Comparative Study of Single-task and Multi-task Learning on Research Protocol Document Classification

Abid Famasya Abdillah
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
abid.famasya@gmail.com

Mohammad Zaenuddin Hamidi
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
mzaenuddinhamidi@gmail.com

Ratih Nur Esti Anggraeni
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
ratih_nea@if.its.ac.id

Riyanarto Sarno
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
riyanarto@if.its.ac.id

Abstract— **Research protocol is an important document to be scrutinized by the ethical committee. As the research proposal is growing, the necessity for quick and concise protocol review is rising. This study undergoes a comparative study of multi-task learning (MTL) and single-task learning (STL) to classify research protocol documents. We try to carry out the classification process from the summary of health research. We represent research documents as multi-label classification problems and develop a deep learning model based on MTL and STL strategies. In our evaluation, multi-task learning achieved a better result with 0.125 loss and 0.785 Jaccard score than 0.182 and 0.720 in single-task learning. In consequence, MTL has a 27% slower computation time than STL.**

Keywords— multilabel document classification, research protocol classification, single-task learning, multi-task learning

I. INTRODUCTION

The advancement of human knowledge is led by numerous researches, including studies in the healthcare domain. Healthcare research is conducted to collect and disseminate information needed to support medical and clinical activities, which finally aims to improve human life. Every health research must be through an ethical review to respect the life, health, privacy, and dignity of research subjects [1]. To attest to the feasibility of health-related research, the ethical committee will conduct a careful examination of the research protocol before issuing an ethical clearance. A protocol is marked as passed if it fulfills seven ethical issue standards which contain social and scientific values, risk and benefits information, preserving privacy, and informed consent as described by [2]. Commonly, ethical committees require a summary that includes descriptive background, methods, treatments, and relevant information. The summary is intended to provide readers relevant information in a straightforward manner [3].

There are keywords that represent ethical issues in the summary section, although it is not possible to include all keywords of ethical issues in the document's summary. Let $L = \{\lambda_1, \lambda_2, ..., \lambda_n\}$ is a finite set of document labels and $\lambda = (x_1, x_2, ..., x_m)$ where x is keywords associated with labels. Hence, finding λ sets of labels associated with summary document D based on sets of x is essentials. This particular problem is known as multilabel classification (MLC), which is quite popular to represents in various real-world problems [4].

Many algorithms have been conducted to solve this particular problem, from regular machine learning algorithms to deep learning-based approaches. Earlier methods that use machine learning algorithm i.e., decision tree or SVM is known to suffer from handling complex data and require extensive feature engineering. To handle the caveats, deep learning is a favorable method due to its ability to learn from abstract concepts and generalize to solve tasks. Further investigation of deep learning also indicates that weights learned from the model can also be generalized over tasks to learn data features faster, which is also called multi-task learning [5]. This approach is reported to be helpful to boost deep learning performance.

In this research, we present a comparative study to examine the performance of single-task and multi-task learning strategies to the protocol documents classification. The study will focus on the applicability of deep learning in the health domain. In particular, we will identify and evaluate what deep learning strategy is the best fit for research protocol documents classification.

II. RELATED WORKS

A. Multilabel classification

Multi-label classification (MLC) is a subset of classification problems where particular data is being labeled based on its characteristics [4] [6]. Multi-label classification is a generalization of multiclass classification. Instead of categorizing an instance into one of the multiple classes, MLC has no limit on how many labels an instance can be assigned. The origin of MLC was originally inspired by classification problems where one data may belong to more than one conceptual class [7]. For example, a newspaper article about the reaction of the release of The Da Vinci Code film can be classified into the categories of society/religion and art/film. Examples of MLC can be seen in Table 1.

TABLE 1. EXAMPLE OF A MULTI-LABEL DATA SET

Doc.	Sports	Religion	Politics	Science
1	Yes	No	No	Yes
2	No	Yes	Yes	No
3	Yes	No	Yes	No
4	No	Yes	No	Yes

978-1-6654-4059-2/21/$31.00 ©2021 IEEE

MLC is different from multiclass, where labels are given in non-exclusive manners. Multiple systems are being proposed to solve this problems from general machine learning algorithm e.g. tree-based approach to Neural Network and Deep Neural Network based. International Classification of Diseases (ICD) is an example of multi-label problems. Yet, [8] has managed to solve it by using bidirectional GRU based approach resulted in 63.1% F1 score. The same domain of problems has also been solved by [9] using LSTM and Attention mechanism capped at 0.815 F1 score. In the legal domain, the Thompson Reuters research team compared Random Forest with multiple DNN models to achieve best F1 score of 0.820 [6]. In the text classification, earlier research tends to use TF-IDF as feature representation. Recently, embedding based approach has been considered as state-of-the-art as it gives better feature representation [10].

Multi-label in machine learning algorithms can be grouped into two main categories: problem transformation and algorithm adaptation methods [7]. Problem transformation is a method where MLC is transformed into single-labeled problems before being processed with an algorithm. On the other hand, algorithm adaptation is a modification of existing algorithms, e.g. Random Forest, k-NN to classify multi-labeled data. The performance, however, could not improve well where features representations are not well prepared. At this point, deep learning is considered well suited for such problems.

B. Single-task and Multi-task learning

Most models in machine learning and DNN use single task learning (STL) or ensemble of models to solve a particular problem. The feed-forward neural network and backpropagation neural network are some of typical STL architecture. While hyperparameters tuning could improve the result, some information from related tasks might be disregarded during model development by using this approach. Multi-task Learning (MTL) is an optimization of the previous approach (STL) to maximize outcomes by sharing weights and inductive biases between tasks [11]. Several conditions must be ensured to ensure multi-task learning work, i.e., model capacity, task covariance, and optimization performance [12]. As of it, we consider this reference to build our model.

The aim of multi-task learning is to exploit the correlation of related tasks by extracting correlation between common features such as sub-vectors and shared sub-spaces. This is done as an effort to improve the classification process by studying tasks in parallel [10]. Supposed that we have T tasks with t sub-tasks, for each t task, we have m_t models and a_t parameters in dimension of d. This also can be written by Equation (1).

$$a_t = \begin{bmatrix} a_{1,t} \\ \vdots \\ a_{d,t} \end{bmatrix}^{\mathrm{T}} \quad (1)$$

Column vectors are then stacked into $a_1, \ldots, a_{\mathrm{T}}$ to form matrix $A \in \mathbb{R}^{d \times t}$ which also be called as block sparse regularization. By assuming that all models have some numbers of shared features, MTL can be computed over matrix parameters A to regularize on all related tasks [5]. There are two points of view of multi-task learning architecture, namely hard parameter sharing and soft parameter sharing. Hard parameter sharing is most commonly used since it learns common space representations for all tasks by fully sharing the weights between tasks. It acts as a regularization and reduces the risk of overfitting. As of this, MTL is claimed to benefit in some cases, such as implicit data augmentation and eavesdropping between tasks [5] [13].

Various domains use multi-task mechanisms to optimize measurements. The MTL architecture combined with encoder-decoder mechanism has been reported achieving Jaccard score of 0.76 and Dice score of 0.87 on organs segmentation [14]. In the text processing, Marc Djandji, et al applied multi-task learning to detect hate and offensive speech in Arabic using BERT architecture with F1 score of 90% on hate and 82% on offensive language [15].

III. Experiments

This section describes our research procedures to undertake multi-label classification via single-task and multi-task learning approaches. We use algorithm adaptation method and hard parameter sharing for the MTL learning strategy.

A. Datasets

Two multilabel datasets are used in the research to validate the robustness of the system. The first dataset contains summaries of ethical protocol documents, which consists of 36 documents and 9,302 words. The documents were gathered from educational institutions and official healthcare web resources to maintain the validity of documents. The example of documents is presented in Table 2.

Each document has seven columns contains one summary and six labels. To preserve privacy, mentions on private information (identity, location, and institution) are masked or replaced by random names. An expert was assigned to annotate documents into six labels corresponds to potential ethical issues mentioned in Table 3. As the guideline, keywords are used during the annotation step. For an instance, when particular document mention students and workers, it can be assumed that corresponding protocol take account of

TABLE 2. RESEARCH PROTOCOL DOCUMENTS EXAMPLE

Doc.	Summary	L1	L2	L3	L4	L5	L6
1	Banyak studi menunjukkan bahwa tidak sedikit staf organisasi (seperti perawat yang bekerja di RS) … *Lots of study show that many organization staff (i.e., nurse working in hospital)…*	Y	N	Y	N	N	Y
2	Peneliti ingin meneliti cara-cara individu pengunjung yang berobat ke RS. … *Researcher wants to know how individuals obtain health service in the hospital…*	N	Y	N	N	N	N
3	Penelitian dilakukan untuk mengamati keselamatan pasien dan memahami penyebab kesalahan manusia … *Research is aimed to observe the patients safety and what cause human error…*	Y	Y	Y	Y	N	Y

TABLE 3. ETHICAL ISSUES LABEL AND KEYWORDS

No	Issues	Keywords
L1	Research is onsidering vurnerable group	Students, workers, oldster, pregnant women, kids, etc.
L2	Privacy is guaranteed	Identity, privacy, confidential, anonymity, etc.
L3	Informed consent is presented	Approval, agreements, etc.
L4	Research subject is not induced	Compulsion, compensation, voluntary, etc.
L5	Risks are informed	Emotional pressure, psychology, law, etc.
L6	Risks are kept minimal	Minimize risk, benefits, adverse events, etc.

vulnerable group. Otherwise, when no "anonymity" is mentioned in the document, it is implied that the protocol do not guarantee the privacy of research subject. Hence, the document is marked as *N* in L2 column.

The second dataset is a research papers dataset obtained from [16] contain title, summary and six labels of topics (Computer Science, Physics, Math, Statistics, Quantitative Biology, Quantitative Finance). Documents are written in English and made of 20972 research papers. Reviewer committees have annotated each of the documents. For convenience, ethical protocol dataset is marked as $D1$ and research paper dataset is marked as $D2$.

As we use two datasets to test the robustness of our experiments, $D1$ and $D2$ will be treated equally. Data preprocessing is performed to standardize and transform text data into a numerical representation. Initially, a summary was extracted from dataset by tokenization into individual words. Then, summaries are converted into sequences of sentences based on token index to maintain word order. The order of word sequence is essential as our model learns syntactic representation and word order in sentences. At this step, we also take note of the longest sentence length in a corpus.

Each sequence is padded into the longest sentence to keep feature length consistent for all sentences in a summary document. In this context, D1 has 576 max lengths while D2 has 436. Padding is a mechanism to fill zero value for sentences less than maximum length. Otherwise, it will be truncated. After padding mechanism is carried out, data is split into training set and *testing set* by ratio of 80:20 followed by modeling step which take input from training data into classifier model. We build two deep neural network classifier model to fit with our scenario where STL and MTL strategy applied. After the model had developed, it will be trained using training set and predicts labels in testing set. Lastly, the predicted labels are evaluated to ground truth data.

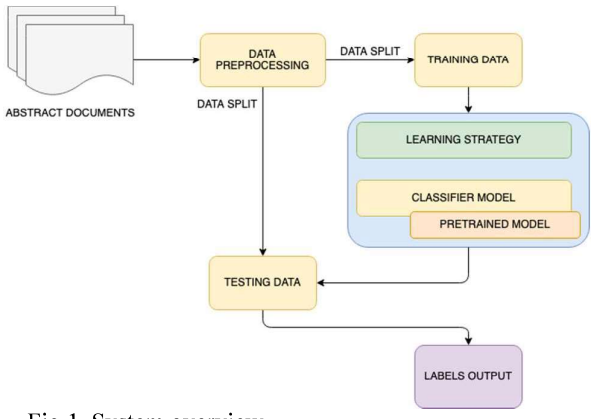

Fig 1. System overview

B. STL and MTL architecture

Besides STL, this research use MTL approach, i.e., instead of having single output layer, we treat each label independently. The use of MTL in this experiment is possible since our tasks have similar task covariance and consistent model capacity to perform in multi-task learning setting [17]. Task covariance means that each task must have a similar feature representation to align and share weights to each other while model capacity requires that the tasks must have a shared module to let transfer knowledge happen. The first aspect is fulfilled in our model since it has the exact same word vector representations. Furthermore, the second aspect is satisfied by the design of the network model.

There are three layers group in MTL learning strategy: input layer, shared layer and task-specific layer. Input layer is a layer to process tokenized string, shared layer contains Bi-LSTM hidden layer, and task-specific layer consists of multiple layers for specific tasks, which in our case are representing six document labels. ReLU is used in the hidden layers and sigmoid is used in output layers. The main difference between STL and MTL lies in their classifier output layers. In STL, output layers represented by single output layer with independent hidden layer while MTL composed of six different layers, one shared layer and one neuron in output layer. An overview of STL and MTL architecture is presented in Fig 2.

Input layer receives sequences of sentences from corpus. In the embedding layer, input sequence will be converted into vector space via language model to be fed into hidden layers. Pretrained Word2Vec with model with 300 dimensions is chosen as the language model for its simple and concise architecture capturing words semantic similarity [10]. Bidirectional Long-Short Term Memory (Bi-LSTM) is used as hidden layer with 64 neurons. The use of Bi-LSTM is based on its popular architecture to capture important information in long context of paragraph [6] [9].

In the end of model both STL and MTL has classifier layer with different implementation. While STL use single layer with six neurons, MTL use six task-specific layers with one neuron in each layer. Sigmoid is used as activation function considering each label has binary property. Our neural network model is developed by using Keras as the deep learning library. For loss evaluation, we use accuracy value.

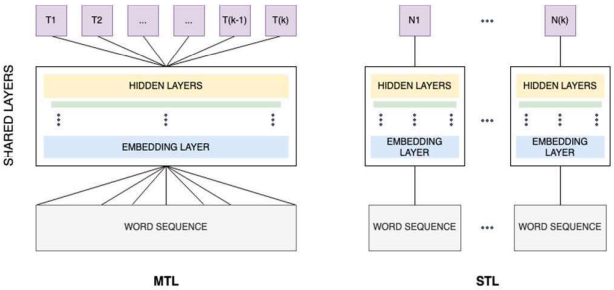

Fig 2. Comparison of MTL and STL architecture

IV. Results and Analysis

Training phase was performed in the cloud environment with following configuration: Intel Xeon CPU @ 2.20GHz, 16 Gb RAM, Tesla K80 12Gb GDDR5 GPU. For the model hyperparameter, following parameters are used: learning rate 1e-4, batch size 128, validation split 20%.

Table 3. Evaluation Results

Exp.	HL	JS	Prec.	Rec.	F1	Time
STL-D1	0.182	0.720	0.621	0.727	0.791	**91**
MTL-D1	**0.125**	**0.785**	**0.875**	**0.875**	**0.878**	126
STL-D2	0.113	0.571	**0.673**	0.679	0.702	**442**
MTL-D2	**0.097**	**0.635**	0.594	**0.868**	**0.773**	762

In the performance measurements, we use average hamming distance between incorrect labels (hamming loss) and Jaccard similarity index (Jaccard score) as standard evaluation metrics of MLC. Hamming loss can be seen as correctly labeled data compared to whole labeled data. In addition, we also calculate precision, recall and F1 score training time into our metrics.

As can be seen in Table 3, lowest hamming loss is achieved by MTL-D1 and MTL-D2 with 0.125 and 0.097 point consecutively. From Jaccard score, multi-task learning is also outperforming single task learning in both *D*1 and *D*2 documents of 0.785 and 0.635. In term of percentage, MTL has 24% lower hamming loss and 9% higher Jaccard score compared to STL. Looking at recall and F1 score the result is also consistent where MTL is better to label documents. However, precision score of STL scenario in *D*2 is higher compared to MTL. Execution time for STL strategy is also faster compared to MTL with 27% on D1 and 41% for D2 dataset. This can be inferred from its architecture in which MTL has more complexity factor than STL. The degradation of all evaluation metrics is also apparent in D2 documents since it processes much larger data than D1 (20972 compared to 36). From statistical point of view, the data distribution of D1 is more homogenous which lead to lesser misclassification [18]. However, to truly attest the robustness of model, *D2* could be better representation of real-world data. Despite the degradation, MTL still has a higher score compared to STL in most evaluation metrics.

Loss value between MTL and STL in Fig. 3. indicates that MTL has steeper loss value than the other. When STL only converge between 0.7 to 0.08, loss (*y* axis) of MTL is converging between 4.6 to 0.5. The huge loss value in MTL comes from accumulation of loss in each task-specific layer which also lead into more training time as seen in Table 3.

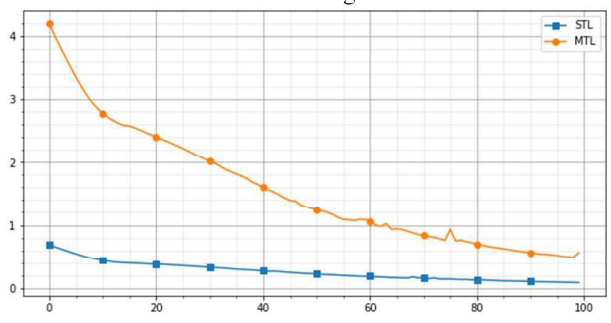

Fig. 3. Training loss comparison between MTL and STL

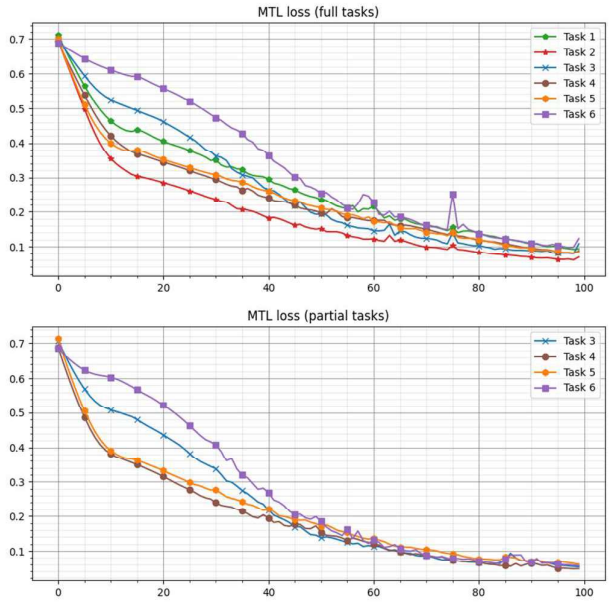

Fig. 3. MTL task-specific layer loss in training

From the graph it also can be seen that STL has lower starting point. It means that for lower epochs, STL could generalize better.

Visualization from MTL loss (Fig. 3.) can be analyzed to learn how our model behaves. In our experiment with 100 epochs (*x* axis), it can be seen that loss values (*y* axis) are started from 0.75 and reduced into 0.02. Further information from the full tasks graph (top figure) indicates that although all task-specific layers are converging, Task 3 and Task 6 are having gentler loss than the rest. It means that those tasks are having difficulty to generalize across training phase. Furthermore, interesting events happened at epochs 58-62 and 75 where spikes on one task is affecting all tasks. We hypothesize based on [5], this phenomenon can be explained by inductive theory where feature representations are induced and generalized across different tasks. We conducted further experiments to remove some tasks and analyze the plot (bottom figure). It can be seen that two major spikes are no longer noticeable afterwards.

V. Conclusion and Future Works

Our experiment to classify research protocol documents finds multi-task learning has some advantages than single-task learning. MTL is outperforming STL strategy in all of our hamming loss, Jaccard score and F1 score measurement in all datasets. This is based on from the theory of MTL where its shared layer could improve generalization of classifier outputs. However, this performance is also come with consequence where computation cost of MTL is also significantly higher than STL (762 seconds compared to 442 seconds on D2). Loss plot of MTL also indicates that with fewer iterations, STL can perform better.

Further improvement of this research can be extended into examination with more data to reduce disparity between labels. The shift to contextual and sub-word level word embedding such as could also be incorporated into this study to generate better word vector [10]. Finally, better handling to data imbalance should also considered as of many real-world data contains imbalance data particularly in multilabel classification problem.

REFERENCES

[1] Kementrian Kesehatan Republik Indonesia, "Program Indonesia Sehat dengan Pendekatan Keluarga," [Online]. Available: https://www.kemkes.go.id/article/view/17070700004/program-indonesia-sehat-dengan-pendekatan-keluarga.html.

[2] Komisi Etik Penelitian dan Pengembangan Kesehatan nasional, Pedoman dan Standar Etik Penelitian dan Pengembangan Kesehatan Nasional, Kementrian Kesehatan Republik Indonesia, 2017.

[3] A. C. Persch and S. J. Page, "Protocol Development, Treatment Fidelity, Adherence to Treatment, and Quality Control," *American Journal of Occupational Therapy,* 2013.

[4] M. S. Sorower, "A Literature Survey on Algorithms for Multi-label," Oregon State University, 2010.

[5] S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," *arXiv preprint arXiv:1706.05098,* 2017.

[6] D. Song, A. Vold, K. Madan and F. Schilder, "Multi-label legal document classification: A deep learning-based approach with label-attention and domain-specific pre-training," *Information Systems,* 2021.

[7] V. S. Tidake and S. S. Sane, "Multi-label Classification: a survey," *International Journal of Engineering & Technology,* 2018.

[8] A. Blanco, O. Perez-de-Viñaspre, A. Pérez and A. Casillas, "Boosting ICD multi-label classification of health records with contextual embeddings and label-granularity," *Computer Methods and Programs in Biomedicine,* 2020.

[9] J. Du, Q. Chen, Y. Peng, Y. Xiang, C. Tao and Z. Lu, "ML-Net: multi-label classification of biomedical texts with deep neural networks," *Journal of the American Medical Informatics Association,* 2019.

[10] J. Camacho-Collados and M. T. Pilehvar, "From Word to Sense Embeddings: A Survey on Vector Representations of Meaning," *Journal of Artificial Intelligence Research ,* 2018.

[11] Y. Zhang and Q. Yang, "A Survey on Multi-Task Learning," *IEEE Transactions on Knowledge and Data Engineering,* 2021.

[12] S. Wu, H. R. Zhang and C. Ré, "Understanding and Improving Information Transfer in Multi-Task Learning," *International Conference on Learning Representations,* 2020.

[13] J. Bingel and A. Søgaard, "Identifying beneficial task relations for multi-task learning in deep neural networks," in *Association for Computational Linguistics*, 2017.

[14] T. He, J. Hu, Y. Song, J. Guo and Z. Yi, "Multi-task learning for the segmentation of organs at risk with label dependence," *Medical Image Analysis,* 2020.

[15] M. Djandji, F. Baly, W. Antoun and H. Hajj, "Multi-Task Learning using AraBert for Offensive Language Detection," in *4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 2020.

[16] Shivanand, "Kaggle," 2021. [Online]. Available: https://www.kaggle.com/shivanandmn/multilabel-classification-dataset/version/1. [Accessed June 2021].

[17] S. Wu, H. R. Zhang and C. Re, "Understanding and Improving Information Transfer in Multi-Task Learning," in *International Conference on Learning Representations*.

[18] A. Althnian, D. AlSaeed, H. Al-Baity, A. Samha, A. B. Dris, N. Alzakari, A. A. Elwafa and H. Kurdi, "Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain," *Applied Science,* 2021.