CS224u lit review

Lara Thompson
Principle Data Scientist @ Salesforce
lara.thompson@salesforce.com

March 7, 2023

1 General problem/task definition

For my final project, I want to explore using large language models in a recommender system for deployment at scale with limited resources. As mentioned in the course, recommender systems are similar to retrieval systems albeit with a different aim: rather than searching for a few key results to answer a query, a recommender system has no query and should sometimes be less reliant on a current context (even if some information is given, e.g. last book read, we still don't know what the reader is in the mood for next). Regardless, there is much to learn from information retriever systems such as ColBERT (Khattab and Zaharia, 2020), particularly for its introduction of late interactions.

Sentence-BERT (Reimers and Gurevych, 2019) is a good paper to compare with ColBERT in their usage of pre-trained BERT and arguably is a better starting place for ColBERT's fine-tuning that BERT itself. "A Simple but Tough-to-Beat Baseline for Sentence Embeddings" (Arora et al., 2017) is a very interesting word embeddings aggregation approach that may suffice, especially if system performance is more important than the last few points of accuracy.

Rec-BERT fine-tunes BERT to improve item-to-item text content-based recommendations (Malkiel et al., 2020); at that time, recommenders with text content that also considered user/item attributes and interactions used custom text embeddings that must be trained separately for each application / domain.

More recently, researchers from Alibaba showcase M6-Rec (Cui et al., 2022) for all their downstream tasks (recommendations, chat, personal product design) by converting these tasks to text prompts. Although, this may a preview of where recommenders are headed, as we'll see, the LLMs may not be ready in their current state.

2 Concise summaries of the articles

2.1 A Simple but Tough-to-Beat Baseline for Sentence Embeddings (Arora et al., 2017)

In contrast to Sentence-BERT, Arora et. al. from Princeton developed a very simple sentence embedding motivated from corpus generation theory. They take a weighted average of the word embeddings (downweighting common words) then subtract out the projection to the principal component (as estimated across several sentences; to remove another common discourse component).

The motivation for their approach is a modified random walk model of corpus generation. Rather than generating only words near a slowly changing discourse vector $c_s(t)$, they allow for the large deviations to common words ('the', 'and', etc.) in two ways: first, by separating a common discourse component c_0 and, second, by allowing any word to appear out of context in proportion to their typical frequency $p(w)$. Note that c_s, c_0 and v_w, the word embedding of w all reside in the same embedding space. The original probability of observing a word w is:

$$\Pr[w(t)|c_s(t)] \propto \exp\left(\langle c_s(t), v_w\rangle\right) \quad (1)$$

After the two modifications, this becomes:

$$\Pr[w(t)|c_s(t)] = \alpha p(w) + (1-\alpha)\frac{\exp\left(\langle \tilde{c}_s(t), v_w\rangle\right)}{Z_{\tilde{c}_s}} \quad (2)$$

where $\tilde{c}_s = \beta c_0 + (1-\beta)c_s$. The authors go on to show how this relates to a $p(w)$-weighted embedding average that then has its component along the principle component $\sim c_0$.

They test their sentence embeddings across the STS benchmarks and compare against other sentence embedding approaches (not Sentence BERT since it came out later!). They find that they beat many more complicated approaches in text similarity tasks but that the RNN approaches outperform them on sentiment tasks. It seems a bag-of-words approach cannot capture the combined sentiment; in fact, this approach may specifically downweight various negation terms (e.g. "not") because they are common.

2.2 Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks (Reimers and Gurevych, 2019)

Reimers and Gurevych develop sentence embeddings that improve over various word embeddings-based approaches for text similarity tasks, that, nevertheless, scaled far better than BERT / RoBERTa that must process every pair individually.

They use pre-trained BERT and RoBERTa in two main configurations (usually SRoBERTA lagged SBERT and will be dropped from the summary): in a siamese network fine-tuned on SNLI (Bowman et al., 2015), Multi-Genre NLI (Williams et al., 2018) and (separately) on STS Argument Facet Similarity (Misra et al., 2016) and STS benchmarks (Cer et al., 2017); and in a triplet network using the Wikipedia sections distinction dataset (Dor et al., 2018). They test both BERT-base and BERT-large.

The SNLI and multi NLI datasets are familiar from the course; the AFS dataset similarly has labelled pairs with a 0-5 similarity rating but, being excerpts from dialogue/arguments, the notion of similarity extends to the line reasoning of the argument. While SBERT-NLI[1] typically outperforms direct application of BERT on STS benchmarks; on the AFS dataset, the excerpt-spanning attention in BERT appears to be important and SBERT-AFS lags by several points in this task.

The Wikipedia sections distinction dataset involves triplets: with two passages from one section of a Wikipedia article and a third from another section. They train with a triplet network (the three passages go through a single BERT model) and use a triplet loss that pushes the embeddings for similar passages closer and the dissimilar passages further in embedding space. This must be a challenging dataset to learn from: the sentences within a given section are not always semantically related; but it's an ingenious way to generate a very large dataset. SBERT-WikiSec does quite well.

As a final evaluation, SBERT-NLI is tested in a transfer learning setting (Conneau and Kiela, 2018): even though SBERT is intended to be fine-tuned to each task (here it is tested on held out tasks), it outperformed many other sentence embeddings in all but a few tasks, even using BERT-base.

2.3 ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT (Khattab and Zaharia, 2020)

ColBERT sets out to leverage the full quality of BERT for contextualized information retrieval while reducing query time by two orders of magnitude. Their design lands in the happy middle ground for query-document interactions in the network: late enough that the bulk of the expensive computations can be precomputed and indexed and yet there is an interaction layer outperforms completely separate query/document embeddings.

Like SBERT, the same BERT model is used to encode both query and document ([Q]/[D] markers are prepended to denote which a sequence is). The query is padded with the special [mask] token that effectively augment the query in a differentiable way (effectively, their embeddings are infilled while attending to the rest of the query). The output layer is fed into a linear layer to reduce the final embedding dimensionality. These embeddings are used in the late interaction by taking the sum of maximum similarity between each query token and set of document tokens. The BERT layers are fine-tuned while the linear layer and [Q]/[D] marker's embeddings are trained from scratch using a pairwise softmax cross-entropy loss on triples of $\langle q, d^+, d^- \rangle$ (a query and positive + negative document match).

The form of this late interaction allows for very efficient querying: from faster loading of embeddings into the GPU to a form amenable to optimized large-scale vector-similarity search (specifically, faiss (Johnson et al., 2017)).

ColBERT was evaluated on two information retrieval benchmark datasets: MS Marco Ranking (Nguyen et al., 2016) and TREC Complex Retrieval (Dietz et al., 2017). On MS Marco in "re-ranking" mode, the MRR@10 of ColBERT using BERT-base outperformed the far more costly direct adaptation of BERT-base to ranking (Nogueira and Cho, 2019). In "end-to-end" mode, Colbert with BERT-large approaches the performance of BERT-large re-ranker, despite being 500x faster.

In the ablation study, the biggest improvements in MRR@10 were for maximum similarity in the late interaction (versus taking the average query

[1] Notation: SBERT-DATASET denotes SBERT fine-tuned on DATASET.

to document embeddings) and for adding query augmentation via [mask] token padding.

2.4 RecoBERT: A Catalog Language Model for Text-Based Recommendations (Malkiel et al., 2020)

RecoBERT is an early example of using a pre-trained LLM for text-based item recommendations, a pure item-to-item recommender system. Earlier attempts that also incorporated context and/or user-item interactions (e.g. (Djuric et al., 2015), (Zheng et al., 2017) or (de Souza Pereira Moreira et al., 2018)) trained text embeddings from scratch with custom networks, making that most likely more costly to develop and less transferable.

Title, document pairs are input to BERT-large as the sequence

[CLS][title tokens][SEP][document tokens]

15% of the tokens are masked. The output embeddings for the title and document are separately averaged to give title and document level embeddings. The title-description model (TDM) loss is:

$$\mathcal{L}_{TDM} = -\frac{1}{2}\sum_{i=1}^{n} \left(y_i \log\left(C^i_{TDM}\right) + \right. \\ \left. (1-y_i)\log\left(1-C^i_{TDM}\right)\right) \tag{3}$$

where C_{TDM} is the cosine distance between the title/document embeddings. A mask language model (MLM) component (following (Devlin et al., 2019)) includes a classification layer mapping the BERT [CLS] embedding back to vocabulary space; the total loss becomes

$$\mathcal{L} = \mathcal{L}_{TDM} + \mathcal{L}_{MLM} \tag{4}$$

For inference, to test the similarity of a candidate title′, document′ to a known title, document they compute: cosine similarity of titles, cosine similarity of documents and the TDM model cosine distance crossed title, document′ and title′, document pairs; the sum of which defines their similarity metric.

They evaluate RecoBERT on two datasets: a wine review catalogue (zackthoutt@Kaggle, 2017) (with an expert annotated test set[2]), and a fashion catalog (no further details provided, presumably proprietary). RecoBERT is trained separately for each. They compare against other sentence embeddings, pre-trained BERT without fine-tuning, and BERT fine-tuned to each domain (on the same reviews that RecoBERT trains on but without the TDM head). RecoBERT outperformed the other approaches, often by a large margin, as quantified by either mean reciprocal ratio (MMR) or hit ratio (HR) for various top-k.

2.5 M6-Rec: Generative Pretrained Language Models are Open-Ended Recommender Systems (Cui et al., 2022)

M6 is a multi-modal (text and image) multilingual (Chinese and English) model and is already in wide use in Alibaba (Lin et al., 2021) (of the same generation as GPT-3). M6-Rec uses a pre-trained M6 model and reformulates all many downstream tasks as prompts by representing user behavior data as natural language plain text as a preface to recommended content using the special input format of M6. The plausibility of that recommendation is quantified using the probability of the output tokens for the recommendation half of the input sequence.

They further optimize is several ways. Inspired by ColBERT, late interactions segment the user attributes (e.g. "Male") and actions ("Clicked X.") to be precomputed for the bulk of the M6 layers. They are concatenated for the final 3 layers (with additionally learned positioning embeddings); interactions are only modelled in those last layers. They propose *option tuning* (a modified prompt tuning (Lester et al., 2021)); combined with *adaptation* (Hu et al., 2022), they achieve better CTR prediction than a fully fine-tuned M6. They need less than a million samples to out-perform the baseline model, DIN (Zhou et al., 2018), given >100x more samples. Their one-shot performance on Amazon movie[3] and Amazon Cloth[4] is only matched by DIN after 40-50k samples, and they match DINs maximum performance with 400-shot learning.

To further downsize, they distill, prune and quantize M6 down to from 300M parameters to the 2M parameter M6-edge model. They use early-exiting to further minimize inference time. The option tuning and adaptation can be fine-tuned on the M6-edge model directly on customers phones. They

[2] See the author's Github page for this paper, https://github.com/r-papso/recobert

[3] Presumably https://snap.stanford.edu/data/web-Movies.html. They have an unrelated reference in their paper: an error?

[4] Not sure which dataset, since again an unrelated reference is given and many such datasets exist.

report M6-edge performance on Alibaba specific tasks and report only slight degradation from a full M6-base (they do not compare with M6-Rec).

The example they give of a personalized product design shows gender bias inherent in LLMs: given the context that a user

> "clicked product of category flowers and plants named Stephanotis floribunda, potted plants, evergreen, absorbing formaldehyde"

and

> "clicked product of category seasonings named Jiangxi dry fermented soybeans, handmade, black soybeans, Jiujiang speciality"

the model predicts that this example user is a middle-aged housewife and suggests

> "clicked product of category clothing named dress, middle-age housewife, summer clothing, chiffon, mid-length dresses, short sleeves"

Sadly, the authors do not note this gender bias and think this is a wonderful suggestion. A recent study explores how even a bias-mitigated LLM transfers bias into harmful task-specific behavior after fine-tuning (Steed et al., 2022).

3 Compare and contrast

These papers all pertain to sentence/passage embeddings: the first gives a baseline that should be tested against; SBERT developed versatile sentence embeddings in a python library; ColBERT optimized passage embeddings for fast information retrieval; RecoBERT fine-tuned BERT for item-to-item text recommendations using intermediate title/document embeddings; M6-Rec prompts their LLM in a context/recommendation format so that they can extract the probabilities of the recommendation embedding.

The SBERT paper didn't include "a simple but tough-to-beat baseline for sentence embeddings" (Arora et al., 2017) in their comparisons. Strangely, the two papers state fairly different performance for the same embeddings on the same tasks (e.g. simple average GloVe embeddings (Pennington et al., 2014) on the STS benchmarks); either the GloVe embeddings used were trained on different corpuses or they're reporting different summaries of results across the subtasks ("a simple..." gave mean metrics across STS tasks and used the glove.840B.300d embeddings; the SBERT paper doesn't say). Since both state a 5-20 point improvement over unweighted GloVe embedding averages, it would have been very interesting to see how SBERT performs against "a simple" baseline.

Like ColBERT, RecoBERT must to fine-tuned to each domain; in both cases, much of the heavy computation can be precomputed and indexed, although the RecoBERT authors did not consider this. RecoBERT is trained with a cosine loss to discriminate positive vs negative pairs; while both SBERT and ColBERT trained using a triplet loss.

At the opposite extreme, M6-Rec does not fine-tune their LLM at all, but "augments" the model and trains only those additional terms (<1% of the parameters of the entire model; presumably they must be trained per task). Each downstream task encodes the user and item context in a text prompt.

Another way to have one model for all tasks involves developing user embeddings that include all their attributes and interactions across all services. Every subtask works with these rich user embeddings and encode their products in the same vector space; each recommender can be much lighter weight. As a further benefit, the user feature store need only be accessed by the model training user embeddings.

Many companies are adopting this pattern according to (Yan, 2021): TripAdvisor creates user embeddings by aggregating item embeddings; YouTube aggregates video embeddings then concatenates other user attributes (geography, demographics, the age of the video, etc.); and Spotify learns session-level user embeddings on activity within and across sessions. At StitchFix, their users' clothing preferences and size may change together in time. They record attributes as a function of time so that, for example, a liked shirt style is recalled with the size the user had on file at that time (Zielnicki et al., 2022).

4 Future work

The simplest sentence embeddings may be improved by starting with BERT in-context word embeddings (e.g. taken from the first layer or two as we did in our first assignment). Their performance on sentiment tasks would be a good test since they did relatively poorly at them.

Many studies suggest that a contrastive loss with triplets performs better; triplet loss works with the embeddings directly rather than forcing them first through bottleneck layers (Schroff et al., 2015). If, as in M6-Rec and RecoBERT, the bottleneck is required for downsizing the embeddings, contrastive loss may be fine; but it's worth comparing with triplet loss for any improvements.

Furthermore, the efficacy of triplet training (and presumably contrastive training) relies on finding "good" positive/negative pairs (distinguishable but barely so), though gradient clipping must ease learning from too-hard pairs; in image and speech domains, much attention is given to choosing better pairs (Moindrot, 2018). Adapting these techniques to NLP may improve some of these approaches.

The M6-Rec approach of one model for every subtask is a direction many companies are taking their recommender systems. From their paper, it doesn't seem wise to use a bias-ridden "wisdom of the crowd" for potentially rather specialized domains. To validating the wine recommender, the RecoBERT authors collected expert recommendations for a collection of wines; even though a user-item recommender trained within the wine community may not have that expertise either, it may still outperform the internet.

For better scaling of recommendations, user embeddings seem like the best approach for now. Next is to add more modes of interaction (text first, then image, sound and video) and to formulate in time by learning from sequences of sessions.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. *5th International Conference on Learning Representations, ICLR 2017*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. https://nlp.stanford.edu/projects/snli/.

Daniel Cer, Mona Diab, Eneko Agirre, Iigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. *Proceedings of the 11th International Workshop on Semantic Evaluation*, pages 1–14. https://github.com/brmson/dataset-sts.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. https://github.com/facebookresearch/SentEval.

Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084*.

Gabriel de Souza Pereira Moreira, Felipe Ferreira, and Adilson Marques da Cunha. 2018. News session-based recommendations using deep neural networks. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*, DLRS 2018, pages 15–23, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*.

Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. 2017. Trec complex answer retrieval overview. *Proceedings of Text REtrieval Conference (TREC)*. http://trec-car.cs.unh.edu/.

Nemanja Djuric, Hao Wu, Vladan Radosavljevic, Mihajlo Grbovic, and Narayan Bhamidipati. 2015. Hierarchical neural language models for joint representation of streaming documents and their content. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 248–255, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Liat Ein Dor, Yosi Mass, Alon Halfon, Elad Venezian, et al. 2018. Learning thematic similarity metric from article sections using triplet networks. *Proceedings of the 56th Annual Meeting of the As- sociation for Computational Linguistics*, 2:49–54. https://research.ibm.com/haifa/dept/vst/debating_data.shtml.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, and other. 2022. Lora: Low-rank adaptation of large language models. In *ICLR 2022*.

Jeff Johnson, Matthijs Douze, and Herve Jegou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*. https://github.com/facebookresearch/faiss.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *SIGIR '20: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059. Association for Computational Linguistics.

Junyang Lin, Rui Men, An Yang, Chang Zhou, et al. 2021. M6: A chinese multimodal pretrainer. *arXiv preprint arXiv:2103.00823*.

Itzik Malkiel, Oren Barkan, Avi Caciularu, Noam Razin, and other. 2020. Recobert: A catalog language model for text-based recommendations. pages 1704–1714.

Amita Misra, Brian Ecker, and Marilyn A. Walker. 2016. Measuring the similarity of sentential arguments in dialogue. *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 276–287. https://nlds.soe.ucsc.edu/node/44.

Olivier Moindrot. 2018. Triplet loss and online triplet mining in tensorflow.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, and other. 2016. MS MARCO: A Human-Generated MAchine Reading COmprehension Dataset.

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

Ryan Steed, Swetasudha Panda, Ari Kobren, and Michael Wick. 2022. Upstream Mitigation Is *Not* All You Need: Testing the Bias Transfer Hypothesis in Pre-Trained Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3524–3542, Dublin, Ireland. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1:1112–1122. https://cims.nyu.edu/~sbowman/multinli/.

Eugene Yan. 2021. Patterns for personalization in recommendations and search.

zackthoutt@Kaggle. 2017. Wine reviews. https://www.kaggle.com/datasets/zynicide/wine-reviews.

Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 425–434, New York, NY, USA. Association for Computing Machinery.

Guorui Zhou, Chengru Song, Xiaoqiang Zhu, and other. 2018. Deep interest network for click-through rate prediction. In *KDD 2018*.

Kevin Zielnicki, Dirk Sierag, and Patrick Foley. 2022. Client time series model: a multi-target recommender system based on temporally-masked encoders.