Representation-Focused Long Document Retriever for Auto ICD Coding

1. Introduction

Automatic International Classification of Diseases (ICD) coding aims to assign multiple ICD code labels to a medical note with an average of 3,000+ tokens (Larkey and Croft, 1996) (Li and Yu, 2020). Specifically, take ICD code heart cardiac catheterization in Figure 1 as an example, the task is to find a similar mention of cardiac catheterization in the medical note. This is one of the most popular works in bio-NLP that has hundreds of citations. Previous study found that this multi-label auto ICD coding could be seen as an information retrieval task such as passage ranking (Chen and Ren, 2019). Instead of ranking top-k similar passages for each query in passage ranking, we would rank top-k similar codes for each medical document. Recent work has shown that cross-encoder outperforms bi-encoder in passage ranking (Zhou and Devlin, 2021). Thus we believe that such cross-encoder models would achieve higher accuracy compared to bi-encoder, which is a key component on auto ICD coding current SOTA model (Yuan et al., 2022).

2. Background

2.1. Information retrieval

The goal of information retrieval is to find relevant documents from a large corpus of documents when a query is given. Most of the classical retrieval models were lexical retrievers based on sparse representations where sentences are quantified as vectors in a way that each element represents a word or a token. These vector representations are mostly filled with 0, indicating that particular vocabularies were not used in the sentences. Some examples of lexical retrievers that use sparse representation are TF-IDF, bag-of-words (Zhang et al., 2010), and BM-25 (Robertson and Zaragoza, 2009). Generally, scoring functions of lexical retrieval systems can be written as

$$s = \sum_{t \in q \cap d} \sigma_t(h_q(q,t), h_d(d,t))$$

where q and d are the sets of query tokens and document tokens respectively and h_q and h_d are functions that extract term information, and σ_t is a term scoring function.

A major drawback of using sparse representation is that the contexts of texts are not taken into account since only word or token matchings are used to calculate the scores. As an example, a lexical retriever may consider 'run' and 'ran' as perfectly irrelevant words while giving high relevance score between a text related to "river bank" and a text related to "financial bank" due to the word "bank" even though the meanings are unrelated to each other.

Dense representations are vectors that encode texts into a vector with dimensions usually much lower than the number of vocabularies in the language. Recently, in the past few years, many information retrieval models that use PLMs such as BERT (Devlin et al., 2018) to encode sentences into dense representations have been proposed.

2.2. Cross-Encoder

Cross-Encoder models use cross-attention between document and query vectors to encode semantically meaningful vectors based on both document and query. (Nogueira and Cho, 2019) utilized BERT to encode both passage and query and used the $<CLS>$ vector output to quantify the relevance. This model significantly outperformed BM-25, but it had a major drawback of having a long computation time.

2.3. Bi-Encoder

Bi-encoder is a variation of the siamese neural network where two different inputs are passed through a neural network so that the output vectors can be compared to each other using a similarity function such as Euclidean distance, dot product, or cosine similarity. Using bi-encoders instead of cross-encoders has the advantage that it reduces the computation time significantly. Reimers and Gurevych (2019) showed that using BERT as a bi-encoder architecture can reduce

the time to find the pair of sentences with the highest similarity among 10,000 sentences from 65 hours to 5 seconds compared to using BERT as a cross-encoder.

2.4. Negative Sampling

Many studies have shown that the performance of dense information retrieval models based on neural networks varies based on how to sample negative samples, which are documents used in the training steps as samples irrelevant to a query. Nogueira and Cho (2019) used passages that are irrelevant but gives high BM-25 scores as negatives. Karpukhin et al. (2020) showed that in Dense Passage Retriever, using both in-batch negatives and irrelevant passages that are highly ranked as additional hard negatives gave considerable performance gain. Xiong et al. (2020) proposes Approximate nearest neighbor Negative Contrastive Learning (ANCE) that further improves the performance than in-batch negatives. It periodically re-indexes all documents for each query in separate GPUs using a recent checkpoint during the training and uses irrelevant passages or documents that are highly ranked to train the model. Gao et al. (2021b) uses irrelevant passages or documents that are highly ranked by a retrieval model for negative samples to train a model for the reranking stage.

3. Approach

We adopt variants of two bi-encoder models, ColBERT and COIL. To encode both the discharge notes and the code synonyms, we use RoBERTa-PM, a transformer-based language model pre-trained using biomedical domain texts including the MIMIC-III dataset.

3.1. Preprocessing

We use the code synonyms used in ICD-MSMN. For each ICD code, we concatenate the code description and three synonyms of each code and then truncate the text to have a size of N_C.

Similarly, each discharge note is truncated to have a maximum length of N_S tokens after being tokenized using the RoBERTa-PM tokenizer.

For COIL and ColBERT, we divide the tokens for discharge notes into a minimal number of chunks such that the size fits with the maximum input size of RoBERTa-PM, which is 512. The final relevance score between each discharge note and each ICD code is the maximum score among the relevance scores calculated between each chunk of the discharge note and the ICD code.

3.2. Inputs

In general, queries in the information retrievals are relatively short, and the texts to be retrieved are long. However, this is different in the ICD coding task as it has an objective to retrieve and rank code descriptions for each discharge summary. Using discharge summary as a query did not give good results, especially in ColBERT where we append $<mask>$ tokens up to N_S tokens. As a result, we switched the input of the query and documents when inputting to the models compared to the original ColBERT and COIL.

3.3. A variant of COIL

COIL (Gao et al., 2021a) uses contextualized exact matches, which use the classical lexical retrieval methods on dense vectors.

3.3.1. Encoding

We prepend the $<CLS>$ token and append the $<SEP>$ token to Q and each chunk of D and then encode them using RoBERTa-PM.

The output of the CLS vector goes through a linear layer without activation layer with output dimension of 768. Similarly, the output of other encoded vectors are passed through a linear layer that has same shape and different parameters.

3.3.2. Interaction

For each query vector that corresponds to a token in Q and each chunk of D, the maximum dot product between the query vector and the document vectors of the chunk with exact token matches are used to calculate the token scores. The token scores can be written as

$$s_{tok}(q,d) = \sum_{q_i \in Q \cap D} \max_{d_j = q_i} (v_i^{q^T} v_j^d)$$

where q_i, d_j, v_i^q, and v_j^d the ith query token, the jth document token, the encoded vector of the ith query token, and the encoded vector of the jth document token respectively.

The dot product between the CLS vectors is added to the token score to calculate the relevance score as

$$s(q,d) = s_{tok}(q,d) + {v_{cls}^q}^T v_{cls}^d$$

where v_{cls}^q, v_{cls}^d are the encoded CLS vectors of Q and D respectively.

3.4. Variant of ColBERT

ColBERT uses an all-to-all match between the encoded vectors corresponding to document tokens and query tokens.

3.4.1. Encoding

To acquire vector representations for each code description, Q is padded using the RoBERTa-PM's special token <*mask*> such that the length of Q has a pre-defined length of N_C. Q is truncated to have a maximum length of N_C if the length of Q is bigger than N_C. Also, a special token <Q> is prepended to N_C.

For each chunk of D, a special token <D> is prepended.

Both modified chunks of Q and D are encoded using the RoBERTa-PM and passed through a linear layer with an output dimension of 768 and without activations. Finally, the encoded query is normalized so that each token vector has an L2 norm of one.

3.4.2. Interaction

The final relevance score is the sum of the maximum dot product between each vector that corresponds to a token in C_p and vectors in E_D as follows:

$$s(q,d) = \sum_{q_i \in Q} \max_j (v_i^{qT} \cdot v_j^d)$$

using the same denotation used for the interaction of COIL.

3.5. Loss

The loss function is calculated as the cross-entropy loss when the text for one relevant ICD code d^+ and texts for at least one negative ICD code $\{d_1^-, d_2^-, \ldots, d_l^-, \ldots\}$ are given for a discharge note in each training step.

$$\mathcal{L} = -\log \frac{\exp\left(s(q,d^+)\right)}{\exp\left(s(q,d^+)\right) + \sum_l \exp\left(s(q,d_l^-)\right)}$$

4. Experiment

4.1. Dataset

MIMIC-III dataset (Johnson et al., 2016) was used for the experiment. MIMIC-III is a freely accessible critical care database with about 53 thousand de-identified medical records. In particular, the dataset had 47723, 1631, and 3372 medical notes and their corresponding ICD-9 codes for the dev and training dataset. The number of the ICD-9 code is 8922. The average number of tokens among medical notes when tokenized using the tokenizer used for RoBERTa-PM (Lewis et al., 2020) was about 3351 tokens. Also, the 0, 25, 50, 75, and 100 percentile of the number of tokens were 225, 2114, 3060, 4226, and 25079, respectively. About 29.1% of the medical notes were tokenized to more than 4000 tokens. The average number of relevant ICD codes for each medical note was about 15.9.

4.2. Evaluation Metrics

We evaluated results based on AUC Macro, AUC Micro, F1 Macro, F1 Micro, precision@5, precision@8, and precision@15.

4.3. Configuration

We truncate the texts for code to $N_C = 30$ tokens and discharge summaries to $N_S = 4000$ tokens.

For negative sampling, we used random selection. For the MIMIC3-full dataset, we used 479 negative codes in each training step, which is about 5.4% of the code descriptions. For the MIMIC3-50 dataset, we used all negative codes in each training step. The performance of using in-batch negative was good likely because of the small batch size available with the same GPU memory constraint.

For both ColBERT and COIL, we train with a learning rate of 1e-6 with a 0.1 warmup ratio and linear decay. The models were trained for three epochs for the MIMIC3-full dataset and ten epochs for the MIMIC3-50 dataset.

4.4. Results

The results for the full MIMIC3 dataset and the MIMIC3-50 dataset are shown in Table 1 and Table 2 respectively.

ColBERT performed better than COIL, suggesting that the relevance scores are calculated better based

Table 1: Results on the full MIMIC-III test dataset

Model	Negative Sampling	AUC		F_1		P@k	
		Macro	Micro	Macro	Micro	P@8	P@15
COIL	random 479	90.6	97.8	3.2	37.0	51.1	39.8
ColBERT	random 47						
‘ ColBERT	random 479	94.6	98.8	9.5	52.2	68.5	54.2
ColBERT	ANCE + random 479						

Table 2: Results on the full MIMIC-III 50 test dataset

Model	AUC		F_1		
	Macro	Micro	Macro	Micro	P@5
COIL	87.2	90.5	54.5	62.5	61.5
ColBERT	91.9	94.1	66.7	71.0	66.8

more on contextualized semantic meanings than by exact word matches.

Also, using a bigger number of negatives gave a better performance for ColBERT. This is expected because using more negatives will more likely allow the model to be trained with meaningful negatives.

Table 3 and Table 4 show the comparison of the state-of-the-art models against ColBERT. While our best model underperformed compared to the state-of-the-art models in the MIMIC3 full dataset, it performed better than many of the listed models in the MIMIC3-50 dataset. The difference in the relative performance can be explained by the fact all of the negative samples were used to train the MIMIC3-50 dataset in each training step, while a small portion was used for the MIMIC3 full dataset.

5. Explainability

The results of both ColBERT and COIL are explainable by finding the token in the discharge notes that gave the best cosine similarity and dot product respectively for each query. In particular, the text of a discharge note that showed highest relevance score with the biggest number of query tokens seemed to show big relevance with the query. Some examples are shown below.

6. Limitations

The performance of our models under-performed compared to the state-of-the-art models. However, this is reasonable based on two reasons that show potential for further improvements. The first reason is that we used bi-encoder models, while cross-encoder models typically outperform bi-encoder models. The second reason is that using better negative samples could give better results. Using ANCE to sample negative ICD codes seems promising based on its performance in the MS Marco leaderboard.

The main drawback of using cross-encoder models and ANCE is that they require multiple times of computation to train models, which is why we used bi-encoders in this paper. However, considering the high cost of manual ICD coding tasks, which require expertise, we speculate that the approach will be worth it.

7. Conclusion

We tested variants of two bi-encoder models used for the information retrieval task to tackle the multi-label task of the auto ICD coding. The information retrieval approach is different from other multi-label classification approaches mainly because, typically, only a small portion of text data to be retrieved is used for training in each step due to computation or memory limitations. As a natural result, the performance of information retrieval models depends highly on how to choose texts irrelevant to the query for each training step. In particular, we showed that the performance varied on the number of irrelevant ICD codes used in each training step when the irrelevant ICD codes were sampled randomly.

Table 3: Results on the full MIMIC-III test dataset

Model	AUC		F_1		P@k	
	Macro	Micro	Macro	Micro	P@8	P@15
CAML	89.5	98.6	8.8	53.9	70.9	56.1
MSATT-KG	91.0	**99.2**	9.0	55.3	72.8	58.1
MultiResCNN	91.0	98.6	8.5	55.2	73.4	58.4
HyperCore	93.0	98.9	9.0	55.1	72.2	57.9
LAAT	91.9	98.8	9.9	57.5	73.8	59.1
JointLAAT	92.1	98.8	10.7	57.5	73.5	59.0
PLM-ICD	92.6	98.9	10.4	**59.8**	**77.1**	**61.3**
ISD	93.8	99.0	11.9	55.9	74.5	
RAC	94.8	**99.2**	**12.7**	58.6	75.4	60.1
MSMN	**95.0**	**99.2**	10.3	58.4	75.2	59.9
ColBERT	94.6	98.8	9.5	52.2	68.5	54.2

Table 4: Results on the MIMIC-III 50 test dataset

Model	AUC		F_1		
	Macro	Micro	Macro	Micro	P@5
CAML	87.5	90.9	53.2	61.4	60.9
MSATT-KG	91,4	93.6	63.8	68.4	64.4
MultiResCNN	89.9	92.8	60.6	67.0	64.1
HyperCore	89.5	92.9	60.9	66.3	63.2
LAAT	92.5	94.6	66.6	71.5	67.5
JoinLAAT	92.5	94.6	66.1	71.6	67.1
MSMN	**92.8**	**94.7**	**68.3**	**72.5**	**68.0**
ColBERT	91.9	94.1	66.7	71.0	66.8

We have shown that a bi-encoder model with a very light interaction between texts for each query and each document could perform well when we use all negative samples to train the model in the MIMIC-III 50 dataset. While it is not in the scope of this paper, we speculate that using cross-encoders and the ANCE or LCE methods will significantly improve the performance.

References

Yuwen Chen and Jiangtao Ren. Automatic icd code assignment utilizing textual descriptions and hierarchical structure of icd code. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 348–353. IEEE, 2019.

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.

Luyu Gao, Zhuyun Dai, and Jamie Callan. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. *arXiv preprint arXiv:2104.07186*, 2021a.

Luyu Gao, Zhuyun Dai, and Jamie Callan. Rethink training of bert rerankers in multi-stage retrieval pipeline. In *European Conference on Information Retrieval*, pages 280–286. Springer, 2021b.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(160035), 2016. doi: https://doi.org/10.1038/sdata.2016.35.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

Leah S Larkey and W Bruce Croft. Combining classifiers in text categorization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 289–297, 1996.

Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov. Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 146–157, 2020.

Fei Li and Hong Yu. Icd coding from clinical text using multi-filter residual convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8180–8187, 2020.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*, 2020.

Zheng Yuan, Chuanqi Tan, and Songfang Huang. Code synonyms do matter: Multiple synonyms matching network for automatic icd coding. *arXiv preprint arXiv:2203.01515*, 2022.

Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1):43–52, 2010.

Giulio Zhou and Jacob Devlin. Multi-vector attention models for deep re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5452–5456, 2021.

Appendix A. First Appendix

This is the first appendix.

Appendix B. Second Appendix

This is the second appendix.