STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LXVII**, Number 1, 2022
DOI: 10.24193/subbi.2022.1.01

CAN YOU GUESS THE TITLE? GENERATING EMOJI SEQUENCES FOR MOVIES

ANNA BAJCSI, BARBARA BOTOS, PÉTER BAJKÓ, AND ZALÁN BODÓ

Abstract. In the culture of the present emojis play an important role in written/typed communication, having a primary role of supplementing the words with emotional cues. While in different cultures emojis can be interpreted and thus used differently, a small set of emojis have clear meaning and strong sentiment polarity. In this work we study how to map natural language texts to emoji sequences, more precisely, we automatically assign emojis to movie subtitles/scripts. The pipeline of the proposed method is as follows: first the most relevant words are extracted from the movie subtitle, and then these are mapped to emojis. In order to perform the mapping, three methods are proposed: a lexical matching-based, a word embedding-based and a combined approach. To demonstrate the viability of the approach, we list some of the generated emojis for a randomly selected movie subset, showing also the deficiencies of the method in generating guessable sequences. Evaluation is performed via quizzes completed by human participants.

1. Introduction

Emojis and emoticons are commonly used to express emotions in online written communication. They are preferred tools, because in written communication mimics and tone are hard to convey, however, it is much more easily achieved by emojis.

In this paper we try to tackle a creative problem, to generate emoji sequences describing a movie. While guessing the movie title from – usually

Received by the editors: 3 November 2021.

2010 *Mathematics Subject Classification.* 68T50, 68T30.

1998 *CR Categories and Descriptors.* I.2.7 [**ARTIFICIAL INTELLIGENCE**]: Natural Language Processing – *Text analysis*; I.2.m [**ARTIFICIAL INTELLIGENCE**]: Miscellaneous.

Key words and phrases. natural language processing, emoji, keyword extraction, movie scripts, lexical matching, word embedding.

human-generated – emoji sequences is a popular game for movie enthusiasts[1], generating emojis describing a movie can be considered a task similar to automatic (text) summarization [19]. The main idea is to extract keywords from the movie subtitle and match them with emojis. In the present work we calculate *tf-idf* scores to obtain the most important, most representative words of the movie script. The most difficult part of the emoji sequence generation is mapping the words to emojis. We describe two main approaches in this paper: lexical matching between keywords and emoji names, as well as a word embedding-based method. We also try to improve on the results by combining these two approaches, as well as by taking into account the title of the movie and the chronological order of the keywords.

The rest of the paper is organized as follows. In Section 2 we briefly describe how emojis shape the online media today, and Section 3 presents the natural language processing problems where emojis can be useful instruments. The structure of our system generating emoji sequences from movie scripts is presented and detailed in Section 4. The experiments and the results obtained are described in Section 5, and the paper concludes with Section 6, discussing the results and specifying potential future research directions.

2. Emojis in online media

Emoticons are inventions of the 19th century, but the first recorded online use occurred only in 1982 [11]. The word emoticon stems from "emotion icon" [6], referring to a pictorial representation of a facial expression, gesture using characters (e.g. punctuation marks, parentheses, etc.). While in North America the horizontal representation of such faces became prevalent, e.g. `:-)`, in Japan the so-called kaomojis had been used [24], which are vertical emoticons like `(^_^)`. Emojis – meaning pictographs in Japanese (*e* = picture, *moji* = characters) [2] – appeared in the late 1990s at the NTT Docomo telecommunications company as the work of the designer Shigetaka Kurita [24]. The resemblance to the English word "emotion" or even "emoticon" is merely coincidental [25]. As pointed out in [24], the appearance of emojis in Japan could be explained by the complexity of the predecessor kaomojis.

[1] See for example the BuzzFeed movie quiz *"If You Can Identify 8/10 Of These Movies From The Emojis, You're Officially A Cinephile"* (`https://www.buzzfeed.com/hayleyrochelletillett/identify-movies-by-emojis`).

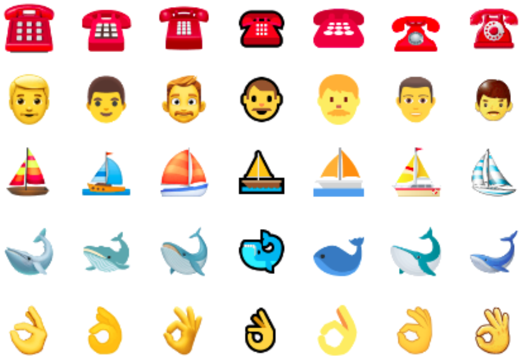


Figure 1. "Call me Ishmael." – the opening sentence of Herman Melville's *Moby Dick* represented as a (vertical) list of emojis in Fred Benenson's *Emoji Dick*, reproduced after [21]. The 7 columns represent different vendors: Apple, Google, Facebook, Windows, Twitter, JoyPixels, and Samsung.

The standardization of emojis took place in Unicode 6.0, containing 722 emojis, Kurita's initial set having only 176 pictograms [12]. The current Unicode 14.0 standard includes 3633 emojis.[2]

Emojis became popular worldwide only in the 2010s, by appearing in various mobile operating systems and web browsers. To emphasize their increasing popularity, we mention the first ever published book written using emojis only, appeared in 2010: Fred Benenson's *Emoji Dick*, the crowdsourced and crowdfunded translation of Herman Melville's famous novel *Moby Dick*.[3] In 2014 Microsoft added emoji support to Bing search.[4] In 2013 Katy Perry released a lyric video for her hit song *Roar*, showing the lyrics as a mixture of words and emojis.[5] We end the series of examples by a statistics from a few years ago: by 2015, already half of the Instagram posts contained at least one emoji.[6] Fig. 1 shows the opening emoji sequence (U+260E, U+1F468, U+26F5, U+1F40B, U+1F44C) of *Emoji Dick* using different sets of emojis.

[2] https://www.unicode.org/emoji/charts-14.0/emoji-counts.html

[3] http://emojidick.com/

[4] https://blogs.bing.com/search/2014/10/27/do-you-speak-emoji-bing-does/

[5] http://www.mtv.com/news/1712176/katy-perry-roar-lyric-video/

[6] https://instagram-engineering.com/emojineering-part-1-machine-learning-for-emoji-trendsmachine-learning-for-emoji-trends-7f5f9cb979ad

3. Emojis in natural language processing applications

With the growing popularity of emojis, these are used worldwide in numerous apps and platforms alongside text in different languages. Emojis play different roles, they offer “both complementary and supplementary relations to words” [15]. Because of this reason, many research communities took interest in emojis and their roles in written language. Although most research can be found in computer and communication science, they represent a popular research topic in marketing, behavioral science, psychology, linguistics, education, etc. as well [2].

In [7] the authors created the emoji2vec model, which – similarly to word2vec [17] – assigns continuous vector representations to emojis, based on their description: the generated emoji embeddings are the sum of the word2vec embeddings of the words from the description. The obtained emoji embeddings outperform word embeddings on the task of sentiment analysis, using a large collection of tweets.

Since in most cases emoticons and emojis are used to express emotions, attitudes, it is not surprising that they are applied most frequently in sentiment analysis, opinion mining. One of the conclusions of the analysis carried out in [26] is that a small set of emoticons have strong and clear polarity, but the rest of it, a much larger set maintain more complicated sentiments. The authors of [8] apply distant supervision to perform sentiment analysis, i.e. they use the emoticons as noisy labels, achieving high classification accuracies in the experiments. In [11] the construction of the first emoji sentiment lexicon, called Emoji Sentiment Ranking, is presented. The lexicon – which is similar to SentiWordNet [1] – contains the 751 most frequently used emojis in Twitter messages, the scores being relative frequencies in tweets having different polarities. The work [29] discusses the importance of emoticons in sentiment analysis, summarizes the existing methods, and it also briefly addresses issues such as sarcasm detection. In [10] sarcasm detection is performed by comparing the polarity of the text and of the emojis.

A video search system is presented in [4], in which video retrieval is accomplished using emojis to formulate the query. Emojis are assigned to a video based on the title, as well as by object recognition on the video frames. The closest research to ours is the Image2Emoji model presented in [3], which assigns emoji sequences to real-world images. Similarly to [4], the word2vec embeddings of the recognized objects in the pictures, as well as embeddings

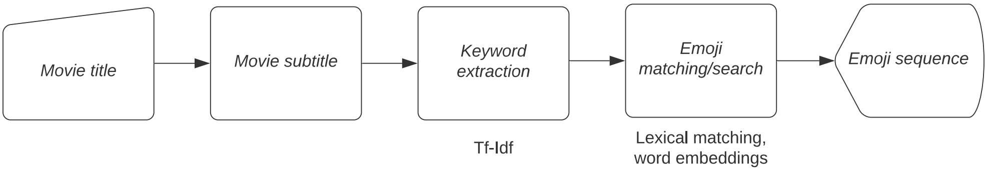


Figure 2. Scheme of the emoji sequence generation algorithm: given a movie title the movie subtitle is retrieved, from which the most relevant keywords are extracted, and finally, the keywords are matched with emojis using lexical matching and word embeddings to build the proper sequence.

of the picture title, description and tags are compared against the emoji embeddings to get a ranking of the most relevant emojis for a given picture.

The system proposed in this paper assigns emoji sequences to movies, using the movies' subtitles, first by extracting the keywords from the subtitle file, and then searching for the semantically nearest emojis for these.

4. Generating emojis from movie scripts

Emojis can be a powerful tool for natural processing algorithms to utilize. For example, the presence of a certain emoji can clearly indicate certain emotions in sentiment analysis, making such a prediction much more accurate. Sentiment analysis, however, is not the goal of this paper. We instead intend to summarize larger texts using emojis. For this purpose, we chose movie subtitles as our data source, and tried to produce a sequence of emojis to describe their plots. While at first this may seem a simple game intended only to entertain the user, summarizing a movie via summarizing its script by a handful of emojis involves many challenges. One such challenge is caused by the very small number of emojis: the retention ratio usually grows with the compression ratio [9], and in this scenario we are dealing with very low compression ratios, values around 0.0014.[7] Thus, although the goal is *just* to guess the title of the movie, it is difficult to achieve high retention ratios at such low compression rates, however, longer emoji sequences will similarly confuse the respondents.

[7] Using a random sample of 50 movie subtitles, we obtained an average number of candidate keywords of 4169.54, where a candidate keyword means that all of its characters are alphabetic, not a stopword, and having a minimum length of 3 letters. If we consider an average emoji sequence length of 6, we obtain a compression ratio of 0.0014.

In the following we will discuss the three main methods used to achieve the above-mentioned goal, our conclusions about the results and also possible future directions to expand this project. All three methods presented are based on keyword extraction and matching these keywords with existing emojis.

The scheme of the algorithm is shown in Fig. 2: we start with a movie title, obtain the subtitle of the given movie, extract its most relevant keywords, then match these keywords with emojis using different methods, and finally create the emoji sequence representing the movie.

4.1. **First approach: lexical matching.**

The basis of all methods proposed is keyword extraction. For this we used *tf-idf* (term frequency × inverse document frequency) scores, known as a weighting scheme for the bag-of-words representation of documents in text categorization [23, 22], but also as a successful keyword extraction approach as well [16, 27]. This method assigns a value to every word, based on how many times it appears in the given document, and how many times it appears in other documents. A frequently used version of *tf-idf* is the following,

$$tfidf(t,d) = f_{t,d} \cdot \log\left(\frac{N}{n_t}\right),$$

where $f_{t,d}$ is the frequency of word t in document d, N is the total number of documents in the corpus, and n_t denotes the number of documents containing word t. The more the word appears in the current document, and the fewer times in other documents, the higher the *tf-idf* value.

By this procedure, each word from a movie script will have a *tf-idf* assigned to it: the words with the highest such values are considered the keywords for a given movie. The keyword extraction step is common to all three methods presented below.

In the first method we check for emojis with names that match fully or partially with the extracted keywords. To measure the similarity between a keyword and an emoji, the Sørensen—Dice coefficient [5] was used,

$$DSC(X,Y) = \frac{2|X \cap Y|}{|X| + |Y|},$$

where X and Y denote two sets. More precisely, we used the convex combination of the Dice coefficients between the emoji name, treated as a set of tokens, and the extracted keywords, as well as the Dice coefficient of the keywords officially assigned to the emoji and the extracted keyword. For each extracted keyword the emoji with the highest Dice value is chosen, provided that this value exceeds a given threshold. At the end, duplicate emojis are removed from the sequence, keeping the first occurrence only.

This approach, however, has one major drawback: if none or just a very few keywords match the emojis, the generated sequence will not be a satisfactory one. A possible solution to this would be to extract a larger set of keywords from the documents, but then the relevancy of these could be questioned.

4.2. **Second approach: word embeddings.** This approach uses vector representations to define the similarities between words and emojis. The vector representation of a word is a series of real values, obtained by applying a computational framework providing continuous word representations, e.g. *word2vec* [17], *GloVe* [20] or *fastText* [18]. Using pretrained models we can get the vector representations of the words (keywords). To do the same for the emojis, the emojis' names are used by taking the average of the vectors of the words present in the emoji name.

Similarity is computed by simply calculating the cosine of the angle enclosed by the resulting vectors [22],

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|},$$

where $\mathbf{x}^T$ denotes the transpose of $\mathbf{x}$ and $\| \cdot \|$ is the Euclidean (ℓ_2) norm. Similarly to the previous approach, an emoji is selected if its cosine similarity is above a certain threshold, and duplicates are removed.

4.3. **Third approach: the combination of the previous two.** This third method simply combines the outcomes of the previous two methods by concatenating the resulting sequences one after the other, the first being the sequence obtained via lexical matching. Duplicate emojis are again removed from the output.

Potential improvements. During keyword extraction, the chronological order of the keywords is also taken into account, since a better ordering might result in a more logical enumeration. Using this method, the keywords (and later the emojis) will appear in the sequence in the same order as they appear in the movie (see Fig. 3(b)).

Furthermore, the movie title may hold a lot of information, therefore we considered *emojizing* the title too, i.e. placing the words of the movie title into the keyword list as well, applying the procedure described in Section 4.3

to the movie title (see Fig. 4). The resulting emojis – if any – are placed at the beginning of the final sequence.

5. Experimental results

As discussed in Section 4, we extracted the keywords from a given movie subtitle using *tf-idf* scores. Given a movie title, its subtitle is downloaded via the OpenSubtitles API[8]. The content of the subtitle file was preprocessed: tokenized, lowercased and the stopwords were removed. Next, the term frequency of each word in this document was calculated. For calculating the idf scores we used the OpenSubtitles dataset [14] (2018 version)[9]. The full English dataset contains a total of 446 612 subtitle files, but a single movie can have multiple subtitles corresponding to different versions (e.g. different authors/transcribers/translators). We used one subtitle per movie, resulting in 140 045 files processed, where the first file for each movie was taken. The lowercased dictionary of idf scores contains 1 716 310 different tokens.

To define the vector representation of the extracted keywords and the emojis the GloVe[10] model was used. It contains 400K tokens, extracted from the Wikipedia 2014 dump[11] and English Gigaword Fifth Edition[12], and their 100-dimensional vector representations. Since our method extracts no collocations but only single-word expressions as keywords, it was straightforward to calculate their GloVe representations – provided these were known tokens. In order to get continuous vector representations for the emojis, their name was tokenized, lowercased and stopword filtered, and the average of the tokens' GloVe vectors were considered. The emoji names, Unicodes and emoji keywords were obtained from EmojiNet [28]. When searching for the best matching emojis the Unicode list v13.1 was considered, omitting emojis with skin-tones,[13] resulting in a set of 1816 pictographs.

The proposed methods have a relatively large set of hyperparameters that should be selected following a systematic procedure (e.g. cross-validation), but since we did not have a usable benchmark dataset for this, we selected the following parameters in a trial-and-error fashion. Lexical matching is based on

[8] `http://www.opensubtitles.org/`

[9] `https://opus.nlpl.eu/OpenSubtitles-v2018.php`

[10] `https://nlp.stanford.edu/projects/glove/`

[11] `https://archive.org/details/enwiki-20141106`

[12] `https://catalog.ldc.upenn.edu/LDC2011T07`

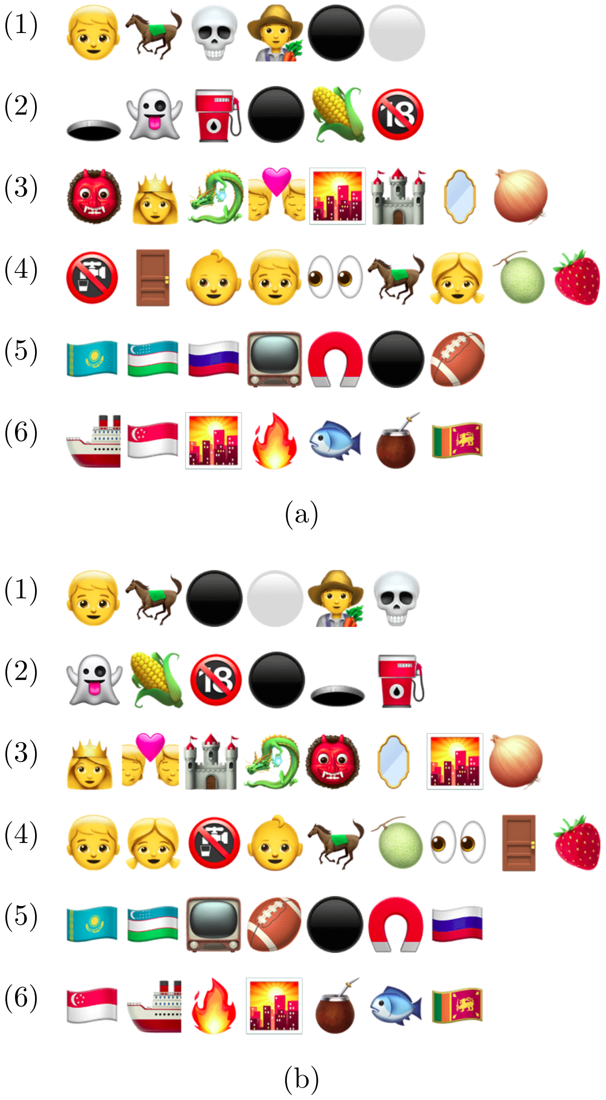
[13] `http://unicode.org/emoji/charts/emoji-list.html`

Table 1. Table showing the extracted keywords, the most similar emojis' Unicodes and the obtained scores with the combined model + keyword ordering.

Interstellar			*Pirates of the Caribbean*		
Keyword	**Unicode**	**Score**	**Keyword**	**Unicode**	**Score**
ghost	U+1F47B	0.8	singapore	U+1F1F8 U+1F1EC	0.8
corn	U+1F33D	0.8385	ship	U+1F6A2	0.8
twelve	U+1F51E	0.8228	fire	U+1F525	0.8
black	U+26AB	0.8971	sunset	U+1F307	0.8
hole	U+1F573	1.0	mate	U+1F9C9	1.0
fuel	U+26FD	0.918	fish	U+1F41F	0.8
			sri	U+1F1F1 U+1F1F0	0.9125

the Dice coefficient of an extracted keyword and the name and keywords of an emoji: 80% percent of the resulting score is given by the Dice similarity of the extracted keyword and the name of the emoji, and 20% of it is calculated as the Dice similarity between the extracted keyword and the keywords assigned to the emoji. The name and the emoji keywords are treated as sets, while the extracted keyword constitutes a single element set. To select a keyword/emoji into the generated sequence, a similarity threshold needs to be exceeded; we set this to 0.8 in our experiments. When working with word embeddings, the similarity threshold for cosine was also set to 0.8. The length of the generated emoji sequence is also important: being too short, there is no room for showing all the pictographs that would be good indicators for guessing the movie, while being too long could confuse the user. Both for the lexical matching and the word embedding-based methods we set this limit parameter to 6. We also used a similar limit parameter when considering the titles, setting it to 2 in both cases.

The emoji sequences obtained for the selected movies are shown in Fig. 3 and 4. Based on the test performed on a randomly chosen set of movie subtitles, the best results were obtained using the combined method, therefore we show some of the generated emoji sequences only by this approach. In order to present not only the bright side of the proposed method, we selected 3 movies for which quite decent sequences are generated, and also 3 other movies for which the extracted keywords and the generated emojis do not really make



(a)

(b)

Figure 3. Emoji sequences generated for 6 selected movies: (a) using the combined method, i.e. lexical maching combined with word embeddings, and (b) the same sequences shown in chronological order of the extracted keywords.

the movie guessable. The selected movies are the following: (1) *Django Unchained*[14], (2) *Interstellar*[15], (3) *Shrek*[16], (4) *Bird Box*[17], (5) *Borat*[18], (6) *Pirates of the Caribbean: at World's End*[19]. The first three of the sequences we

[14] `https://www.imdb.com/title/tt1853728`

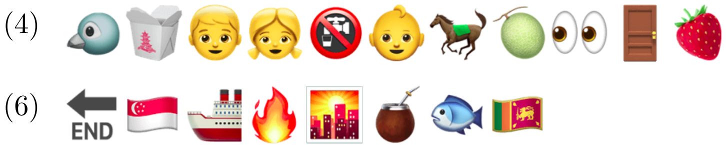


Figure 4. Emoji sequences obtained with the combined method applying chronological ordering and placing the emojis generated from the title at the beginning of the list. Only those movies are shown here, for which considering the title as well introduced new pictographs.

consider relevant and guessable, however, the other three are less successful. Fig. 3(a) shows the emoji sequences obtained using the combined method, i.e. lexical matching combined with the word embedding-based method, Fig. 3(b) displays the same sequences but in chronological order, while Fig. 4 shows the two motion pictures for which the introduction of the title affected the results. Table 1 lists the most relevant keywords found by the combined method, together with the Unicode code points of the best fitting emojis, as well as the similarity scores obtained for two movies.

Emoji sequence generation for movie scripts was implemented in Python, and its source code can be found at `https://github.com/bajcsianna/movie2emoji`.

5.1. **Evaluating the emoji sequences.** Evaluating the generated emoji sequences, that is evaluating the performance of the proposed method proved to be a difficult task. Since generating pictograms that show the main events and motifs of a movie can be considered a summarization task, using the ROUGE metric might seem appropriate [13]. The problem with the application of this measure is twofold: (i) no sufficiently large dataset of human-generated movie emoji sequences is available, (ii) the sequences – usually containing 2 to 6 pictograms – are too short for this metric.

In order to obtain an evaluation of the proposed methods, we randomly generated 10 + 10 pictograph sequences using the combined method (without

[15] `https://www.imdb.com/title/tt0816692`

[16] `https://www.imdb.com/title/tt0126029`

[17] `https://www.imdb.com/title/tt2737304`

[18] `https://www.imdb.com/title/tt0443453`

[19] `https://www.imdb.com/title/tt0449088`

TABLE 2. Accuracies obtained for the two movie emoji quizzes, the last line showing the overall accuracy.

	1st quiz	**2nd quiz**
Movie #1	13.15%	31.88%
Movie #2	0%	23.18%
Movie #3	0%	57.97%
Movie #4	0%	30.43%
Movie #5	0%	50.72%
Movie #6	36.82%	26.08%
Movie #7	2.63%	68.11%
Movie #8	65.78%	30.43%
Movie #9	26.31%	46.37%
Movie #10	44.73%	27.53%
	18.94%	39.27%

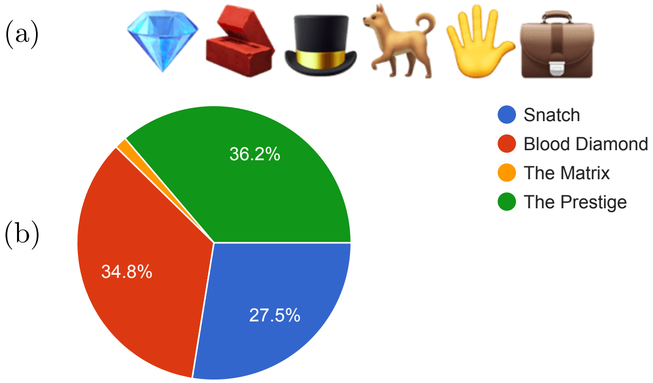


FIGURE 5. Example for an overly successful obfuscation: (a) emoji sequence obtained for *Snatch* and (b) the answers given by the participants. The results show that three of the four possible answers seemed equally acceptable for the participants of the survey.

taking into account the movie title), and assembled two surveys: (s_1) for guessing the movies by typing in the title, and (s_2) for guessing the movies in form of multiple choice questions with four possible answers. The selected movies contain only motion pictures and animated films, not necessarily blockbusters, released between 1994 and 2019. The surveys have been sent to students of the Babeş–Bolyai University and members of the academic staff via learning management systems, business communication platforms and social networking sites. For the two quizzes we received 38 and 69 answers, respectively.

The accuracy results are shown in Table 2. As it was anticipated by us, the second quiz produced better results, since the possible answers were narrowed down to four, as in the case of the first quiz all existing films – from the above-mentioned period, as this was brought to the attention of the respondents – had to be considered. For the first quiz the majority answers correctly determined 3 out of 10 movies, while this number was 4 for the second quiz.

In survey (s_2), in addition to the correct title we selected three incorrect ones, but not in a random manner: the incorrect titles were chosen such that at least one emoji matched the main motifs of the movie. However, this obfuscation sometimes worked too well. In Fig. 5(a) the emojis generated for the movie *Snatch*[20] are shown, while the pie chart in (b) shows the distribution of the participants' answers. The answers for other movies show the signs of too successful obfuscation as well, which certainly affects the results. Choosing the movies to generate the emoji sequences for is also a difficult task, since one cannot assure that all movies are known by the participants. Similarly, one cannot expect the same degree of seriousness of quiz completion from all participants. Therefore, we suggest to consider the obtained accuracy scores as lower limits of guessability of the generated emoji sequences.

The quizzes used in the evaluation process – with the correct answers – are available at the following links: movie2emoji I.: `https://forms.gle/YHGeky6oCcxAwjkd8`, movie2emoji II.: `https://forms.gle/XFkjCHtZXbcbkjbN8`.

6. Discussion and future work

In this paper we presented a system that is able to assign emoji sequences to movies, based on the movie's subtitle. The pipeline of the proposed method is simplistic but rather effective: extraction of the most relevant keywords from the subtitle (or script) of the movie, and then assigning emojis to these. We experimented with three approaches: (i) lexical matching using Dice coefficients to determine similarity, (ii) a word embedding-based approach using cosine similarity, and (iii) the combination of these two.

[20] `https://www.imdb.com/title/tt0208092`

While the obtained results are promising, there is room for improvements and further experiments too. Since the keyword extraction model applied is a central component of this approach, we consider experimenting with other such models important. In the present system information is acquired only from the movie subtitle, providing an efficient means to generate the emojis, which could be supplemented by object recognition models considering also the movie frames when available, similarly to [3]. Studying the effect of stemming/lemmatization of the extracted keywords on the output is also left as a future work. Part of speech tagging and selection of words belonging to important parts (nouns, verbs, adjectives), as well as considering word collocations or neighborhoods of the selected keywords can also positively affect the performance of our system.

Acknowledgements

We would like to thank the students and the academic staff of the Babeş–Bolyai University who participated in our experiments to evaluate the proposed models.

References

[1] Baccianella, S., Esuli, A., and Sebastiani, F. SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *LREC* (2010), vol. 10, pp. 2200–2204.

[2] Bai, Q., Dan, Q., Mu, Z., and Yang, M. A systematic review of emoji: Current research and future perspectives. *Frontiers in Psychology 10* (2019), 2221.

[3] Cappallo, S., Mensink, T., and Snoek, C. G. Image2emoji: Zero-shot emoji prediction for visual media. In *Proceedings of the 23rd ACM International Conference on Multimedia* (2015), pp. 1311–1314.

[4] Cappallo, S., Mensink, T., and Snoek, C. G. Query-by-emoji video search. In *Proceedings of the 23rd ACM International Conference on Multimedia* (2015), pp. 735–736.

[5] Dice, L. R. Measures of the amount of ecologic association between species. *Ecology 26*, 3 (1945), 297–302.

[6] Dresner, E., and Herring, S. C. Functions of the nonverbal in CMC: Emoticons and illocutionary force. *Communication Theory 20*, 3 (2010), 249–268.

[7] Eisner, B., Rocktäschel, T., Augenstein, I., Bošnjak, M., and Riedel, S. emoji2vec: Learning emoji representations from their description, 2016.

[8] Go, A., Bhayani, R., and Huang, L. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford 1*, 12 (2009), 2009.

[9] Hovy, E. Text summarization. In *The Oxford Handbook of Computational Linguistics*, R. Mitkov, Ed. Oxford University Press, Oxford, 2004, ch. 32.

[10] Karthik, V., Nair, D., and Anuradha, J. Opinion mining on emojis using deep learning techniques. *Procedia Computer Science 132* (2018), 167–173.

[11] Kralj Novak, P., Smailović, J., Sluban, B., and Mozetič, I. Sentiment of emojis. *PloS One 10*, 12 (2015), e0144296.

[12] Kumari, R., and Gangwar, R. Use of expression based digital pictograms in interpersonal communication: a study on social media and social apps. *International Journal of Innovative Knowledge Concepts 6* (2018), 11.

[13] Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out* (2004), ACL, pp. 74–81.

[14] Lison, P., and Tiedemann, J. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (Portorož, Slovenia, May 2016), European Language Resources Association (ELRA), pp. 923–929.

[15] Mei, Q. Decoding the new world language: Analyzing the popularity, roles, and utility of emojis. In *Companion Proceedings of The 2019 World Wide Web Conference* (New York, NY, USA, 2019), WWW '19, Association for Computing Machinery, p. 417–418.

[16] Mihalcea, R., and Csomai, A. Wikify! Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management* (2007), pp. 233–242.

[17] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space, 2013.

[18] Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)* (2018).

[19] Nenkova, A., and McKeown, K. A survey of text summarization techniques. In *Mining Text Data*. Springer, 2012, pp. 43–76.

[20] Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 1532–1543.

[21] Radford, W., Chisholm, A., Hachey, B., and Han, B. :telephone::person::sailboat::whale::okhand:; or "Call me Ishmael" – How do you translate emoji? In *Proceedings of Australasian Language Technology Association Workshop* (2016), pp. 150–154.

[22] Schütze, H., Manning, C. D., and Raghavan, P. *Introduction to information retrieval*. Cambridge University Press, 2008.

[23] Sebastiani, F. Machine learning in automated text categorization. *ACM Computing Surveys (CSUR) 34*, 1 (2002), 1–47.

[24] Stark, L., and Crawford, K. The conservatism of emoji: Work, affect, and communication. *Social Media + Society 1*, 2 (2015), 2056305115604853.

[25] Taggart, C. *New words for old: Recycling our language for the modern world*. Michael O'Mara Books, 2015.

[26] Wang, H., and Castanon, J. A. Sentiment expression via emoticons on social media. In *International Conference on Big Data* (2015), IEEE, pp. 2404–2408.

[27] Wartena, C., Brussee, R., and Slakhorst, W. Keyword extraction using word co-occurrence. In *International Workshops on Database and Expert Systems Applications* (2010), IEEE, pp. 54–58.

[28] Wijeratne, S., Balasuriya, L., Sheth, A., and Doran, D. EmojiNet: An open service and API for emoji sense discovery. In *Proceedings of the International AAAI Conference on Web and Social Media* (2017), vol. 11.

[29] Yadav, P., and Pandya, D. Sentireview: Sentiment analysis based on text and emoticons. In *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)* (2017), pp. 467–472.

Faculty of Mathematics and Computer Science, Babeş–Bolyai University, Cluj-Napoca, Romania

Email address: anna.bajcsi@stud.ubbcluj.ro
Email address: barbara.botos@stud.ubbcluj.ro
Email address: peter.bajko@stud.ubbcluj.ro
Email address: zbodo@cs.ubbcluj.ro