Future Generation Computer Systems 125 (2021) 715–725

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Representing emotions with knowledge graphs for movie recommendations

Arno Breitfuss, Karen Errou, Anelia Kurteva *, Anna Fensel

University of Innsbruck, Department of Computer Science, Semantic Technology Institute (STI) Innsbruck Technikerstr. 21a, A-6020 Innsbruck, Austria

ARTICLE INFO

Article history:
Received 23 December 2019
Received in revised form 3 May 2021
Accepted 1 June 2021
Available online 5 June 2021

Keywords:
Knowledge graph
Artificial intelligence
Machine learning
Bias
Emotions
Ontology
Movies
Recommender system

ABSTRACT

Consumption of media, and movies in particular, is increasing and is influenced by a number of factors. One important and overlooked factor that affects the media consumption choices is the emotional state of the user and the decision making based on it. To include this factor in movie recommendation processes, we propose a knowledge graph representing human emotions in the domain of movies. The knowledge graph has been built by extracting emotions out of pre-existing movie reviews using machine learning techniques. To show how the knowledge graph can be used, a chatbot prototype has been developed. The chatbot's reasoning mechanism derives movie recommendations for the user by combining the user's emotions, which have been extracted from chat messages, with the knowledge graph. The developed approach for movie recommendations based on sentiment represented as a knowledge graph has been proven to be technically feasible, however, it requires more information about the emotions associated with the movies than currently available online.

1. Introduction

Knowledge graphs are a powerful technology that is useful in numerous domains [1]. Their typical usage scenarios include information integration and making the information explicable to the end users. Recent examples include employment of knowledge graphs for recommendation of services [2] such as in the tourism and entertainment sectors [3,4]. The aim of this work is to develop the knowledge graph technology and demonstrate its usage by developing a chatbot prototype, which can perform simple emotional reasoning. According to Wikipedia[1] *"Emotional reasoning is a cognitive process by which a person concludes that his/her emotional reaction proves something is true, regardless of the observed evidence"*. This work collects, reuses and designs ontologies in order to represent human emotions and to design reasoning mechanisms employing these. The developed knowledge graph used for emotion representation can be inbuilt in practical scenarios e.g. recommender systems. This paper focuses on a specific domain, namely movie recommendations.

Emotions are extracted from pre-existing movie reviews and mapped to the corresponding movies. For the chatbot prototype, chat messages provide the needed data about the user by letting the user talk to an Artificial Intelligence (AI). Next, with the help of reasoning mechanisms the most suitable movie recommendations for the user are discovered. The whole process is transparent, explainable and involves user interaction thus making the user feel involved in the process.

Chatbots are programs, which allow communication via text with an artificial agent [5]. These programs are usually designed to simulate how humans will react or behave in a conversation. In a "perfect" world, chatbots should be capable of recognizing human emotions in order to respond more smoothly with situation awareness thus the need for emotionally intelligent AI arises.

It is important to note that one's emotions could easily affect one's choices. For example, a choice of a movie is also dependent on the user's mood. However, emotions such as happiness and sadness affect people in different ways. Someone who feels sad might not want to be recommended a 'happy' movie and vice versa, which shows the presence of bias. Definitions of bias vary but in a general sense it is defined as "any form of preference, fair or unfair" [6,7]. Without having enough context and experience AI would not be able to understand this difference and make the most optimal movie recommendation.

Various machine-learning methods for classification of extracted movie reviews such as Naïve Bayes could be applied to analyse the data and construct relevant knowledge graphs that represent emotions. Using such methods could help the detection of bias in both the provided data and the results. The complex notion of bias and its implications are of great importance as they could change the overall results without one even realizing.

* Corresponding author.
E-mail addresses: Arno.Breitfuss@student.uibk.ac.at (A. Breitfuss), Karen.Errou@student.uibk.ac.at (K. Errou), anelia.kurteva@sti2.at (A. Kurteva), anna.fensel@sti2.at (A. Fensel).

1 https://www.wikipedia.org.

https://doi.org/10.1016/j.future.2021.06.001

Thus, in this work we rely on key technologies mentioned up to now, namely knowledge Graphs, natural language processing, machine learning. The main research question addresses *creation of a knowledge graph representing human emotions for the domain of movie recommendation and its application*.

The paper is structured as follows. Section 1 is an introduction to the subject of this research. Section 2 presents a brief discussion of relevant work. Section 3 describes the implementation, while Section 4 presents a possible application of the knowledge graph. Section 5 deals with the evaluation and testing of the given solutions. Conclusion and future work can be found in Section 6.

2. Related work

Nowadays, with the rise of AI developers become more and more aware of the seemingly unlimited power and capacity of this technology. The question regarding the exact areas where AI can replace people in the future is still present.

Even though, various AI platforms and tools such as Microsoft Azure Machine Learning,[2] Google Cloud Prediction API,[3] Cortana,[4] Siri[5] and Alexa[6] have been developed and are fully functional, more design specifications need further addressing here. One such area is emotional intelligence. Could a machine learn how to recognize emotions based on data provided by the users and can it use that knowledge to make correct recommendations? How to represent emotions? The following sections provide a review of existing recommendation systems that take into account one's emotions and discuss bias, machine learning techniques and knowledge graphs for representing emotions. It is important to note that the main focus of this paper is the development of a knowledge graph that represents emotions thus existing machine learning approaches will not be discussed in full detail.

2.1. Recommendation systems using emotions

Based on their core algorithm, recommender systems can mainly be categorized into collaborative filtering (CF), content-based (CB) and hybrid systems [8]. CF works by looking at user–item interactions (such as clicks, views, etc.) which are then used to model user preferences. These preferences are compared to other user preferences in order to recommend items that similar users preferred. CB systems, on the other hand, recommend items based on item features such as movie genre and director and use those features to recommend similar items. In order to work, CB systems need side information or rich data sets thus one advantage of CF systems over CB systems is not requiring the efforts of feature extraction [8]. However, this is also where the problems of CF systems stem from. Without any data to begin with and the recommendation relying on user–item interactions, recommendations can only be made after some data has been gathered. And even after the system has been used and some data has been accumulated, the user–item matrix may still be sparse. One special case of this data sparsity problem is the cold-start problem, where a new user without any item interactions joins and thus cannot be recommended any items [8].

Hybrid systems are one possible answer to those problems and may be implemented in various ways. One possible implementation is to use a CB recommender system for a new user and switch to a CF system after enough user–item interaction data has been gathered. The following paragraphs briefly describe five emotion-based recommender systems. Possible improvements to those proposals by making use of a knowledge graph are discussed later in Section 5.

Ferwerda and Schedl [9] propose enhancing recommendations in the music domain by using one's personality traits and emotional states. The research [9] consists of an online user study followed by emotion acquisition and emotion classification. During the online study users are exposed to various film clips which is a frequently used method for inducing emotions such as anger, disgust, fear, happiness, sadness or neutral emotion. Upon watching each movie clip and hearing emotionally laden music a series of control questions were asked and an emotion profile for each participant was built. Emotional states were also extracted from textual data on Twitter. In order to improve music recommendations Ferwerda and Schedl [9] conclude that songs need to be emotionally annotated either by direct human annotation (online surveys, tags, games), indirect human annotation (lyrics, web documents) or by performing content-based analysis.

The emotional factor is also discussed by Gonzáles et al. [10] who propose a new approach for modelling content-based and collaborative-filtering recommender systems based on capturing the emotions of both individuals and the community as a whole. Similar to [9] users are presented with tests which help evaluate one's emotional state and extract emotions such as love, fondness, happiness, restlessness, desperation, depression etc. The extracted emotions are later mapped to service features, which makes them operational in any domain. The mapping itself is similar to the one presented in [11] and depends strongly on context. Gonzáles et al. conclude that emotions could be beneficial for recommender systems if appropriately represented and integrated [10].

This is the topic of interest for Zheng [12]. The work in [12] builds upon the existing context-aware matrix (CAMF) [13] factorization approach and extends it with user-personalized metrics (also referred to as CAMF_CU). Zheng [12] extends the CAMF_CU approach with the individuals' emotional states and compares its performance to the original version, which does not include emotions. Both approaches are evaluated by using the mean absolute error (MAE) as an evaluation metric. The results validate the author's idea that emotions can be useful for context-aware recommender systems.

Ho et al. [14] present a hybrid approach that combines CF and CB systems to enable emotion-based movie recommendations. In order to build individual profiles, users were handed a questionnaire, asking which kind of movies (based on genre) they would like to watch when they are in a certain emotional state. The user emotions chosen in [14] were love, joy, anger, and sadness, while for movie genres- love, horror, action, humour, fiction and documentary. The user's current emotional state was extracted using a colour-based survey where the user picks predefined colours for an avatar. The interpretation of the result of this survey solely relies on colours representing emotions (e.g. "light red" represents love, while brown, grey or black represent sadness, etc.) [14]. The resulting emotion-genre matrix then can be used as a rule set for recommendations.

The findings in [9] and [10] bring to light the issue of emotion representation, annotation and mapping. This issue may be addressed by a new knowledge representation model for emotions developed with semantic technology such as knowledge graphs. As we will later demonstrate in this paper, representing the extracted emotions with a knowledge graph provides structure, semantic annotations and allows faster and easier knowledge discovery from which any recommendation system could benefit. However, when dealing with emotions and AI bias is highly likely to be present and could affect the reasoning mechanisms and the final results.

2 https://azure.microsoft.com/en-us/services/machine-learning/.
3 https://cloud.google.com/ai-platform.
4 https://www.microsoft.com/en-us/cortana.
5 https://www.apple.com/siri/.
6 https://www.alexa.com.

2.2. Artificial intelligence and bias

During the past few years, AI chatbots such as Alexa[6], Siri[5] and Google Assistant[7] have been successfully deployed and are continuously being tested for bias. Tackling bias in AI has also been an interest of big corporations such as Google,[8] which introduced the Google What-If Tool [15] that enables watching changes in the model's performance in real-time, visualizing data and code-free machine learning experience. Further, algorithmic fairness analysis could also be used to detect AI bias [16].

Silberg and Manyika [6] from the McKinsey Global Institute[9] define bias as *"Any form of preference, fair or unfair"*. As per Oxford Learner's Dictionary*"Bias is a strong feeling in favour of or against one group of people, or one side in an argument, often not based on fair judgement"* [17].

How AI is used and why plays an important role in bias discovery and tackling unfairness [6]. One of the main reasons for bias could be the underlying data [18,19]. In most cases the data consists of human information that has been provided and as discussed in Section 2.3, people are prone to be biased [20]. In order to deal with this issue, fairness measures have been introduced. For example, defining separate decision thresholds for each specific group of people or just one threshold that would apply to all [20].

Various machine learning algorithms and methods that deal with mitigating bias have been developed during the years. Such example is the debiasing algorithm developed by Amini et al. [21], which takes into consideration the distribution of all underlying features in the training data. Testing of the developed solution was done in the case of detecting gender and race by facial recognition software. Potentially bias data was provided so it could be checked whether or not the developed algorithm mitigates bias. The achieved results were sufficient and proved that the debiasing algorithms work and most of all reduces bias. Several other machine learning approaches could be used to deal with bias. For example, pre-processing as defined in [22,23] and post-processing techniques as described in [24,25].

Dong et al. [26] present one of the many methodologies for data analyses that have been developed. The method in [26] includes Support Vector Machine, Naïve Bias and machine learning methods such as pre-processing [23]. In addition, as proposed in [27] data categorization based on time, date, chat message authors etc. could be performed in order to improve the granularity of the results. However, at the end of the day the results still need to be reviewed by a human. This leads to questions such as: *"Who decides when and AI system has minimized unfair bias so that it could be safely realized for use?"* [6].

2.3. People and bias

In the context of the present research, several studies regarding movie review bias have been done. One such paper is by DellaVigna and Hermle [28]. The study uses data from two major movie rating platforms: Rottentomatoes[10] and Metactrictic[11] which differ in their data summarization techniques. Metacritic[11] uses a 0–100 scale and computes the average score of all reviews for each movie while Rottentomatoes[10] uses a non-numerical scale, which classifies a movie either as 'fresh' or 'rotten'. In the case of quantitative reviews, Rottentomatoes[10] classifies movies with 2 stars as 'rotten', with 3 or more stars as 'fresh' and movies with 2.5 stars are split based on a subjective judgement [28]. The research was aimed at proving that specific movie production companies such as 20th Century Fox[12] have specific features associated with them. By selecting pre-assigned features which are not changed by user reviews, it was concluded that Fox[12] movies are most likely to be action, comedy, drama and unlikely to be documentaries [28]. This leads to the conclusion that bias is present everywhere and it could not be fully eliminated.

Koh et al. [29] look further into the issue of bias reviews. Their paper [29] uses behavioural theory to analyse the reviews left by users from two different countries — China and the USA. In the case of the USA, movie reviews were gathered from IMDb[13] and in the case of China - Douban.[14] One of their main interests was to discover whether or not the cultural differences could affect user's reviews. In their proposed model for evaluation several reasons for writing reviews were derived: Attitude, Social Norms and Motivation. Theories such as Theory of Planned Behaviour (TPB) [30] and Theory of Reasoned Action (TRA) [31] were examined and it was concluded that the user's attitude affects the reviews they leave. In order to test their hypothesis, Koh et al. [29], retrieved 1000 reviews of random movies from both IMDb[13] and Douban[14]. However, having random selection meant that different movies would be picked from each website thus a second dataset was retrieved. The data was then divided into categories: Top 100, Bottom 100 and Random 1000. First the data was retrieved from imdb.com and then from douban.com so that the same movie titles were selected. After analysing the data, it was concluded that the main differences in opinions is in the category Bottom 100 [29]. The differences showed that American reviewers were more open to show their dissatisfaction while Chinese reviewers were more generous. The study [29] as stated *"…validates the application of cultural dimensions theory proposed by Hofstede's in explaining online movie reviewers' behavioural difference across culture"*. This again leads to biased data, that even when used by a highly trained AI which has adopted various machine learning techniques to cope with unfairness, might produce biased results.

2.4. Knowledge graphs and emotional bias

Once extracted emotions need to be represented in a way that enables their processing. Many solutions such as [9] and [10] represent emotions as lexicons which is effective, however, scalability, annotations and cross-domain usage could be a problem. Ontologies are a good solution to this problem as they provide a unified structure in both human-readable and machine-readable format and could be integrated in any system. For example, the Emotion Ontology,[15] which in addition describes moods, appraisals and subjective feelings, is designed to support interdisciplinary research by providing unified annotations. Another ontology is the EmotionOnto, which is able to model contexts [32].

Dragoni et al. [33] present the OntoSenticNet[16] ontology for sentiment analysis. The ontology enables defining a specific sentiment by analysing multi word expressions that are related to concepts connected to emotions. The ontology's development process included graph-mining and multi-dimensional scaling techniques, which in addition enabled the integration of reasoning in order to define the sentiment in detail. With the help of rich terminologies and common-sense expressions high granularity is achieved. Ontologies can help implementing emotion-aware

7 https://assistant.google.com.
8 https://www.google.com.
9 https://www.mckinsey.com/mgi/overview.
10 https://www.rottentomatoes.com.
11 https://www.metacritic.com.
12 https://www.foxmovies.com.
13 https://www.imdb.com.
14 https://www.douban.com.
15 https://bioportal.bioontology.org/ontologies/MFOEM/?p=summary.
16 https://sentic.net/downloads/.

applications in different domains and enhance the recommender systems. As discussed in the previous sections, bias is a factor that should be considered when dealing with emotions.

To address bias in knowledge graphs, Janowicz et al. conducted a research [34] which presented biasness from three perspectives: bias arising from available data; bias from embedded ontologies and as a result of drawing inferences. Having biased data means that the developed knowledge graph will also be biased thus increasing dissimilarity between less prototypical classes. Moreover, this would affect the defined rules and lead to unfair question answering. In addition, bias in schema is also present as schemas are developed by engineers and experts in the domain each one of them is prone to bias. The research [34] found that some biases could be spotted only when comparing the different ontologies and datasets which means that one could work with a given ontology and not accept it as being unbiased only because it has not been compared to others. Last but not least, inferential bias, which as concluded is a result by simply using different endpoints to query the same ontology [34]. With the help of various data processing methods, bias could be mitigated to a certain degree, however, it could as Janowicz et al. stated: *"come at a cost of increased variances"* [34].

3. Implementation

Fensel et al. [35] analyse existing tools and methods for building knowledge graphs and derive a step-by-step process model for knowledge graph generation. The model [35] consists of the following four main steps: **Knowledge Creation, Knowledge Hosting, Knowledge Curation and Knowledge Deployment**. During the knowledge creation step, manual and semi-manual editing of information is performed, which is then followed by mapping according to domain specifications and the automatic generation of annotations. Tools for knowledge collection, storage and retrieval are addressed in the knowledge hosting stage. Once a knowledge graph is built, the main focus turns to improving its quality. This is also known as knowledge curation and as described in [35] includes assessment, cleaning and enrichment processes. The final step (Knowledge Deployment) is about making data available and useable.

The following sections present how the knowledge graph was built based on the process model by Fensel et al. [35] and a possible application in a recommender system.

3.1. Knowledge creation

The used domain area is movie reviews. Existing vocabularies from the Wikidata[21] knowledge base as well as from metacritic were reused and thus creation of new knowledge was not needed. Links to Wikidata[21] for each of the movie entries, and to DBpedia[22] whenever possible were maintained using **owl:sameAs** relations. **Schema.org**[23], a widespread vocabulary for structuring data on the web [36] was reused. Mainly its movie and review model was utilized to describe the scraped data sets. A REST-API was implemented, which on /api/v1/movie/:id returns a potentialAction. It is a "watch" action whose target property leads to the IMDb video gallery site for a specific movie. The sample solution looks as follows:

```
"@context": "http://schema.org", "@type": "Movie",
"@id": "/api/v1", "title": "Pulp Fiction", "potentialAction": {
"@type": "WatchAction",
"target": "https://www.imdb.com/title/tt0110912/videogallery
?ref_=tt_pv_vi_sm"}
```

Further, the **Onyx**[24] ontology for modelling emotions, which have been extracted from text and the **WNAffect**[25] ontology, were reused. WNAffect[25] was particularly useful as it provided different labelling for emotions, moods and situations [37] which were used for mapping words to affects (sentiment).

Emotions were extracted from both movie reviews and the user chat log. Natural language processing was used for extracting and annotating emotions from the restored user chat logs. To minimize dependencies and to provide faster data retrieval, the GraphDB[19] graph database was used to store the result.

Regarding extracting emotions from movie reviews, nearly one million reviews were analysed and their assumed emotional values were extracted with a Bayesian Classifier. Based on the assumption that all attributes are independent given the class variable the Bayesian Classifier reduces the complexity of the calculations [38,39]. The classifier was also used to extract the emotional values out of the user's chat messages.

Stemming [36] and map reduce [37] techniques were adopted in order to enhance the results. While stemming was used for removing grammatical forms and simplifying words to their root [40], the map reduce technique improved the scalability of the data by mapping it into tuples and then combining those tuples into a smaller set of tuples [41]. By using these techniques, a more even distribution of emotion was produced.

The Ontotext Semantic Tagging Service[17] was used for annotating natural language. Ontotext[18] provides a REST-API to their service, which in this case was not as powerful as initially expected when it comes to computational load capacities and thus not many reviews were annotated. There are several different data types provided by Ontotext[18]. Parts of natural language texts are then assigned to those data types, according to the underlying algorithms. For example, the annotation: "…Disney bought everything from marvel comic…" was classified as ontotext:RelationAcquisition. However, the linked ontology is nowhere to be found, so the types were kept and used an ad-hoc set up namespace.

3.2. Knowledge hosting

Capturing the data in the current knowledge graph was done by mapping which as defined by Fensel et al. [35] maps data sources to schema.org. The data is also saved in a database and could be accessed by request. GrapbDB[19] was used as a Triple-Store as it provides a simple way of uploading .ttl files, and a neat interface for testing SPARQL[20] queries. Another advantage of GraphDB[19] is its REST API, which is language independent and was used in this research. This is important, since Node.js was for handling the JSON formatted data and HTTP requests. A SPARQL API was used for requests to the server, however, problems were encountered when trying to run "insert" queries thus custom HTTP requests were needed.

3.3. Knowledge assessment

There was not one single data set to be found that provided everything that was needed. In order to categorize movies, pre-processing steps were undertaken. Information about movies from IMDb[13] and Metacritic[11] was gathered using web scraping. Since different resources (movies) have different URLs, the movie's internal IDs for both web sites needed to be retrieved. This was done by using the Wikidata[21] knowledge base, which provides the P345 property for IMDb movie IDs and the P1712 property for Metacritic IDs. The following query enabled the

17 https://www.ontotext.com/solutions/semantic-tagging/.

18 https://www.ontotext.com.

19 https://www.ontotext.com/products/graphdb/.

20 https://www.w3.org/TR/rdf-sparql-query/.

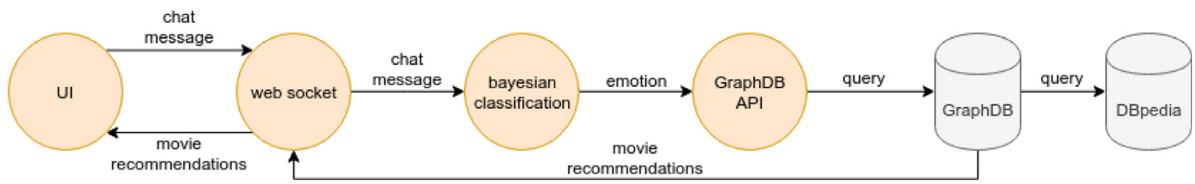


Fig. 1. Chatbot process.

retrieval of a list of all IMDb IDs and existing Metacritic IDs for the corresponding movie:

```
SELECT ?item ?imdb ?metacritic
WHERE
{
?item wdt:P345 ?imdb.
OPTIONAL { ?item wdt:P1712 ?metacritic }
}
```

The web scraping service iterates over every movie ID and constructs the necessary URLs for sending an HTTP request. The movies' IDs were gathered using the Wikidata[21] knowledge base. For every movie in the present database there is an equivalent on Wikidata.[21] Due to that, linking to Wikidata[21] was done with the help of federated SPARQL queries. DBpedia[22] maintains links in the form: owl:sameAs relations to Wikidata,[21] thus it was possible to enrich the database with links to DBpedia[22]. Example SPARQL query that enables linking to Wikidata[21] by using the predicate sameAs:

```
PREFIX schema: <http://schema.org/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/> SELECT
?film ?wdlink ?dblink WHERE {
SERVICE <http://dbpedia.org/sparql> {
    ?dblink a dbpedia-owl:Film .
    ?dblink owl:sameAs ?wdlink
}
?film a schema:Movie .
?film owl:sameAs ?wdlink
}
```

Next, the HTTP requests were used for retrieving the corresponding HTML files. From these HTML files a virtual Document Object Model (DOM) is generated, with respective predefined tags. Next by using jQuery on the DOM data of interest is extracted.

3.4. Knowledge cleaning and enrichment

The developed knowledge graph was enhanced by extracting emotions from the user's chat log and mapping them to movie reviews. A lightweight tool, which automatically maps the extracted data from JSON to RDF format was built. Since one could theoretically add any kind of triple to a knowledge graph, by using the SHACL-JS[23] library on the backend, it was possible to perform automatic SHACL validation of the produced RDF files, which helped the knowledge cleaning and enrichment processes.

3.5. Knowledge deployment

The data used for the creation of the knowledge graph is stored on a server and is publicly available.[24] The data comprises the extracted movie reviews (from Metacritic[11]), as well as the user's chat logs. The final knowledge graph structure is shown in Fig. 4.

[21] https://www.wikidata.org/wiki/Wikidata:Main_Page.
[22] https://wiki.dbpedia.org.
[23] https://www.w3.org/TR/shacl-js/.
[24] https://github.com/aneliamk/ontology.

4. Application of the knowledge graph within a recommender system

An example recommender system can consist of a pre-existing AI chatbot that interacts with the user and extracts emotions from the chat logs, a collection of movie reviews that were categorized based on emotions and a server that handles the data processing and the mapping of emotions to movies, which is stored as a knowledge graph in a graph database. Further, an example of how the knowledge graph could be integrated in a recommender system by developing a simple chatbot and its possible user interface are provided.

4.1. Chatbot prototype

A chatbot prototype was developed with Node.js.[25] The chatbot uses Markov chains to generate natural language text and transitions between states using a probabilistic model [42]. For natural language, a Markov chain checks which word can follow another in order to form sentences and is thus not context aware. For a chatbot this means that the generated chat messages do not necessarily correlate with the topic of the user's chat messages.

The user's emotion was extracted by classifying one's chat messages on the fly with the same Bayesian classifier which was used to extract emotions from movie reviews as described in Section 3.1. Based on this emotion, the reasoning mechanism then looks for a movie classified as causing a similar emotion by querying the knowledge graph to look for emotions of the same category. In particular, if a chat message is classified as describing "happiness" the recommended movie might be classified as causing "cheerfulness" since the category of both emotions is "Joy". The whole process can be seen in Fig. 1.

While this process may not be as sophisticated as modern recommender systems, it fulfils its purpose of showcasing how the knowledge graph might be utilized by a more sophisticated system. If one were to create user preferences for CB or CF systems, the watchAction described in Section 3.1 could be utilized to mark which movies have been watched by the user. By extracting the user's emotion after watching the movie, the user–item interaction for a CF system could be generated by specifying which emotion the movie has caused, which could then be used to recommend movies to other users who had similar emotional responses to other movies the user has watched. The improvements to modern recommender systems by making use of the knowledge graph are discussed in Section 5.

4.2. User interface

A simple version of a user interface (Fig. 2), for bidirectional communication between client and server, was developed with an express.js[26] web server, using socket.io.[27] Fig. 3 presents the AI chatbot, which analyses user inputs based on the approach described in this paper and recommends a movie.

[25] https://nodejs.org/en/.
[26] https://expressjs.com.
[27] https://socket.io.

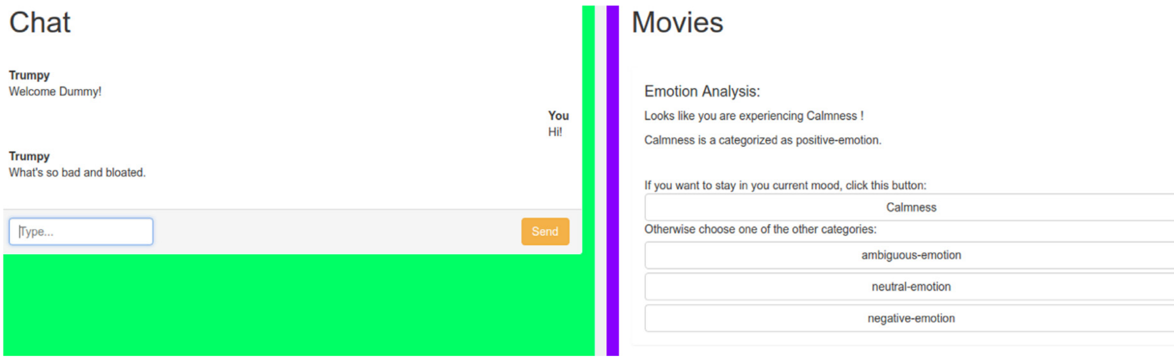


Fig. 2. Chatbot user interface.

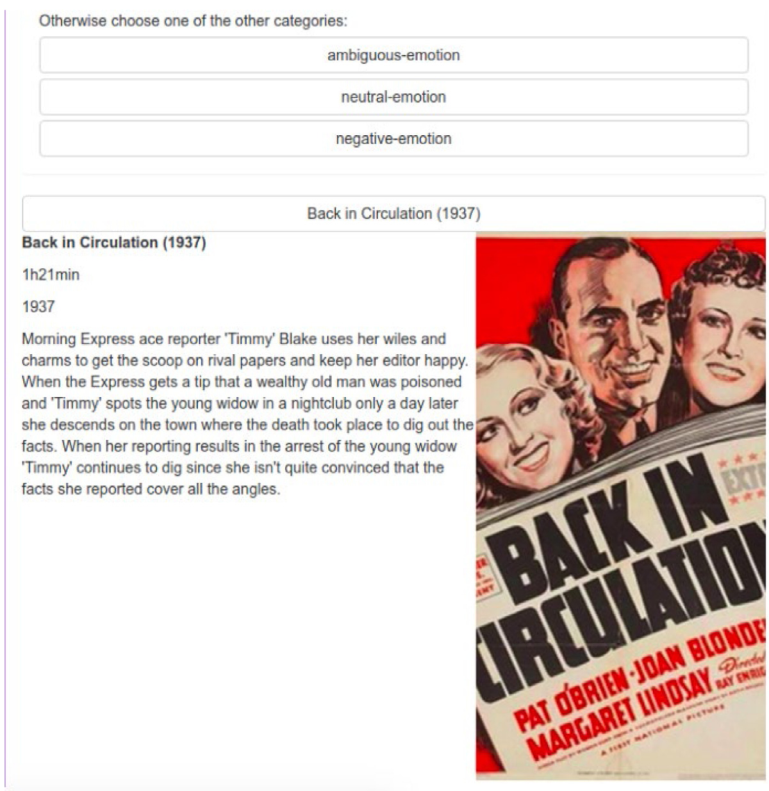


Fig. 3. Example of a movie recommendation based on emotions.

5. Evaluation

We extracted and analysed 2,627,476 movie reviews, namely, 122,557 from metractitic.com[11] and 2,504,919 from imdb.com[13] which resulted in 37,798,497 triples and 3,566,761 individuals. The assumed emotional values were extracted by a Bayesian classifier and a thesaurus, which did the training. The classifier was also used for extracting the emotional values from the user's chat messages. The emotion extraction delivered outcomes, which were not well distributed (Fig. 5a). Particularly, the emotions of apathy, joy, neutral and positive-expectation have been present excessively (over 30000 entries). This can be explained with the de-facto situation i.e. that these are the main emotions experienced when watching the movies. Or, one could assume that the result may still look somewhat different when further technological refinements of the developed system such as better training and other method(s) are applied. Data from social media platforms and direct human sensing data could be used as well.

In order to provide running efficiency the experiment was repeated. This time by extracting and analysing 166,630 movies from imdb.com[13] and 10725 movies from metractitic.com[11] as well as the corresponding reviews. Table 1 lists the run time for all tasks necessary to build the knowledge graph: (1) extracting data (2) analysing the reviews, (3) creating the knowledge graph, (3) uploading the graph to a suitable engine (in this case GraphDB[19]). All steps have been executed on a Lenovo ThinkPad running Windows 10 with 16 GB of RAM and an Intel i7-8550U (x64-based) CPU. It is also important to note that the interval between requests while extracting the data had to be increased from last time since IMDB as well as Metacritic would have blocked the requests otherwise. An interval of 50 s was chosen for both websites since it did not result in any rejected request after increasing it in steps of 5s starting at 30 s. As can be seen in Fig. 5b, the emotion distribution is similar to the one of the original experiment. The spike in movies categorized with "Apathy" (from 30000 to 34136) (Fig. 5a) may stem from the content of the reviews since the data of the new and the original

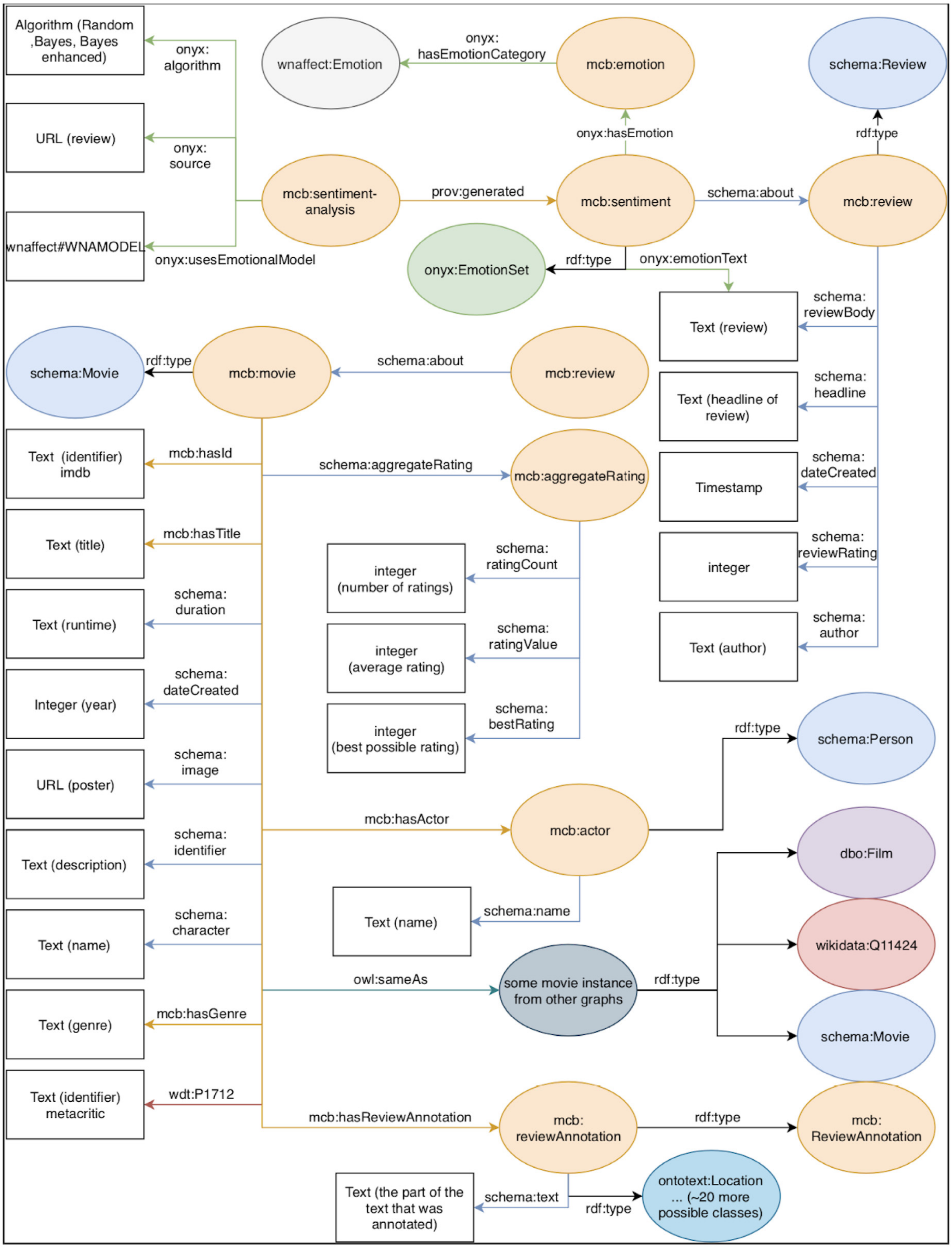
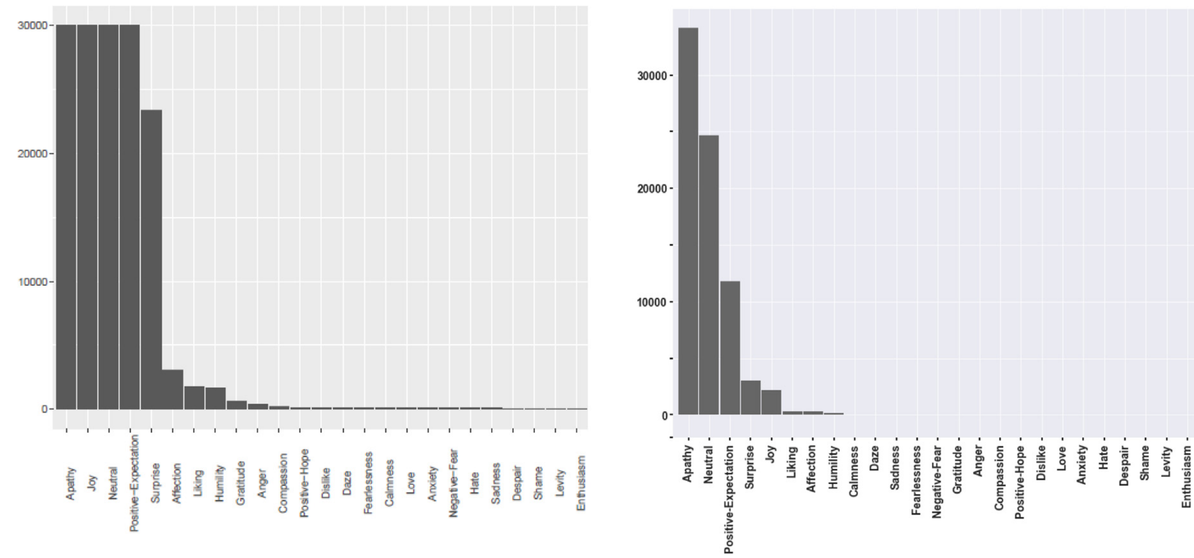


Fig. 4. Final knowledge graph.

experiment differ and both do not properly represent the whole volume of the data available. For the original experiment the data was gathered during November 2018 while for the repeated experiment was gathered during March 2021. Comparing both Fig. 5a and 5b further shows that the number of movies categorized with "Joy" has significantly decreased (from 30000 to 2156). A decline can also be noticed for the category "Positive Expression" (from 30000 to 11794).

In the related work section, the limitations of existing emotion-based recommender systems, one of which is the representation of emotions, have been discussed. Table 2 presents a comparison of several recommender systems based on their underlying algorithms (collaborative filtering, context-based and



5(a) 5(b)

Fig. 5. Initial distribution of the extracted emotions (5a) and the distribution of extracted emotions after repeating the experiment (5b).

Table 1
Running efficiency for all tasks necessary to build the knowledge graph.

Task	Time (in ms)	Time (human readable)	Notes
Extract data from IMDB	485871388.197	5 d 14 h 57 m 51 s	Request interval: 50 s
Extract data from Metacritic	55959325.627	15 h 32 m 39 s	Request interval: 50 s
Classify IMDB data	14690303.249	4 h 4 m 50 s	–
Classify Metacritic data	3472884.929	57 m 52 s	–
Create the knowledge graph	2185847.584	36 m 25 s	–
Insert into GraphDB	109035.741	1 m 49 s	On local system

hybrid) and the benefits that a knowledge graph could bring: extensibility, improved recommendation quality, better emotion discoverability and elimination of redundancy.

An example content-based system such as in [9] would benefit from a knowledge graph representation as it will provide improved emotion categorization and will allow recommendations of movies from other emotional domains. Further, in [9] emotions are stored as lexicons which could lead to redundancy and low latency [43]. When it comes to recommendation systems that are mainly based on human-interaction, time is of essence and such issues could ruin the user experience. These issues could be solved by using a knowledge graph representation of the emotions and reasoning.

Similar issues are found in collaborative-filtering recommender systems that use databases and knowledge bases to store sentiment. In [44] and [45] only the general emotions such as happiness, sadness and anger are annotated. A knowledge graph such as ours provides a formal, explicit specification of emotions and allows different levels of granularity. For example, to differ between positive/negative emotions or between all the emotion sub-categories. Additionally, a knowledge graph would provide the side information needed for context-based systems as well as for hybrid systems. Transforming any collaborative-filtering system into a hybrid system would further fix the data sparsity and cold-start problem as described by Guo et al. [8].

Hybrid systems such as [10] and [14], which are a combination of collaborative-filtering and context-based systems, would benefit from the ability of a knowledge graph to be reused and further extended. Replacing the used database and lexicon with a graph representation of emotions will improve the response time of the recommendation system as well. Any recommender system could benefit from the conversational support that a knowledge graph offers and the focus on specific conversational aspects that could be achieved [35]. For example, the emotion-based recommender developed by Costa and Macedo [45] which aims to solve the issue of information overload.

Testing of the emotion-based recommender system showed that introducing emotional features (stored in a knowledge base) to the machine learning algorithm, improves the precision of the recommendations with up to 6% in comparison to using only machine learning [45]. Precision could be improved even further if the emotional features are represented with a knowledge graph such as the one presented in this paper. This is confirmed by the work of Guo et al. [8] who discuss the benefits of knowledge graphs in recommender systems. The authors in [8] analyse recommendation systems in the book, movie, news, product, music and social media domains and conclude that integrating a knowledge graph improves both accuracy and interpretability [8]. Further, according to Gao et al. [67], using a knowledge graph enables recommender systems to capture not only the user–item interactions but also the rich item–item/user–user relations, which could be used to provide more accurate recommendations.

6. Conclusions and future work

In conclusion, a knowledge graph based on the users' emotions was created and emotional reasoning via a chatbot was showcased. The resulting knowledge graphs have been published in open access and studies on performance and the prevalent emotions have been made. We have also shown that the representation of emotions changes over time, and thus it is valuable to repeatedly create new versions of the knowledge graph. Linking methods were used and additional information such as movie genre was displayed. By retraining the chatbot on movie reviews,

Table 2
Benefits of using a knowledge graph.

Type of a system	Representation of emotions	Source of emotions	Benefits of using a knowledge graph representation of emotions				
			Data representation (e.g. sparsity impact)	Granularity of emotions	Extensibility	Recommendation quality	Additional characteristics
Content-based Ferweda et al. [9],	Lexicon	Surveys, social media.	Moderate impact	Extend further or provide better categorization of emotions.	Allow recommendations of movies which are categorized with different emotions but have similar characteristics (e.g. producer, country, actors).	Improved quality	Interpretability, discovering similar emotions, eliminating redundancy and better response time.
Collaborative-filtering Zheng et al. [44], Costa et al. [45]	Database, Knowledge Base	LDOS-CoMoDa data set (900 items and 1600 ratings [46]), web-crawling.	High impact	Discovery and annotation of more specific emotions (e.g. pride, disgust, shame, embarrassment).	Allow future reuse of emotions.	Improved quality	Interpretability, discovering similar emotions, eliminating redundancy and better response time.
Hybrid (content based and collaborative filtering) Gonzáles et al. [10], Ho et al. [14]	Lexicon, Database	Experiments carried out on emotional intelligence, colour surveys.	Moderate impact	Better classification of emotions.	Allow future reuse of emotions.	Improved quality	Interpretability, discovering similar emotions, eliminating redundancy and better response time.

it was able to make movie recommendations relying on emotions represented in the created knowledge graph. Further developments can be made in the user interface and user interaction. Further research is needed in the area of natural language generation, since Markov Chains are not context aware [47]. The classification tasks could be improved, and the employed Naïve Bayes approach could have been suboptimal. However, data about movies was acquired successfully, which led to a large knowledge base, allowing a more advanced movie recommendation process. For the chatbot prototype different techniques e.g. collaborative filtering and content-based filtering as defined in [48] and [49] could be applied in the future better showcase the utilization. The data acquisition part of the presented approach will benefit from extension of the knowledge base with the different types of methods such as image analysis and feature extraction from the video files of the movies [50].

The whole process of developing and implementing a knowledge graph may be used as a guide, which also answers the main research question *on how to create a knowledge graph representing human emotions in the domain of movie recommendation*. Using our approach, it should also be possible to develop knowledge graphs for other domains and thus, as discussed in Section 5, a knowledge graph proves to be a feasible improvement to emotion-based recommender systems. Emotion-aware reasoning and recommendation is and will be a prevailing area of interest as developers aim to create emotionally intelligent machines that make decisions which cater to the users' genuine interests on their own.

With the availability of the knowledge graphs and a possibility to easily construct them, it becomes clear that more and more semantic approaches towards explainable AI are possible in real life settings. An example application of this research could be an improvement to the already existing customizable chatbot solutions such as the one offered by Onlim GmbH[28] [35]. Existing chatbots, such as developed by Onlim, could make use of the emotional reasoning that were elaborated on in this research. Emotion-aware reasoning and recommendation is and will be a prevailing area of interest as developers aim to create emotionally intelligent machines that make decisions that cater to the users' genuine interests.

28 https://onlim.com.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been partly supported by an Interreg funded project KI-Net (AB292). We thank our colleague Juliette Opdenplatz for collaborating on previous early versions of this research.

References

[1] J.Z. Pan, G. Vetere, J.M. Gomez-Perez, H. Wu, Exploiting Linked Data and Knowledge Graphs in Large Organizations, Springer, Heildelberg, 2017, p. 281.
[2] H. Wang, Z. Wang, S. Hu, X. Xu, S. Chen, Z. Tu, DUSKG: A fine-grained knowledge graph for effective personalized service recommendation, Future Gener. Comput. Syst. (2019) 600–617.
[3] Z. Akbar, A. Fensel, D. Fensel, On generating stories from semantically annotated tourism-related content, in: OTM Confederated International Conferences On the Move to Meaningful Internet Systems, 2018, pp. 481-497.
[4] A. Cimmino, N. Mihindukulasooriya, F. Priyatna, M. Rico, Linked-fiestas: A knowledge graph to promote cultural tourism in spain, in: International Conference of Web Engineering, 2018, pp. 202-205.
[5] D. Chaudhuri, M. Rony, S. Jordan, S.J. Lehmann, Using a KG-copy network for non-goal oriented dialogues, in: International Semantic Web Conference, Springer, Cham, 2019, pp. 93–109.
[6] J. Manyika, J. Silberg, Notes from the AI Frontier: Tackling Bias in AI (and in Humans), McKinsey Global Institute, 2019.
[7] McKinsey Global Institute, available at: https://www.mckinsey.com/mgi/overview.
[8] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems, 2020, https://arxiv.org/abs/2003.00911.
[9] B. Ferwerda, M. Schedl, Enhancing music recommender systems with personality information and emotional states: A proposal, in: User Modeling, Adaptation and Personalization Workshops, 2014.
[10] G. González, B. López, J.L. de la Rosa, The emotional factor: an innovative approach to user modelling for recommender systems, in: Proceedings of AH 2002 Workshop on Recommendation and Personalization in e-Commerce, 2002.
[11] R. Imbert, A. Antonio, J. Segovia, The bunny dilema: stepping between agents and avatars, in: Proceedings of the 17th Twente Workshop on Language Technology, The Netherlands, 2000.

[12] Y. Zheng, Adapt to emotional reactions in context-aware personalization, in: Proceedings of the 4th Workshop on Emotions and Personality in Personalized Systems (EMPIRE) held in conjunction with the 10th ACM Conference on Recommender Systems (ACM RecSys), Boston, USA, 2016.
[13] L. Baltrunas, B. Ludwig, F. Ricci, Matrix factorization techniques for context aware recommendation, in: Proceedings of the fifth ACM conference on Recommender systems, New York, USA, 2011, pp. 301–304.
[14] A.T. Ho, I.L.L. Menezes, Y. Tagmouti, E-MRS: emotion-based movie recommender system, in: Proceedings of IADIS eCommerce Conference, 2006, pp. 1–8.
[15] Google, Introducing the What-If Tool, available at: https://cloud.google.com/blog/products/ai-machine-learning/introducing-the-what-if-tool-for-cloud-ai-platform-models.
[16] K. Wiggers, Google's What-If Tool for TensorBoard helps users visualize AI bias, available at: https://venturebeat.com/2018/09/11/googles-what-if-tool-for-tensorboard-lets-users-visualize-ai-bias/.
[17] Oxford University Press, Oxford Dictionary, available at: https://www.oxfordlearnersdictionaries.com/definition/english/bias_1.
[18] R. Bellamy, K. Dey, M. Hind, S.C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K.N. Ramamurthy, J.T. Richards, D. Saha, P. Sattigeri, M. Singh, K.R. Varshney, Y. Zhang, AI fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias, IBM J. Res. Dev. 63 (2019) 15, 4:1-4.
[19] A. Koene, L. Dowthwaite, S. Seth, IEEE P7003 standard for algorithmic bias considerations: work in progress paper, in: the International Workshop, 2018, pp. 38-41.
[20] J. Kleinberg, J. Ludwig, S. Mullainathan, A. Rambachan, Advances in big data research in economics: algorithmic fairness, in: AEA Papers and Proceedings, Vol. 108, 2018, pp. 22-27.
[21] A. Amini, A. Soleimany, W. Schwarting, N.S. Bhatia, D. Rus, Uncovering and mitigating algorithmic bias through learned latent structure, in: AIES, 2019, pp. 290–294.
[22] P. Pandey, Data Preprocessing concepts, available at: https://towardsdatascience.com/data-preprocessing-concepts-fa946d11c825.
[23] D. Kumar, Introduction to data preprocessing in machine learning, 2018, available at: https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d.
[24] M. Hardt, E. Price, N. Srebo, Equality of opportunity in supervised learning, in: the 30th Conference on Neural information Processing Systems, 2016.
[25] I. Bruha, Pre and post-processing in machine learning and data mining, in: Lecture Notes in Computer Science, Vol. 2049, Springer, Berlin, Heidelberg, 2001.
[26] H. Dong, S.C. Hui, Y. He, Structural analysis of chat messages for topic detection, Online Inf. Rev. 30 (5) (2006) 496–516.
[27] J. Bengel, S. Gauch, E. Mittur, M. Vijayaraghavan, Chattrack: Chat room topic detection using classification, Intell. Secur. Inf. 3073 (2004).
[28] J. Hermle, S. DellaVigna, Does media concentration lead to biased coverage? Evidence from movie reviews, Rev. Econom. Stud. 84 (4) (2017) 1510–1550.
[29] N.S. Koh, N. Hu, E.K. Clemons, Do online reviews reflect a product's true perceived quality? An investigation of online movie reviews across cultures, Electron. Commer. Res. Appl. 9 (5) (2010).
[30] Wikipedia, Theory of Planned Behaviour, available at: https://en.wikipedia.org/wiki/Theory_of_planned_behavior.
[31] Wikipedia, Theory of Reasoned Action, available at: https://en.wikipedia.org/wiki/Theory_of_reasoned_action.
[32] R. Gila, J. Virgili-Gomáa, R. Garcíaa, C. Masonb, Emotions ontology for collaborative modelling and learning of emotional responses, Comput. Hum. Behav. 51 (Part B) (2015) 610–617.
[33] M. Dragoni, S. Poria, E. Cambria, Ontosenticnet: A commonsense ontology for sentiment analysis, IEEE Intell. Syst. (2018).
[34] K. Janowicz, B. Yan, B. Regalia, R. Zhu, G. Mai, Debiasing knowledge graphs: why female presidents are not like female popes, in: the International Semantic Web Conference (P & D/Industry/BlueSky), CEUR-WS 2180, 2018.
[35] D. Fensel, U. Simsek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich, A. Wahler, Knowledge Graphs: Methodology, Tools and Selected Use Cases, Springer, 2020.
[36] J.B. Lovins, Development of a stemming algorithm, in: Mechanical Translation and Computational Linguistics, Vol. 11, Cambridge, Massachusetts, 1968.
[37] J. Dean, S. Ghemawat, MapReduce: simplified data processing on large clusters, in: the Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, 2004, pp. 137-150.
[38] L.E. Sucar, Bayesian Classifiers: Probabilistic Graphical Models, Springer, 2015.
[39] J.G.R. Cheng, Comparing bayesian network classifiers, in: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, 1999, pp. 101–108.
[40] A.G. Jivani, A comparative study of stemming algorithms, Int. J. Comput. Technol. Appl. 2 (2011) 1930–1938.
[41] IBM, Apache MapReduce, available at: https://www.ibm.com/analytics/hadoop/mapreduce.
[42] A. Jayant, Data Science and Machine Learning Series: Markov Chains for Text Generation (NLP), Technics Publications, 2020.
[43] T. Trippel, The lexicon graph model: A generic model for multimodal lexicon development, 2006, pp. 119–132.
[44] Y. Zheng, B. Mobasher, R. Burke, Emotions in context-aware recommender systems, in: Emotions and Personality in Personalized Services, Human–Computer Interaction Series, Springer, Cham, 2016.
[45] H. Costa, L. Macedo, Emotion-based recommender system for overcoming the problem of information overload, in: The 11th International Conference on Practical Applications of Agents and Multi-Agent Systems, Volume: Highlights on Practical Applications of Agents and Multi-Agent Systems, Communications in Computer and Information Science, Springer, 2013.
[46] A. Košir, A. Odić, M. Kunaver, M. Tkalčič, J.F. Tasič, Database for contextual personalization, English Edition, Elektrotehniskš Vestnik 78 (5) (2011) 270–274.
[47] C.T. Lenard, K.C. Chan, T.M. Mills, An introduction to Markov chains, in it's my maths: Personalized mathematics learning, Math. Assoc. Vic. Cliveden (2012) 40–47.
[48] A.T. Ho, I.L.L. Menezes, Y. Tagmouti, E-mrs: Emotion-based movie recommendation system, in: Proceedings of IADIS e-Commerce Conference, 2006, pp. 1-8.
[49] K. Wakil, R. Bakhtyar, K. Ali, K. Alaadin, Improving web movie recommender system based on emotions, Int. J. Adv. Comput. Sci. Appl. (6) (2015) 218–226.
[50] M.H. Rimaz, M. Elahi, F.B. Moghadam, C. Trattner, R. Hosseini, M. Tkalčič, Exploring the power of visual features for the recommendation of movies, in: Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization, 2019, pp. 303-308.

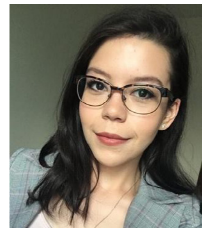
Further reading

[1] M. Alasaarela, The rise of emotionally intelligent AI, 2017, available at: https://machinelearnings.co/the-rise-of-emotionally-intelligent-ai-fb9a814a630e.
[2] Google, Google Assistant, available at: https://assistant.google.com.
[3] Google, Google, available at: https://www.google.com.
[4] F. Calmon, D. Wei, B. Vinzamuri, K.N. Ramamurthy, K.R. Varshney, Optimized data pre-processing for discrimination prevention, in: the 31st Conference on Neural Information Processing Systems, 2017.
[5] B. Hu Zhang, B. Lemoine, M. Mitchell, Mitigating unwanted biases with adversarial learning, in: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics and Society, 2018, pp. 335-340.
[6] B. Zafar, I. Valera, K. Gummadi, Fairness constraints: A mechanism for fair classification, in: the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2015.
[7] Rottentomatoes, available at: https://www.rottentomatoes.com.
[8] Metacritic, available at: https://www.metacritic.com.
[9] Twenty Century Fox, Fox Corporation, 20th Century Fox, available at: https://www.foxmovies.com.
[10] IMDb, available at: https://www.imdb.com.
[11] Douban, available at: https://www.douban.com.
[12] D.C. Ong, J. Zaki, N.D. Goodmanc, Computational models of emotion inference in theory of mind: A review and roadmap, in: Topics in Cognitive Science, 2017.
[13] A.M. Florea, M. Puică, Emotional belief-desire-intention agent model: Previous work and proposed architecture, Int. J. Adv. Res. Artif. Intell. 2 (2) (2013).
[14] G.M. Rao, Modeling rational agents within a BDI-architecture, in: Proceedings of Knowledge Representation and Reasoning (KR & R-91), 1991, pp. 473–484.
[15] A. Damasio, Descartes' Error: Emotion, Reason and the Human Brain, Gosset/Putnam Press, 1994.
[16] J. Russel, A circumplex model of affect, J. Personal. Soc. Psychol. 39 (6) (1980) 1161–1178.
[17] J.M. Gomez-Perez, J.Z. Pan, G. Vetere, H. Wu, Enterprise knowledge graph: An introduction, in: Exploiting Linked Data and Knowledge Graphs in Large Organizations, Springer, Cham, 2017.
[18] A. Schultz, A. Matteini, R. Isele, P.N. Mendes, C. Bizer, C. Becker, LDIF - A framework for large-scale linked data integration, in: 21st International World Wide Web Conference: Developers Track, Lyon, France, 2011.
[19] Google, Microsoft, Yahoo and Yandex, schema.org, available at: https://schema.org/.
[20] C. Strapparava, A. Valitutti, Wordnet-affect: an affective extension of wordnet, in: Proceedings of the 4th International Conference on Language Resources and Evaluation, 2004.

[21] R. Zemel, Y.L. Wu, K. Swersky, T. Pitassi, C. Dwork, Learning fair representations, in: Proceedings of the International Conference of Machine Learning, 2013, pp. 325–333.
[22] F. Kamiran, T. Calders, Data preprocessing techniques for classification without discrimination, Knowl. Inf. Syst. 33 (1) (2012) 1–33.
[23] B. Galitsky, Developing Enterprise Chatbots: Learning Linguistic Structures, Springer International Publishing, 2019.

Arno Breitfuss received his B.Sc. in Computer Science in 2017 from Innsbruck University. Currently he is pursuing his M.Sc. in Computer Science at the same university while working half-time at REDcatch GmbH. His current research interests are Natural Language Processing, Physical Simulations and Deep Learning for Audio Data.

Karen Errou received her B.Sc. in Computer Science from Lebanese University in 2017. Currently she is perusing M.Sc. in Computer science at the University of Innsbruck. She has experience in various programming languages and in working as Software and Front-End Developer.



Anelia Kurteva received her M.Sc. in Advanced Computer Science in 2019 from Cardiff University where she also completed her B.Sc. in Computer Science and Visual Computing. Currently she is pursuing a Ph.D in Computer Science at STI Innsbruck, Department of Computer Science, Austria. Her current research interests are Data Visualization, Semantic Technology, Data Mining and Pattern Recognition.

Anna Fensel is Associate Professor at STI Innsbruck, Department of Computer Science, University of Innsbruck, Austria. Earlier she worked as a Senior Researcher at FTW – Telecommunications Research Center Vienna, Austria, a Research Fellow at the University of Surrey, UK, and as a project employee at DERI Innsbruck, University of Innsbruck, Austria. Anna has earned both her habilitation (2018) and her doctoral degree (2006) in Computer Science at the University of Innsbruck. Prior to that, she has received a diploma in Mathematics and Computer Science equivalent to Master degree in 2003 from Novosibirsk State University, Russia. Anna has been working in more than 20 European and nationally funded projects related to Semantic technologies, as a consortium coordinator, a principal investigator and a technical contributor. She has been a co-organizer/Program Committee member of more than 100 scientific events including being in chair roles at large conferences in her field such as SEMANTiCS and ESWC, a reviewer for numerous journals and a project proposal evaluator for funding agencies. She has co-authored more than 110 refereed publications.