IR Based Chatbot: By tuning BERT architecture

Puneet Mangla*
cs17btech11029@iith.ac.in
IIT Hyderabad
India

Yash Khasbarge*
cs17btech11044@iith.ac.in
IIT Hyderabad
India

1 PROBLEM STATEMENT

The problem statement is to build a Information Retrieval (IR) based chat-bot which assigns a correct response (Utterance) among set of responses (Utterance + Distractors) for a given context of information. Following are some definitions needed to formulate the objective of IR-based Chat-bot.

- **Context**: Conversation up-to this point.
- **Utterance**: Correct response to context.
- **Distractors**: Incorrect response to context.

The objective is to develop a ranking model $\mathcal{R}$ which takes a context c and a response r as an input and outputs a relevance score $\mathcal{R}(c, r)$ such that

$$\mathcal{R}(c, r_u) > \mathcal{R}(c, r_d^i) \; \forall i$$

Here r_u is the utterance and r_d^i is the i^{th} distractor in set of responses. In other words, the ranking model should assign highest score to true utterance and lower scores to distractors. Figure 1 illustrates the objective pictorially.

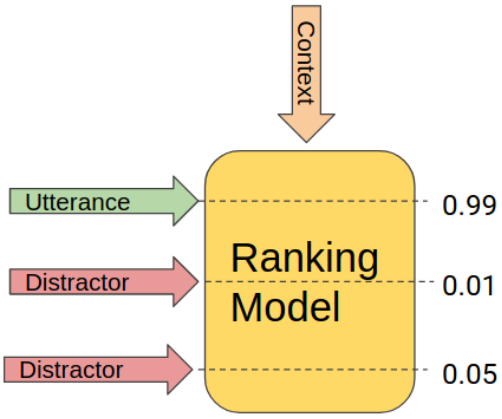

Figure 1: Illustration of IR based chatbot as a Ranking model.

2 OUR APPROACH: USING BERT FOR BINARY CLASSIFICATION

We pose the task of learning a Ranking model as a binary classification task wherein we have a set of triplets $\mathcal{D} = \{(c, r, y)^i\}_{i=1}^N$ as our training examples. Here c is the context, r is a response and $y = 1$ iff corresponding response is correct (utterance) response to context, else 0. Figure 2 illustrates our approach.

At first, we use sate-of-the-art transformer architecture, BERT [2] $\mathcal{B}$ to learn semantically rich - low dimensional embeddings, $\mathcal{B}(c)$ and $\mathcal{B}(r)$ for context c and response r respectively. Since context embedding, $\mathcal{B}(c)$ and response embedding, $\mathcal{B}(r)$ lies in different subspaces, we use a projection matrix/MLP M to transform $\mathcal{B}(c)$ to same space as $\mathcal{B}(r)$. Thus the final context embedding is calculated and represented as $M(\mathcal{B}(c))$.

After projecting both embeddings to same subspace, we take the dot product between $M(\mathcal{B}(c))$ and $\mathcal{B}(r)$ to obtain a scalar score, $M(\mathcal{B}(c)) \circ \mathcal{B}(r)$. The rationale behind taking the dot product, is that a higher positive dot product will mean that the context c and response r are similar whereas a lower negative dot product will mean the opposite.

Since we pose the ranking model as a binary classification task, we pass the obtained dot-product, $M(\mathcal{B}(c)) \circ \mathcal{B}(r)$ through a Sigmoid activation function σ to obtain the probability ($P(r = utterance|c) = \sigma(M(\mathcal{B}(c)) \circ \mathcal{B}(r))$) of response r being the correct response (utterance) to context c. After obtaining the probability, we use the Binary Cross Entropy (BCE) Loss to optimize the modules, $\mathcal{B}$ and M as follows:

$$\begin{aligned}\mathcal{B}^*, M^* &= \arg\min_{\mathcal{B},M} \mathbf{E}_{(c,r,y)\sim\mathcal{D}} \; BCE(P(r = utterance|c), y) \\ &= arg\min_{\mathcal{B},M} \mathbf{E}_{(c,r,y)\sim\mathcal{D}} \; -y \cdot log(\sigma(M(\mathcal{B}(c)) \circ \mathcal{B}(r))) \\ &\quad - (1-y) \cdot log(1 - \sigma(M(\mathcal{B}(c)) \circ \mathcal{B}(r)))\end{aligned} \quad (1)$$

Like shown in Figure 2, in hope to further improve the performance of our pipeline, we also propose to regularize the BERT embeddings through an additional self-supervision task which tries to predict the randomly masked tokens in the input text. We found that a recent version of BERT called as ALBERT [3] offers same regularization with almost 16 times less parameters and computation requirement as compared to vanilla BERT. Hence, we decide to fine-tune it.

3 EXPERIMENTS

3.1 Dataset Details

The Ubuntu Dialog Corpus (UDC)[4] is one of the largest public dialog datasets available. It is based on chat logs from the Ubuntu channels on a public Freenode Ubuntu IRC network. The training data consists of 1,000,000 examples, 50% positive (label 1) and 50% negative (label 0). Each example consists of a context, the conversation up to this point, and an utterance, a response to the context. A positive label means that an utterance was an actual response to a context, and a negative label means that the utterance wasn't – it was picked randomly from somewhere in the corpus. After necessary tokenization, stemming, and lemmatization, the maximum context is 86 words long and the maximum utterance is 17 words long. Check out following public Jupyter notebook to see the data analysis. The test set contains 11-tuples $\{(c, r_1, r_2, ..., r_{10})^i\}_{i=1}^N$. The

*Both members contributed equally.

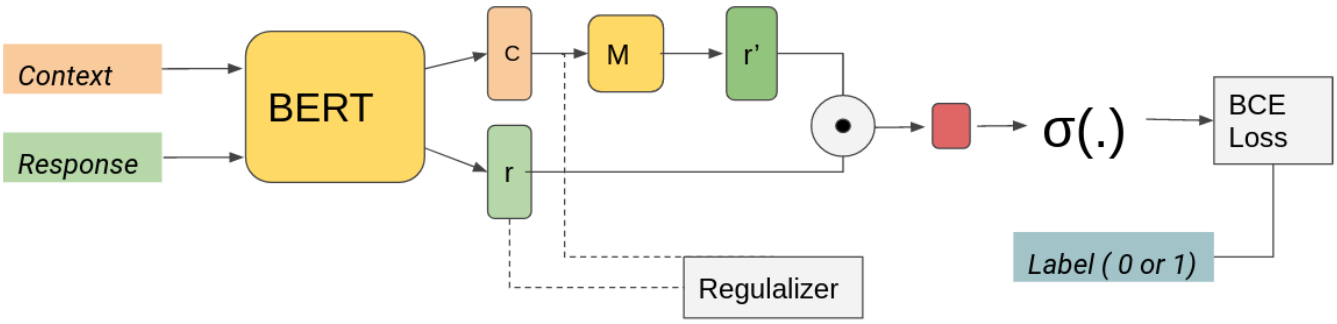

Figure 2: Pipeline of Our Approach

c^i's are contexts and the $r_1^i, r_2^i, ..., r_{10}^i$ are the available responses. The r_1^i is the correct response and rest 9 are distractors.

3.2 Implementation Details

We use the PyTorch framework [5] for implementation. The BERT and ALBERT models along with their pretrained weights were obtained from [6]. We trained using a single RTX 1080 GPU with batch size of 64. We used AdamW optimizer with base learning rate 2×10^{-5}.

The vocabulary present in UDC dataset is more related to Computer Science and Operating Systems. Accordingly, naively using default BERT vocabulary was found to be deteriorating. We thus, concatenated the default BERT vocabulary with the UDC vocabulary. We selected the tokens with minimum frequency of 10, and chose the alphabet limit to be 1000 and vocabulary size to be 30000. Finally, the concatenated vocabulary had around 50K tokens. The importance of the UDC vocabulary can be seen form the fact that the vocabulary was increased by 20000 tokens. Effect of this was further reflected in the improved results using the concatenated vocabulary.

3.3 Evaluation Metrics

We evaluate our models using *Top-K* metric. Top-K measures if utterance lies among the top-k relevant documents retrieved by the learning algorithm. Mathematically, if $\{r_1^i, r_2^i ..., r_{10}^i\}$ be the set of given response choices corresponding to a context c^i, r_1^i is utterance and $r_2^i, r_3^i ... r_{10}^i$ are distractors, the Top-K score is given as

$$\text{Top(k) Score} = \frac{\sum_{i=1}^{N} (\pi(r_1^i) \leq k)}{N}$$

where $\pi(r)$ gives the relevance index of document r in ordered list (descending order of relevance) obtained using learning algorithm.

3.4 Baselines and Variants

We train following variants of our approach:

- *Ours-BERT (50K)* is pipeline trained using our approach using BERT transformer on 50K examples.
- *Ours-BERT (500K)* is pipeline trained using our approach using BERT transformer on 500K examples.
- *Ours-ALBERT (50K)* is pipeline trained using our approach using ALBERT transformer on 50K examples.

We compare above variants of our approach with following baselines which are trained on 1M datapoints.

- *Random Baseline* selects the utterance among all responses given
- *Dual-Encoder LSTM [4]*
- *Dual Encoder RNN [1]*
- *TF-IDF model*

3.5 Results, Observations and Conclusions

Table 1 reports the performance of baselines and our approach when evaluated using Top-1, Top2 and Top-5 metrics. From table, we draw following observations/conclusions:

- With just half training examples (500K) our approach is:
  - better than Random, Dual-Encoder RNN and TF-IDF model is most cases.
  - Comparable to Dual-Encoder LSTM which gives best performance
- Performance of Ours-BERT on 50K and 500K examples suggest that more training data can improve performance [1].
- By comparing ours-BERT and ours-ALBERT:
  - Top-2 performance of ours-ALBERT is fairly better that ours-BERT. Top1 and Top5 are comparable.
  - We observe that convergence of Ours-ALBERT was slow - possibly because of additional SSA task.
  - While BERT achieved 90% training accuracy in just 15 epochs, ALBERT only achieved 60%. This and previous observations suggest that more epochs and data might be needed to better tune ALBERT embeddings.

4 EXAMPLES OF CLASSIFICATION ON TEST SET

We have noted some correctly classified examples from test set by fine-tuned ALBERT. We report the top-1 correctly classified example in Table 2 and top-3 correctly classified example in Table 3. The **bold** font denotes the correct response.

In Table 2, the context deals with issues in installing wubi, and the model has correctly identified that re-downloading is the most feasible option from the responses. Since, re-downloading in case

[1] Due to compute limitations we are unable to train on complete 1M dataset

Method	#Samples	Top-1	Top-2	Top-5
Random	-	0.1	0.2	0.5
Dual-Encoder LSTM	1M	**0.49**	**0.68**	**0.91**
Dual-Encoder RNN	1M	0.38	0.56	0.83
TF-IDF	1M	0.49	0.587	0.76
Ours-BERT	50K	0.235	0.371	0.71
Ours-BERT	500K	0.43	0.61	0.84
Ours-ALBERT	50K	0.22	0.4680	0.68

Table 1: Our approach and baseline evaluated using Top-1,2,5 metrics.

context	Hello guys. I have a problem installing ubuntu with wubi. At the end this error occurs - > http://shrani.si/f/1s/4o/1n9nru75/capture.png My only idea here be you run wubi as admin? Hmm what window do you use ? Help shoot me in the foot I'm out of idea I tried with cd boot and installation gets terminated by signal 15 [9028]
1	**Try re-download?**
2	Try the nvidia-one
3	I've never done it myself
4	I find /usr/share/x11/xerrordb through the manpage
5	It is not valgrind by any means but it help sometimes

Table 2: Correctly classified example in Top-1

of issues is a general solution, we believe that the example was quite easy and hence, was classified correctly.

In Table 3, the context is about dualboot and SSDs, but hardware specific words are used more frequently. We see that the correct response was ranked as 3^{rd}. It can also be seen that the rank 1 response is also related to hardware. We believe that our model is not able to sufficiently utilize the domain specific vocabulary.

5 FUTURE STEPS

We believe that a major step in future will be to use whole 1M data-points in the training split to train models using both variants of our approach. Secondly, we feel the need to explore other self-supervision tasks which can be added in order to improve our existing pipeline.

context	Can someone help me out? I'm having a problem with dualboot a sony vaio in raid 0 How did you create the raid 0? It's hardware raid - two 256 GB SSDs so which raid control do you use then ?
1	Whatever process can run the video. So it may be vlc or the browser for flash. A co-worker said that legacy hardware will usually generate a warning upon OS installation. I get no such warning.
2	You noticed that's an IPv6 port only? Do you expect an IPv4 port too? Indeed. Does the vpn host operate a firewall independent of the guest ?
3	**Um I'm not entirely sure... That's why I can use some help. I tried dualboot ubuntu, but after installing 12.04 the Windows 7 loader didn't appear in grub.**
4	Space is space, is all I mean.
5	NTPD won't work if your clock be too far out.

Table 3: Correctly classified example in Top-3

REFERENCES

[1] Alessandro Bay and Biswa Sengupta. 2017. Sequence stacking using dual encoder Seq2Seq recurrent networks. *CoRR* abs/1710.04211 (2017).

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[3] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1eA7AEtvS

[4] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, Prague, Czech Republic, 285–294. https://doi.org/10.18653/v1/W15-4640

[5] Adam Paszke, S. Gross, Francisco Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, Alban Desmaison, Andreas Köpf, E. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, B. Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *ArXiv* abs/1912.01703 (2019).

[6] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6