It's complicated 🎄

Matija Bertović, Antun Magdić, Ante Žužul

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{matija.bertovic,antun.magdic,ante.zuzul2}@fer.hr

Abstract

Abstract...

1. Introduction

2. Related work

3. Dataset

We frame the task of emoji prediction as a supervised learning task. Each example is made of a tweet labelled by the emoji it contains which is removed from the tweet body as in (Barbieri et al., 2017).

We gathered ten million tweets from the period between November 1, 2018 and December 31, 2018. From those tweets we extracted only the ones which contain a single emoji. In the final dataset we keep only the tweets where one of the 20 most frequent emojis occurs. We split the data in train, validation and test sets containing 120 000, 40 000 and 40 000 tweets, respectively. Classes in all sets are perfectly balanced[1].

4. Models and Representations

We experimented with various models. Different models use different input representations which include binary bag of words vectors, TF-IDF vectors (Manning et al., 2008), as well as100 dimensional GloVe word embeddings pretrained on Twitter data (Pennington et al., 2014).

In the following subsections $\hat{y}$ is used to denote the predicted class, and $\mathcal{Y}$ is used to denote the set of all classes. The classes are labelled by integers ranging from 1 to 20, so $\mathcal{Y} = \{1, 2, 3, \ldots, 20\}$.

4.1. Naïve Bayes

Naïve Bayes (Manning et al., 2008) is a probabillistic model for classification. It takes advantage of the Bayes rule to compute the probability

$$P(y|\mathbf{x}) = \frac{\mathcal{L}(y|\mathbf{x})P(y)}{P(\mathbf{x})},$$

where y is the class label, $\mathbf{x}$ is the example to be classified and $\mathcal{L}$ is the likelihood function. Example is then assigned to the class $\hat{y}$ with the highest probability:

$$\hat{y} = \operatorname*{argmax}_{y \in \mathcal{Y}} P(y|\mathbf{x}).$$

When using this model, we represent each tweet with a binary bag of words vector and we use multivariate Bernoulli distribution as the likelihood function, where we make the naïve assumption of conditional independence of words in a tweet, given the tweet's class label.

[1] ... as all things should be.

4.2. Logistic regression

Logistic regression (Murphy, 2012) is a simple discriminative model. We train a logistic regression classifier for each class. The output of the classifier trained for the class y is the predicted probability that the given example belongs to the class y. The probability is given by

$$P(y|\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^\top \mathbf{x} + b)}},$$

where $\mathbf{w}$ and b are learned parameters. We then use OVR strategy (Bishop, 2006) to make the final classification. Class with the maximum predicted probability is assigned to the input example, that is

$$\hat{y} = \operatorname*{argmax}_{y \in \mathcal{Y}} P(y|\mathbf{x}).$$

We use this model with two different input representations: TF-IDF vectors and mean vectors of GloVe word embeddings of all the words in the tweet. In both cases we set the regularization parameter to 0.1.

4.3. Feed forward neural network

Neural networks have shown to be strong performers at solving various problems, so we also use them for the task of emoji prediction.

We train two feed forward neural networks. One uses TF-IDF vector of a tweet as the input representation, while the other uses the mean vector of GloVe word embeddings of all the words in the tweet.

We use one hidden layer with size 100 in the network with TF-IDF input representation and we use three hidden layers with sizes 150, 100, 50 in the network with mean GloVe input representation. We set the regularization parameter to 10^{-5} for both networks.

4.4. LSTM

A class of neural networks that performs remarkably well on NLP tasks are recurrent neural networks. Hence, we also use a Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997).

LSTM is a type of recurrent neural network that is able to capture long-term dependencies. Fully-connected layer is added after the LSTM cell to map the output of the LSTM cell to the vector of class logits. The final output of the network, i. e. the predicted class $\hat{y}$, is the class with the highest logit value:

$$\hat{y} = \operatorname*{argmax}_{y \in \mathcal{Y}} (\mathbf{W}\mathbf{o} + \mathbf{b})_y,$$

Table 1: Accuracy various models on test data.

Model	Accuracy (%)
NB	0
LR GloVe	0
LR TF-IDF	0
NN GloVe	0
NN TF-IDF	0
LSTM	0

where $\mathbf{o}$ is the output of the LSTM cell and $\mathbf{W}$ and $\mathbf{b}$ are learned parameters of the fully-connected layer. y is used to index the output vector of logits, so y for which the highest logit is obtained is selected as the predicted class.

Two bidirectional LSTM (BLSTM) layers with hidden state size of 300 are used in the LSTM cell. A single bidirectional LSTM layer is composed of two standard LSTM layers, where one is processing the input sequence from the first word to the last word and the other is going the opposite way. This way, both past and future context is available at every time step. Both of those layers' outputs are then concatenated into a single output vector of size 600. After the first BLSTM layer, a dropout layer with dropout probability 0.2 is used. In the end, a fully-connected layer with output size of 20 is used, because there are 20 different classes.

Parameters are optimized using ADAM (Kingma and Ba, 2014) with the initial learning rate of 10^{-3}. The model is trained for 20 epochs over the train set with the batch size of 32.

Each input tweet is represented by a sequence of GloVe word embeddings.

5. Results

We run two experiments. In the first experiment we compare various models and their performances on the task of emoji prediction and in the second we try to gain some insight about the use of emojis in tweets and the difficulties in their prediction.

5.1. Experiment 1

5.2. Experiment 2

In the second experiment...

6. Conclusions

References

Francesco Barbieri, Miguel Ballesteros, and Horacio Saggion. 2017. Are emojis predictable?

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80, 12.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge university press.

Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.