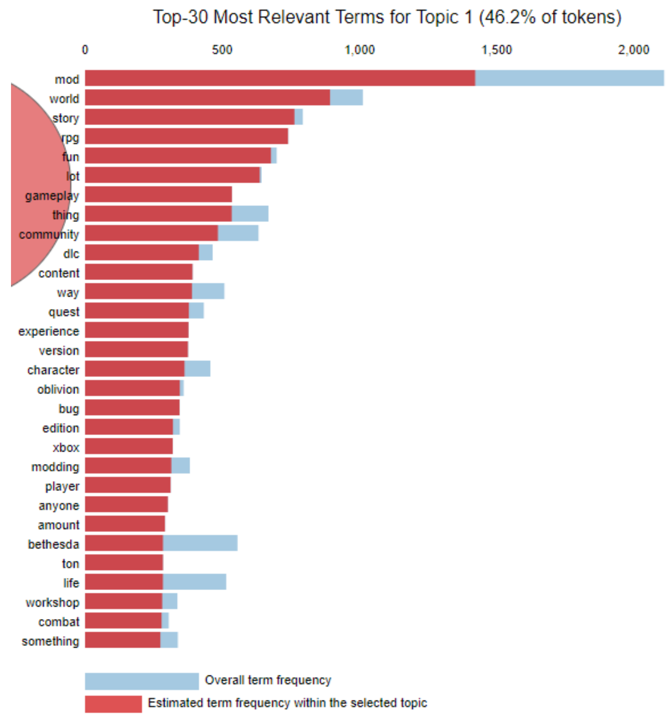


Fig. 3: Example visualization of most relevant terms for topic 1 (Content & Authenticity) using pyLDAvis [13].

Figure 4 shows the distribution of the topics based on the most dominant topic in each review. The most popular topic is the generic *Content & Authenticity* as expected, since most users usually write a short review of whether they generally liked the game or not and what was the most important feature.

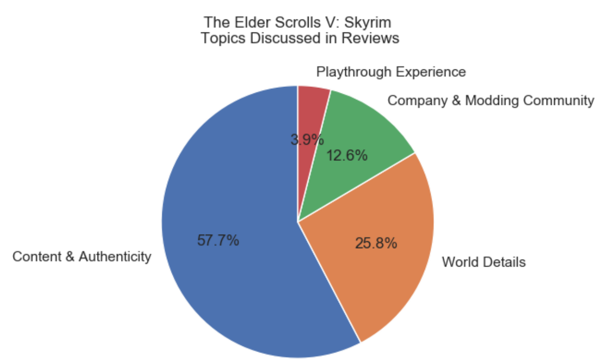


Fig. 4: Topics distribution based on the most dominant topic in each review.

2) Sentiment Analysis: As shown in Figure 5 we can just analyze the total sentiment of reviews in a straightforward manner and see that 73.8% of all reviews are positive 8.1% are neutral and 18.1% are negative. This simple analysis certainly

TABLE III
TOP 15 MOST RELEVANT TERMS FOR TOPICS DISCUSSED IN SKYRIM REVIEWS

Content & Authenticity	World Details	Company & Modding Community	Playthrough Experience
mod	dragon	mod	value
world	arrow	bethesda	replay
story	knee	valve	ing
rpg	life	review	endless
fun	chicken	community	card
lot	people	money	nuff
gameplay	sword	modders	badge
thing	horse	edit	regret
community	house	shit	campaign
dlc	world	pay	playthroughs
content	man	fuck	star
way	guard	company	hey
quest	thing	store	dovahkiin
experience	town	paywall	life
version	mountain	people	switch

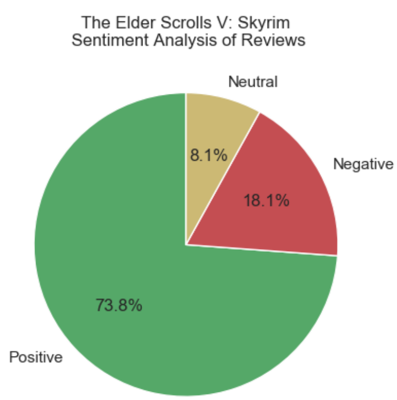


Fig. 5: Sentiment distribution in Skyrim reviews.

gives us useful information as we can see that the game has mostly good feedback, but it is more important to explore the sentiment based on what users are writing about in the reviews. This way, we can get a better idea about what users like or dislike and potentially discover important issues. Figure 6 shows the sentiment analysis of reviews based on each topic. We can see that when the users generally write about the game in the topic *Content & Authenticity* they give the most positive feedback. In the other topics we observe more negative reviews as users usually complain about specific things such as bugs or other inconveniences.

It is important to note that there are quite a few reviews that are falsely characterised as negative by VADER, which tends to happen when users use inappropriate language to describe their enthusiasm or excitement like the phrase ”dawnguard shit looks neat”, which calls ”shit” (humoristically) the content of the Dawnguard DLC, but the user actually likes it. Even though the review is positive, VADER recognised it as negative.

From the perspective of the developers the most important topic to analyze is probably the ”Company & Modding Community” where in fact we can see that there is an important issue. Table IV shows some example key-phrases in negative reviews of this topic which make clear that users complain about paid mods. The phrases were extracted using rake_nltk. As described in the Fandom Wiki[4], the paid mods were introduced at April 24th in 2015, but a few days later the company made them free again as the feature received negative backlash.

TABLE IV
KEY-PHRASES IN NEGATIVE REVIEWS ABOUT THE TOPIC ”COMPANY & MODDING COMMUNITY”

Key-phrase
stop paid modding
valve introduces paid mods
let cash grab mods gain
bethesda attempting paid mods
new paid mod system implemented
paid mods greedfest idea
fuck paid mods dont fuck bethesda
modders almost get nothing
new paid mod scam
system would discourage free mods

[4]tes-mods.fandom.com

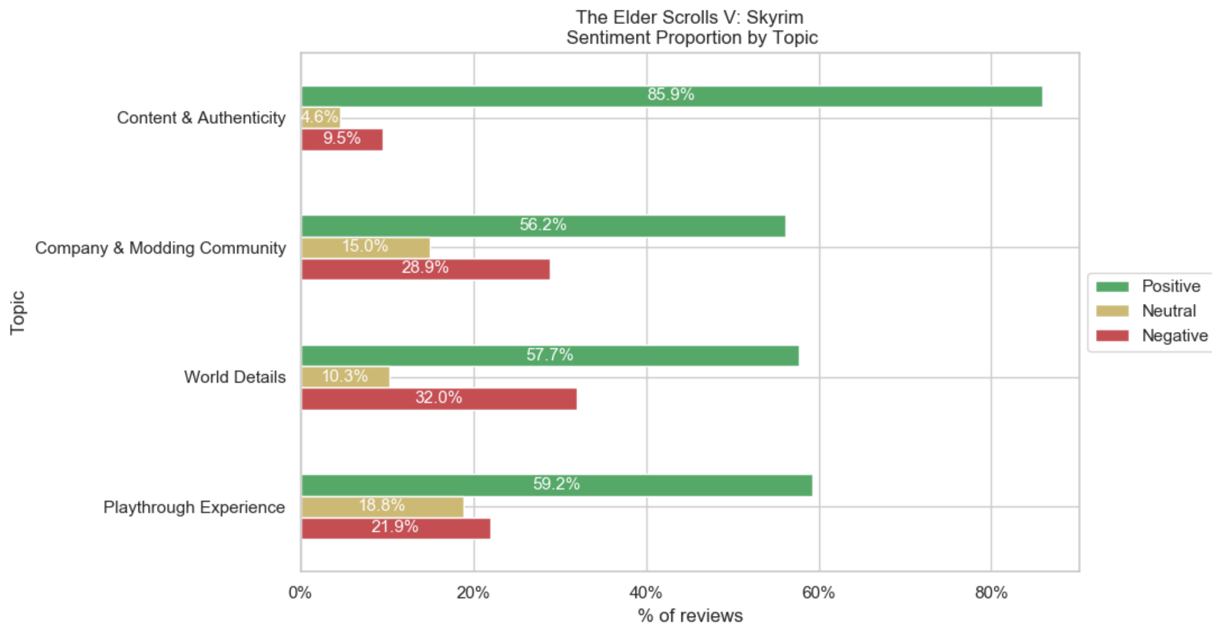


Fig. 6: Sentiment distribution per topic discussed in Skyrim reviews.

IV. Conclusions and future work

In this work, we created a content-based recommender system for Steam games and we analyzed Skyrim's reviews. Based on our results, it is safe to assume that our recommender system provides reasonable recommendations and we could proceed to deploy it and further evaluate and improve it based on user feedback. The analysis of Skyrim's reviews showed that the the majority of users speak positively of the game and we discovered an important issue where users gave a lot of negative feedback due to the introduction of paid mods.

Our work could be extended in the following directions. First, we could create a user-based collaborative filtering recommender system and combine it with the content-based one. Finally, we could aggregate the review analysis by periods to identify emerging topics and extract entities to keep track of specific points of interest.

References

[1] Carlos A. Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4), December 2016.

[2] Brent Smith and Greg Linden. Two decades of recommender systems at amazon.com. *IEEE Internet Computing*, 21:12–18, 05 2017.

[3] Dylan Wang, Melody Moh, and Teng-Sheng Moh. Using deep learning and steam user data for better video game recommendations. In *Proceedings of the 2020 ACM Southeast Conference*, ACM SE '20, page 154–159, New York, NY, USA, 2020. Association for Computing Machinery.

[4] Germán Cheuque, José Guzmán, and Denis Parra. Recommender systems for online video game platforms: The case of steam. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 763–771, New York, NY, USA, 2019. Association for Computing Machinery.

[5] Dayi Lin, Cor-Paul Bezemer, Ying Zou, and Ahmed E. Hassan. An empirical study of game reviews on the steam platform. *Empirical Software Engineering*, 24(1):170–207, June 2018.

[6] Lukas Eberhard, Patrick Kasper, Philipp Koncar, and Christian Gutl. Investigating helpfulness of video game reviews on the steam platform. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, October 2018.

[7] 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, May 2019.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[10] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. *Automatic Keyword Extraction from Individual Documents*, pages 1 – 20. 03 2010.

[11] Matthew D. Hoffman, David M. Blei, and Francis Bach. Online learning for latent dirichlet allocation. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, page 856–864, Red Hook, NY, USA, 2010. Curran Associates Inc.

[12] C. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1), May 2014.

[13] Carson Sievert and Kenneth Shirley. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.