MATH 6380P Advanced Topics in Deep Learning: EmoContext

Andrea Madotto, Genta Indra Winata, Zhaojiang Lin, Jamin Shin
{amadotto,giwinata,zlinao,jmshinaa}@connect.ust.hk

Abstract

In this project, given a textual dialogue, i.e., a user utterance along with two turns of context, the system has to classify the emotion of the last user utterance from one of the following emotion classes: *Happy*, *Sad*, *Angry*, and *Others*. The training dataset contains 15K records for all emotional classes, (i.e., Happy, Sad, and Angry combined) and another 15K sample for not belonging to any of the aforementioned emotion classes (Others). To solve this challenge, we benchmark several features based classifiers, end-to-end solutions, and we tried an automatic hyper-parameter search using Gaussian Processes.

1 Introduction

Many of us have gone through terrible customer service experience at least once and often get emotional during it. Even human operators who are trained to deal with such situations often struggle to do so, partly because of their own emotions. For obvious reasons, neither do automated systems succeed in such scenarios. *What if we could teach machines how to react under these emotionally stressful situations of dealing with angry customers?*

This SemEval2018 shared task aims to bring more research to the first part of the above problem, teaching machines to be empathetic, namely contextual emotion detection in the text. Given a textual dialogue with two turns of context, the system has to classify the emotion of the next utterance into one of the following emotion classes: Happy, Sad, Angry, or Others. The training dataset contains 15K records for emotion classes, i.e., Happy, Sad and Angry combined, and contains 15K records not belonging to any of the aforementioned emotion classes. An example of the dataset is shown in Table 1.

The most naive first step would be to recognize emotion from a given flattened sequence, which has been researched quite extensively despite the very abstract nature of emotion (Socher et al., 2013; Felbo et al., 2017a; McCann et al., 2017; Xu et al., 2018). However, these *flat* models do not work very well on dialog data as we have to concatenate the turns and flatten the hierarchical information merely. Not only the sequence gets too long, but also the hierarchy between sentences will be destroyed (Hsu and Ku, 2018; Kim et al., 2018). We believe that the natural flow of emotion exists in dialogs and using such hierarchical information will allow us to predict the last utterance emotion better.

Naturally, the next step is to be able to detect emotion with a hierarchical structure. To the best of our knowledge, this task of extracting emotional knowledge in a hierarchical setting has not been extensively explored in the literature yet. Therefore, in this project, we to investigate this problem in depth with several strong hierarchical baselines, but with an emphasis on the novel application of *Transformers* (Vaswani et al., 2017; Dehghani et al., 2018), as we hypothesize that these self-attentive models are well suited for modeling hierarchical sequential structures.

2 Methodology

We mainly try two approaches for this task: 1) Feature-based, and 2) End-to-End. The former compares several well-known pre-trained embeddings including *GloVe* (Pennington et al., 2014), *ELMo* (Peters et al., 2018), *BERT* (Devlin et al., 2018), etc. We combine these pre-trained features with a simple Logistic Regression model as

Table 1: The table shows the example dialogues for 4 different classes. Notice that the prediction is refereed to the last User 2's responce.

# True Label	User 1's	User 2's	User 1's
1 Angry	get lost	I know you guys want to loose to me always.	I don't want to talk u any more
2 Sad	I don't read books	Reading is for rich people	But I am poor
3 Happy	Very good chocolate	I prefer orange flavoured chocolates	I like this chocolate
4 Others	Don't worry,I'm girl	hmm how do I know if you are	What's ur name?

the classifier to compare the effectiveness of them. The latter approach is to train a model fully end-to-end with back-propagation. We mainly compare the performances of *Flat* models and *Hierarchical* models, which also take into account the sequential turn information of dialogs.

2.1 Feature-based Approach

The pre-trained feature-based approach can be subdivided into two categories: 1) Word Embeddings pre-trained only on semantic information, and 2) Emotional Embeddings that augment Word Embeddings with emotional/emoji information. We also examine the use of both categories in conjunction.

Word Embeddings include the standard pre-trained non-contexutalized GloVe (Pennington et al., 2014), the contextualized embeddings from biLSTM language model ELMo (Peters et al., 2018), and the more recent *Transformer* based embeddings from bidirectional language model BERT (Devlin et al., 2018).

Emotional Embeddings refer to two types of features equipped with emotional knowledge. The first is word embeddings augmented with emotional information, namely *DeepMoji* (Felbo et al., 2017b), which utilizes a biLSTM with attention model to predict emoji from the text. Similarly, Xu et al. (2018) (*Emo2Vec*) uses a CNN model to predict emotion classes obtained with distant supervision using emojis from tweets. Finally, we also use *Emoji2Vec* (Eisner et al., 2016) which directly maps emojis to continuous representations.

2.2 End-to-End Approach

We mainly consider four models for this task: fine-tuning ELMo (Peters et al., 2018), Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), Universal Transformers (UTRS) (Dehghani et al., 2018), and fine-tuning BERT (Devlin et al., 2018).

For ELMo and BERT, which are pre-trained models, instead of individually extracting the embeddings for each word, we take the entire model as an encoder and attach to a new Softmax layer, and back-propagate the error signals for fine-tuning. While LSTM is the widely known model used almost ubiquitously in the literature, UTRS is a recently published recurrent extension of the multi-head self-attention based model, *Transformers* (Vaswani et al., 2017).

Finally, for all models, we consider the hierarchical extension which considers the turn information as well. We add another instance of the same model to also encode sentence-level information on top of the word-level representations.

Gaussian Processes (GP) are widely used Bayesian optimization techniques used to optimized unknown functions, especially, in functions where the evaluation of a sample is expensive. This is precisely the case of a hyper-parameter search for deep learning models; indeed a single configuration requires to run for several epoch before converging, which may take days. Thus, we model hyper-parameter search with a GP where the function f to optimize gets as input a set of hyper-parameter, both continuous and discrete, and it returns the validation set F1 score after training.

We used a standard Expected Improvement (EI) (Jones et al., 1998) acquisition function with 0.05 jitter. We explore the tuning of Hierarchical Universal Transformer since is the model with the most hyper-parameters to tune, 12 including dropout in the different part of the model and architectural design such hidden size and number of layers (i.g. hops). The type of variable and their range are shown in Table 2.

3 Evaluation

In this section, we present our evaluation metrics used in the experiment, followed by results on

Table 2: Selected hyper-parameters used for the GP search

Name	Type	Range
learning rate	cont.	(0.00001, 0.005)
input dropout	cont.	(0.0, 0.3)
layer dropout	cont.	(0.0, 0.3)
attention dropout	cont.	(0.0, 0.3)
relu dropout	cont.	(0.0, 0.3)
embeddings dim	disc.	(60, 500)
layers	disc.	(1, 10)
heads	disc.	(1, 10)
depth key	disc.	(20, 80)
depth val	disc.	(20, 80)
filter	disc.	(60, 300)
batch size	disc.	(32, 64)

Table 3: The table shows the F1 score on Logistic Regression with different features.

Feature	F1
Deepmoji	**0.687**
ELMo	0.617
GLoVe	0.547
Emo2Vec	0.501
BERT	0.38
Emoji2Vec	0.31
ELMo + Emo2Vec	0.599
Emoji2Vec + GLoVe	0.573

feature-based, end-to-end approaches and Gaussian Process Search. We run 10-fold cross validation by randomly partition the dataset to form training and validation set.

3.1 Evaluation Metrics

The task is evaluated with micro F1 score for the three emotion classes i.e. Happy, Sad and Angry, and by taking the harmonic mean of the precision and the recall.

$$P = \frac{\sum_i TP_i}{\sum_i (TP_i + FP_i)} \forall i \in \{H, S, A\} \quad (1)$$

$$R = \frac{\sum_i TP_i}{\sum_i (TP_i + FN_i)} \forall i \in \{H, S, A\} \quad (2)$$

$$F1 = 2\frac{P \times R}{P + R} \quad (3)$$

This scoring function has been provided by the challenge organizers.

Table 4: The table shows F1 score on flat and hierarchical End-to-end models

Model	Flat	Hierarchical
ELMo	0.543 ± 0.018	0.713 ± 0.014
LSTM	0.666 ± 0.017	0.717 ± 0.029
UTRS	0.674 ± 0.023	0.739 ± 0.018
BERT	0.685 ± 0.021	0.703 ± 0.018
LSTM+GLoVe	0.692 ± 0.02	0.74 ± 0.019
BLSTM+GLoVe	$\mathbf{0.699 \pm 0.031}$	$\mathbf{0.749 \pm 0.009}$
UTRS+GLoVe	0.669 ± 0.034	0.736 ± 0.024
UTRS+GP	-	0.736 ± 0.015

3.2 Experimental Results

For feature-based approach, we run logistic regression models using Scikit-Learn toolkit (Pedregosa et al., 2011). For time constraint reason, we could not try more complex classifiers (e.g., SVM, XGboost) since the feature space and the number of the samples are pretty large (1K and 30K). We report the average of the f1-score from all splits as the measure since the test set labels have not been released yet. From Table 3, we can see that Deepmoji outperforms other features with a large margin. Indeed, Deepmoji has been trained using large emotional corpus, which is compatible with the current task. Emoji2Vec get a very low F1-score because it only has emojis, and indeed by adding GLoVe, a more general embedding, we achieve better performance. For end-to-end approach, a hierarchical bidirectional LSTM (BLSTM) with GLoVe word embedding achieves the highest score with 0.749 F1-score. We achieve similar results with a unidirectional LSTM and UTRS. Adding a pre-trained provide a small improvement except for UTRS which does not gain any performance. In general, hierarchical models are better than flatten models because they can capture both in-sentence and cross-sentence information of the history and the query (also defined as User 1's response).

3.3 Gaussian Processes

We implement the GP model using an existing library called GPyOpt[1], which provides high-level function call for Bayesian Optimization especially GPs. We run the GP for 100 iterations giving as a starting point the hyper-parameter used in Table 3. We use a Hierarchical Universal Transformer as a base model and a single split for estimating the

[1] http://sheffieldml.github.io/GPyOpt/

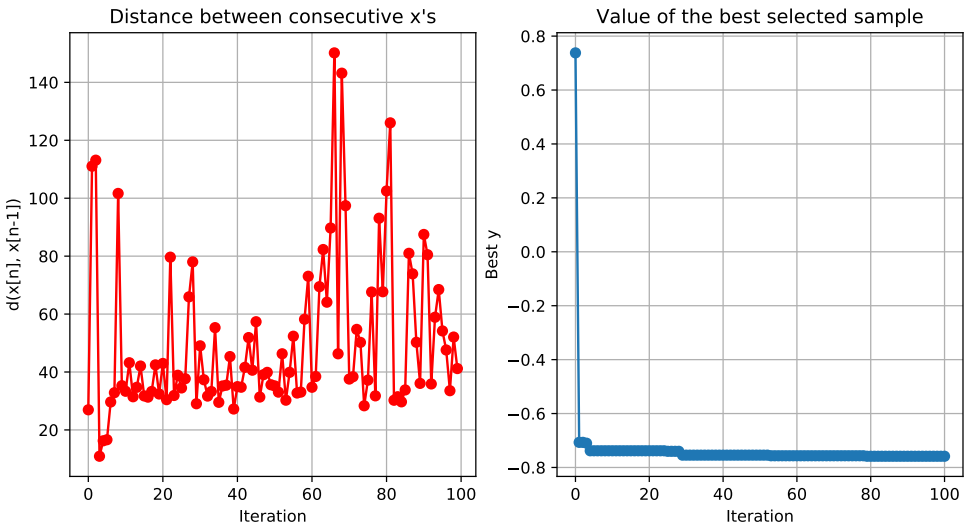

Figure 1: GP convergence.

model F1. The best hyper-parameter we achieve is 0.758 F1-score in the single split. However, when we use this hyper-parameter in the ten-fold, the average F1 is 0.7357, which basically it very close to the starting point. Interestingly, the GP finds a model with a single layer and the maximum number heads in the attention (i.e., 10), which is very different from the starting point which has six layer and few attention heads. On the other hand, dropout is always kept very low, so it seems that GP tried to regularize the model by reducing layer numbers and by increasing the number of heads. Figure 1 shows the F1-score during GP iteration.

4 Conclusion

In this project, we compare different pre-trained word embedding features by using logistic regression along with flat and hierarchical architectures trained in an end-to-end manner. We further explore the Gaussian Processes for faster hyper-parameter search. Our experiments show that hierarchical architectures give significant improvements and we further gain higher accuracy by combining the pre-trained features with end-to-end models.

References

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2018. Universal transformers. *arXiv preprint arXiv:1807.03819* .

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

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.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017a. 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*. pages 1615–1625.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017b. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *2017 Conference on Empirical Methods in Natural Language ProcessingEmpirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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

Chao-Chun Hsu and Lun-Wei Ku. 2018. Socialnlp 2018 emotionx challenge overview: Recognizing emotions in dialogues. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*. pages 27–31.

Donald R Jones, Matthias Schonlau, and William J Welch. 1998. Efficient global optimization of ex-

pensive black-box functions. *Journal of Global optimization* 13(4):455–492.

Yanghoon Kim, Hwanhee Lee, and Kyomin Jung. 2018. Attnconvnet at semeval-2018 task 1: Attention-based convolutional neural networks for multi-label emotion classification. In *Proceedings of The 12th International Workshop on Semantic Evaluation*. pages 141–145.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*. pages 6294–6305.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. volume 1, pages 2227–2237.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1631–1642. http://www.aclweb.org/anthology/D13-1170.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. pages 5998–6008.

Peng Xu, Andrea Madotto, Chien-Sheng Wu, Ji Ho Park, and Pascale Fung. 2018. Emo2vec: Learning generalized emotion representation by multi-task training. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. pages 292–298.