# Are Emojis Predictable?

**Francesco Barbieri**◇ **Miguel Ballesteros**♠ **Horacio Saggion**◇
◇Large Scale Text Understanding Systems Lab, TALN Group
Universitat Pompeu Fabra, Barcelona, Spain
♠IBM T.J Watson Research Center, U.S
{francesco.barbieri, horacio.saggion}@upf.edu
miguel.ballesteros@ibm.com

## Abstract

Emojis are ideograms which are naturally combined with plain text to visually complement or condense the meaning of a message. Despite being widely used in social media, their underlying semantics have received little attention from a Natural Language Processing standpoint. In this paper, we investigate the relation between words and emojis, studying the novel task of predicting which emojis are evoked by text-based tweet messages. We train several models based on Long Short-Term Memory networks (LSTMs) in this task. Our experimental results show that our neural model outperforms two baselines as well as humans solving the same task, suggesting that computational models are able to better capture the underlying semantics of emojis.

## 1 Introduction

The advent of social media has brought along a novel way of communication where meaning is composed by combining short text messages and visual enhancements, the so-called *emojis*. This visual language is as of now a *de-facto* standard for online communication, available not only in Twitter, but also in other large online platforms such as Facebook, Whatsapp, or Instagram.

Despite its status as language form, emojis have been so far scarcely studied from a Natural Language Processing (NLP) standpoint. Notable exceptions include studies focused on emojis' semantics and usage (Aoki and Uchida, 2011; Barbieri et al., 2016a; Barbieri et al., 2016b; Barbieri et al., 2016c; Eisner et al., 2016; Ljubešic and Fišer, 2016), or sentiment (Novak et al., 2015). However, the interplay between text-based messages and emojis remains virtually unexplored. This paper aims to fill this gap by investigating the relation between words and emojis, studying the problem of predicting which emojis are evoked by text-based tweet messages.

Miller et al. (2016) performed an evaluation asking human annotators the meaning of emojis, and the sentiment they evoke. People do not always have the same understanding of emojis, indeed, there seems to exist multiple interpretations of their meaning beyond their designer's intent or the physical object they evoke[1]. Their main conclusion was that emojis can lead to misunderstandings. The ambiguity of emojis raises an interesting question in human-computer interaction: how can we teach an artificial agent to correctly interpret and recognise emojis' use in spontaneous conversation?[2] The main motivation of our research is that an artificial intelligence system that is able to predict emojis could contribute to better natural language understanding (Novak et al., 2015) and thus to different natural language processing tasks such as generating emoji-enriched social media content, enhance emotion/sentiment analysis systems, and improve retrieval of social network material.

In this work, we employ a state of the art classification framework to automatically predict the most likely emoji a Twitter message evokes. The model is based on Bidirectional Long Short-term Memory Networks (BLSTMs) with both standard lookup word representations and character-based representation of tokens. We will show that the BLSTMs outperform a bag of words baseline, a baseline based on semantic vectors, and human annotators in this task.

[1] https://www.washingtonpost.com/news/the-intersect/wp/2016/02/19/the-secret-meanings-of-emoji/

[2] http://www.dailydot.com/debug/emoji-miscommunicate/

*Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 105–111,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 😂 | ❤️ | 😍 | 🔥 | 💯 | 😊 | 🙌 | 😘 | 🎄 | 💕 |
| 100.7 | 89.9 | 59 | 33.8 | 28.6 | 27.9 | 22.5 | 21.5 | 21 | 20.8 |
| 🎉 | 😭 | 💙 | ✨ | ❄️ | 😎 | 💪 | 🙏 | 👌 | 💋 |
| 19.5 | 18.6 | 18.5 | 17.5 | 17 | 16.1 | 15.9 | 15.2 | 14.2 | 10.9 |

Table 1: The 20 most frequent emojis that we use in our experiments and the number of thousand tweets they appear in.

## 2 Dataset and Task

**Dataset:** We retrieved 40 million tweets with the Twitter APIs[3]. Tweets were posted between October 2015 and May 2016 geo-localized in the United States of America. We removed all hyperlinks from each tweet, and lowercased all textual content in order to reduce noise and sparsity. From the dataset, we selected tweets which include *one and only one* of the 20 most frequent emojis, resulting in a final dataset[4] composed of 584,600 tweets. In the experiments we also consider the subsets of the 10 (502,700 tweets) and 5 most frequent emojis (341,500 tweets). See Table 1 for the 20 most frequent emojis that we consider in this work.

**Task**: We remove the emoji from the sequence of tokens and use it as a label both for training and testing. The task for our machine learning models is to predict the single emoji that appears in the input tweet.

## 3 Models

In this Section, we present and motivate the models that we use to predict an emoji given a tweet. The first model is an architecture based on Recurrent Neural Networks (Section 3.1) and the second and third are the two baselines (Section 3.2.1 and 3.2.2). The two major differences between the RNNs and the baselines, is that the RNNs take into account sequences of words and thus, the entire context.

### 3.1 Bi-Directional LSTMs

Given the proven effectiveness and the impact of recurrent neural networks in different tasks (Chung et al., 2014; Vinyals et al., 2015; Dzmitry et al., 2014; Dyer et al., 2015; Lample et al., 2016; Wang et al., 2016, inter-alia), which also includes modeling of tweets (Dhingra et al., 2016), our emoji prediction model is based on bi-directional Long Short-term Memory Networks (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005). The B-LSTM can be formalized as follows:

$$\mathbf{s} = \max\{\mathbf{0}, \mathbf{W}[\mathbf{fw}; \mathbf{bw}] + \mathbf{d}\}$$

where $\mathbf{W}$ is a learned parameter matrix, $\mathbf{fw}$ is the forward LSTM encoding of the message, $\mathbf{bw}$ is the backward LSTM encoding of the message, and $\mathbf{d}$ is a bias term, then passed through a component-wise ReLU. The vector $\mathbf{s}$ is then used to compute the probability distribution of the emojis given the message as:

$$p(e \mid \mathbf{s}) = \frac{\exp\left(\mathbf{g}_e^\top \mathbf{s} + q_e\right)}{\sum_{e' \in \mathcal{E}} \exp\left(\mathbf{g}_{e'}^\top \mathbf{s} + q_{e'}\right)}$$

where $\mathbf{g}_{e'}$ is a column vector representing the (output) embedding[5] of the emoji $e$, and $q_e$ is a bias term for the emoji $e$. The set $\mathcal{E}$ represents the list of emojis. The loss/objective function the network aims to minimize is the following:

$$Loss = -log(p(e_m \mid \mathbf{s}))$$

where $m$ is a tweet of the training set $\mathcal{T}$, $\mathbf{s}$ is the encoded vector representation of the tweet and $e_m$ is the emoji contained in the tweet $m$. The inputs of the LSTMs are word embeddings[6]. Following, we present two alternatives explored in the experiments presented in this paper.

**Word Representations**: We generate word embeddings which are learned together with the updates to the model. We stochastically replace (with $p$ = 0.5) each word that occurs only once in the training data with a fixed represenation (out-of-vocabulary words vector). When we use pretrained word embeddings, these are concatenated with the learned vector representations obtaining a final representation for each word type. This is similar to the treatment of word embeddings by Dyer et al. (2015).

**Character-based Representations**: We compute character-based continuous-space vector embeddings (Ling et al., 2015b; Ballesteros et al., 2015) of the tokens in each tweet using, again, bidirectional LSTMs. The character-based approach learns representations for words that are orthographically similar, thus, they should be able to handle different alternatives of the same word type occurring in social media.

[3] https://dev.twitter.com

[4] Available at http://sempub.taln.upf.edu/tw/eacl17

[5] The output embeddings of the emojis have 100 dimensions.

[6] 100 dimensions.

### 3.2 Baselines

In this Section we describe the two baselines. Unlike the previous model, the baselines do not take into account the word order. However, in the second baseline (Section 3.2.2) we abstract on the plain word representation using semantic vectors, previously trained on Twitter data.

#### 3.2.1 Bag of Words

We applied a bag of words classifier as baseline, since it has been successfully employed in several classification tasks, like sentiment analysis and topic modeling (Wallach, 2006; Blei, 2012; Titov and McDonald, 2008; Maas et al., 2011; Davidov et al., 2010). We represent each message with a vector of the most informative tokens (punctuation marks included) selected using term frequency−inverse document frequency (TF-IDF). We employ a L2-regularized logistic regression classifier to make the predictions.

#### 3.2.2 Skip-Gram Vector Average

We train a Skip-gram model (Mikolov et al., 2013) learned from 65M Tweets (where testing instances have been removed) to learn Twitter semantic vectors. Then, we build a model (henceforth, AVG) which represents each message as the average of the vectors corresponding to each token of the tweet. Formally, each message $m$ is represented with the vector $V_m$:

$$Vm = \frac{\sum_{t \in T_m} S_t}{|T_m|}$$

Where $T_m$ are the set of tokens included in the message $m$, $S_t$ is the vector of token $t$ in the Skip-gram model, and $|T_m|$ is the number of tokens in $m$. After obtaining a representation of each message, we train a L2-regularized logistic regression, (with $\varepsilon$ equal to 0.001).

## 4 Experiments and Evaluation

In order to study the relation between words and emojis, we performed two different experiments. In the first experiment, we compare our machine learning models, and in the second experiment, we pick the best performing system and compare it against humans.

### 4.1 First Experiment

This experiment is a classification task, where in each tweet the unique emoji is removed and used as a label for the entire tweet. We use three datasets, each containing the 5, 10 and 20 most frequent emojis (see Section 2). We analyze the performance of the five models described in Section 3: a bag of words model, a Bidirectional LSTM model with character-based representations (char-BLSTM), a Bidirectional LSTM model with standard lookup word representations (word-BLSTM). The latter two were trained with/without pretrained word vectors. To pretrain the word vectors, we use a modified skip-gram model (Ling et al., 2015a) trained on the English Gigaword corpus[7] version 5.

We divide each dataset in three parts, training (80%), development (10%) and testing (10%). The three subsets are selected in sequence starting from the oldest tweets and from the training set since automatic systems are usually trained on past tweets, and need to be robust to future topic variations.

| | **5** | | | **10** | | | **20** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **P** | **R** | **F1** | **P** | **R** | **F1** | **P** | **R** | **F1** |
| **BOW** | .59 | .60 | .58 | .43 | .46 | .41 | .32 | .34 | .29 |
| **AVG** | .60 | .60 | .57 | .44 | .47 | .40 | .34 | .36 | .29 |
| **W** | .59 | .59 | .59 | .46 | .46 | .46 | .35 | .36 | .33 |
| **C** | .61 | .61 | .61 | .44 | .44 | .44 | .36 | .37 | .32 |
| **W+P** | .61 | .61 | .61 | .45 | .45 | .45 | .34 | .36 | .32 |
| **C+P** | **.63** | **.63** | **.63** | **.48** | **.47** | **.47** | **.42** | **.39** | **.34** |

Table 2: Results of 5, 10 and 20 emojis. Precision, Recall, F-measure. BOW is bag of words, AVG is the Skipgram Average model, C refers to char-BLSTM and W refers to word-BLSTM. +P refers to pretrained embeddings.

Table 2 reports the results of the five models and the baseline. All neural models outperform the baselines in all the experimental setups. However, the BOW and AVG are quite competitive, suggesting that most emojis come along with specific words (like the word *love* and the emoji ❤️). However, considering sequences of words in the models seems important for encoding the meaning of the tweet and therefore contextualize the emojis used. Indeed, the B-LSTMs models always outperform BOW and AVG. The character-based model with pretrained vectors is the most accurate at predicting emojis. The character-based model seems to capture orthographic variants of the same word in social media. Similarly, pretrained vectors allow to initialize the system with unsuper-

[7] https://catalog.ldc.upenn.edu/LDC2003T05

vised pre-trained semantic knowledge (Ling et al., 2015a), which helps to achieve better results.

| Emoji | P | R | F1 | Rank | Num |
|---|---|---|---|---|---|
| 😂 | 0.48 | **0.74** | **0.58** | 2.12 | 783 |
| ❤️ | 0.32 | **0.74** | 0.45 | **1.59** | 757 |
| 😍 | 0.35 | 0.22 | 0.27 | 3.58 | 470 |
| 😊 | 0.31 | 0.15 | 0.21 | 4.2 | 260 |
| 😎 | 0.24 | 0.1 | 0.14 | 4.39 | 212 |
| 🔥 | 0.46 | 0.49 | 0.47 | 3.76 | 207 |
| 💕 | 1 | 0 | 0.01 | 4.69 | 206 |
| 💯 | 0.44 | 0.19 | 0.27 | 5.15 | 200 |
| 💪 | 0.44 | 0.54 | 0.48 | 4.71 | 165 |
| 🙌 | 0.33 | 0.11 | 0.17 | 5.79 | 150 |
| 😘 | 0.3 | 0.12 | 0.17 | 5.78 | 148 |
| 💙 | 0.54 | 0.11 | 0.18 | 6.73 | 131 |
| ✨ | 0.45 | 0.19 | 0.27 | 6.43 | 120 |
| 💋 | **0.56** | 0.09 | 0.15 | 7.58 | 112 |
| 👌 | 0.2 | 0.01 | 0.02 | 9.01 | 110 |
| 🙏 | 0.46 | 0.33 | 0.39 | 5.83 | 108 |
| 😭 | 0.5 | 0.08 | 0.13 | 4.9 | 105 |
| 🎉 | 0.32 | 0.25 | 0.28 | 6.13 | 89 |
| ❄️ | 0.44 | 0.53 | 0.48 | 5.35 | 34 |
| 🎄 | 0.22 | 0.67 | 0.33 | 1.67 | 3 |

Table 3: Precision, Recall, F-measure, Ranking and occurrences in the test set of the 20 most frequent emojis using char-BLSTM + Pre.

**Qualitative Analysis of Best System:** We analyze the performances of the char-BLSTM with pretrained vectors on the 20-emojis dataset, as it resulted to be the best system in the experiment presented above. In Table 3 we report Precision, Recall, F-measure and Ranking[8] of each emoji. We also added in the last column the occurrences of each emoji in the test set.

The frequency seems to be very relevant. The Ranking of the most frequent emojis is lower than the Ranking of the rare emojis. This means that if an emoji is frequent, it is more likely to be on top of the possible choices even if it is a mistake. On the other hand, the F-measure does not seem to depend on frequency, as the highest F-measures are scored by a mix of common and uncommon emojis (😂, ❤️, 🔥, and ❄️) which are respectively the first, second, the sixth and the second last emoji in terms of frequencies.

The frequency of an emoji is not the only important variable to detect the emojis properly; it is also important whether in the set of emojis there are emojis with similar semantics. If this is the case the model prefers to predict the most frequent emojis. This is the case of the 💕 emoji that is almost never predicted, even if the Ranking is not too high (4.69). The model prefers similar but most frequent emojis, like ❤️ (instead of 💕). The same behavior is observed for the 💙 emoji, but in this case the performance is a bit better due to some specific words used along with the blue heart: "blue", "sea" and words related to childhood (e.g. "little" or "Disney").

Another interesting case is the Christmas tree emoji 🎄, that is present only three times in the test set (as the test set includes most recent tweets and Christmas was already over; this emoji is commonly used in tweets about Christmas). The model is able to recognize it twice, but missing it once. The correctly predicted cases include the word "Christmas"; and it fails to predict: *"getting into the holiday spirit with this gorgeous pair of leggings today ! #festiveleggings"*, since there are no obvious clues (the model chooses ❤️ instead probably because of the intended meaning of "holiday" and "gorgeous".).

In general the model tends to confuse similar emojis to ❤️ and 😂, probably for their higher frequency and also because they are used in multiple contexts. An interesting phenomenon is that 😭 is often confused with 😂. The first one represent a small face crying, and the second one a small face laughing, but the results suggest that they appear in similar tweets. The punctuation and tone used is often similar (many exclamation marks and words like *"omg"* and *"hahaha"*). Irony may also play a role to explain the confusion, e.g. *"I studied journalism and communications , I'll be an awesome speller! Wrong.* 😭 *haha so much fun"*.

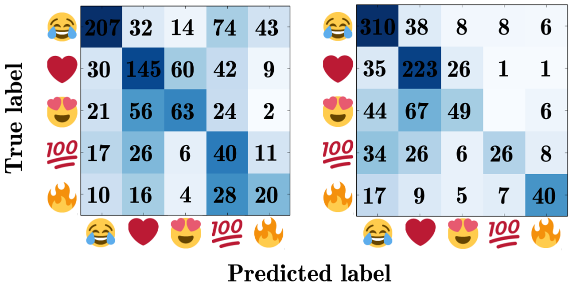
## 4.2 Second Experiment

Given that Miller et al. (2016) pointed out that people tend to give multiple interpretations to emojis, we carried out an experiment in which we evaluated human and machine performances on the same task. We randomly selected 1,000 tweets from our test set of the 5 most frequent emojis used in the previous experiment, and asked

[8] The Ranking is a number between 1 and 20 that represents the average number of emojis with higher probability than the gold emoji in the probability distribution of the classifier.

|  | Humans |  |  | B-LSTM |  |  |
|---|---|---|---|---|---|---|
| **Emo** | **P** | **R** | **F1** | **P** | **R** | **F1** |
| 😂 | 0.73 | 0.56 | 0.63 | 0.7 | 0.84 | **0.77** |
| ❤️ | 0.53 | 0.51 | 0.52 | 0.61 | 0.78 | **0.69** |
| 😍 | 0.43 | 0.38 | **0.4** | 0.52 | 0.3 | 0.38 |
| 💯 | 0.19 | 0.4 | 0.26 | 0.62 | 0.26 | **0.37** |
| 🔥 | 0.24 | 0.26 | 0.25 | 0.66 | 0.51 | **0.58** |
| **Avg** | 0.53 | 0.48 | 0.50 | 0.65 | 0.65 | **0.65** |

Table 4: Precision, Recall and F-Measure of human evaluation and the character-based B-LSTM for the 5 most frequent emojis and 1,000 tweets.



| True label | 😂 | ❤️ | 😍 | 💯 | 🔥 |
|---|---|---|---|---|---|
| 😂 | 207 | 32 | 14 | 74 | 43 |
| ❤️ | 30 | 145 | 60 | 42 | 9 |
| 😍 | 21 | 56 | 63 | 24 | 2 |
| 💯 | 17 | 26 | 6 | 40 | 11 |
| 🔥 | 10 | 16 | 4 | 28 | 20 |

| True label | 😂 | ❤️ | 😍 | 💯 | 🔥 |
|---|---|---|---|---|---|
| 😂 | 310 | 38 | 8 | 8 | 6 |
| ❤️ | 35 | 223 | 26 | 1 | 1 |
| 😍 | 44 | 67 | 49 |  | 6 |
| 💯 | 34 | 26 | 6 | 26 | 8 |
| 🔥 | 17 | 9 | 5 | 7 | 40 |

Figure 1: Confusion matrix of the second experiment. On the left the human evaluation and on the right the char-BLSTM model.

humans to predict, after reading a tweet (with the emoji removed), the emoji the text evoked. We opted for the 5 emojis task to reduce annotation efforts. After displaying the text of the tweet, we asked the human annotators "What is the emoji you would include in the tweet?", and gave the possibility to pick one of 5 possible emojis 😂, ❤️, 😍, 💯, and 🔥. Using the crowdsourcing platform ''CrowdFlower", we designed an experiment where the same tweet was presented to four annotators (selecting the final label by majority agreement). Each annotator assessed a maximum of 200 tweets. The annotators were selected from the United States of America and of high quality (level 3 of CrowdFlower). One in every ten tweets, was an obvious test question, and annotations from subjects who missed more than 20% of the test questions were discarded. The overall inter-annotator agreement was 73% (in line with previous findings (Miller et al., 2016)). After creating the manually annotated dataset, we compared the human annotation and the char-BLSTM model with the gold standard (i.e. the emoji used in the tweet).

We can see in Table 4, where the results of the comparison are presented, that the char-BLSTM performs better than humans, with a F1 of 0.65 versus 0.50. The emojis that the char-BLSTM struggle to predict are 😍 and 💯, while the human annotators mispredict 💯 and 🔥 mostly. We can see in the confusion matrix of Figure 1 that 😍 is misclassified as ❤️ by both human and LSTM, and the 💯 emoji is mispredicted as 😂 and ❤️. An interesting result is the number of times 💯 was chosen by human annotators; this emoji occurred 100 times (by chance) in the test set, but it was chosen 208 times, mostly when the correct label was the laughing emoji 😂. We do not observe the same behavior in the char-BLSTMs, perhaps because they encoded information about the probability of these two emojis and when in doubt, the laughing emoji was chosen as more probable.

## 5 Conclusions

Emojis are used extensively in social media, however little is known about their use and semantics, especially because emojis are used differently over different communities (Barbieri et al., 2016a; Barbieri et al., 2016b). In this paper, we provide a neural architecture to model the semantics of emojis, exploring the relation between words and emojis. We proposed for the first time an automatic method to, given a tweet, predict the most probable emoji associated with it. We showed that the LSTMs outperform humans on the same emoji prediction task, suggesting that automatic systems are better at generalizing the usage of emojis than humans. Moreover, the good accuracy of the LSTMs suggests that there is an important and unique relation between sequences of words and emojis.

As future work, we plan to make the model able to predict more than one emoji per tweet, and explore the position of the emoji in the tweet, as close words can be an important clue for the emoji prediction task.

## Acknowledgments

We thank the three anonymous reviewers for their time and their useful suggestions. First and third authors 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

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

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*, pages 349–359, Lisbon, Portugal, September. Association for Computational Linguistics.

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*, pages 326–332, Barcelona, Spain, December.

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*, pages 531–535, Amsterdam, Netherlands, October. ACM.

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*, pages 526–534, Portoroz, Slovenia, May.

David M. Blei. 2012. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, April.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

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*, pages 107–116, Uppsala, Sweden, July. Association for Computational Linguistics.

Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William Cohen. 2016. Tweet2vec: Character-based distributed representations for social media. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 269–274, Berlin, Germany, August. Association for Computational Linguistics.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July. Association for Computational Linguistics.

Bahdanau Dzmitry, Cho Kyunghyun, and Bengio Yoshua. 2014. Neural machine translation by jointly learning to align and translate. In *In Proceeding of the third International Conference on Learning Representations*, Toulon, France, May.

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*, pages 48–54, Austin, TX, USA, November. Association for Computational Linguistics.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June. Association for Computational Linguistics.

Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2015a. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, Denver, Colorado, May–June. Association for Computational Linguistics.

Wang Ling, Chris Dyer, Alan W. Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015b. 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*, pages 1520–1530, Lisbon, Portugal, September. Association for Computational Linguistics.

Nikola Ljubešic and Darja Fišer. 2016. A global analysis of emoji usage. In *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 82–89, Berlin, Germany, August. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June. Association for Computational Linguistics.

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

Hannah Miller, Jacob Thebault-Spieker, Shuo Chang, Isaac Johnson, Loren Terveen, and Brent Hecht. 2016. "Blissfully Happy" or Ready to Fight: Varying Interpretations of Emoji. In *In Proceeding of the International AAAI Conference on Web and Social Media (ICWSM)*, pages 259–268, Cologne, Germany, July. AAAI.

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

Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*, pages 111–120, Beijing, China, April. ACM.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Proceeding of the conference on Neural Information Processing Systems*, Montreal, Canada, December.

Hanna M. Wallach. 2006. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 977–984, Pittsburgh, USA, June. ACM.

Peilu Wang, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao. 2016. Learning distributed word representations for bidirectional lstm recurrent neural network. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 527–533, San Diego, California, June. Association for Computational Linguistics.