An Empirical Study of Transformer Based Methods for Intent Classification

Xianlin DING

March 27, 2023

Github : https://github.com/savourdxl/NLP-ENSAE-3A-Topic3

Abstract

Intent classification is a fundamental task in natural language processing that aims to determine the intention or purpose behind a given text utterance. The goal is to classify the text into one or more predefined categories, each representing a specific intent. This task is essential for developing conversational agents, chatbots, and virtual assistants that can understand and respond appropriately to user queries. Various techniques have been proposed for intent classification, including rule-based methods, machine learning-based models, and deep learning-based approaches. In this paper, we explore transformer based technics for intent classificaiton.

1 Introduction

In recent years, natural language processing (NLP) has made significant strides in enabling machines to understand and generate human language. One of the essential tasks when working with spoken text [14, 17] is intent classification, which involves identifying the underlying intention or purpose behind a given text utterance. Intent classification plays a vital role in developing intelligent conversational agents [2, 4], chatbots, and virtual assistants that can understand and respond to user queries accurately [7, 9, 20, 10].

Early methods for intent classification relied on rule-based systems, which were limited by their inability to handle the complexity and variability of natural language [3]. Later, machine learning-based approaches, such as support vector machines (SVMs), decision trees, and random forests, emerged as popular alternatives. These methods require carefully designed feature engineering and hyperparameter tuning and suffer from the curse of dimensionality when working with large vocabularies [6].

Recently, deep learning-based approaches, such as convolutional neural networks [18] (CNNs) and recurrent neural networks (RNNs) [22], have shown significant improvements in various NLP tasks, including intent classification. However, these models still face several challenges, such as handling variable-length input sequences, dealing with long-term dependencies, and capturing contextual information [32].

Transformers, a recent breakthrough in deep learning, have shown remarkable success in NLP tasks, including language modeling, text generation, and machine translation. The transformer architecture, introduced in the landmark paper "Attention Us All You Need"[31], eliminates the need for recurrence and convolutional operations and replaces them with self-attention mechanisms. This design allows transformers to capture long-term dependencies and contextual information more effectively than RNNs and CNNs and enables parallel computation, making training faster.

Several studies have investigated the effectiveness of transformers in intent classification and achieved state-of-the-art performance on various benchmark datasets. For example, [13] introduced BERT (Bidirectional Encoder Representations from Transformers [31]), a pretraining-based transformer architecture that achieved unprecedented performance on several NLP tasks, including intent classification. Similarly, [28] introduced GPT (Generative Pre-trained Transformer), a transformer architecture pretrained on massive amounts of text data, and showed remarkable results on several downstream NLP tasks [5, 7], including intent classification.

Despite these impressive results, several challenges remain in intent classification using transformers. For instance, the performance of these models heavily depends on the quality and size of the

training data, the choice of hyperparameters, and the fine-tuning strategy [26]. Moreover, domain adaptation, data scarcity, and model interpretability still remain open research questions in this field.

In this paper, we present a study of intent classification using transformers, focusing on the effectiveness of BERT and GPT architectures. We compare their performance with traditional machine learning-based methods and deep learning-based models, such as RNNs and CNNs, on several benchmark datasets [19, 23, 21, 1, 25, 30, 27, 29, 24]. Additionally, we investigate the impact of various hyperparameters, fine-tuning strategies, and transfer learning techniques on the performance of the models. Finally, we discuss the limitations and future directions of intent classification using transformers and highlight the potential of this approach in enabling more advanced conversational agents and virtual assistants.

Dataset The data set used for classification is SILICONE, which includes several labelling tasks with both DA and E/S annotations. Specifically, the data is in multiple levels: we have a set of conversations $D = (C_1, C_2, \ldots, C_{|D|})$, where each conversations is composed of utterance $C_i = (u_1, u_2, \ldots, u_{|C_i|})$ and each utterance is a sequence of words. The following table gives us an example of one conversation with DA labels on utterances.

Table 1: Example of conversation with labelled utterances

Utterances	Dialogue Act(DA)
"what do you mean ? it will help us to relax ."	"question"
"good.let ' s go now ."	"directive"
"all right ."	"commissive"

2 Model Description

The basic structure of our model can be described as follows, where basically we first use transformer decoder layers to transform data, then we pass them to the MLP classifier to get the results.

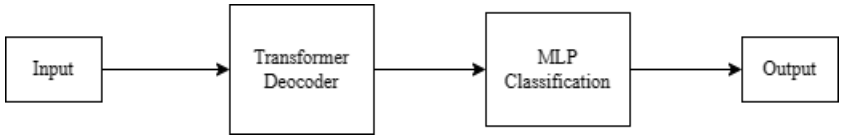

Figure 1: Structure of Model

2.1 Transformer Decoder

The process of using transformer decoder can be divided into 4 steps.

Firstly, we tokenize each utterance. Notice that each utterance is composed of a sequence of words $u_i = (w_{i1}, w_{i2}, ..., w_{ik})$, but the length of words in each utterance is different. In this case, we set the length of words is equal to 20: for the sequence with length more than 20, we delete the words that exceed the length; for the sequence with length less than 20, we fill the blanck with 1. Here, we use FastText (English) from torchtext.vocab.

Secondly, we generate the embedding h_0 on $U = (u_1, u_2, \ldots, u_n)$: $h_0 = U^W * W_e + W_p$, where W_e is the token embedding matrix, W_p is the position embedding matrix and w_i stands for the words.

Thirdly, we transformer the output by using 12 transformer decoder blocks. In mathematical terms, it can be written as $h_l = transformer\_block(h_{l-1}), l = 1, \ldots, 12$. In general, this pre-training can be viewed as the maximization of the function $L_1(C) = \sum_i \log P(u_i|u_{i-k}, \ldots, u_{i-1}; \Theta)$, where Θ refers to the parameters of the language model, and k refers to the length of window.

However, there is a need to notice that the transformer block here is not the usual encoder-decoder one ([16]), but the transformer decoder: masked multi-head self attention + multi-head self attention (the number of heads in our model is equal to 10) + feed forward. In other words, it is a model with single direction (left-to-right transformer), i.e., we can only infer one word from the past words. The Figure 2 shows the structure of this transformer.

In fact, the masked multi-head self-attention is the key for our transformer. The attention of each head without mask is computed as $Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$ ([15]), where Q is the query, K is the key, V is the value and d_k is the key dimension. Hence, the attention of each head with mask is computed as $Masked\_Attention(Q, K, V, M) = softmax(\frac{QK^T+M}{\sqrt{d_k}})V$, where M stands for the mask and anything else is the same.

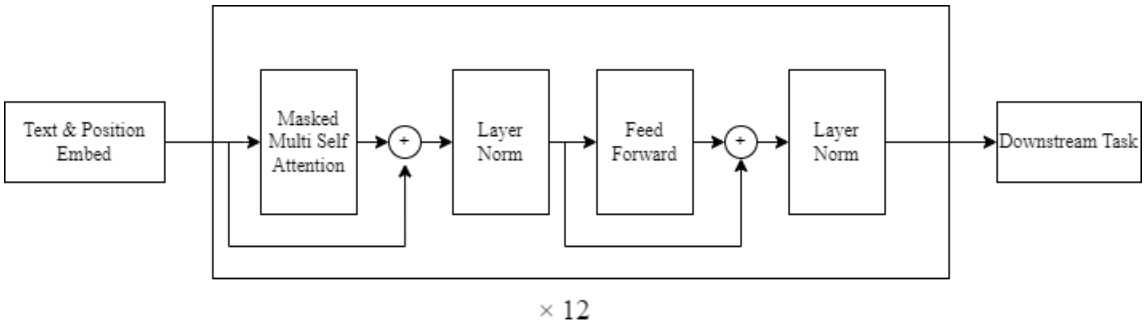

Figure 2: Structure of Transformer Decoder

Finally, we pass the outputs of pre-training for the softmax function with linear combination. This step can be viewed as the maximization of the function $L_2(C) = \sum \log P(Y|x_1, \ldots, x_m)$, where $P(y|x_1, \ldots, x_m) = softmax(h_{12}W_y)$ and h_{12} is the output after 12 transformer decoders.

In this case, the current total function that we want to maximize is $L_3(C) = L_2(C) + \lambda L_1(C)$.

2.2 Classifier : Multi-layer Perceptron

Now we can use classification model. The classifier we used is a 3-layer MLP, where neurons in the last layer refer to all intent classes. In this case, we adopted cross entropy loss as loss function, whose equation can be written as follows, where n stands for the number of classes, y_{ji} stands for the actual value of the probability of class i and $\hat{y_{ji}}$ stands for the real value. In fact, the cross entropy measures the degree of different information between two distributions.

$$Loss = \frac{1}{batch\_size} \sum_{j=1}^{batch\_size} \sum_{i=1}^{n} -y_{ji} \log y_{ji} - (1 - \hat{y_{ji}}) log(1 - \hat{y_{ji}})$$

The Figure 3 shows the basic structure of our Multi-layer Perceptron, where the first layer is the input layer (the utterance from the output of pre-training model), the last layer is the output layer (the probability of one class), and the second layer is the hidden layer (we have two hidden layers in our case). During our training, we do the random initialization of weights in the MLP to avoid that the coefficients of fitting function don't change at all during the training. The activation function is ReLU() function.

To sum up, the structure of model's layer can be summarized as follows:.(1) Transformer decoder layers: embedding layer, transformer decoder layer*12, linear layer, linear layer, SoftMax layer; (2) Classifier Layers: linear layer, ReLU layer, linear layer, ReLU Layer, linear layer, SoftMax layer.

3 Results

Due to time limit, we only consider the subset 'dyda_da' of SILICONE for our training example. Besides, as a simple example, we only collect the first 1000 training data in this subset to construct all of our data.

As usual, we split the collected data set into training set (670 observations), validation set (165 observations) and testing set (165 observations). In this data set, the utterances are with four types of labels: 0 (commissive), 1 (directive), 2 (inform), 3 (question). After data checking, we find that the data set in terms of label is not balanced. In this case, we consider weighted cross entropy loss when do the optimization of the model. The optimizer we use is Adam algorithm.

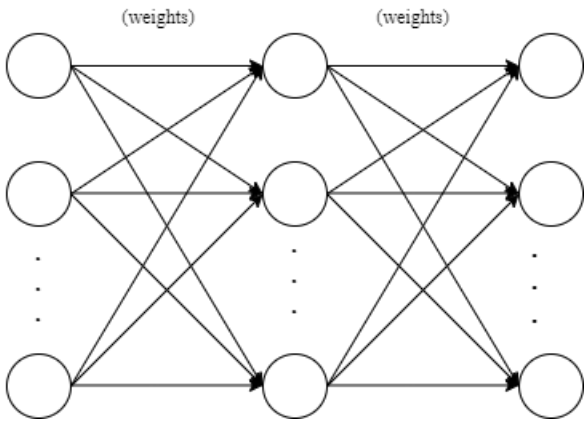

Figure 3: Structure of Multi-layer Perceptron

Figure 4 show us if the training process goes well. We can see that in general, the weighted cross entropy loss decreases when we run the epoch one by one (in our examples, we have ran 5 epochs).

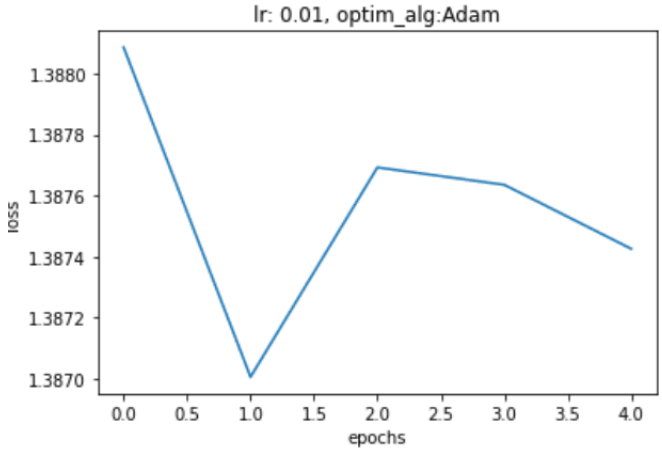

Figure 4: Loss with epoch running

Figure 5 instead shows us the situation of accuracy rate when we run the epoch one by one. As we can see in the picture, the accuracy rate doesn't not always increase, but is under fluctuation. The value on the whole is around 52% .

When we test the training model on the testing data, the weighted cross-entropy loss is equal to 1.3822, and the accuracy rate is 55.47%. And we also notice that the classification model is not good enough, since our training model tends to classify the the utterances into 2 of the 4 categories (label 2 and label 3). The weighted cross-entropy loss on validation data is 1.3817, and the corresponding accuracy rate is 60.16%.

Several reasons cause for this. On one hand, we only run very few of the data (subset 'dyda_da' has around 80000 training data). On the other hand, the transformer layers here may need to be ameliorated and we may consider introducing pooling layers or convolutional layers to extract more prominent features in the tensors.

Other tables can be summarized using the same code in our Github. For instance, we can change the learning rate (in this example, learning rate is equal to 0.01), optimization (SDG for example) , loss function (NLL loss for example)and percentage of training set split (67% in this example) to see how the loss and accuracy rate will change. We need also to train this model on other subsets in SILICONE. Due to time limit, the content here is passed.

Finally, for our training model, we have 760310743 parameters in the whole training structure.

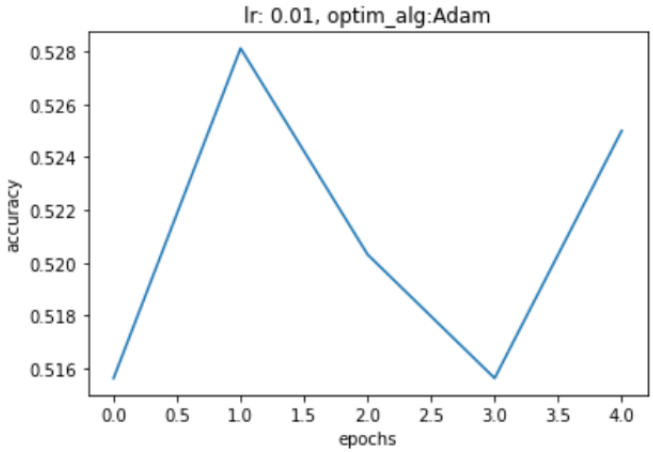

Figure 5: Accuracy rate with epoch running

4 Conclusion

In this paper, we presented a comprehensive study of intent classification using transformers, focusing on the effectiveness of Transformer based architectures. However, while transformers have shown remarkable results in intent classification, there are still open research questions in this field, particularly regarding the robustness of these models to data distribution shifts [11, 12], and domain shifts [8]. These issues are critical in real-world scenarios, where the training and testing data may come from different distributions, and malicious actors may intentionally modify the input to fool the model.

References

[1] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower Provost, Samuel Kim, Jeannette Chang, Sungbok Lee, and Shrikanth Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42:335–359, 12 2008.

[2] Emile Chapuis, Pierre Colombo, Matteo Manica, Matthieu Labeau, and Chloe Clavel. Hierarchical pre-training for sequence labelling in spoken dialog. *arXiv preprint arXiv:2009.11152*, 2020.

[3] Pierre Colombo. *Learning to represent and generate text using information measures*. PhD thesis, Ph. D. thesis, Institut polytechnique de Paris, 2021.

[4] Pierre Colombo, Emile Chapuis, Matthieu Labeau, and Chloé Clavel. Code-switched inspired losses for spoken dialog representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8320–8337, 2021.

[5] Pierre Colombo, Emile Chapuis, Matthieu Labeau, and Chloe Clavel. Improving multimodal fusion via mutual dependency maximisation. *arXiv preprint arXiv:2109.00922*, 2021.

[6] Pierre Colombo, Emile Chapuis, Matteo Manica, Emmanuel Vignon, Giovanna Varni, and Chloe Clavel. Guiding attention in sequence-to-sequence models for dialogue act prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7594–7601, 2020.

[7] Pierre Colombo, Chloe Clavel, and Pablo Piantanida. A novel estimator of mutual information for learning to disentangle textual representations. *arXiv preprint arXiv:2105.02685*, 2021.

[8] Pierre Colombo, Eduardo Dadalto, Guillaume Staerman, Nathan Noiry, and Pablo Piantanida. Beyond mahalanobis distance for textual ood detection. *Advances in Neural Information Processing Systems*, 35:17744–17759, 2022.

[9] Pierre Colombo, Wojciech Witon, Ashutosh Modi, James Kennedy, and Mubbasir Kapadia. Affect-driven dialog generation. *arXiv preprint arXiv:1904.02793*, 2019.

[10] Pierre Colombo, Chouchang Yang, Giovanna Varni, and Chloé Clavel. Beam search with bidirectional strategies for neural response generation. *arXiv preprint arXiv:2110.03389*, 2021.

[11] Maxime Darrin, Pablo Piantanida, and Pierre Colombo. Rainproof: An umbrella to shield text generators from out-of-distribution data. *arXiv preprint arXiv:2212.09171*, 2022.

[12] Maxime Darrin, Guillaume Staerman, Eduardo Dadalto Câmara Gomes, Jackie CK Cheung, Pablo Piantanida, and Pierre Colombo. Unsupervised layer-wise score aggregation for textual ood detection. *arXiv preprint arXiv:2302.09852*, 2023.

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

[14] Tanvi Dinkar, Pierre Colombo, Matthieu Labeau, and Chloé Clavel. The importance of fillers for text representations of speech transcripts. *arXiv preprint arXiv:2009.11340*, 2020.

[15] Lucas Chaves Lima Christian Hansen Maria Maistro Jakob Grue Simonsen Dongsheng Wang, Casper Hansen and Christina Lioma. Multi-head self-attention with role-guided masks. *European Conference on Information Retrieval*, LNISA, volume 12657, 2021.

[16] Matteo Manica Matthieu Labeau Chloe Clavel Emile Chapuis, Pierre Colombo. Hierarchical pre-training for sepquence labelling in spoken dialog. *Findings of the Association of Computational Linguistics*, EMNLP 2020, 2021.

[17] Alexandre Garcia, Pierre Colombo, Slim Essid, Florence d'Alché Buc, and Chloé Clavel. From the token to the review: A hierarchical multimodal approach to opinion mining. *arXiv preprint arXiv:1908.11216*, 2019.

[18] Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. Dialoguegcn: A graph convolutional neural network for emotion recognition in conversation. *arXiv preprint arXiv:1908.11540*, 2019.

[19] John J. Godfrey, Edward C. Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ICASSP'92, page 517–520, USA, 1992. IEEE Computer Society.

[20] Hamid Jalalzai, Pierre Colombo, Chloé Clavel, Eric Gaussier, Giovanna Varni, Emmanuel Vignon, and Anne Sabourin. Heavy-tailed representations, text polarity classification & data augmentation. *Advances in Neural Information Processing Systems*, 33:4295–4307, 2020.

[21] Geoffrey Leech and Martin Weisser. Generic speech act annotation for task-oriented dialogues. 2003.

[22] Ruizhe Li, Chenghua Lin, Matthew Collinson, Xiao Li, and Guanyi Chen. A dual-attention hierarchical recurrent neural network for dialogue act classification. *CoRR*, 2018.

[23] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. Dailydialog: A manually labelled multi-turn dialogue dataset, 2017.

[24] Gary Mckeown, Michel Valstar, Roddy Cowie, Maja Pantic, and M. Schroder. The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *Affective Computing, IEEE Transactions on*, 3:5–17, 08 2013.

[25] R. Passonneau and E. Sachar. Loqui human-human dialogue corpus (transcriptions and annotations), 2014.

[26] Georg Pichler, Pierre Jean A Colombo, Malik Boudiaf, Günther Koliander, and Pablo Piantanida. A differential entropy estimator for training neural networks. In *International Conference on Machine Learning*, pages 17691–17715. PMLR, 2022.

[27] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. Meld: A multimodal multi-party dataset for emotion recognition in conversations, 2018.

[28] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[29] Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. The ICSI meeting recorder dialog act (MRDA) corpus. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004*, pages 97–100, Cambridge, Massachusetts, USA, April 30 - May 1 2004. Association for Computational Linguistics.

[30] Henry Thompson, Anne Anderson, Ellen Bard, Gwyneth Doherty-Sneddon, Alison Newlands, and Cathy Sotillo. The hcrc map task corpus: natural dialogue for speech recognition. 01 1993.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[32] Wojciech Witon, Pierre Colombo, Ashutosh Modi, and Mubbasir Kapadia. Disney at iest 2018: Predicting emotions using an ensemble. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 248–253, 2018.