Emoji Prediction for Hebrew Political Domain

Chaya Liebeskind
Department of Computer Science, Jerusalem College of Technology, Lev Academic Center
Jerusalem, Israel
liebchaya@gmail.com

Shmuel Liebeskind
Department of Computer Science, Jerusalem College of Technology, Lev Academic Center
Jerusalem, Israel
israellieb@gmail.com

ABSTRACT

In this study, we aim to predict the most likely emoji given only a short text as an input. We extract a Hebrew political dataset of user comments for emoji prediction. Then, we investigate highly sparse n-grams representations as well as denser character n-grams representations for emoji classification. Since the comments in social media are usually short, we also investigate four dimension reduction methods, which associates similar words to similar vectorial representation. We demonstrate that the common Word Embedding dimension reduction method is not optimal. We also show that the character n-grams representations outperform all the other representation for the task of emoji prediction for Hebrew political domain.

CCS CONCEPTS

• **Information systems → Social tagging systems**; **Data analytics**; **Data mining**; **Content analysis and feature selection**; • **Computing methodologies → Machine learning approaches**; **Machine learning algorithms**.

KEYWORDS

Emoji prediction, Machine learning, Semantic analysis, Social media, Supervised learning.

ACM Reference Format:
Chaya Liebeskind and Shmuel Liebeskind. 2019. Emoji Prediction for Hebrew Political Domain. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3308560.3316548

1 INTRODUCTION

In recent years there has been increasing usage of emojis in social media. Emoji is defined as "a digital image that is added to a message in electronic communication in order to express a particular idea or feeling" (Cambridge dictionary). Any system that aims to address the task of modeling social media communication need to deal with the usage of emojis.

Following Novak et al. [36], Barbieri et al. [3] argued that efficient prediction of the most likely emoji given the short text of a message may help to improve different Natural Language Processing (NLP) tasks. These NLP tasks include both objective tasks, such as information retrieval, social media content generation and emoji suggestion, and subjective tasks like sentiment analysis and emotion recognition. They introduced the task of emoji prediction in Twitter and showed that their neural network model can outperform humans.

Inspired by the promising results of Barbieri et al. [3], Barbieri et al. [4] proposed the first shared task on multilingual emoji prediction. Previous findings about the idiosyncrasy of use of emojis across languages motivated Barbieri et al. [4] to focus on emoji prediction for two languages, English and Spanish. Additional researchers followed them and investigated the task for other languages including Italian [40], Japanese [42], Hindi, Bengali and Telugu [12, 13]. We are interested in emoji prediction for Hebrew. Hebrew is characterized by highly productive morphology and, as far as we know, has not been investigated before.

Since most of the Hebrew speakers live in Israel, we examined the statistics of social media usage in Israel (see Figure 1, generated by http://gs.statcounter.com/social-media-stats/all/israel). Facebook is leading with 77.45% of the users. Twitter is in the fourth place with only 3.99% of the users. Therefore, even though, most prior work used Twitter data, in order to gather a sufficient amount of data we used another social network, Facebook.

In this paper, we first describe emoji usage in our political Facebook dataset. Then, we explore the emoji prediction task. Given only a comment text, we aim to classify it to one of the twenty emojis that occur most frequently in the dataset. To perform this task, we adopt a supervised Machine Learning (ML) approach. We compare four semantic vector representations, i.e., Latent Semantic Analysis (LSA) [15, 25], Latent Dirichlet Allocation (LDA) [10, 22], Random projection (RP) [9] and Word Embedding [31, 32], to represent each comment as a vector of features. Each of the representations is generated by a different dimension reduction method, which associates similar words to similar vectorial representation. Moreover, all of the representations are built using entirely unsupervised distributional analysis of unlabeled text. We demonstrate that, in the case study of Hebrew Facebook, the Word Embedding dimensional reduction method achieves the lowest F1.

Additionally, we analyze two text representation approaches, i.e., n-grams and character n-grams. Both of these representations outperform all the semantic representations. We also show that the best character n-grams representation outperforms the FastText baseline [24], which is often on par with deep learning classifiers in terms of accuracy.

The rest of this paper is organized as follows: Section 2 introduces relevant background about emoji prediction. Section 3 presents the emoji prediction task, dataset, and the representation methods. Section 4 introduces the experimental setting, the experimental

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316548

results and a deep analysis of the best method. Finally, Section 5 summarizes the main findings and suggests future directions.

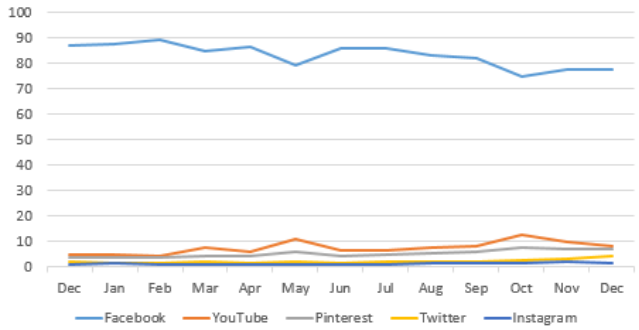

Figure 1: Social Media Stats Israel, Dec 2017 - Dec 2018

2 BACKGROUND

Recently, there is growing interest in distributional embeddings of emojis. Barbieri et al. [8] used the skip-gram neural embedding model [32] to train emoji embeddings from a large Twitter dataset of over 100 million English tweets. They evaluated these pre-trained emoji representations on two tasks: a pair similarity and relatedness, and clustering. They showed that the representations improve accuracy on both tasks. The same emoji representation methodology was adopted to compare the meaning and usage of emojis across two Spanish cities: Barcelona and Madrid [5], and across different languages [6].

A similar neural embedding model for emoji similarity was explored by Pohl et al. [38]. They claimed that search is a critical problem of emoji entry, and thus emoji keyboards need to be optimized for search. Being able to compute the level of similarity between two emojis allows to place related emojis close to each other and support users' search. They showed that the model obtains good performance in capturing detailed relationships between emojis.

Eisner et al. [16] argued that Barbieri et al. [8] method is unable to learn robust representations for infrequent emojis. They suggested estimating the representation of emojis directly from their description in the Unicode emoji standard. They found that their representation generally outperforms the emoji embeddings trained by Barbieri et al. [8] on the task of Twitter sentiment analysis.

To improve emoji embedding models, Wijeratne et al. [44] suggested incorporating more words by using longer emoji definitions. Using the information in EmojiNet [43], they considered three different ways to represent the meaning of an emoji, namely emoji descriptions, emoji sense labels, and the emoji sense definitions. They showed that their models outperform the previous best-performing emoji embedding model of Eisner et al. [16] on the sentiment analysis task. They also created a new publicly available dataset called EmoSim508 [43], which assigns human-annotated semantic similarity scores to a set of 508 carefully selected emoji pairs.

Currently, neural models have been used to model the semantics of emojis. Most of these models are based on neural networks with word embeddings representations which are generated by word2vec models [32] or gradient descent based learning algorithm.

Xie et al. [47] investigated the task of emoji recommendation in multi-turn dialogue systems. They proposed a hierarchical Long Short-Term Memory (LSTM) model to construct better dialogue representations by encoding the contextual information located in conversations. Their method significantly outperforms other LSTM models and a baseline bag-of-words model which takes logistic regression as the classifier and tf-idf as features.

Barbieri et al. [3] employed state-of-the-art Recurrent Neural Network (RNN) for emoji prediction. Their model is based on Bidirectional Long Short-term Memory Networks (BLSTMs) [18, 21] with two types of embedding representations: word embeddings and character-based continuous-space vector embedding [30]. They showed that the BLSTMs outperform a bag-of-words baseline, a baseline based on semantic vectors, and human annotators in this task.

Another variant of LSTM Networks [21, 41] was utilized by Felbo et al. [17] to predict sentiment, emotions and irony using an emoji prediction classifier as pre-trained system. Their pretrained model includes an attention mechanisms [48] to increase its sensitivity to individual words during prediction. Their model obtained state-of-the-art performance on 8 benchmark datasets. Based on Felbo et al. [17] architecture, Barbieri et al. [2] proposed a label-wise attention mechanism that is suitable for underrepresented emojis. They observed a performance improvement over Felbo et al. [17] model and over the efficient FastText [24] classifier.

By comparing emoji embeddings trained on a corpus of different seasons, Barbieri et al. [7] showed that some emojis are used differently depending on the time of the year . They proposed a method that considers temporal information for emoji prediction systems. The method extracts two types of embeddings: character BLSTM [30] and word embeddings. Then it concatenates the two representations (as in [3]) and passes them to a word LSTM and word attention units. They showed that their method outperforms state-of-the-art systems.

Guibon et al. [20] developed an automatic recommendation system based on user message analysis and real emoji usage. They showed that a multi-label RandomForest classifier with a bag-of-words/characters representation and computed features outperforms [3] BLSTMs.

Wu et al. [46] also addressed the multi-label setting of the emoji prediction task. They proposed a hierarchical neural model with attention mechanism. The model contains three modules, a character encoder to learn hidden representations of words using Convolutional Neural Networks (CNNs), a word encoder to learn sentence representations using a combination of CNN and LSTM, an emoji classifier to predict emojis for tweets. Their approach outperforms several baselines, including K independent bag-of-word Support Vector Machine (SVM) models, CNN as the word encoder, and a hierarchical model with LSTM in both character and word encoders, as well as humans in this task.

Recently, Barbieri et al. [4] introduced a new SemEval task, i.e., the multilingual emoji prediction task. The task was divided into two subtasks respectively dealing with the prediction of the emoji associated to English and Spanish tweets. For each subtask the

tweets that include one of the twenty emojis that occur most frequently in the Twitter data were selected. Therefore, the task can be viewed as a multi-label classification problem with twenty labels. The task required to predict the emoji by relying exclusively on the textual content of the text message. In total, 49 teams participated in the English subtask and 22 teams submitted a system run to the Spanish subtask. Even though many of the participating teams preferred neural architectures, mostly LSTMs and CNNs, the best performing system [14] on both English and Spanish datasets used a SVM classifier with bag-of-n-grams features (both characters and words). The task organizers suggested to extend the problem of modeling emoji semantics by incorporating more and more diverse languages.

One step in this direction was taken by Ronzano et al. [40]. They proposed the shared task also for the Italian language (ITAMoji) in the context of the Evalita 2018 evaluation campaign [11]. Twelve runs were submitted at ITAmoji by five teams. Generally, systems which employ neural network architecture obtained good performances in this task, especially when relying on BLSTM model.

The work of Tomihira et al. [42] was the first attempt to focus on Japanese. They collected Japanese tweets from Twitter and proposed a new model that learns from sentences. They investigated multiple models based on Encoder-Decoder model of RNN and CNN. In contrast to Zhao and Zeng [49] who showed that the CNN model have higher classification accuracy than the RNN model, they showed that in their case study the Encoder-Decoder model with attention was higher than the CNN model in accuracy and F1 score.

Another step in this direction was the study by Choudhary et al. [13] who created corpus for multilingual sentiment analysis and emoji prediction in Hindi, Bengali and Telugu. They addressed resource-poor languages because such discourse is available on resource-rich languages like English and Spanish, whereas resource-poor languages are largely ignored. Choudhary et al. [12] introduced a twin BLSTM RNNs model to learn emoji-based representations of resource-poor languages. They jointly train the resource-poor languages (Hindi and Telugu) with resource-rich languages (English and Spanish) in a common emoji space by using a similarity metric based on the emojis present in sentences from both languages. Their model outperforms the state-of-the-art emoji prediction approaches based on distributional semantics, semantic rules, lexicon lists and deep neural network representations without shared parameters. In this research, we also investigate emoji prediction in a resource-poor language, Hebrew.

3 EMOJI PREDICTION

3.1 Task

Emoji prediction is an anticipation task of predicting the emojis that appear in a given text message by relying exclusively on the textual content of that message.

Practically, we remove the emojis from the text of the messages and use them as labels. Therefore, the task can be viewed as a multi-label classification problem.

Following previous works on emoji prediction [3, 3, 40], we selected only messages with a single emoji, so that the challenge can be cast as a single-label classification problem, detailed examples from our Hebrew dataset are shown in Table 1[1].

Table 1: Examples from our Hebrew dataset

#	Comment	Label
1	איזו הפתעה יוזמה מבורכת aizw hpt`h iwzmh mbwrkt what a surprise, a blessed initiative	👏
2	שאפו כל הכבוד šapw. kl hkbwd well done! hats off to	👍
3	שבת מבורכת אשת חייל šbt mbwrkt ašt xiil a blessed Sabbath (Saturday) woman of valor	🌹
4	דייייי נקרע הלב diiii nqr` hlb enough, the heart is rending	💔
5	יהי זכרו ברוך ihi zkrw brwk of blessed memory	😢

3.2 Dataset

We have created a dataset for the task of emoji prediction using a political dataset by Liebeskind et al. [27–29]. All posts of Members of Knesset (MKs) between 2014-2016 (n=130 MKs, m=33,537 posts) have been downloaded via Facebook Graph API . The data included also the comments to these posts (n=5.37M comments posted by 702,396 commentators).

We have analyzed the emoji usage of the commentators. There are 786 types of emojis and 98,865 of the comments include at least one of them. There are 50,243 comments with a single emoji.

As mentioned in the previous section, only Facebook comments with a single emoji should be included in the task dataset. However, to increase the number of comments, we include comments with multiple appearances of the same emoji, such as חג שמח יא מלך 👑👑👑 (xg šmx ia mlk - happy holiday, yo king). After limiting our dataset to comments that contain one and only one **type** of emojis, our dataset contains 78,147 comments that include 593 emoji types.

Further analysis of the data reveals that there are 240 types of emojis which never appear alone. They either have multiple appearances or appear with additional types of emojis. The ten most frequent emojis that never appear alone with their frequencies and top-5 co-occurring emojis are presented in Table 2. In example #2 the emoji and all its co-occurring emojis are of the same *event* group[2]. In examples #6-#7, #9, and #10, the emoji's group (*geometric*, *transport-ground*, and *family*, respectively) is dominant in its top co-occurring emojis. In examples #3 and #4, two Japanese's buttons co-occur in all comments. In example #1, 🎤 belongs to the *music* group and its co-occurring emoji 🎹 belongs to the *musical-instrument*

[1]To facilitate readability, we used a transliteration of Hebrew using Roman characters; the letters used, in Hebrew lexico-graphic order, are *abgdhwzxTiklmns `pcqr št.*
[2]The group names are from the Full Emoji List, v11.0 of Unicode Emoji: https://unicode.org/emoji/

group. Since music is often played at wedding celebrations, it is also understandable that 🎤 and 💒 often co-occur in the same comment. In examples #5 and #8, it is difficult to interpret the connection between the emoji and its top co-occurring emojis, the connection can probably be explained by the comments' text. It is interesting to mention that in example #8 most of the top co-occurring emojis are from the *plant-flower* group (🌹(16), 🌷(14), 🌺(13), 🌸(13), 🌼(12), 🌻(12)).

Table 2: The 10 most frequent emojis that never appear alone with their frequencies and top-5 co-occurring emojis

#	Emoji	Frequency	Co-occurring emojis
1	🎤	40	😇(25) 💒(25) 👭(24) 🎹(24) 🍇(24)
2	🌄	37	🌆(23) 🎉(15) 🎈(9) 🎊(8) 🎁(6)
3	🈲	36	😇(35) 💒(35) 👭(34) 🉑(33) 🌹(30)
4	🉑	35	😇(35) 💒(35) 👭(34) 🈲(33) 🌹(29)
5	🌍	30	🏏(29) 🙌(28) 💩(28) 🍼(28) 👑(28)
6	⚪	27	🔵(17) ❤️(6) ⚫(3) 🔴(2) 💚(2)
7	🔵	27	⚪(17) ❤️(4) 🔴(3) 🔯(2) 👍(2)
8	💫	25	🌹(16) 😇(14) 🌷(14) ❤️(14) 🌺(13)
9	🚲	21	🚓(21) 🚑(21) 🚂(20) 🙌(20) 🌍(20)
10	👭	20	👭(13) 👭(8) 👰(7) 👴(4) 👍(4)

Emojis are used by 41,789 of the commentators. In this analysis that is focused on commentators we take under consideration only the top-100 commentators with high usage of emojis (above 50 comments) in our dataset. In Figure 2, we compare the average number of words in comments with emojis (the blue line) with the average number of words in comments without emojis (the green line). The x-axis' values denote the commentators, while the y-axis' values denote the average number of comments. Most of the commentators use a similar number of words with and without emojis. This implies that emojis are not used as a substitute for words. The second comparison in Figure 2 between the average number of words in comments with emojis (the blue line) and the average number of emojis (the orange line) also shows that more emojis do not mean less words.

Figure 3 shows the diversity of emojis in the comments. The x-axis' values denote the commentators, while the y-axis' values denote the number of comments (blue bars) or the number of emoji types (orange bars). The figure illustrates that the diversity of the emojis depends on user preferences rather than on the number of comments.

For completeness, we ranked the list of all the 702,396 commentators of our dataset by the number of their comments. The distribution of the top-100 commentators with high usage of emojis over this ranked list is presented in Figure 4. 6% of the emoji users are part of the top-100 heavy users of our dataset. Most of the emoji users are ranked between 1,000 to 10,000 (60%). Non of them is ranked below 15,583, all the top-100 commentators with high usage of emojis are part of the top-15,583 heavy users of our dataset.

For the task of emoji prediction, we selected the comments that include one of the twenty emojis that occur most frequently in the dataset we have described (see Table 3). After removing fake comments with more than 15 emojis, our dataset contains 45,410 comments.

Table 3: The 20 most frequent emojis with their comment frequency, total frequency, and average comment length

#	Emoji	Comment Freq	Total Freq	Avg Comment Len
1	👍	12,537	25,486	9.21
2	❤️	5,767	10,234	15.37
3	👏	4,206	14,572	7.73
4	🙏	3,728	6,359	16.62
5	😂	3,108	9,553	10.77
6	😊	2,624	3,216	15.74
7	❤️	1,880	3,639	14.98
8	🌹	1,787	3,254	11.93
9	😉	1,618	1,835	17.94
10	😍	1,453	2,662	8.48
11	👎	1,333	3,581	13.5
12	😉	1,324	2,016	15.12
13	👑	1,280	2,949	11.05
14	😊	1,222	1,532	20.2
15	💙	1,195	2,420	9.88
16	😀	1,189	1,938	14.12
17	😄	1,146	1,799	29.73
18	😘	1,114	1,886	11.89
19	✌️	995	1,979	11.43
20	😢	961	1,606	16.15

3.3 Method

In this research, we adopt a supervised Machine Learning (ML) approach for emoji prediction. The first step in training a classifier is deciding what features of the text are relevant, and how to encode these features. First, We investigate two types of text representations:

(1) N-grams representation - An n-gram is a contiguous sequence of n words. Each of the n-grams in the comment is considered as a feature. The score of the feature is the n-gram tf-idf
The n-grams representation is a high-dimensional sparse representation for documents of any length. However, the sparsity problem is much more critical for short texts, such as comments, where most words have only one occurrence.

(2) Character n-grams representation - Character n-grams are strings of length n. For example, the character 3-grams of the string "prediction" would be: "pre", "red", "edi", "dic", "ict", "cti", "tio" and "ion". Each of the character n-grams of the comment is considered as a feature and scored by its tf-idf. Since there is much less character combinations than n-gram combinations, character n-grams representation overcomes the problem of sparse data that arises when using n-grams

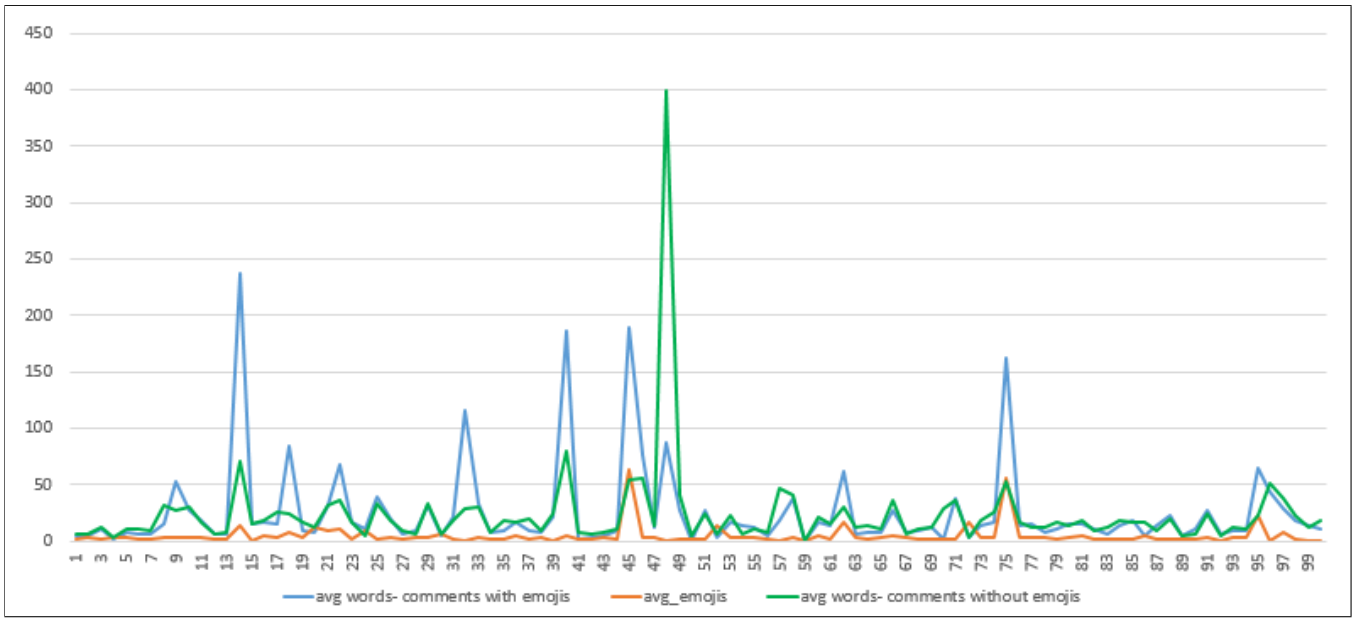

Figure 2: A comparison between the average number of words in comments with and without emojis

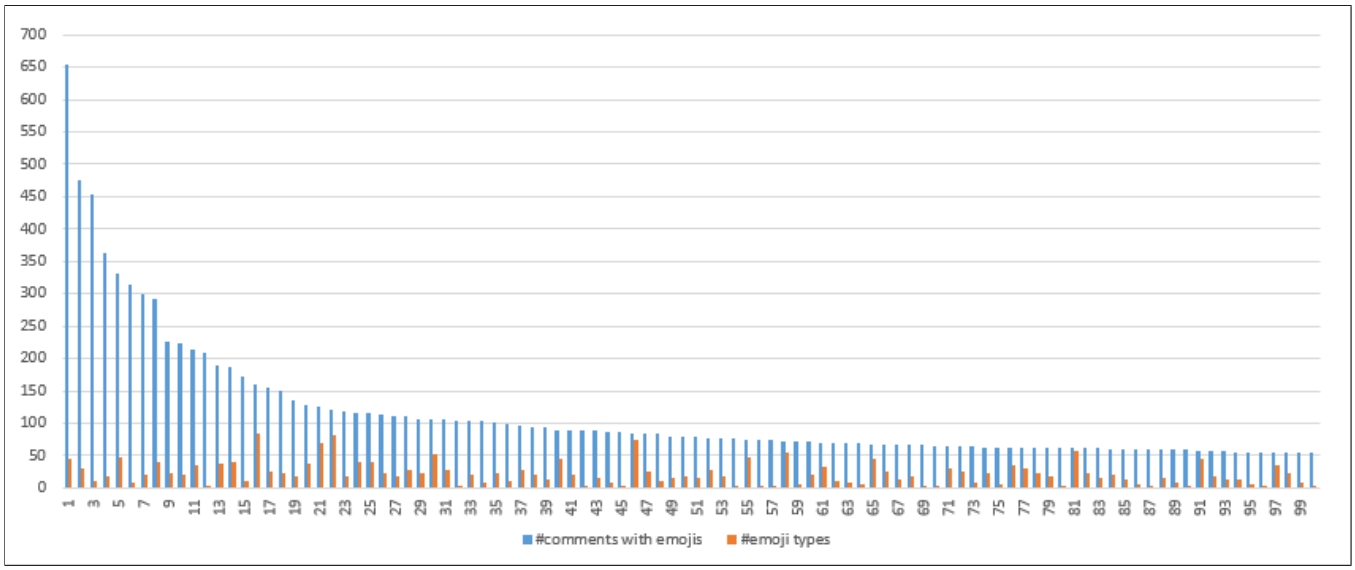

Figure 3: The diversity of emojis in the comments

representation. However, it still produces a considerably larger feature set.

Due to the tendency of noise and misspellings to have smaller impact on substring patterns than on n-gram patterns, character n-gram features can be quite effective for short informal text classification [1, 39].

We did not lemmatize the comments because previous work on Hebrew short text classification [29, 35] has indicated that lemmatization (using a Part-of-speech tagger) do not improve the performance.

Another approach to overcome the sparsity problem of the n-grams representation is to apply dimensional reduction methods for semantic analysis.

Next, we examine four semantic vector representations. Each of the representations is generated by a different dimension reduction method, which produces condensed vectorial representation of words in which similar words are described by similar vectors (with respect for instance to their cosine similarity). All of the representations are built using entirely unsupervised distributional analysis of large amount of unlabeled text.

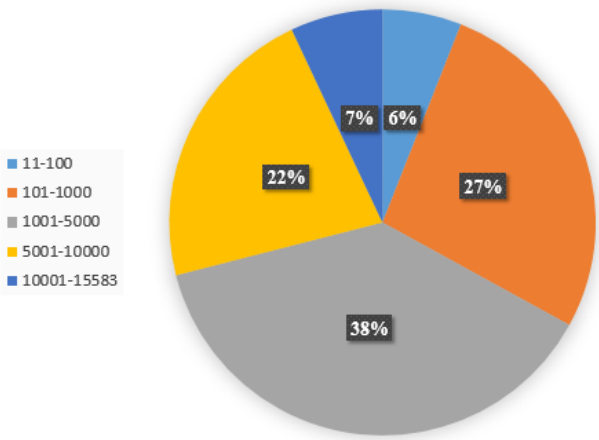

Figure 4: The distribution of the top-100 commentators with high usage of emojis over the ranked list all the dataset commentators

(1) **Latent Semantic Analysis (LSA)**[15, 25] constructs a semantic space from a large matrix of term-document association data by Singular Value Decomposition (SVD). SVD is a linear algebra procedure of decomposing an arbitrary matrix into three matrices, two of which are orthonormal and the third is a diagonal matrix whose diagonal values are the singular values of the matrix.

(2) **Latent Dirichlet Allocation (LDA)**[10, 22] is a generative statistical model for detecting latent semantic topics in large corpora. Documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over all the words. The topic probabilities provide an explicit dense representation of a document. The main challenge of building an LDA model is how to estimate the distribution information of latent topics within the document. Different algorithms, such as Expectation Maximization (EM)[34] and Gibbs sampling [19] are used to face this challenge.

(3) **Random Projection (RP)**[9] projects the original high-dimensional data onto a lower-dimensional subspace using a random matrix whose columns have unit lengths. The main idea behind random projection is given in the Johnson-Lindenstrauss lemma [23]: if points in a vector space are projected onto a randomly selected subspace of suitably high dimension, then the distances between the points are approximately preserved.

(4) **Word Embedding** [31, 32] approaches reduce the high dimensionality of words using neural probabilistic language models. A d-dimensional vector of real numbers models the contexts of each word. The vectors are meaningless on their own, but semantically similar words have similar vectors. Inspired by the methods for learning word vectors using neural networks, documents are also mapped to vectors (doc2vec) [26]. Given many contexts sampled from the document, the document vectors predict the next word in the given context. We use doc2vec to generate the word embedding based vectorial representation of the Facebook comments.

As far as we know, only the last semantic vector representation, Word Embedding, has been investigated for the task of emoji prediction, as detailed in Section 2.

4 EVALUATION

4.1 Evaluation Setting

While the supervised classification was performed on the comments that include one of the twenty emojis that occur most frequently in the Facebook data we collected (see Section 3.2), the unsupervised learning of dense vector representations was performed on the large corpus of 5.37M comments.

Table 4: The results of eight ML methods for all the four semantic vector representations

Rep.	Alg.	Precision	Recall	F1	Accuracy
RP	NB	0.1522	0.162	0.144	0.2045
	DT	0.1212	0.1228	0.1219	0.1851
	LR	0.233	0.1274	0.1302	**0.3065**
	RF	0.1514	0.1256	0.1277	0.2299
	MLP	0.1724	0.1652	**0.166**	0.2641
	SVC	0.443	0.0556	0.0298	0.2341
	AdaBoost	0.1186	0.0812	0.0655	0.2584
	Bagging	0.1559	0.1291	0.1316	0.2344
LDA	NB	0.1383	0.1016	0.0849	0.2536
	DT	0.1355	0.1313	0.1323	0.2174
	LR	0.1606	0.0919	0.077	0.2833
	RF	0.1618	0.1364	0.1401	0.2487
	MLP	0.2016	0.1308	0.1293	**0.2934**
	SVC	0.0114	0.05	0.0186	0.2291
	AdaBoost	0.1442	0.0901	0.0766	0.272
	Bagging	0.1691	0.1377	**0.1421**	0.2569
LSA	NB	0.156	0.1758	0.1464	0.1968
	DT	0.1472	0.1484	0.1477	0.2139
	LR	0.2255	0.1113	0.1139	**0.2712**
	RF	0.1726	0.1538	0.1559	0.2516
	MLP	0.1755	0.1449	0.1483	0.2622
	SVC	0.3429	0.0538	0.0261	0.2328
	AdaBoost	0.1629	0.1047	0.0967	0.2668
	Bagging	0.1767	0.1573	**0.1596**	0.2566
doc2vec	NB	0.1119	0.113	0.0911	0.1447
	DT	0.0865	0.0862	0.0863	0.1585
	LR	0.2522	0.0926	0.0797	0.3029
	RF	0.1114	0.1007	0.0971	0.2288
	MLP	0.1699	0.115	**0.1107**	**0.3061**
	SVC	0.1297	0.0707	0.0457	0.2784
	AdaBoost	0.1932	0.087	0.067	0.2901
	Bagging	0.1132	0.1016	0.0986	0.2305

Table 5: LogisticRegression and RandomForest results for the N-grams and Character n-grams representations

Representation	Alg.	Precision	Recall	F1	Accuracy
Char 2-grams	LR	0.3622	0.1664	0.1752	0.3718
	RF	0.2121	0.1713	0.1771	0.3271
Char 3-grams	LR	0.4365	0.1984	**0.2176**	**0.3847**
	RF	0.249	0.2072	0.2128	0.3412
Char 4-grams	LR	0.4588	0.191	0.2109	0.3792
	RF	0.2558	0.2069	0.215	0.3407
Char 5-grams	LR	0.4492	0.1855	0.204	0.3752
	RF	0.2482	0.2025	0.2112	0.336
Unigrams	LR	0.4284	0.184	0.2022	0.3712
	RF	0.2398	0.1943	0.2036	0.3279
Bigrams	LR	0.4335	0.1359	0.1509	0.3111
	RF	0.2106	0.1508	0.1645	0.2759
Trigrams	LR	0.3963	0.1047	0.1114	0.282
	RF	0.2484	0.113	0.1233	0.269

For classification, Scikit-learn[3] machine learning python module [37] was used. We used 5-fold cross-validation to estimate the classification performance, which is a resampling procedure of partitioning the dataset into a training set to train the model, and a test set to evaluate it. For dimensionality reduction, GenSim[4] python library with the default settings of 300 dimensions for the LSA, RP ,and Word Embedding (doc2vec) and 100 dimensions for LSA was used (due to computational limitations).

In our experiments, we compared the performance of our algorithms by four commonly used classification measures: precision, recall, F1, and accuracy. The scores are *macro-averaged*; we first calculate the measure for each label/emoji and then take the average of these scores.

4.2 Results

In our experiments, we combined the features in a supervised classification framework using eight ML methods: Bernulli Naive Bayes (NB), Decision Tree (DT), Logistic Regression (LR), Random Forest (RF), Multilayered Perceptron (MLP), Support Vector Classification (SVC), Adaboost, and Bagging. Table 4 presents the results of all the dimensional reduction methods, extracted by the unlabeled data. When using the RP representation, MLP and Logistic Regression achieve the best F1 and accuracy, respectively. When using the LDA representation, Bagging and MLP achieve the best F1 and accuracy, respectively. When using the LSA representation, Bagging and Logistic Regression achieve the best F1 and accuracy, respectively. When using the doc2vec representation, the best F1 and accuracy are achieved by the same method, the MLP.

As detailed in Section 2, in previous works, the Word Embedding approach was exclusively selected as a dimensional reduction method. However, our results show that all the other reduction methods that we have suggested achieve a higher F1 score. The F1 advantage of the LDA representation, which is lower than the F1 of the RP and LDA representations, is statistically significant[5] at the 0.05 level.

The text representations generated a large feature set. Therefore, we filtered out features that have less than 8 appearances or appear in more than 5% of the comments in our dataset. We empirically selected two classifiers; LR and RF (out of the eight) that achieve good performance with reasonable run time.

Table 5 presents the results of the two types of text representation: N-grams representations (unigram, bigram, and trigram) and Character n-grams (character 2-grams, 3-grams, 4-grams, and 5-grams).

Generally, the LR classifier outperforms the RF classifier with all the text representation. The unigram representation and the

Table 6: The precision, recall, and F1 measure of the character 3-grams representation for each emoji

#	Emoji	Precision	Recall	F1
1	👍	0.33	0.84	0.47
2	❤️	0.62	0.69	0.65
3	🙏	0.45	0.44	0.44
4	👏	0.3	0.1	0.15
5	😂	0.36	0.39	0.38
6	😊	0.18	0.06	0.09
7	❤️	0.67	0.11	0.2
8	🌹	0.26	0.13	0.17
9	😉	0.16	0.04	0.06
10	😍	0.28	0.02	0.03
11	😜	0.17	0.02	0.04
12	😊	0.81	0.06	0.11
13	👑	0.58	0.34	0.43
14	😀	0.33	0.01	0.01
15	😄	0.95	0.11	0.2
16	😘	0.37	0.01	0.02
17	💙	0.53	0.24	0.33
18	👎	0.38	0.09	0.14
19	✌️	0.35	0.06	0.1
20	😢	0.64	0.22	0.33

[3]https://scikit-learn.org/stable/index.html
[4]https://radimrehurek.com/gensim/index.html
[5]In all the reported experiments, statistical significant was measured according to the two-sided Wilcoxon signed-rank test [45]

Table 7: Confusion matrix of the character 3-grams representation

	😘	😀	😜	😉	❤️	😢	🌹	😊	😄	👍	👑	👏	👎	😊	🙏	😂	✌️	💙	😍	❤️
😘	10	0	1	9	3	0	39	4	0	694	17	23	5	24	48	61	3	3	9	101
😀	0	6	7	23	4	2	46	0	0	762	4	29	7	35	41	127	3	5	0	57
😜	0	2	28	25	1	3	7	0	0	848	5	28	16	26	24	263	0	3	3	17
😉	1	0	21	65	0	1	7	0	0	1131	3	28	8	47	35	218	1	2	2	36
❤️	0	0	1	9	201	4	54	1	1	870	16	26	5	20	197	36	7	19	3	281
😢	0	0	7	18	1	186	3	0	0	390	0	7	7	8	108	84	1	0	1	28
🌹	2	3	3	5	6	17	223	2	0	1021	15	28	1	76	176	14	5	8	1	95
😊	0	0	8	13	7	0	39	73	0	637	5	13	2	50	55	62	5	5	3	211
😄	0	1	3	21	3	2	21	0	124	649	4	18	7	21	34	158	1	8	1	40
👍	1	3	13	43	12	8	104	0	0	8742	92	278	20	134	355	254	25	69	11	242
👑	0	0	3	4	1	0	21	0	0	587	399	31	3	17	38	21	0	8	1	41
👏	3	1	9	11	1	3	23	2	0	2515	41	320	8	26	89	103	3	19	1	93
👎	0	0	12	19	0	5	0	0	0	623	1	16	86	17	14	171	0	0	0	14
😊	0	2	7	45	6	3	107	3	1	1685	6	42	7	144	163	141	8	9	2	187
🙏	1	0	5	17	14	32	68	2	1	1438	11	34	12	64	1541	38	11	15	6	222
😂	2	0	25	49	0	12	3	0	2	1440	2	48	21	18	25	1111	4	5	2	45
✌️	1	0	2	5	5	0	4	1	0	426	3	7	4	2	16	28	59	1	1	392
💙	0	0	0	0	11	1	11	0	0	441	6	12	0	8	130	15	0	242	2	145
😍	1	0	2	7	4	2	31	1	0	844	24	44	2	29	64	79	4	16	22	137
❤️	5	0	6	16	19	10	55	1	1	1140	31	43	5	51	262	63	27	18	8	3889

Table 8: Error analysis of five emojis' comments with the percentage of misclassified comments that seem to fit the comments content in 30 randomly selected examples

#	Emoji	% in 30	Example #1	Example #2	Example #3
1	👑	100	את מלכה!!! at mlkh!!! you are a queen	מלדדדדדדדדך אוהבים אותך mlkkkkkkkkkk awhbim awtk king, we like you	יצאת נסיך icta nsik you came out a prince
2	🌹	96	מזל טוב mzl Twb congratulations!	שנה טובה šnh Twbh happy new year!	בוקר טוב וחג שמח bwqr Twb wxg šmx good morning and happy holiday
3	😢	83	יהי זכרם ברוך ihi zkrm brwk of blessed memory	עצוב מאוד `cwb mawd very sad	משתתפת בצערך mšttpt bc`rk to express one's condolence
4	🙏	73	החלמה מלאה hxlmh mlah full recovery	רפואה שלמה ובשורות טובות rpwah šlmh wbšwrwt Twbwt get well soon! may we hear only good news	יהי זכרם ברוך ihi zkrm brwk of blessed memory
5	👏	70	כל הכבוד מגיע לך kl hkbwd mgi` lk well done! you deserve it	אתה אלוף העולם ath alwp h`lm you are the world champion	בראבו brabw bravo

character n-grams representations are better than all the semantic vector representations, presented in Table 4.

The unigram representation outperforms all the other n-grams representations significantly. However, the best representations are the character 3-grams and character 4-grams. The F1 advantage of the character 3-grams representation over the unigram representation is statistically significant at the 0.01 level. This finding is interesting since the Hebrew root, which is the most basic form

of the word, to which other parts, such as affixes, can be added, contains three letters.

We also compared our results to a common FastText baseline [24], which is often on par with deep learning classifiers in terms of accuracy. Hyperparameters were set as default (as in Barbieri et al. [4]). We obtained an accuracy of 36.28, and precision, recall and a F1 scores of 0.2113 0.1504 0.1489, respectively. We observed that the character 3-grams representation outperforms the FastText baseline and its F1 advantage is statistically significant at the 0.01 level.

We note that in the SemEval shared task on multilingual emoji prediction the best performing system [14] on both English and Spanish datasets was based on text representations.

In Table 6, we show the precision, recall, and F1 measure of the character 3-grams representation for each emoji, sorted by the emoji frequency. Except for the 👏 emoji, the frequent emojis have higher F1 scores. Additional emojies with good performances are: 👑, 💙, and 😢. In general, the precision is higher than the recall. The classifier learned only part of the emojis features. Its default choice was 👍

4.3 Error Analysis

To better understand the challenges of the emoji prediction task, we analyzed the classification errors of the character 3-grams representation. First, in Table 7, we present the classification confusion matrix. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class.

Most of the classification errors were due to incorrect classification of comments into the most frequent emoji 👍. On average 55% of the wrong classification per emoji were classified as 👍. In the case of the emoji 👏 77% of the comments that contain it were classified as 👍. According to the Emojipedia [6] 👍 indicates approval and 👏 can be used as a round of applause, when used multiple times. Therefore, they might express a similar meaning.

12% of the comments with 😢 were wrongly classified as 🙏. It might mean being sad and praying for a quick recovery from illness. The 9% of the comments with 😢 which were wrongly classified as 😂 are probably a misinterpretation of the emoji as crying out of sadness instead of joy.

14% of the comments with 💙 and 16% of the comments with ❤️ were wrongly classified as ❤️. Due to the high frequency of the emoji ❤️, classifications error in the opposite direction were not frequent.

Next, in Table 8, we present five emojis that we have analyzed their misclassified comments. Since the emoji prediction for specific emojis could be subjective, we observed that in many of the cases both the classifier decision and the emoji that was chosen by the commentator seem to fit the comment content (the percentage of such cases in 30 randomly selected examples are given in the table). This can be explained by the finding of recent studies that people interpret emoji characters inconsistently [3, 33].

We note that in examples #3 and #5 there is a common expression יהי זכרם ברוך -ihi zkrm brwk (of blessed memory). This expression can express sadness (😢) or a prayer for the dead (🙏). In this paper, we simplified the prediction setting to single-label classification. However, in real life, it is a multi-label prediction task.

[6]https://emojipedia.org

5 CONCLUSIONS AND FUTURE WORK

We explored the task of emoji prediction as a single-label classification problem. We investigated four dimensional reduction methods which learn the distributional representations unsupervisingly using large unlabeled data. By applying 8 ML methods for emoji prediction, we compared the effectiveness of different semantic vector representations. The Word Embedding common representation for emoji prediction achieved the lowest F1 score. The ML algorithm that should be chosen depends on what you want to optimize; F1 or accuracy.

We showed that both of the text representations, i.e., n-grams and character n-grams, outperform all the semantic vector representations significantly. The best representation character 3-grams achieve an accuracy of 38.47% and a F1 score of 0.2176. It also significantly outperforms the efficient FastText algorithm.

In practice, more than one emoji may appear in a given text message. Thus, we plan to address the multi-label setting of the emoji prediction task. In addition, we plan to investigate the deep learning approach for our classification task as it has been shown to be effective [3, 12, 17, 42, 46]. Word Embedding is often the selected representation for neural network models. However, since it did not perform well in our setting, we would probably need to apply more sophisticated models than the deep learning baselines.

REFERENCES

[1] Fotis Aisopos, George Papadakis, Konstantinos Tserpes, and Theodora Varvarigou. 2012. Content vs. context for sentiment analysis: a comparative analysis over microblogs. In *Proceedings of the 23rd ACM conference on Hypertext and social media*. ACM, 187–196.

[2] Francesco Barbieri, Luis Espinosa Anke, Jose Camacho-Collados, Steven Schockaert, and Horacio Saggion. 2018. Interpretable emoji prediction via label-wise attention LSTMs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4766–4771.

[3] Francesco Barbieri, Miguel Ballesteros, and Horacio Saggion. 2017. Are Emojis Predictable?. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Vol. 2. 105–111.

[4] Francesco Barbieri, Jose Camacho-Collados, Francesco Ronzano, Luis Espinosa Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, and Horacio Saggion. 2018. SemEval 2018 Task 2: Multilingual Emoji Prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*. 24–33.

[5] Francesco Barbieri, Luis Espinosa-Anke, and Horacio Saggion. 2016. Revealing patterns of Twitter emoji usage in Barcelona and Madrid. *Frontiers in Artificial Intelligence and Applications. 2016;(Artificial Intelligence Research and Development) 288: 239-44.* (2016).

[6] Francesco Barbieri, German Kruszewski, Francesco Ronzano, and Horacio Saggion. 2016. How cosmopolitan are emojis?: Exploring emojis usage and meaning over different languages with distributional semantics. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 531–535.

[7] Francesco Barbieri, Luis Marujo, Pradeep Karuturi, William Brendel, and Horacio Saggion. 2018. Exploring Emoji Usage and Prediction Through a Temporal Variation Lens. *arXiv preprint arXiv:1805.00731* (2018).

[8] Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2016. What does this Emoji Mean? A Vector Space Skip-Gram Model for Twitter Emojis.. In *LREC*.

[9] Ella Bingham and Heikki Mannila. 2001. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 245–250.

[10] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[11] Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso. 2018. EVALITA 2018: Overview of the 6th evaluation campaign of natural language processing and speech tools for Italian. In *Proceedings of Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018),*

Viviana Patti Tommaso Caselli, Nicole Novielli and Paolo Rosso (Eds.), Vol. 2263. 1–6.
[12] Nurendra Choudhary, Rajat Singh, Ishita Bindlish, and Manish Shrivastava. 2018. Contrastive Learning of Emoji-based Representations for Resource-Poor Languages. *arXiv preprint arXiv:1804.01855* (2018).
[13] Nurendra Choudhary, Rajat Singh, Vijjini Anvesh Rao, and Manish Shrivastava. 2018. Twitter corpus of Resource-Scarce Languages for Sentiment Analysis and Multilingual Emoji Prediction. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1570–1577.
[14] Çağrı Çöltekin and Taraka Rama. 2018. Tübingen-Oslo at SemEval-2018 Task 2: SVMs perform better than RNNs in Emoji Prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*. 34–38.
[15] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41, 6 (1990), 391–407.
[16] Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bosnjak, and Sebastian Riedel. 2016. emoji2vec: Learning Emoji Representations from their Description. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*. 48–54.
[17] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1615–1625.
[18] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM networks. In *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint conference on*, Vol. 4. IEEE, 2047–2052.
[19] Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences* 101, suppl 1 (2004), 5228–5235.
[20] Gaël Guibon, Magalie Ochs, and Patrice Bellot. 2018. Emoji recommendation in private instant messages. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. ACM, 1821–1823.
[21] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[22] Xia Hu, Nan Sun, Chao Zhang, and Tat-Seng Chua. 2009. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 919–928.
[23] William B Johnson and Joram Lindenstrauss. 1984. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics* 26, 189-206 (1984), 1.
[24] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Vol. 2. 427–431.
[25] Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse processes* 25, 2-3 (1998), 259–284.
[26] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. 1188–1196.
[27] Chaya Liebeskind, Shmuel Liebeskind, and Yaakov HaCohen-Kerner. 2017. Comment relevance classification in Facebook. In *18th International Conference on Computational Linguistics and Intelligent Text Processing*. Budapest, Hungary.
[28] Chaya Liebeskind and Karine Nahon. 2018. Challenges in Applying Machine Learning Methods: Studying Political Interactions on Social Networks. In *Semantic Keyword-Based Search on Structured Data Sources*, Julian SzymaÅ„ski and Yannis Velegrakis (Eds.). Springer International Publishing, Cham, 136–141.
[29] Chaya Liebeskind, Karine Nahon, Yaakov HaCohen-Kerner, and Yotam Manor. 2017. Comparing Sentiment Analysis Models to Classify Attitudes of Political Comments on Facebook (November 2016). *Polibits* 55 (2017), 17–23.
[30] Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1520–1530.
[31] Chenglong Ma, Weiqun Xu, Peijia Li, and Yonghong Yan. 2015. Distributional representations of words for short text classification. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. 33–38.
[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[33] Hannah Miller, Daniel Kluver, Jacob Thebault-Spieker, Loren Terveen, and Brent Hecht. 2017. Understanding emoji ambiguity in context: The role of text in emoji-related miscommunication. In *Eleventh International AAAI Conference on Web and Social Media*.
[34] Thomas Minka and John Lafferty. 2002. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 352–359.
[35] Dror Mughaz, Tzeviya Fuchs, and Dan Bouhnik. 2018. Automatic Opinion Extraction from Short Hebrew Texts using Machine Learning Techniques. *Computación y Sistemas* 22, 4 (2018).
[36] Petra Kralj Novak, Jasmina Smailović, Borut Sluban, and Igor Mozetič. 2015. Sentiment of emojis. *PloS one* 10, 12 (2015), e0144296.
[37] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.
[38] Henning Pohl, Christian Domin, and Michael Rohs. 2017. Beyond Just Text: Semantic Emoji Similarity Modeling to Support Expressive Communication. *ACM Transactions on Computer-Human Interaction (TOCHI)* 24, 1 (2017), 6.
[39] Stephan Raaijmakers and Wessel Kraaij. 2008. A shallow approach to subjectivity classification. In *ICWSM*.
[40] Francesco Ronzano, Francesco Barbieri, Endang Wahyu Pamungkas, Viviana Patti, Francesca Chiusaroli, et al. 2018. Overview of the EVALITA 2018 Italian Emoji Prediction (ITAMoji) Task. In *6th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop, EVALITA 2018*, Vol. 2263. CEUR-WS, 1–9.
[41] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.
[42] Toshiki Tomihira, Atsushi Otsuka, Akihiro Yamashita, and Tetsuji Satoh. 2018. What Does Your Tweet Emotion Mean?: Neural Emoji Prediction for Sentiment Analysis. In *Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services*. ACM, 289–296.
[43] Sanjaya Wijeratne, Lakshika Balasuriya, Amit Sheth, and Derek Doran. 2016. Emojinet: Building a machine readable sense inventory for emoji. In *International Conference on Social Informatics*. Springer, 527–541.
[44] Sanjaya Wijeratne, Lakshika Balasuriya, Amit Sheth, and Derek Doran. 2017. A semantics-based measure of emoji similarity. *arXiv preprint arXiv:1707.04653* (2017).
[45] Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics bulletin* 1, 6 (1945), 80–83.
[46] Chuhan Wu, Fangzhao Wu, Sixing Wu, Yongfeng Huang, and Xing Xie. 2018. Tweet Emoji Prediction Using Hierarchical Model with Attention. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM, 1337–1344.
[47] Ruobing Xie, Zhiyuan Liu, Rui Yan, and Maosong Sun. 2016. Neural Emoji Recommendation in Dialogue Systems. *arXiv preprint arXiv:1612.04609* (2016).
[48] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1480–1489.
[49] Luda Zhao and Connie Zeng. [n. d.]. Using Neural Networks to Predict Emoji Usage from Twitter Data.