XLNet applied to the task of Twitter Emotion Analysis

Gaoussou Youssouf Kebe
Progress Report
mb888814@umbc.edu

Abstract

Emotion analysis is a very active research field in modern NLP. The complex nature of emotions makes it also one of the hardest text classification tasks. In this project, I will attempt a multi-label classification of tweets between 5 basic emotions (anger, joy, fear, surprise and sadness) using a novel language representation model with state of the art performance in text classification: XLNet.

1 Motivation

Emotional analysis is the logical follow-up to sentiment analysis or opinion mining. While sentiment analysis has achieved tremendous results especially in marketing, the potential perks of effective emotion analysis are even more remarkable. A near-future consequence of the progress currently being made in artificial intelligence is that autonomous agents will be expected to interact with non expert users. A major key to successful agent-human interaction is the ability of the agent to understand human affect. Picard (2000) goes so far as to say the Turing test, despite its textual nature, cannot be passed by a machine incapable of "perceiving and expressing emotions" arguing that emotions can be detected from the content and form of a text. NLP is therefore a cornerstone of affective computing and successful emotion analysis in text is a key aspect. The problem is however very complicated and good solutions in literature tend to achieve very average results. (Strapparava and Mihalcea, 2008) and (Alm et al., 2005) are good examples. XLNet (Yang et al., 2019) is a novel transformer model with state of the art performance in text classification including sentiment analysis. Since XLNet has never been used in emotion analysis literature, one of the major goals of this project is to evaluate its performance in this complex task.

2 Proposed solution

2.1 Emotion modeling

A major detail in emotion classification is the choice of the emotion model. A popular model in literature is the Ekman model (Ekman, 1999) of six basic emotions: anger, disgust, fear, happiness, sadness, and surprise. The classification task in this project will be done according to the Ekman model.

2.2 Dataset

Finding a large enough dataset is a challenge in emotion analysis. I initially intended to use the 7097 tweets dataset proposed in (Mohammad and Bravo-Marquez, 2017). They propose a dataset of tweets labeled not only based on the class of emotion expressed but also the intensity in 0-1 range. For each emotion, they provide hundreds of tweets with different intensities. The annotations were done manually using a technique called best-worst-scaling (BWS) and evaluated using split-half reliability (SHR). But after trying to use the dataset, I noticed they only model fear, joy, sadness, and anger. Follow-up works (Mohammad and Kiritchenko, 2018) on that paper led to a new dataset for SemEval-2018 Task 1: Affect in Tweets (Mohammad et al., 2018). This dataset was annotated using the same BWS technique but it models eleven different emotions including the six I consider in my classification task. The English classification (E-c) portion of the dataset also has a multi-label component which corresponds to the goals of my project. The training, validation and test sets are respectively comprised of 6838, 886 and 3260 sentences. But because of the lack of labels in the test set, the validation set is used as a

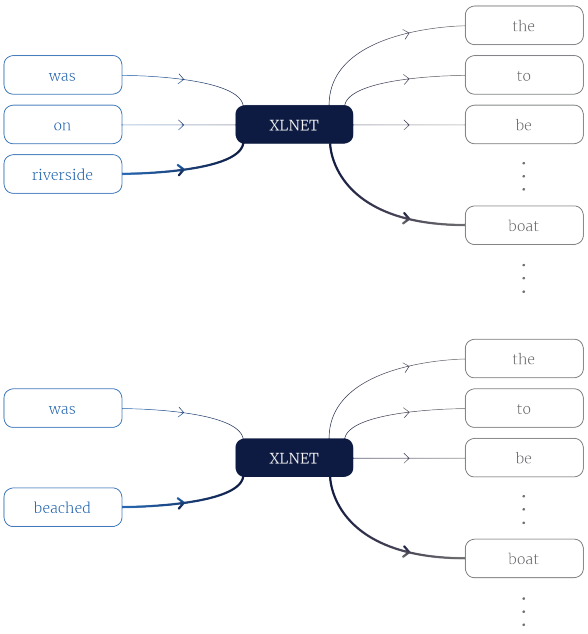

Figure 1: Depiction of the XLNet model. (Simon and Cao, 2019)

test set for this project and 10-fold cross validation is done on the training set for hyper-parameter tuning. The English classification (E-c) files (training, validation) of the dataset are used. Emotions are represented as one hot encoded vectors. A value of 1 means the emotion represented by the column is expressed in the sentence while a value of 0 means the opposite.

2.3 XLNet

Presented in (Vaswani et al., 2017), a transformer is a novel type of neural network developed to solve machine translation like sequence to sequence problems. XLNet (Yang et al., 2019) is one of the latest transformer models and is considered state-of-the-art in many text classification tasks. Similarly to other transformer models, XLNet's language modeling considers non-adjacent tokens to generate the probability of a given token using an attention mechanism. This allows them to learn higher level contextual information that other language modeling techniques cannot. As seen in Figure 1, XLNet takes this concept farther by considering different combinations of the tokens in the sequence.

2.4 Challenges

Emotion analysis is a field of many challenges. Among others, negation handling is a big obstacle to any classification attempt. Like in this project, many emotion analysis tasks deal with raw human generated textual data like tweets. These documents are unstructured and tend to contain typos and other types of syntactical unreliability. Eliminating this noise and cleaning the data to focus on the emotion-related features of the text is also a challenge in emotion analysis. An additional goal of this project consist of experimenting different solutions from literature to face these hurdles and maximize performance.

3 Related work

Emotional analysis is well studied in natural language processing. Hasan et al. (2014) used classic supervised learning algorithms (KNN, Naive Bayes, Decision Tree and SVM) to classify tweets with hashtags based on the circumplex model of emotions introduced by Russell (1980). Strapparava and Mihalcea (2008) implemented different Latent Semantic Analysis (LSA) based methods to classify news headlines according to Ekman's model while, Alm et al. (2005) focused on the problem of classifying sentences from children stories.

Transformers have recently established themselves as the state of the art in many text classification tasks. Consequently, they have been also been used in recent attempts to tackle the emotion analysis problem. Balazs et al. (2018) use a pre-trained ELMo (Peters et al., 2018) model among other components to classify tweets and Luo and Wang (2019) use a pre-trained BERT (Devlin et al., 2018) model to classify dialogues from the TV show Friends and Facebook chat logs. In contrast to my project, both works consider a multi-class classification task.

My work is closest to (Ying et al., 2019) and (Kant et al., 2018) where a pre-trained BERT model and an attention-based transformer respectively are used to classify tweets from the same SemEval-2018 Task 1 dataset used in this project. My approach follows this trend of using transformers for emotion classification by exploiting the recently deployed XLNet (Yang et al., 2019).

4 Methodology

4.1 Pre-processing

The first step of emotion analysis is pre-processing the data. It involves applying a series of techniques which would result in a better performing

	Epochs = 2		Epochs = 3		Epochs = 4	
	Micro	Macro	Micro	Macro	Micro	Macro
Precision	0.778	**0.794**	**0.783**	0.789	0.778	0.769
Recall	0.697	0.609	0.690	0.616	**0.701**	**0.628**
F1 score	0.735	0.649	0.733	0.665	**0.737**	**0.669**

Table 1: Grid search for the number of epochs: 4 epochs result in the optimal value for macro and micro f1 scores

	Batch size = 32		Batch size = 48	
	Micro	Macro	Micro	Macro
Precision	**0.783**	**0.789**	0.778	0.778
Recall	0.690	0.616	**0.696**	**0.618**
F1 score	**0.733**	**0.665**	0.733	0.658

Table 2: Grid search for the batch size: Batches of size 32 result in the optimal value for macro and micro f1 scores

	LR = 2e-05		LR = 3e-05	
	Micro	Macro	Micro	Macro
Precision	**0.778**	**0.769**	0.764	0.747
Recall	0.701	0.628	**0.716**	**0.640**
F1 score	0.737	0.669	**0.738**	**0.671**

Table 3: Grid search for the learning rate: A 3e-05 learning rate results in the optimal value for macro and micro f1 scores

model. In this project the following techniques are applied:

Negation handling: As mentioned earlier, negation handling is a big challenge in emotion analysis and sentiment analysis. Pang et al. (2002) dealt with negation by adding _NOT to all the words succeeding negation words like "not". Similarly in this project, a regex function was used to add the prefix NOT_ to every words in a sentence that comes after negation words (not, never and any word containing the characters "n't").

Links and mentions: A regex function is used to delete all links and mentions as they do not provide significant information.

Stop-words: For similar reasons, commonly used words like pronouns and articles are filtered out as they are of little value in helping discriminate between documents.

4.2 XLNet Fine-tuning

The fine-tuning process was done using pytorch-transformers (Wolf et al., 2019) which provides a pytorch implementation of different transformer models including XLNet. Fine-tuning a transformer model can take a very long time if performed on a CPU. In order to accelerate the process, I used Google Colab notebooks throughout the project.

The XLNetForSequenceClassification model of pytorch-transformers is used for training and evaluation. While the model is perfect for binary and multi-class classification, its inability to take one-hot encoded label vectors prevent it from performing multi-label classification. Using inheritance, a new XLNetForMultiLabelSequenceClassification class is defined to override this light inconvenience. The model takes the following inputs:

- An input matrix containing the features of each sentence.
- A label matrix containing the one-hot encoded label vector of each sentence.
- An attention mask matrix.

The features of a sentence are obtained through tokenization. The tokenization process involves breaking apart the words in a sentence into sub-word units and is done through SentencePiece (Kudo and Richardson, 2018). SentencePiece considers sentences to be series of unicode characters without any sort of language-dependent logic and generates sub-word units that boost the accuracy of the transformer model. [SEP] and [CLS] tokens are added to the end of every sentence to represent beginning and end of sentences. XLNet requires every tokenized feature vector in the input to be of the same size. Therefore, every vector is padded to have a predefined length of 128 tokens. The tokens are then replaced by numerical

	T = 0.3		T = 0.5		T = 0.8	
	Micro	Macro	Micro	Macro	Micro	Macro
Precision	0.692	0.672	0.764	0.747	**0.847**	**0.852**
Recall	**0.771**	**0.687**	0.716	0.640	0.536	0.474
F1 score	0.729	0.662	**0.738**	**0.671**	0.656	0.580

Table 4: Grid search for the probabilistic threshold T: T = 0.5 results in the optimal value for macro and micro f1 scores

values indicating their index. The attention mask input is a matrix of binary vectors representing the sentences where 1 designate an actual token and 0 designate a padding. Traditional deep learning hyper-parameters such as learning rate, batch size and number of training epochs are also defined.

4.3 Classification Strategy

The output of the fine-tuned XLNet model is turned into a probability using a sigmoid function:

$$P_j(i) = \frac{1}{1 + e^{-\beta_{ij}}}, \tag{1}$$

where β_{ij} is the output of the XLNet model for label j and tweet i. A fixed threshold T is used for multi-label classification such that the value of the one-hot encoded prediction vector at label j and tweet i is:

$$Y_{i,j} = \begin{cases} 1 & \text{if } P_j(i) \geq T \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

4.4 Hyper-parameter Tuning

The traditional deep learning hyper-parameters (learning rate, batch size and number of training epoch) and the probabilistic threshold T are tuned. Based on the examples discussed in (Yang et al., 2019), the following values are considered for the learning rate, batch size and number of training epochs:

- Learning rate: 2e-05 or 3e-05
- Batch size: 32 or 48
- Number of epochs: 2, 3 or 4

A high threshold of 0.8, medium threshold of 0.5 and low threshold of 0.3 are considered for T. The hyper-parameters are tuned using grid search and 10-fold cross validation. Table (1-4) display the results of the grid search. When tuning one hyper-parameter, everything else is kept constant. Micro and macro averages of the precision, recall and f1 scores are computed for every model. I consider F1 scores to choose the best model. In Table 1, 4 epochs is shown to be the optimal number of epochs despite the fact that 2 and 3 epochs respectively lead to better macro and micro precision. In Table 2, A batch size of 32 is shown to be optimal while the optimal learning rate is 3e-05 per Table 3.

Table 4 shows that a higher probabilistic threshold (0.8) corresponds to higher precision and lower recall while a lower threshold (0.3) results in higher recall and lower precision which makes intuitive sense. A medium value (0.5) achieves a good compromise between the metrics and results in higher f1-score. Therefore, the best model that is subsequently used for the classification process uses 4 training epochs, batches of size 32, a learning rate of 3e-05 and a probabilistic threshold of 0.5.

	Micro	Macro
Precision	0.795	0.799
Recall	0.723	0.662
F1 score	0.758	0.701

Table 5: Performance metrics when applied to the test set

5 Results

The model is applied to the validation set of the SemEval-2018 Task 1: Affect in Tweets dataset. As mentioned in section 2, I only consider the 6 basic emotions of the Ekman model: anger, disgust, fear, joy, sadness and surprise. (Mohammad et al., 2018). Table 5 displays the performance metrics obtained while Table 6 is the confusion matrix. The model achieves a F1-score of 0.758 which is impressive in a multi-label classification problem as complex as emotion analysis.

	TP	FP	FN	TN
anger	235	53	80	518
disgust	227	69	92	498
fear	97	43	24	722
joy	313	40	87	446
sadness	171	66	94	555
surprise	10	1	25	850

Table 6: Confusion matrix when applied to the test set

The model was particularly efficient in predicting the emotions with more instances in the dataset. It achieves its lowest performance with surprise which had only 26 instances despite an impressive precision of 0.9. I suspect that a more balanced dataset would have led to even more impressive results.

A current limitation of the system is the value of the probability threshold. Setting a global threshold may not be the optimal setting since emotions do not have the same level of prediction difficulty. Learning a different threshold value for the different labels will probably be a more adapted solution. Another limitation is the number of emotions considered. The Ekman model is very popular but one could argue it does not fully represent the scope of human emotions. Overall, I was very impressed by the performance achieved by the model.

6 Future works

Emotion recognition is a multi-modal problem. In addition to textual data, visual and audio features could be included in an interactive emotion recognition task. Recent developments in neural networks and deep learning approaches have greatly advanced the performance of state-of-the-art visual recognition systems. Combining a transformer-based emotion analysis classifier like the one described in this project with a convolutional neural network in a multi-modal emotion recognition task could be an interesting direction for future works.

7 Conclusion

In this project, I applied XLNet, a novel state-of-the-art transformed to the task of multi-label emotion analysis and obtained very encouraging results. The project not only allowed me to explore the field of emotion analysis but it also provided me with valuable experience in using transformer models which are set to be a dominant NLP deep learning architecture for the next few years.

References

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 579–586, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Jorge A. Balazs, Edison Marrese-Taylor, and Yutaka Matsuo. 2018. IIIDYT at IEST 2018: Implicit emotion classification with deep contextualized word representations. *CoRR*, abs/1808.08672.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Paul Ekman. 1999. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16.

Maryam Hasan, Elke A. Rundensteiner, and Emmanuel Agu. 2014. Emotex: Detecting emotions in twitter messages.

Neel Kant, Raul Puri, Nikolai Yakovenko, and Bryan Catanzaro. 2018. Practical text classification with large pre-trained language models. *CoRR*, abs/1812.01207.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Linkai Luo and Yue Wang. 2019. Emotionx-hsu: Adopting pre-trained BERT for emotion classification. *CoRR*, abs/1907.09669.

Saif Mohammad and Felipe Bravo-Marquez. 2017. Emotion intensities in tweets. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 65–77, Vancouver, Canada. Association for Computational Linguistics.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Saif M. Mohammad and Svetlana Kiritchenko. 2018. Understanding emotions: A dataset of tweets to study interactions between affect categories. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference*, Miyazaki, Japan.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

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)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Rosalind W Picard. 2000. *Affective computing*. MIT press.

James A Russell. 1980. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161.

Prince Simon and Yanshuai Cao. 2019. Research.

Carlo Strapparava and Rada Mihalcea. 2008. Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 1556–1560, New York, NY, USA. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Wenhao Ying, Rong Xiang, and Qin Lu. 2019. Improving multi-label emotion classification by integrating both general and domain-specific knowledge. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 316–321, Hong Kong, China. Association for Computational Linguistics.

drive