Towards the Understanding of Gaming Audiences by Modeling Twitch Emotes

Francesco Barbieri[◇] **Luis Espinosa-Anke**[◇] **Miguel Ballesteros**[♠]
Juan Soler-Company [◇] **Horacio Saggion**[◇]
[◇] Large Scale Text Understanding Systems Lab, TALN Group
Universitat Pompeu Fabra, Barcelona, Spain
[♠]IBM T.J Watson Research Center, U.S
{name.surname}@upf.edu, miguel.ballesteros@ibm.com

Abstract

Videogame streaming platforms have become a paramount example of noisy user-generated text. These are websites where gaming is broadcasted, and allows interaction with viewers via integrated chatrooms. Probably the best known platform of this kind is Twitch, which has more than 100 million monthly viewers. Despite these numbers, and unlike other platforms featuring short messages (e.g. Twitter), Twitch has not received much attention from the Natural Language Processing community. In this paper we aim at bridging this gap by proposing two important tasks specific to the Twitch platform, namely (1) *Emote* prediction; and (2) Trolling detection. In our experiments, we evaluate three models: a BOW baseline, a logistic supervised classifiers based on word embeddings, and a bidirectional long short-term memory recurrent neural network (LSTM). Our results show that the LSTM model outperforms the other two models, where explicit features with proven effectiveness for similar tasks were encoded.

1 Introduction

Understanding the language of social media is a mature research area in Natural Language Processing (NLP) and Artificial Intelligence. Not only for the challenges it poses from a linguistic perspective, but also for being a task with a direct impact in relevant sectors like politics, stock market or health (Small, 2011; Bollen et al., 2011; Culotta, 2010). The notion of *understanding* in social media contexts may be divided in more specific AI tasks, including, among others, Sentiment Analysis (Pang and Lee, 2008), Irony Detection (Reyes et al., 2013b), or Event Summarization via Twitter Streams (Chakrabarti and Punera, 2011), as well as other subtasks such as Event (Weng and Lee, 2011) or Stance Detection (Mohammad et al., 2016) in Twitter.

While the study of language in social media typically involves blog posts, comments or product reviews, one of the most interesting areas of research concerns those highly restrictive platforms, e.g. enforcing character limits in each message. One of these platforms, Twitter, has attracted much attention due to its large user base as well as the linguistic idiosyncrasies of its language. It is interesting, therefore, to focus on another growing platform (in number of users) which shares some of the features that made Twitter popular in NLP. This platform is TWITCH.TV (henceforth, Twitch), the largest videogame video streaming service, currently a subsidiary of Amazon. Inc.

Twitch is used by a large community of individual *gamers* to broadcast themselves playing a game (Smith et al., 2013), but also by companies to broadcast live videogame and *electronic sports* (competitive video gaming) events, as well as releasing footage of new products, such as consoles or games. An outstanding feature of Twitch broadcasts is that they run alongside a permanent chat platform. Properly analyzing the content of Twitch chat messages can be useful for understanding the opinion of the community towards any industry product or stakeholder, in addition to its industrial relevance (Kaytoue et al., 2012). Moreover, analyzing this platform is fundamental for informing a number of AI-related applications such as behaviour prediction or Information Retrieval.

Interpreting Twitch language, however, is a challenging problem, as it features a vast amount of *Internet memes*, slang and gaming-related

Proceedings of the 3rd Workshop on Noisy User-generated Text, pages 11–20

lingo. In addition, Twitch language is characterized by combining short text messages with small pictures known as *emotes*. These emotes generally serve a different communicative purpose than most visual aids (e.g. Twitter *emojis*), and therefore require specific modeling.

In this paper, we put forward an approach for the understanding of Twitch messages by means of modeling the underlying semantics of Twitch emotes, and a dataset of Twitch chat messages. Building up on previous research on predicting paralinguistic elements (e.g. emojis) (Barbieri et al., 2017), we target the *Emote Prediction* problem, i.e. the task of, given a collection of chatroom messages, predicting which *emote* the user is more likely to use. Second, *Trolling Detection*, which we reformulate as the task to detect a specific set of emotes which are broadly used by Twitch users in troll messages. For both tasks, we evaluate models which consider sequences of words (bidirectional recurrent neural networks (Graves and Schmidhuber, 2005)), and compare against order-agnostic baselines which have proven to be highly competitive in similar tasks.

2 Twitch Language

An essential feature in a Twitch live broadcast is the chatroom alongside the gameplay. This component enables interaction among viewers and between viewers and streamers. This interaction is in general expressed via short messages, although in larger channels with higher activity, the majority of users may only use *emotes* in their messages for conveying emotions (Olejniczak, 2015). While not entirely arbitrary, the language and the content of conversations are remarkably diverse. In a very short time span, users may comment on the game that is being played, make an out-of-context joke, or discuss an unrelated event like a football game.

2.1 Twitch Emotes

Twitch messages can be enhanced with Twitch *emotes*, "small pictorial glyphs that fans pepper into text"[1]. These emotes range from the more regular *smiley* faces, to others such as game-specific, channel-specific, or even sponsored emotes which are introduced to the platform during the promotion of an event or a videogame. They constitute a core element in Twitch language and therefore their interpretation is essential to fully understand a message.

2.2 The *kappa* emote as a trolling indicator

The most used Twitch emote is known as 'Kappa' ( )[2]. It is a black and white emote based on the face of a former Twitch employee, and is freely available to any registered user (unlike other emotes, which are behind a paywall). There is wide agreement in the online community that this emote "represents sarcasm, irony, puns, jokes, and trolls alike"[3].

3 Tasks

In this section we describe the two tasks we propose. Similarly to Barbieri et al. (2017) we focus on, given a Twitch message, predicting its associated emote. We argue that predicting the emote is similar to understanding the intended meaning of the message (Hogenboom et al., 2013, 2015; Castellucci et al., 2015), regardless of how it was phrased.

3.1 Predicting Twitch Emotes

This is a generic task, consisting in predicting any of the 30 most used emotes in our Twitch dataset. Our aim is to classify messages that only include one and only one type of emote, even if it appears repeatedly, and which constitutes the classification label.

3.2 Trolling Detection

The availability and general usage of the 'kappa' emote enables a potential test bed for performing experiments on detecting troll messages in Twitch chatrooms. We approach this task under the assumption that adding 'kappa' at the end of a message has a similar effect as it would be to add `#irony` or `#sarcasm` at the end of a Twitter message (see (Davidov et al., 2010; Reyes et al., 2013b; Barbieri and Saggion, 2014) for extensive research on irony and sarcasm detection in Twitter under this assumption). Thus, for the trolling prediction experiments, we benefit from this particularity and construct an evaluation dataset where messages are split by considering presence or absence of this emote. In an additional experiment, we further investigate the properties of derivations

[1] http://www.cnet.com/news/learn-the-secret-language-of-twitchs-rogue-emotes/

[2] It is possible to track the usage of the most popular Twitch emotes live at `http://kappa.ws/`.

[3] http://www.urbandictionary.com/define.php?term=Kappa

Dataset	Chars	Tokens	Mentions
30 emotes	28,7M (57.4)	5,5M (10.9)	58M (0.12)
M-Kappa	22,7M (45.6)	4,4M (8.9)	68,5M (0.13)

Table 1: Statistics of the two datasets used in the emote prediction experiments.

of the 'kappa' emote, e.g. 'keepo'  , 'kappaross' or 'kappapride'  .

4 Data Gathering and Preprocessing

Our Twitch corpus was gathered thanks to a crawler of chat messages applied in the 300 most popular Twitch channels from September 2015 to February 2016. From this initial corpus, we only keep messages from the streams of the five most popular Twitch games[4] at the time (by viewer numbers).

For preprocessing, we benefit from a modified version of the CMU TWEET TOKENIZER (Gimpel et al., 2011), and removed all hyperlinks and non-ASCII characters, and also lower cased all textual content in order to reduce noise and sparsity. We also removed messages that where sequentially repeated (a common spamming practice in Twitch). We also remove messages with less than four tokens. This process yields a corpus of 62 million messages (Counter-Strike 15M, Dota 6M, Hearthstone 15M, League 20M, and World of Worcraft 6M).

We restrict our dataset to chat messages with one and only one emote.

The final dataset used in the experiments is obtained by keeping only those messages including one of the 30 most frequent emotes. From this large corpus, two datasets were derived for the experiments we report in this paper. The first one (30 Emote Dataset) is composed of 100,000 messages per game that have only one type of emote, resulting in 500,000 messages in total. Messages were randomly selected to avoid topic bias. The second dataset (Multi Kappa dataset) is composed of 100,000 messages per game that contain 'kappa' emotes, hence a total of 500,000 messages. Due to the similarity of some emotes to 'kappa' we considered five different emotes as 'kappa', namely 'kappa', 'kappapride', 'keepo', 'kappaross' and 'kappaclaus'.

[4]These games are: *Counter Strike: Global Offensive*, *Dota 2*, *Hearthstone: Heroes of Warcraft*, *League of Legends* and *World of Warcraft*.

Table 1 displays statistics of the datasets. For each dataset we show the total number of characters, the total number of tokens, the total number of user mentions, and for each statistics we also show in parenthesis the ratio per message. We can see that the 30 Emotes Dataset includes slightly longer messages (with in average 57.4 chars against 45.6 chars).

5 Models Description

In this section we describe the methodology followed to construct the three models we evaluate, namely (1) a bidirectional LSTM; (2) a BOW-based classifier; and (3) a Skipgram classifier based on vector average.

5.1 Bi-Directional LSTMs

Given the proven effectiveness of recurrent neural networks in different tasks (Chung et al., 2014; Vinyals et al., 2015; Bahdanau et al., 2014, inter-alia), which also includes modeling of tweets (Dhingra et al., 2016; Barbieri et al., 2017), our Emote prediction model is based on RNNs, which are modeled to learn sequential data. We use the word based B-LSTM architecture by Barbieri et al. (2017), designed to model emojis in Twitter.

The forward LSTM reads the message from left to right and the backward one reads the message in the reverse direction.[5] The learned vector of each LSTM, is passed through a component-wise rectified linear unit (ReLU) nonlinearity (Glorot et al., 2011); finally, an affine transformation of these learned vectors is passed to a softmax layer to give a distribution over the list of emotes that may be predicted given the Twitch chat message.

The inputs of the LSTMs are word embeddings (100 dimensions). We use a lookup table to learn word representations. For out-of-vocabulary words (OOVs), the system uses a fixed vector that is handled as a separate word. In order to train the fixed representation for OOVs, we stochastically replace (with p = 0.5) each word that occurs only once in the training data with the fixed representation in each training iteration.

5.2 Baselines

Two baselines were compared to the performance of the B-LSTM model. We chose two common algorithms for text classification, which unlike

[5]LSTM hidden states are of size 100, and each LSTM has two layers.

LSTMs, do not take into account the entire sequence of words.

5.2.1 Bag of Words

We designed a Bag-of-Words (Bow) classifier as such model has been successfully employed in several classification tasks, like sentiment analysis and irony detection (Davidov et al., 2010; Gonzalez-Ibanez et al., 2011; Reyes et al., 2013a). We represent each message with a vector of the most informative tokens (punctuation marks are included as well). Words are selected using term frequency-inverse document frequency (TF-IDF), which is intended to reflect how important a word is to a document (message) in the corpus. After obtaining a vector for each message we classify with a L2-regularized logistic regression classifier to make the predictions[6] with ε equal to 0.001.

5.2.2 Skip-Gram Vector Average

We employ the Skip-gram model (Mikolov et al., 2013) learned from the 62M Twitch dataset (where testing instances have been removed) to learn Twitch semantic vectors. Then, we build a model (henceforth, Vec-AVG) which represents each message as the average of the vectors corresponding to each word included in a given Twitch message. After obtaining a representation of each message, we train a L2-regularized logistic regression classifier, (with ε equal to 0.001).

6 Experimental Results

In this section, we describe the experimental setup for each of the tasks, and present the results of our proposed model.

6.1 Predicting Twitch Emotes

This is a multilabel classification task, where each label corresponds to the 30 emotes listed in Table 3. We compare three models, namely the BoW and Vec-AVG baselines and the B-LSTM model. We report the performance of the models in Table 2, where we also show the results of a majority baseline (where all the prediction are equal to "kappa" in this case).

We further investigate the behavior of the B-LSTM model by analyzing its *emote*-wise performance. Results are summarized in Table 3, where we report Precision, Recall and F-Measure for each *emote*, along with their Ranking and occurrences in the test set. The Ranking is the average number of *emotes* with higher probability than the gold emote in the probability distribution (in each prediction) provided by the classifiers (softmax). For example, a Ranking equal to 3.0 means that the gold *emote* is selected, in average, as the third option (the Ranking goes from 1 to X where X is the number of emotes).

Model	P	R	F1
Majority	0.06	0.25	0.10
BOW	0.35	0.33	0.29
Vec-AVG	0.46	0.38	0.32
B-LSTM	**0.47**	**0.42**	**0.39**

Table 2: Precision, Recall and F-Measure of the two models in the 30 emotes prediction experiment.

6.2 Trolling Detection

We perform two tasks. First, a *Trolling VS Non-Trolling* experiment, which we frame as a classification problem consisting in discriminating between messages with any of the 'kappa'-related emotes, and those without. Second, in the *Multi-Kappa* experiment, we aim at performing a finer-grained classification among similar but different ways of trolling, which Twitch users perform by consciously selecting a specific variation of the 'kappa' emote.

6.2.1 Trolling VS Non-Trolling

We compare the performance of the three competing models, namely BoW, Vec-AVG and B-LSTMs. However, for the purpose of this experiment, we perform modifications in the label set. Our aim is to explicitly perform a coarse and a fine-grained experiment on trolling detection by clustering together labels which are generally used for the same trolling purpose (all 'kappa'-related emotes). Note that the aim of the task is in all cases the same, discerning between trolling and non-trolling messages. The resulting label sets and their associated datasets are:

- **D1** This is the original dataset, with the original 30 emote label set. In this configuration, a true positive occurs when the model correctly assigns any 'kappa' label to a message with a 'kappa'-related emote. Similarly, true negatives come from correctly predicting the

[6] We used the MatLab implementation of Multicore LIBLINEAR `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/multicore-liblinear/`

Emo	Name	P	R	F1	Rank	Te	Tr
	kappa	0.38	0.78	0.52	1.78	25	127
	4head	0.38	0.14	0.21	3	9.2	45
	pogchamp	0.42	0.44	0.43	3.01	9.2	46
	elegiggle	0.43	0.44	0.43	3.34	8.9	43
	biblethump	0.39	0.36	0.37	4.41	5.3	25
	dansgame	0.41	0.31	0.35	4.21	4.8	24
	kreygasm	0.3	0.19	0.23	6.37	4.2	21
	failfish	0.44	0.17	0.25	5.17	3.6	19
	swiftrage	0.57	0.4	0.47	5.61	3.3	15
	wutface	0.62	0.14	0.22	7.26	2.4	14
	keepo	1	0	0.01	9.79	2.2	11
	residentsleeper	0.54	0.3	0.38	7.66	2.2	10
	kappapride	0.48	0.26	0.34	8.54	2.1	11
	trihard	0.75	0.49	0.6	5.79	2.1	10
	kappaross	0.61	0.2	0.3	8.38	1.7	8
	babyrage	0.54	0.28	0.37	8.16	1.6	9
	notlikethis	0.53	0.13	0.21	10.33	1.5	8
	opieop	0.71	0.11	0.19	9.15	1.4	7
	smorc	0.69	0.47	0.56	7.93	1.4	7
	anele	0.42	0.57	0.49	6.38	1.2	6
	seemsgood	0.75	0.33	0.46	9.83	1.2	6
	brokeback	0.8	0.16	0.27	13.85	1.2	5
	osfrog	0.7	0.48	0.57	8.36	1	6
	mrdestructoid	0.64	0.42	0.5	9.98	0.7	4
	heyguys	0.72	0.21	0.32	16.07	0.6	3
	kappaclaus	0.92	0.21	0.34	14.42	0.6	3
	datsheffy	0.72	0.48	0.57	9.65	0.5	2
	coolcat	0.89	0.38	0.53	13.34	0.4	2
	osrob	1	0.91	0.95	2.91	0.3	2
	pjsalt	0.67	0.12	0.21	19.57	0.3	2

Table 3: Detailed results for each class in the Emote prediction experiment. We report the results of the B-LSTMs model. We report Precision, Recall, F-Measure, Rank and thousand of occurrences in the Test (Te) and in the Train (Tr) for each emote.

absence of a non 'kappa' label in a message.

- **D2** In this case, we replace all 'kappa' emotes with an umbrella *super-'kappa'* emote, thus forcing the model to learn a coarser-grained class. Negative examples are the same as in D1.

- **D3** This is the coarsest of the three configurations, where we train with a *super-'kappa'* positive class, and a superclass for negative cases (clustering all the non-'kappa' emotes into a dummy negative label).

D	Model	Class	P	R	F1
-	Majority	Avg	0.47	0.68	0.56
D1	BoW	Iro	0.41	0.73	0.53
		Non-Iro	0.81	0.52	0.63
		Avg	0.68	0.59	0.60
	Vec-AVG	Iro	0.41	0.89	0.56
		Non-Iro	0.89	0.40	0.55
		Avg	0.73	0.55	0.55
	B-LSTM	Iro	0.47	0.79	**0.59**
		Non-Iro	0.86	0.59	0.70
		Avg	0.74	0.66	0.67
D2	BoW	Iro	0.41	0.78	0.53
		Non-Iro	0.82	0.47	0.60
		Avg	0.69	0.57	0.58
	Vec-AVG	Iro	0.39	0.92	0.55
		Non-Iro	0.91	0.34	0.49
		Avg	0.74	0.52	0.51
	B-LSTM	Iro	0.45	**0.85**	**0.59**
		Non-Iro	**0.88**	0.53	0.66
		Avg	**0.75**	0.63	0.64
D1	BoW	Iro	0.44	0.29	0.35
		Non-Iro	0.72	**0.83**	0.77
		Avg	0.63	0.66	0.64
	Vec-AVG	Iro	0.72	0.91	0.80
		Non-Iro	0.52	0.22	0.31
		Avg	0.66	0.69	0.65
	B-LSTM	Iro	**0.58**	0.49	0.53
		Non-Iro	0.78	**0.83**	**0.81**
		Avg	0.72	**0.72**	**0.72**

Table 4: Results of the trolling prediction experiments. The classes are two, trolling and non-trolling.

6.2.2 Multi-'Kappa'

'Kappa'-related emotes are used to express irony or sarcasm and in general troll alike messages. We are interested in investigating if there is a fine-grained pattern in the usage of any of these emotes, as the community does not seem to use them interchangeably. Thus, we perform a *multi-'kappa'* experiment, i.e. an experiment designed to discern among nuanced ironic messages.

In Table 5 we show comparative results of the models under evaluation for this task, in terms of Precision, Recall and F-Measure of the five classes ordered by frequency, from the most frequent ('kappa') to the rarest ('kappaclaus'). Similarly as

Model	Class	P	R	F1
Majority	Avg	0.60	0.78	0.68
BoW	kappa	0.81	0.98	0.88
	kappapride	0.67	0.28	0.39
	keepo	0.20	0.01	0.02
	kappaross	0.60	0.19	0.28
	kappaclaus	0.51	0.14	0.22
	Avg	0.73	0.79	0.74
Vec-AVG	kappa	0.80	0.99	0.89
	kappapride	0.76	0.23	0.36
	keepo	1.00	0.01	0.02
	kappaross	0.68	0.10	0.18
	kappaclaus	0.77	0.16	0.27
	Avg	**0.81**	0.80	0.74
B-LSTM	kappa	0.81	0.99	0.89
	kappapride	0.78	0.32	0.46
	keepo	0.00	0.00	0.00
	kappaross	0.84	0.24	0.38
	kappaclaus	0.82	0.20	0.32
	Avg	0.75	**0.81**	**0.76**

Table 5: Results of the multi 'kappa' prediction experiment.

in the previous experiment, the B-LSTM method outperformed the baselines, this time, however, with a smaller difference. We can see that the three systems show similar F1 in the 'kappa' prediction (0.74, 0.74 and 0.76). However the B-LSTM works better on the other kappa emotes, suggesting that the B-LSTM model is better at modeling the inner semantic of the kappa emotes.

6.3 Discussion

In the first experiment, *Predicting Twitch Emotes*, our B-LSTM model notably outperforms the baselines, showing a 10 point difference. Further analysis on the behavior of our model can be found in Table 3. We observed that the emotes which are best recognized (highest F-Measure) are not necessarily the most frequent. For example, the best predicted emotes are 'osrob', 'osfrog' and 'datsheffy', with F-Measure scores of 0.95, 0.57 and 0.57 respectively. In contrast, the most difficult emote to identify is 'keepo' (F-Measure of 0.01), probably due to its semantic overlap with 'kappa'. On the other hand, specific emotes such as 'trihard'[7], 'mrdestructoid'[8] or 'smorc'[9] are easier to predict, due to their stronger bound to a specific topic and the univocity of their meaning.

However, we found that the model often prioritizes the most frequent emotes. We look into this observation by computing Pearson Correlation (PC) between frequency and Ranking, which yields -0.6, hence, if an emote shows high frequency, it has low Ranking, and vice versa. However, in terms of Recall and F-Measure, these do not show any correlation with frequency (PC of 0.3 and 0.1 respectively), nor Ranking. Finally, let us highlight the fact that Precision is inversely correlated to frequency, with a PC score of -0.54. Again, the model may have high confidence in rare emotes only in very specific cases, and it is then when they are selected.

We provide a visualization of the model's performance with a confusion matrix (Figure 1). As mentioned earlier, the B-LSTM has a bias towards 'kappa', the most frequent emote in Twitch. It is also clear that 'biblethump', 'elegiggle', 'kreygasm' and 'pogchamp' are also very frequent in Twitch language due to the large number of confusions involving these emotes. 'Elegiggle' and 'failfish' are often confused. The main reason behind this confusion might be that they are both used in situations where the streamer has failed ('faifish'), and the audience finds this funny. Interestingly, '4head', one of the most frequent emotes, seems to not be the source of wrong predictions. The reason behind that is that the usual usage of '4Head' is to substitute the word forehead, which clearly restricts the communicative contexts available for it being used. The emote 'pogchamp', moreover, is wrongly selected with notable frequency. We have observed that the use of 'pogchamp' and 'kreygasm' emotes is fairly interchangeable, as in gaming, the notion of positive surprise ('pogchamp') and ecstasy ('kreygasm') are more strongly related to the same events or reactions.

Table 4 shows the performance of our model on the task of differentiating between ironic and non-ironic messages using three different training strategies and comparing these performances with a two baselines. It can be observed that once again our model outperforms the baselines in every case and that it achieves very competitive performance when the system is trained by labeling every message with a 'kappa' emote as trolling and every message with a non-kappa emote as non-trolling.

[7]It refers to the idea of 'trying hard', i.e. putting maximum effort in a task.

[8]General robot emoticon.

[9]Used in scenarios where 'Orc' characters are present.

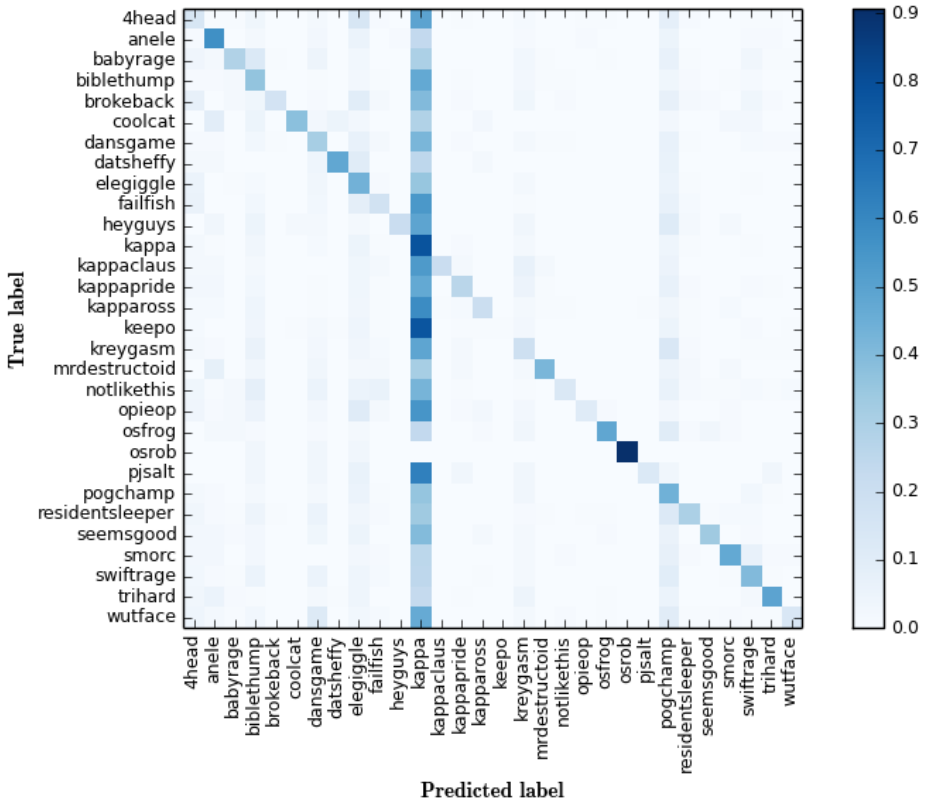

Figure 1: Confusion Matrix of B-LSTMs of the 30 emotes prediction experiment

Even in the first two training strategies, where the messages are labeled with a higher amount of emotes (and as a result, the system can confuse emotes that are used in similar scenarios), the performance is high.

Once we differentiated between trolling and non-trolling messages, we further explored a finer grained classification process over the 'kappa' derivations. Table 5 presents the results in the classification of 'kappa' emotes of our system compared again with the two baselines. From our results, it seems that there are indeed differences in the usage of certain emotes. The emote 'kappa' is a sort of generalisation of each one of its other derivations. Note that there are three cases where the usage of emotes that are not 'kappa' have patterns that are not equivalent: 'kappaclaus', which is a version of kappa with a christmas theme, 'kappapride' which is a kappa face with the characteristic colors of the rainbow flag of the LGBT movement and 'kappaross', which is a Twitch homage to the painter Bob Ross. Even if the underlying intention of the mentioned emotes is trolls alike, it is clear that their intended meaning is not the same as 'kappa'. On the other hand, 'keepo', the 'kappa' emote with cat ears, is always confused with 'kappa', and thus we can conclude that both emotes are used interchangeably.

7 Related Work

The most similar communicative phenomena to emotes are *emojis*. Emojis are used by the vast majority of Social Media services and instant messaging platforms (Jibril and Abdullah, 2013; Park et al., 2013, 2014). Emojis (like the older emoticons) give the possibility to express a variety of ideas and feelings in a visual, concise and appealing way that is perfectly suited for the informal style of Social Media. Several recent works studied Emojis, focusing on emojis' semantics and usage (Aoki and Uchida, 2011; Barbieri et al., 2016a,b,c; Eisner et al., 2016; Ljubesic and Fiser, 2016; Ai et al., 2017; Miller et al., 2017), and sentiment (Novak et al., 2015; Hu et al., 2017). Finally, (Barbieri et al., 2017) presented an emoji prediction model for Twitter, where they use a char based B-LSTM to detect the 20 most frequent emojis.

Most work on irony and sarcasm detection in

Twitter has employed *hashtags* as labels for detecting irony. This approach was introduced by Tsur et al. (Tsur et al., 2010) and (Gonzalez-Ibanez et al., 2011), who used the *#sarcasm* hashtag to retrieve sarcastic tweets. This technique was later validated by various studies (Wang, 2013; Sulis et al., 2016), which analyze the language associated to the use of irony-related *hashtags* (such as *#irony*, and *#not*). Recent years have seen an increase in models for detecting *#irony* and *#sarcasm*. Many of these models adopted hand crafted features (amoung others (Reyes et al., 2013a; Barbieri and Saggion, 2014; Liu et al., 2014; Joshi et al., 2015)), and others employed pretrained word embeddings or deep learning systems such as CNN or LSTMs (Joshi et al., 2016; Ghosh and Veale, 2016; Poria et al., 2016; Amir et al., 2016).

8 Conclusions and Future Work

In this paper we have addressed the problem of modeling the usage of Twitch emotes. This is an important problem in social media text understanding, as the inherent noisy nature of these messages can be alleviated by having robust systems that interpret the semantics of visual aids such as Twitter emojis or Twitch emotes.

Emote understanding is approached in this paper via different approaches, namely a BOW system, a logistic regression classifier based on embedding average, and a bidirectional LSTM. The main conclusion that we draw from our experiments is that the RNN model is more capable to predict Twitch emotes than its competing baselines. In addition, we performed an analysis on the usage of different trolling emotes and studied their usage patterns and differences.

As future work we plan to incorporate more context to the model, providing a representation of previous chat messages where the emote appears. This would allow us to tackle the problem of the emote detection as a sequence modeling task, and this will be more natural as it is not easy to predict an emote of a message with no context. Finally, as Barbieri et al. (2017) we plan to investigate character-based approaches to represent words (Ling et al., 2015; Ballesteros et al., 2015) and/or messages (Dhingra et al., 2016) since Twitch data contain noisy text.

Acknowledgments

We thank the three anonymous reviewers for their time and their useful suggestions. Francesco, Luis and Horacio acknowledge support from the TUNER project (TIN2015-65308-C5-5-R, MINECO/FEDER, UE) and the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

References

Wei Ai, Xuan Lu, Xuanzhe Liu, Ning Wang, Gang Huang, and Qiaozhu Mei. 2017. Untangling emoji popularity through semantic embeddings. In *International AAAI Conference on Web and Social Media, ICWSM*. pages 2–11.

Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976* .

Sho Aoki and Osamu Uchida. 2011. A method for automatically generating the emotional vectors of emoticons using weblog articles. In *Proc. 10th WSEAS Int. Conf. on Applied Computer and Applied Computational Science, Stevens Point, Wisconsin, USA*. pages 132–136.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* .

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 349–359. http://aclweb.org/anthology/D15-1041.

Francesco Barbieri, Miguel Ballesteros, and Horacio Saggion. 2017. Are Emojis Predictable? In *European Chapter of the Association for Computational Linguistics, EACL*. ACL, Valencia, Spain.

Francesco Barbieri, Luis Espinosa Anke, and Horacio Saggion. 2016a. Revealing Patterns of Twitter Emoji Usage in Barcelona and Madrid. In *19 th International Conference of the Catalan Association for Artificial Intelligence*. Barcelona, Spain, pages 326–332.

Francesco Barbieri, German Kruszewski, Francesco Ronzano, and Horacio Saggion. 2016b. 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, Amsterdam, Netherlands, pages 531–535.

Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2016c. What does this emoji mean? a vector space skip-gram model for twitter emojis. In *Language Resources and Evaluation conference, LREC*. Portoroz, Slovenia, pages 526–534.

Francesco Barbieri and Horacio Saggion. 2014. Modelling Irony in Twitter. In *Proceedings of the EACL Student Research Workshop*. ACL, Gothenburg, Sweden, pages 56–64. http://www.aclweb.org/anthology/E14-3007.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science* 2(1):1–8.

Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2015. Acquiring a large scale polarity lexicon through unsupervised distributional methods. In *Natural Language Processing and Information Systems*, Springer, pages 73–86.

Deepayan Chakrabarti and Kunal Punera. 2011. Event summarization using tweets. *International AAAI Conference on Web and Social Media, ICWSM* 11:66–73.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR* .

Aron Culotta. 2010. Towards detecting influenza epidemics by analyzing twitter messages. In *Proceedings of the first workshop on social media analytics*. ACM, pages 115–122.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pages 107–116.

Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William W. Cohen. 2016. Tweet2vec: Character-based distributed representations for social media. In *Proceedings of the ACL*.

Ben Eisner, Tim Rocktaschel, 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*. Association for Computational Linguistics, Austin, TX, USA, pages 48–54.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *WASSA@ NAACL-HLT*. pages 161–169.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, pages 42–47.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Conference on Artificial Intelligence and Statistics (AISTATS)*.

Roberto Gonzalez-Ibanez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: a closer look. In *NAACL*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*.

Alexander Hogenboom, Daniella Bal, Flavius Frasincar, Malissa Bal, Franciska de Jong, and Uzay Kaymak. 2013. Exploiting emoticons in sentiment analysis. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, pages 703–710.

Alexander Hogenboom, Daniella Bal, Flavius Frasincar, Malissa Bal, Franciska De Jong, and Uzay Kaymak. 2015. Exploiting emoticons in polarity classification of text. *J. Web Eng.* 14(1&2):22–40.

Tianran Hu, Han Guo, Hao Sun, Thuyvy Thi Nguyen, and Jiebo Luo. 2017. Spice up your chat: The intentions and sentiment effects of using emoji. *arXiv preprint arXiv:1703.02860* .

Tanimu Ahmed Jibril and Mardziah Hayati Abdullah. 2013. Relevance of emoticons in computer-mediated communication contexts: An overview. *Asian Social Science* 9(4):201.

Aditya Joshi, Prayas Jain, Pushpak Bhattacharyya, and Mark Carman. 2016. Who would have thought of that!': A hierarchical topic model for extraction of sarcasm-prevalent topics and sarcasm detection. *arXiv preprint arXiv:1611.04326* .

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *ACL (2)*. pages 757–762.

Mehdi Kaytoue, Arlei Silva, Loïc Cerf, Wagner Meira Jr, and Chedy Raïssi. 2012. Watch me playing, i am a professional: a first study on video game live streaming. In *Proceedings of the 21st international conference companion on World Wide Web*. ACM, pages 1181–1188.

Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Peng Liu, Wei Chen, Gaoyan Ou, Tengjiao Wang, Dongqing Yang, and Kai Lei. 2014. Sarcasm detection in social media based on imbalanced classification. In *International Conference on Web-Age Information Management*. Springer, pages 459–471.

Nikola Ljubesic and Darja Fiser. 2016. A global analysis of emoji usage. In *Proceedings of the 10th Web as Corpus Workshop and the EmpiriST Shared Task*. Association for Computational Linguistics, Berlin, Germany, pages 82–89.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168* .

Hannah Jean Miller, Daniel Kluver, Jacob Thebault-Spieker, Loren G Terveen, and Brent J Hecht. 2017. Understanding emoji ambiguity in context: The role of text in emoji-related miscommunication. In *International AAAI Conference on Web and Social Media, ICWSM*. pages 152–161.

Saif M Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the International Workshop on Semantic Evaluation, SemEval*. volume 16.

Petra Kralj Novak, Jasmina Smailovic, Borut Sluban, and Igor Mozetic. 2015. Sentiment of emojis. *PloS one* 10(12):e0144296.

Jedrzej Olejniczak. 2015. A linguistic study of language variety used on twitch.tv: Descriptive and corpus-based approaches. In *Proceedings of Redefining Community in Intercultural Context (RCIC15*.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval* 2(1-2):1–135.

Jaram Park, Young Min Baek, and Meeyoung Cha. 2014. Cross-cultural comparison of nonverbal cues in emoticons on twitter: Evidence from big data analysis. *Journal of Communication* 64(2):333–354.

Jaram Park, Vladimir Barash, Clay Fink, and Meeyoung Cha. 2013. Emoticon style: Interpreting differences in emoticons across cultures. In *International AAAI Conference on Web and Social Media, ICWSM*.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. 2016. A deeper look into sarcastic tweets using deep convolutional neural networks. *arXiv preprint arXiv:1610.08815* .

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013a. A multidimensional Approach for Detecting Irony in Twitter. *Language Resources and Evaluation* pages 1–30.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013b. A multidimensional approach for detecting irony in twitter. *Language resources and evaluation* 47(1):239–268.

Tamara A Small. 2011. What the hashtag? a content analysis of canadian politics on twitter. *Information, Communication & Society* 14(6):872–895.

Thomas Smith, Marianna Obrist, and Peter Wright. 2013. Live-streaming changes the (video) game. In *Proceedings of the 11th european conference on Interactive TV and video*. ACM, pages 131–138.

Emilio Sulis, Delia Irazú Hernández Farías, Paolo Rosso, Viviana Patti, and Giancarlo Ruffo. 2016. Figurative messages and affect in twitter: differences between# irony,# sarcasm and# not. *Knowledge-Based Systems* .

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *International AAAI Conference on Web and Social Media, ICWSM*.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Proc. ICLR*.

Po-Ya Angela Wang. 2013. #irony or #sarcasma quantitative and qualitative study based on twitter. *27th Pacific Asia Conference on Language, Information, and Computation* .

Jianshu Weng and Bu-Sung Lee. 2011. Event detection in twitter. *International AAAI Conference on Web and Social Media, ICWSM* 11:401–408.