Intent classification: a performance comparison of small transfomers

Favre Hugo
ENSAE Paris
hugo.favre@ensae.fr

Alice Fabre-Verdure
ENSAE Paris
alice.fabre-verdure@ensae.fr

Abstract

Transformer models like BERT have shown great success in various natural language processing tasks, but they often require a significant amount of time and computational resources to train and deploy due to their large size. In this analysis, we are comparing the accuracy of four small BERT transformers as a solution to reduce the computational requirements while maintaining similar levels of performance. Additionally, we are examining the impact of using an MLP decoder, which seems to have a positive effect on the accuracy for the medium bert. We evaluate our results on a new benchmark we call Sequence labellIng evaLuatIon benChmark fOr spoken laNguagE benchmark (SILICONE). [1]

1 Introduction

Intent classification plays a crucial role in improving the performance of models used for spontaneous dialogue tasks. To achieve this, it's important to identify both the Dialog Acts (DA) and Emotion/Sentiment (E/S) in spoken language (Atmaja and Sasou, 2022; Dinkar* et al., 2020). By accurately identifying the intended action and emotional state of the speaker, the system can avoid generating generic responses that do not address the user's needs. This is a common problem in automatic dialogue systems, where generic responses may not provide specific solutions to the user's query (Yi et al., 2019; Colombo et al., 2019)). Hence, identifying the intent and emotional state of the speaker (Garcia* et al., 2019) helps the system to provide more relevant and personalized responses, leading to an improved user experience.

In this problem, we first need to clearly define the challenge at hand. To define our problematic, we rely on (Chapuis et al., 2020; Colombo et al., 2021a). The purpose of their paper is to achieve competitive results with fewer parameters with hierarchical encoders. To do so they use two data sets: Silicone and Opensubtitles. As we do not have GPU on our computers we can neither load important model nor massive datasets. In this study, we fine-tune pre-trained BERT models on the SILICONE datasets (Godfrey et al., 1992; Li et al., 2017; Leech and Weisser, 2003; Passonneau and Sachar., 2014; Thompson et al., 1993; Poria et al., 2018; Shriberg et al., 2004; Mckeown et al., 2013) for the task of intent classification. Then, we decide to reproduce statistics descriptive of the paper to ensure the right comprehension of the SILICONE database and to compare small pretrained transformer BERT models: with tiny, small and medium pretrained model and multi linear perceptron with the same database.

2 Literature review

Different papers deal with the GPU/TPU memory limitations and longer training times. They try to find some solutions to overcome these issues and then decreasing run time without compromising performances.

Transformer distillation

In their 2019 study, (Jiao et al., 2019) tackle this issue by proposing a novel transformer distillation : a unique approach to knowledge distillation (KD) of Transformer-based models that enabled them to speed up inference, reduce model size, and maintain accuracy. Specifically, they developed a new KD method that was tailored for distilling knowledge from a large "teacher" BERT to a smaller

[1] https://github.com/hugofavre1/NLP_intent_classification

"student" tiny-BERT. This approach allowed the abundant knowledge embedded in the teacher model to be effectively transferred to the student model. Furthermore, they introduced a two-stage learning framework for tiny-BERT that employed transformer distillation during both pre-training and task-specific learning stages. This framework ensured that tiny-BERT could acquire both general-domain and task-specific knowledge from BERT.

Number of paramteres
On the contrary, (Lan et al., 2019) decide to reduce the number of parameters used in their model to lower the memory consumption. Doing so, they get comprehensive empirical evidence which shows that the methode they used lead to models that scale much better compared to the original BERT. However these two papers rely on the database GLUE which is not the one that we use.

The Silicone benchmark
We use Silicone which gather different database. (Chapuis et al., 2020) also use this base and underline that it gather both DA and E/S annotated datasets and it is built upon preexisting datasets which have been considered by the litterature as challenging and interesting. Any model that is able to process multiple sequences as inputs and predict the corresponding labels can be evaluated on SILICONE. In their paper they demonstrate how hierarchical encoders achieve competitive results with consistently fewer parameters compared to state-of-the-art models and they show their importance for both pre-training and finetuning.

3 Presentation of the databases

3.1 Global presentation

Despite a variety of small or medium-sized tagged datasets being available (SwDA, MRDA), they are often overlooked for evaluation purposes due to the high computational cost involved in analyzing the largest corpora ((Li et al., 2018)). To address this limitation, we use the SILICONE benchmark, which is introduced and described in this paper (Chapuis et al., 2020). This is a group of sequence labeling that integrate annotated DA and E/S information from different sources. The scientific community has already employed tough and intriguing datasets that are part of SILICONE. The latter can be used to assess any model that can handle multiple sequences as inputs and forecast the related labels.

Utterance	DA
can you do push-ups ?	Question
of course i can	Inform
really ? i think that's impossible !	Question
it's easy . if you do exercise everyday, you can make it , too	Inform

Table 1: Samples of dialogs with the DA label taken from the dyda_da dataset

3.2 Descriptive statistics

To understand the database and to be able to select interesting models, we realize descriptive statistics by replicating table 2 of (Chapuis et al., 2020).

Dataset	Task	Train	Valid	Test	Labels
swda	DA	1000	115	11	43
mrda	DA	56	6	11	5
dyda_da	DA	11118	1000	1000	4
maptask	N/A	20905	2963	2894	12
oasis	N/A	12076	1513	1478	42
dyda_e	E	11118	1000	1000	7
meld_s	S	1038	114	280	3
meld_e	S	1038	114	280	7
iemocap	E	109	11	31	10
sem	S	61	8	10	3

Table 2: Statistics of datasets composing SILICONE. The Sizes of Train, Val and Test are given in number of conversations.

We observe that results are almost the same compared to the one of the paper mentionned just before. However, it is not really the case for maptask and oasis which do not have the same columns as other datasets and then cannot be used in the same way.
We observe that $Dyda_{da}$, $Dyda_e$ and Maptask are the three bases with the highest number of conversation. On the contrary, Swda has the most important number of unique labels, juste

before oasis.

4 Modelisation

4.1 Data preprocessing

We only used a subset of the training data in order to speed up the experimentation process. We took 10% of the orignal datasets for training, validaiton and test.

In natural language processing, tokenization is an essential preprocessing step that involves breaking down raw text into smaller, meaningful units called tokens. These tokens can then be fed into machine learning models for further processing. For our experiment, we utilized the Hugging Face transformers library (Wolf et al., 2019), which provides a range of tokenizers designed for different use cases and model architectures. We loaded the tokenizer and model for each model architecture, and used them together to process the inputs and generate predictions. Specifically, the tokenized inputs were passed into the model as input features, which were then processed through a series of layers to generate a prediction for each input. The use of tokenization and machine learning models allowed us to effectively analyze and classify text data.

4.2 Encoding - Decoding

The second step of each NLP task consists in transforming language into vectors. There are several ways to do so, and we consider a pre-trained transformer-based language model : the BERT model (Devlin et al., 2018). However, as it is time computing to use Bert-base model, we use tiny, mini, small and medium Bert (Bhargava et al., 2021; Turc et al., 2019).

Model	Layers	Size	Attention heads
Bert-tiny	2	128	2
Bert-mini	4	256	4
Bert-small	6	512	8
Bert-medium	8	512	8

Table 3: Number of parameters of bert models

Advantages of these models :
The use of smaller models have impacts in computation as they have a fast inference time and a low memory usage. However, they also have some weakness : they may not perform as well as larger models on more complex tasks. We can also underline that medium model is more powerfum and more time consuming than the mini one.

MLP decoder
The decoder used in this code is the output layer of the pre-trained transformer-based model, which generates the predicted labels for the input sequences. Several papers have conducted analyses on the performance in Sequence labelling of different decoders like MLP and GRU (Colombo et al., 2020). In order to improve the performance of the pre-trained BERT models, we define a custom MLP decoder using the PyTorchnn.Sequential module. Our MLP decoder consists of three fully connected layers with ReLU activation functions and dropout regularization (Pascual et al., 2021). We replace the original linear classifier layer of the pre-trained BERT model with the MLP decoder. The resulting model is then fine-tuned on aspecific sequence classification task. We hope that this new choice of decoder will introduce non linearity into the model to capture more complex patterns in the data and provide regularization benefits such as dropout, which can help to prevent overfitting and improve generalization.

4.3 Loss Function

We chose the CrossEntropy loss function as it is well-suited for classification. Indeed, it combines a softmax activation function and a negative log-likelihood loss function into a single function. The output of the model's final layer is passed through a softmax activation function, which generates a probability distribution over the possible classes for each input instance. The CrossEntropy loss function then calculates the negative log-likelihood of the correct class based on this probability distribution. For an input X, with a softmax vector $\hat{Y} = (p_1, p_2, p_3, p_4)$, where (p_i) represents the computed probability of i being the right label for X, the value of the loss function is:

$$L(X, \hat{Y}) = \sum_{i}^{4} -y_i log(p_i)$$

where Y = (y_i) is the vector of the real label of X ($y_i = 1$ only if label of X is i).

4.4 Hyper-parameters

In order to chose hyperparameters of the model, we rely on (Chapuis et al., 2020) as they use the same database as the one we use. Then, for the SILICONE dataset we decided to take a learning at 0.0001, 2 epochs and 64 as batch size. These parameters were chosen based on common practices and empirical evidence in the natural language processing (NLP) literature. We selected hyperparameters such as number of epochs, batch size, and optimizer based on a trade-off between model performance, training time, computational efficiency, and model stability. The CrossEntropyLoss is a standard loss function for classification tasks and was chosen because SILICONE is a sequence labeling task that requires predicting one of several possible labels for each token in the input sequence. These parameters were chosen for the list of transformer models because they are reasonable starting values that have been shown to work well for many NLP tasks. However, the optimal values may vary depending on the specific task, dataset, and model architecture.

5 Results

5.1 Evaluation of performances

In machine learning, it is essential to evaluate the performance of the model on unseen data to assess its ability to generalize well new data. To achieve this, it is common to split the available dataset into three parts: training set, validation set, and test set. The training set is used to optimize the model parameters, the validation set is used to tune the hyperparameters and prevent overfitting, and the test set is used to evaluate the final performance of the model.

We evaluated the model's performance using the accuracy metric on both the validation and test sets, where the former was used for monitoring the model's performance during training and adjusting the hyperparameters, while the latter was used to assess the final performance of the model on an unseen dataset.

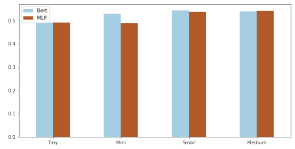

Figure 1: Average test accuracy for each model

On the Figure 1, we compare performances of the four BERT models. As explained before, the larger the pre-trained model, the longer it takes time to run and the better we expect accuracy. We observe that it is true for Bert-tiny and Bert-small for whom performances increase a lot but for the two others models, accuracy does not increase much whereas time to run these code increase much more. For Bert medium, the use of the MLP decoder increases the average accuracy of the model. However, the accuracy gains are generally not significant in our different models for this specific task

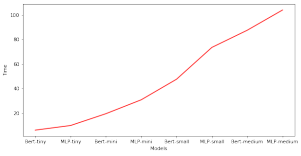

Figure 2: Running average time for each model

We indeed observe on the Figure 2 that the time increase between each model is quite linear, which mean that the increase in performance justify the use of Bert-mini rather than Bert-tiny but not the use of Bert-medium instead of Bert-small. Moreover, we observe that for a given size of model, MLP takes more time to run than BERT.

5.2 Performances comparison

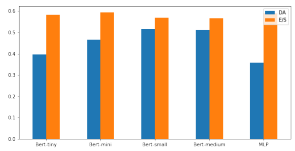

Figure 3: Performance comparison between DA and E/S datasets

When we look at the difference in performance between the two groups of datasets (DA,

E/S) in Figure 3, we observe that E/S group performs better for all models but that the larger the model the smaller is the spread between the two groups.

6 Conclusion and discussion

In this analysis, we are comparing the accuracy of four small BERT transformers as a solution to reduce the computational requirements while maintaining similar levels of performance. Based on our findings, it seems that there is no significant difference in accuracy between the small BERT transformers we tested, but there is a significant difference in training time. The accuracy differences are not proportional enough to the time differences that a larger transformer implies. Our results also show that using an MLP decoder instead of a linear decoder does not have a significant positive impact on model accuracy for this task. Overall, our results show that increasing the size of small transformers does not offer sufficient accuracy compared to the time cost involved.

However, there are limitations to consider when fine-tuning small BERT transformers for intent classification in dialog acts. First, (Chapuis et al., 2020) find that contrary to us, the accuracy is better for DA compared to E/S. As the spread decreases when the model increases we can make the assumption that when using the full BERT model, the spead is in the other side. Moreover, these models may suffer from bias or lack of generalizability to other domains or languages due to training data limitations. In addition, they may struggle with longer sequences and may not be suitable for real-time applications due to their computational requirements. Overall, their effectiveness is dependent on training data, dialog complexity, and computational resources.

There are several areas of future research that could be explored to enhance the use of small BERT transformers in natural language processing tasks. One idea would be to add multimodality to current systems (Colombo et al., 2021b; Garcia* et al., 2019). Another promising avenue is to investigate the effectiveness of small BERT transformers in tasks beyond intent classification in dialog acts. Additionally, the potential of using ensemble models that combine multiple small BERT transformers to improve model performance should be explored. Another area of focus for future research could be to develop more efficient training methods or architectures for small transformers that can reduce their computational requirements while maintaining high levels of accuracy.

Furthermore, it is essential to consider the out of distribution generalization of small BERT transformers (Colombo et al., 2022; Darrin et al., 2023). While these models have shown impressive results in intent classification, their effectiveness on data that differs significantly from the training data is unclear. Therefore, future research should investigate the performance of small BERT transformers on out of distribution data, which could lead to the development of more robust and versatile models.

References

Bagus Tris Atmaja and Akira Sasou. 2022. Sentiment analysis and emotion recognition from speech using universal speech representations. *Sensors*, 22(17):6369.

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics. *arXiv preprint arXiv:2110.01518*.

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

Pierre Colombo, Emile Chapuis, Matthieu Labeau, and Chloé Clavel. 2021a. Code-switched inspired losses for spoken dialog representations. In *EMNLP 2021*.

Pierre Colombo, Emile Chapuis, Matthieu Labeau, and Chloe Clavel. 2021b. Improving multimodal fusion via mutual dependency maximisation. *EMNLP 2021*.

Pierre Colombo, Emile Chapuis, Matteo Manica, Emmanuel Vignon, Giovanna Varni, and Chloe Clavel. 2020. 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.

Pierre Colombo, Eduardo Dadalto Câmara Gomes, Guillaume Staerman, Nathan Noiry, and Pablo Piantanida. 2022. Beyond mahalanobis distance for textual ood detection. In *NeurIPS 2022*.

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

Maxime Darrin, Pablo Piantanida, and Pierre Colombo. 2023. Rainproof: An umbrella to shield text generators from out-of-distribution data. *arXiv preprint arXiv:2212.09171*.

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*.

Tanvi Dinkar*, Pierre Colombo*, Matthieu Labeau, and Chloé Clavel. 2020. The importance of fillers for text representations of speech transcripts. *EMNLP 2020*.

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

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. 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. IEEE Computer Society.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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

Ruizhe Li, Chenghua Lin, Matthew Collinson, Xiao Li, and Guanyi Chen. 2018. A dual-attention hierarchical recurrent neural network for dialogue act classification. *arXiv preprint arXiv:1810.09154*.

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

Gary Mckeown, Michel Valstar, Roddy Cowie, Maja Pantic, and M. Schroder. 2013. 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.

Damian Pascual, Sandro Luck, and Roger Wattenhofer. 2021. Towards bert-based automatic icd coding: Limitations and opportunities. *arXiv preprint arXiv:2104.06709*.

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

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

Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. 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. Association for Computational Linguistics.

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

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

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

Sanghyun Yi, Rahul Goel, Chandra Khatri, Alessandra Cervone, Tagyoung Chung, Behnam Hedayatnia, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. 2019. Towards coherent and engaging spoken dialog response generation using automatic conversation evaluators. *arXiv preprint arXiv:1904.13015*.