Multi-Label Emotion Classification in English Poetry using Song Lyrics and a Dual Attention Transfer Mechanism

Yash Deshpande
NYU CDS
yd1282@nyu.edu

Philip Ekfeldt
NYU CDS
kae358@nyu.edu

Micaela Flores
NYU CDS
mrf444@nyu.edu

Tony Xu
NYU CDS
tx507@nyu.edu

Abstract

In this paper, we attempt to perform multi-label emotion classification on English poetry. This domain presents a number of challenges, chief among which are the subjectivity involved in emotion annotation and the apparent lack of annotated poetry data. We employ a dual attention transfer mechanism and augment emotion learning in poetry by using sentiment learned from song lyrics.

1 Introduction

As online platforms where users share their personal opinions grow in popularity, emotion detection has the potential to be a powerful tool in domains such as social media, political science, marketing, and human-computer interaction. Our intention is to learn and yield accurate emotion classifications as understanding people's emotions can help improve the performance of personalized recommendations and services. Previous literature (Yu et al., 2018; Kim et al., 2018; Baziotis et al., 2018) predominantly makes use of Twitter data. Instead, we choose to use emotion-annotated poetry data, as poetry represents a domain of textual data that has significantly more emotional nuance.

Deep learning methods require large annotated data sets, and we were hence presented with the problem of finding sufficient emotion-annotated poetry data. Extensive efforts to find such a data set were unsuccessful. Subsequently, the annotation task was outsourced to Amazon Mechanical Turk (Crowston, 2012).

Using a bi-directional LSTM model to establish a baseline, we proceed to implement a dual attention transfer mechanism (Yu et al., 2018). We address the lack of emotion-annotated poetry data by using sentiment-annotated song lyrics data to augment the effectiveness of the emotion classification model.

2 Literature Review

Sentiment classification of textual data has been extensively studied and benchmarked, and its predecessors predate even the advent of computing. Stine (2019) notes that the modern age of sentiment analysis began in the 2000s with the explosive growth of online reviews, blogs, and social media. The effect of product reviews on sales brought a commercial need to monitor posts on the Internet, and the scope of the problem implied that the solution needed to be automated.

Modern sentiment classification techniques exploit deep learning and predictive models built from large neural networks to develop a contextual understanding of the text. As focus has predominantly shifted to deep learning, the demand for large annotated data sets has grown. Various data sets have been created to address this demand, such as the IMDb movie review data set and the Music Mood Data set. Çano and Morisio (2017) identify four standards for these data sets: they must be (1) highly polarized to serve as ground truth, (2) labeled using a popular mood taxonomy, (3) large (at least 1000 observations), and (4) publicly available. Based on these standards, they utilize Playlist, Million Song Dataset, and *last.fm* user tags to create two data sets of song lyrics and corresponding emotion labels. For the purposes of this paper, we utilize a down-sampled version of the second data set – a collection of songs annotated as positive or negative.

Moreover, the classification accuracy of neural networks has steadily improved over time. In 2011, Maas et al. (2011) correctly classified 89% of the IMDb test cases. By 2016, Miyato et al. (2016) achieved an accuracy of 94% on the benchmark IMDb data set. As better models and corpora are developed, classification accuracy will continue to improve until it becomes comparable to human readers.

Emotion detection and classification in text, while relatively under-explored, has gained popularity in recent years. Seyeditabari et al. (2018) suggest this may be due to the amount of insight that can be obtained in moving beyond naive sentiment classification and instead focusing on emotion classification. In particular, they underscore how two emotions might convey the same positive or negative sentiment, yet evoke completely different emotional responses. This information can be used to further improve the accuracy and general effectiveness of feedback models and recommendation systems across various social, political, or relational disciplines.

As is discussed in Yu et al. (2018), current methodologies for emotion classification include lexicon-based methods, graphical model-based methods, linear classifier-based methods, neural network models, and attention mechanisms. As we have highlighted in prior sections, the main drawback to these methods is that they rely substantially on large sets of annotated data. These data sets are often costly to obtain and not widely available, especially in our domain of choice: poetry.

Furthermore, Alsharif and Ghneim (2013) suggest the lack of previous work with emotion classification in poetry may be due to how models trained on language tokens often find it challenging to capture the complex emotional language structures that are so inherent to poetry. They note that emotion classification is particularly absent within Arabic poetry, as old poets tended not to display explicit emotion in their text. Though we deal with English poetry, we look to further the limited exploration of emotion classification in poetry, providing a richer platform for future applications and analysis involving other languages.

To address the lack of large data sets annotated with emotion labels, Yu et al. (2018) propose a dual attention transfer mechanism, which aims to improve the performance of multi-label emotion classification models by utilizing information from a sentiment classification task on a similar data set. The mechanism splits the multi-label classification problem into two tasks. The *source* task is trained on a sentiment-annotated data set, and the *target* task is trained on a smaller, emotion-annotated data set. Whereas Yu et al. (2018) train both tasks on Twitter data, our model utilizes sentiment-annotated song lyrics to train the source model.

3 Methodology

3.1 Data

We use a set of 4900 sentiment-annotated song lyrics, scraped from Genius.com as per Çano and Morisio (2017). This data set is balanced in that there is an equal number of positive and negative song labels. We require that the poetry data used be similar in language to song lyrics. For example, poetry that makes use of archaic English would not benefit from sentiment learned on songs, as these make use of modern English. Hence, we use a set of 700 poems scraped from /r/OCPoetry, a Reddit community where users can post original poetry for feedback. Mechanical Turk workers were asked to annotate these poems with 2 emotion labels among a set of 8 emotions: *anger, anticipation, fear, joy, love, optimism, pessimism,* and *sadness*.

The poetry data is subject to a training/development/test split of 0.64 / 0.16 / 0.20. The lyrics data is subject to a train/development split of 0.80 / 0.20.

3.2 Baseline model

To establish baseline results, we use a single-layer bi-directional LSTM with attention, mirroring the baseline model and attention mechanism described by Yu et al. (2018). The words in the poem are processed sequentially without any separating tokens between lines or verses (i.e. the *form* of the poem is not learned):

$$\overrightarrow{\mathbf{h}_j} = \mathrm{LSTM}\left(\overrightarrow{\mathbf{h}_{j-1}}, \mathbf{x}_j, \Theta_f\right) \tag{1}$$

$$\overleftarrow{\mathbf{h}_j} = \mathrm{LSTM}\left(\overleftarrow{\mathbf{h}_{j+1}}, \mathbf{x}_j, \Theta_b\right) \tag{2}$$

where Θ_b and Θ_f represent trainable parameters and $x_j \in \mathrm{I\!R}^{300}$ is the GloVe embedding (Pennington et al., 2014) of the j-th word. The two hidden outputs for each word are then concatenated to create a representation $\mathbf{h}_j = [\overrightarrow{\mathbf{h}_j} : \overleftarrow{\mathbf{h}_j}]$ for each word.

The attention is then calculated as follows, and the output of the attention layer, $\mathbf{H}$, is a representation of the full text of the poem.

$$u_j = \mathbf{v}^\top \tanh\left(\mathbf{W}_h \mathbf{h}_j + \mathbf{W}_z \mathbf{h}_\mathbf{n}\right) \tag{3}$$

$$\alpha_j = \frac{\exp\left(u_j\right)}{\sum_{l=1}^{n}\exp\left(u_l\right)} \tag{4}$$

$$\mathbf{H} = \sum_{j=1}^{n} \alpha_j \mathbf{h}_j \tag{5}$$

where $\mathbf{v}$, $\mathbf{W_h}$, and $\mathbf{W_z}$ are trainable parameters and $\mathbf{h_n}$ is the hidden output of the last word in the poem. $\mathbf{H}$ is then fed through a dropout layer into a multi-layer perceptron (MLP) with one hidden layer and dropout. Softmax is used on the MLP output to produce the final output.

We use KL divergence loss to train the baseline model, as described by Yu et al. (2018):

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbf{e}_k^{(i)}\left(\log\left(\mathbf{e}_k^{(i)}\right) - \log\left(\mathbf{o}_k^{(i)}\right)\right) \tag{6}$$

where $\mathbf{e}_k^{(i)}$ is the true label value for emotion k and sentence i, and $\mathbf{o}_k^{(i)}$ is the predicted probability.

3.3 Dual Attention Transfer Network (DATN)

Described by Yu et al. (2018), the dual attention transfer mechanism is a transfer learning model that makes use of sentiment classification to augment the emotion classification task.

Recall the DATN is made up of two tasks, henceforth referred to as the *source* task and the *target* task. Both task models have the same base structure as that of the baseline, i.e. a bidirectional LSTM module.

The source model is trained on song lyrics, annotated as expressing positive or negative sentiment. The source attention is calculated as described above in Equations 3 and 4. After training, the weights learned by the source model are frozen.

The target model is trained on the poetry data, annotated by emotions. The poetry data is propagated through both the source and target models; the source model utilizes the weights frozen earlier, whereas the target model learns specifically on the poems. As a result, attention for the poetry data is calculated in both networks. The attention calculation in the target model, however, incorporates the source attention as follows:

$$u_j^t = \mathbf{v}^{t^\top}\tanh\left(\mathbf{W}_h^t\mathbf{h}_j^t + w_\alpha\alpha_j^s + \mathbf{W}_z^t\mathbf{z}^t\right) \tag{7}$$

$$\alpha_j^t = \frac{\exp\left(u_j^t\right)}{\sum_{l=1}^{n}\exp\left(u_l^t\right)} \tag{8}$$

$$\mathbf{H_t} = \sum_{j=1}^{n}\alpha_j^t\mathbf{h}_j^t \tag{9}$$

The representations from the source and target models are then concatenated to produce a final representation of the poem, $\mathbf{H} = [\mathbf{H_s} : \mathbf{H_t}]$. A multi-layer perceptron with a single hidden layer is then applied on top of $\mathbf{H}$ and Softmax is applied its output. The emotion labels with the two highest probabilities are then selected as the model outputs.

The final objective function then incorporates the loss described in Equation 6, the sentiment label y_m, and the cosine similarity between the source and target attention vectors α_s and α_t.

$$\begin{aligned}\mathcal{J} = &-\frac{1}{M}\sum_{m=1}^{M}\log p\left(y_m|\mathbf{H_c}\right) + \mathcal{L} \\ &+ \lambda\sum_{i=1}^{N}\cos\mathrm{sim}\left(\alpha_i^s, \alpha_i^t\right)\end{aligned} \tag{10}$$

3.4 Hyper-parameter tuning

As suggested by Bergstra and Bengio (2012), hyper-parameter tuning is carried out using 60 distinct randomized combinations. We tune the hidden layer dimension *(h_dim)*, dimensions of v and v^t *(v_dim)* (Equations 3 and 7), dropout rate *(dropout)* and learning rate *(*η*)*. λ (Equation 10) is set to 0.05, per Yu et al. (2018).

Table 1 reports the hyper-parameter ranges tested. Tables 2 and 3 report the optimal hyper-parameter values for the baseline model and the DATN respectively. The optimization is carried out separately for the source and target models, with *h_dim* consistent between the two.

Parameter	**Min.**	**Max.**
h_dim	50	300
v_dim	5	30
dropout	0.2	0.8
$log_{10}(\eta)$	-4	-2

Table 1: Hyper-parameter ranges

Parameter	Opt.
h_dim	199
v_dim	8
dropout	0.697
$log_{10}(\eta)$	-3.224

Table 2: Hyper-parameter optima (baseline model)

Parameter	Opt. (source)	Opt. (target)
h_dim	183	183
v_dim	21	10
dropout	0.351	0.715
$log_{10}(\eta)$	-3.836	-3.128

Table 3: Hyper-parameter optima (DATN)

3.5 Evaluation

The metric used to evaluate model performance is accuracy, based on the two emotion labels with the highest output probabilities ($top - 2$).

This deviates from the methodology used by Yu et al. (2018) – they utilize a threshold instead of a $top-k$ method, and the number of labels predicted by their model is variable.

4 Results

Table 4 reports the results for both the baseline and DATN models.

Model	Dev. accuracy	Test accuracy
Random	25%	25%
Baseline	46.43%	40.43%
DATN	34.82%	30.31%

Table 4: Results

A random selection of emotion labels results in an accuracy of 25%. The baseline model performs slightly better, achieving a test accuracy of 40.43%.

It is observed that incorporating the dual-attention transfer mechanism does not result in an improvement in classification performance as measured by accuracy. On the contrary, the DATN performs significantly worse than the baseline, achieving a test accuracy 30.31%.

5 Analysis

There are several plausible explanations to the poor performance of both the baseline model and the DATN. Due to the lack of poetry data annotated with either sentiment or emotion, emotion annotations were crowd-sourced using Amazon Mechanical Turk – MTurk (Crowston, 2012). This, however, produced incorrect or badly annotated data. On inspection, we found that MTurk *workers* either (1) completely ignored the explicit directions to annotate each poem with exactly two emotions, or (2) annotated each poem with an obviously incorrect emotion. Due to this, a significant amount of data needed to be re-annotated manually. It should be noted that MTurk does enable users to both cross-reference responses for a single piece of text from multiple *workers* and pre-set *worker* qualifications (education, demographic, etc.). However, we were unable to utilize these services due to budget constraints.

We were able to effectively restrict both data sets to modern English. The songs in the lyrics data set belong to the last five decades, and the choice of publicly available poetry from /r/OCPoetry was motivated by this factor. However, the differences in language and dialect between songs and poetry could have affected the performance of the DATN. For example, the meaning of a word in the context of a song may be different from it's meaning in the context of a poem; sentiment learned from a song may add noise to the emotion learned from a poem instead of additional information.

The metaphorical nature of language common to our choice of domains also makes learning any objective information from the text difficult. Specifically, the emotions that both songs and poems evoke very likely differ across individuals.

Another possible deterrent to model performance are the excessively small data sets – deep learning models typically require large amounts of data to be effective

6 Conclusion

The dual attention transfer mechanism does not seem to improve the performance of multi-label emotion classification on poetry data. Further research could explore building a larger annotated data set, training GloVe embeddings (Pennington et al., 2014) specific to the data, and making use of BERT (Devlin et al., 2018) embeddings, which have been shown to extract contextual complexities.

Code

The code used thus far can be found in this GitHub repository.

Collaboration statement

This project was a challenge to every member of our team, in different respects. Philip, familiar with web-scraping before this project, guided the extraction and pre-processing of the songs data set. Tony and Yash, familiar with Reddit and its intricacies, scraped the poetry data and set up the emotion annotation task on Amazon Mechanical Turk. Both the bi-directional LSTM model and the DATN were extensively discussed as a group. Micaela pre-processed and batchified the cleaned data using the Torchtext package, and Philip and Yash implemented the Bi-LSTM and DATN models using the PyTorch package.

Combing through research papers was an exhaustive task due to its recursive nature, and Tony and Micaela were able to extract the most relevant information from the papers we cited while maintaining concision. Finally, the effort to put together this draft was collective, with multiple passes to ensure the flow of logic and consistency in tense and grammar, and to abide by the limitations imposed by the guidelines and rubric.

References

Ouais Alsharif and Nada Ghneim. 2013. Emotion Classification in Arabic Poetry using Machine Learning. *International Journal of Computer Applications*, 65(16):975–8887.

Christos Baziotis, Athanasiou Nikolaos, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. 2018. NTUA-SLP at SemEval-2018 Task 1: Predicting Affective Content in Tweets with Deep Attentive RNNs and Transfer Learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 245–255, New Orleans, Louisiana. Association for Computational Linguistics.

James Bergstra and Yoshua Bengio. 2012. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Erion Çano and Maurizio Morisio. 2017. Music Mood Dataset Creation Based on Last FM Tags.

Kevin Crowston. 2012. Amazon mechanical turk: A research tool for organizations and information systems scholars. In *Shaping the Future of ICT Research. Methods and Approaches*, pages 210–221, Berlin, Heidelberg. Springer Berlin Heidelberg.

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.

Genius.com. Genius — Song Lyrics & Knowledge.

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, New Orleans, Louisiana. 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. Association for Computational Linguistics.

Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2016. Adversarial Training Methods for Semi-Supervised Text Classification. *arXiv e-prints*, page arXiv:1605.07725.

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.

/r/OCPoetry. Reddit - original composition poetry.

Armin Seyeditabari, Narges Tabari, and Wlodek Zadrozny. 2018. Emotion Detection in Text: a Review.

Robert A. Stine. 2019. Sentiment analysis. *Annual Review of Statistics and Its Application*, 6(1):287–308.

Jianfei Yu, Luis Marujo, Jing Jiang, Pradeep Karuturi, and William Brendel. 2018. Improving multi-label emotion classification via sentiment classification with dual attention transfer network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1097–1102, Brussels, Belgium. Association for Computational Linguistics.