Extremely Fast Fine-Tuning for Cross Language Information Retrieval via Generalized Canonical Correlation

John M. Conroy, Neil P. Molino, and Julia S. Yang

October 31, 2022

Abstract

Recently, work using language agnostic transformer neural sentence embeddings show promise for a robust multilingual sentence representation. Our submission to TREC was to test how well these embeddings could be fine-tuned cheaply to perform the task of cross-lingual information retrieval. We explore the use of the MS MARCO dataset with machine translations as a model problem. We demonstrate that a single generalized canonical correlation analysis (GCCA) model trained on previous queries significantly improves the ability of sentence embeddings to find relevant passages. The dominant computational cost for training is computing dense singular value decompositions (SVDs) of matrices derived from the fine-tuning training data. (The number of SVDs used is the number of languages retrieval views and query views plus 1). This approach illustrates that GCCA methods can be used as a rapid training alternative to fine-tuning a neural net, allowing models to be fine-tuned frequently based on a user's previous queries. This model was then used to prepare submissions for the re-ranking NeuCLIR task.

1 Introduction

Recent work using language agnostic transformer neural sentence embeddings show promise. In this notebook we describe some very low cost methods to fine-tune and adapt sentence embeddings to CLIR. The approach is based on classical statistics and the cost to perform the fine-tuning is considerably less than computing the sentence embeddings.

There are settings where we have a small amount of labeled data, but not enough to even do a fine-tuning of a transformer-based large language model.

2 Methods

Sentences embedding from a modern neural net model, such as produced by a recurrent neural net or a transformer model provide a rich semantic repre-

sentation of the content of a sentence. It is natural to use language agnostic embeddings to find related text. In particular, [6] shows such representations, along with positional information, are very strong methods for finding bitext pairs of across languages. Here, we explore to what extend sentence embeddings can be used for cross-lingual question and answering and more generally cross-lingual information retrieval (CLIR). In the question and answering setting there is some promise in that answers to questions often use much of the same vocabulary and a sentence embedding will reflect this overlap in word usage, especially the overlap include bigrams and trigrams. We test this approach uses translations of the TREC 2002-2004 Novelty track data[1] as well as the MS MARCO data translations [3]. Both of these datasets, provide aligned sentences in a CLIR where the queries are English and the relevant text is in English as well as machine generated translations in Russian, English, and Farsi.

Computing a similarity of a query and all candidate passages can be done by embedding both the query and the passages. For specificity let us denote the embedding query by the vector q and passages p_i with $q, p_i \in \mathbb{R}^n$. For ease of notation we assume the sentence embeddings are on the unit sphere. Thus, the cosine similarity is simply the inner product and we can find the nearest vector to a query q via

$$\tilde{p} = \arg\max_j q^T p_j.$$

If a query has multiple parts the inner product scores can be combined via a reduction operator such as maximum or summation.

The dimension n is generally fairly large, typically 512 to 1024, and it is natural to ask if queries and relevant passages can be made to be closer in a lower dimensional space by employing classical statistics and linear algebra. A natural choice is canonical correlation analysis (CCA) and its generalizations. Previously, the paper [7] uses a correlation matrix method which is a variant of CCA in an image classification. Here, we employ CCA and a *sumcorr* (sums of correlation) which is a generalized canonical correlation (GCCA). There are other generalizations, but we limit our attention to a regularized *sumcorr*. Employing a GCCA allows us to consider embedding multiple parts of a query (e.g., title, description, and narrative) as well as one or more translations of relevant passages into the same space.

The GCCA procedure produce linear transformations $A^{(k)}$ for $k = 1, \ldots, \nu$, which project the n dimensional vectors into d dimensional space, where ν is the number of views. To ease notation, let $X^{(k)}$ be a mean 0 random variable of dimension n for the k-th view. These random variable are, for example sentence embeddings for aligned data, normalized so each component is mean 0. The matrices $A^{(k)}$ are conceptually recovered one row at a time solving the optimization problems

[1]The authors are grateful to Paul McNamee and Liam Dugan of the Johns Hopkins Human Language Technology Center of Excellence who provided these translations.

1. $a_1^{(k)}$ are the solution of

$$\arg\max \sum_{1 \le i < j \le \nu} \mathrm{Cor}(a_1^{(i)} X^{(i)}, a_1^{(j)} X^{(j)})$$

2. for $i = 2, \ldots, m : a_i^{(k)}$ are the solution of 1

subject to:

$$\mathrm{Cor}(a^T X^{(k)}, a_j^T X^{(k)}) = 0 \qquad \forall j < i,\ k = 1, \ldots, \nu.$$

The specific implementation of GCCA is the function `MCCA` from the package Multiview Learn or `mvlearn`[4]. In our application a *view* is a part of a query or a passage. `MCCA` projects the views so as to maximize a regularized sum of all pairs of correlations between the views. `MCCA` gives a number options for regularization. The one we found most useful for CLIR first uses principal component analysis (PCA) on each of the views to reduce the dimension. Each PCA is computed via an SVD on the view.

3 Experiments

3.1 Sentence Retrieval Task on TREC Novelty Track

This section describes results for fine-tuning sentence embeddings for an information retrieval task for the NIST TREC 2002-2004 Novelty track. In each case a baseline is reported which scores sentences for relevance via an inner product between the query vectors and the document sentence vectors. For each document set we used two query vectors, the topic and the first sentence of the description. The score for a baseline sentence is simply the sum of the two inner products.

We can build these models in multiple ways. We consider three ways:

- Training bitext data of parallel in-domain corpora. (Here we used the machine translation data of the TREC training data portion in the 5-fold cross validation.)
- Training bitext data from a large corpus, which may be out-of-domain. (Here we used UN 6-way corpora [2] but just 2M bitext pairs).
- 3-tuples of (query topic, query description, document sentence) for the TREC training data portion in the 5-fold cross validation.)

All the results presented use a generalized canonical correlation analysis, which implements a regularized version of the all-pairs method [2] [5]. The

[2] https://conferences.unite.un.org/uncorpus/en/downloadoverview

package `mvlearn`[3] [4] provides the method `MCCA` and gives an easy and efficient code to compute this decomposition and save the result in class.

The Python package `mvlearn` was used to build lower dimensional approximation of the LaBSE sentence embeddings [1]. The dimension reduction is a generalization of principal component analysis to multiview data.

Results on TREC 2002 and 2004 are given in Tables 1 and 2 the AUCs are reported based on a 5-fold cross validation comparing the baseline LaBSE vectors, the `mvlearn` 200-dimensional representations, as well as naive Bayes model for both of these. We report just the first of the bitext training methods, so as to not belabor the reader, as the other two approaches give comparable results. In each table the English-only task, i.e., the one originally proposed by NIST is compared with an English-Russian (en/ru) and English-Chinese (en/zh) version. In the 2002 data, the `mvlearn` approach is the best of the four and the naive Bayes model reduces the performance for both the baseline and `mvlearn`.

The story is different in the 2004 data. Here, `mvlearn` without naive Bayes, lags behind the baseline; but with naive Bayes performs comparably. The differences were confirmed via a paired Wilcoxon test and indeed Baseline + naive Bayes is significantly better than the Baseline. MCCA(200,0.5) + naive Bayes gives results which are not significantly different than Baseline + naive Bayes.

The 2004 data is perhaps more reliable than the 2002 data for two reasons. First, 2002 was the first year for the NIST Novelty Task and there is usually a "learning curve" in evaluations for the task organizers. Second, the 2004 data has documents which contain no relevant sentences, which is closer to the CLEF task being used at SCALE 2021. So, indeed we can use the 2004 data as a "pre-filtered" data set to evaluate document retrieval. These are documents that were returned by the query engine used by NIST in 2004, so they may be viewed as a re-ranking task, not unlike for the NeuCLIR 2022 task.

Table 1: TREC 2002 Results

CLIR Task	Baseline	MCCA(200,0.5)	Baseline + NB	MCCA(200,0.5) + NB
en/en	0.727	0.751	0.683	0.635
en/ru	0.698	0.726	0.701	0.668
en/zh	0.718	0.729	0.681	0.632

3.2 Document Retrieval Task on TREC Novelty Track 2004

The below table gives results for document reranking problem using the TREC 2004 Novelty data. These data have 50 queries (topics) with 25 to 75 documents

[3] https://mvlearn.github.io

Table 2: TREC 2004 Results

CLIR Task	Baseline	MCCA(200,0.5)	Baseline + NB	MCCA(200,0.5 + NB
en/en	0.707	0.692	0.768	0.765
en/ru	0.710	0.684	0.756	0.758
en/zh	0.713	0.685	0.764	0.752

for each set. Each topic has a total of 25 relevant documents. Depending on the topic there is 0 to 50 non-relevant documents. The sentence scores used in the previous section were "promoted" to document scores by computing the mean sentence score for each document in the cluster. As before a 5-fold cross-validation was done. Here, we compute the average mean precision over the 5-fold experiment. In addition to using the training data bitext sentence pairs en-ru and en-zh, we also include bitext from the UN 6-way corpus to illustrate the degradation for not having genre-specific bitext. Thus we replace building the multiview dimensionality reduction using the training newswire data with 2M bitext pairs from UN parallel corpora. The upshot is the loss is small. As the `MCCA` decomposition is based on only two views, it is a regularized variant of canonical correlation. For comparison sake we compare results using Python sklearn `PLSSVD` in the results following.

Table 3: TREC 2004 Document Retrieval

CLIR Task	Baseline	MCCA[PLSSVD]	Baseline + NB	MCCA[PLSSVD] + NB
en/en	0.783	0.782	0.810	0.803
en/ru	0.785	0.781 [0.777]	0.800	0.803 [0.791]
en/zh	0.779	0.774 [0.776]	0.809	0.789 [0.789]

Table 4: Document Retrieval F1 scores for TREC 2004 Using UN 6-way Parallel Corpora for Training.

4 Application to Passage Retrieval MS MARCO

MS MARCO data has much larger query relevant document training set, approximately 800K query-passage pairs in the training data. So, using these data gives more opportunities. The translation team translated the entire English

[3]https://scikit-learn.org/stable/modules/generated/sklearn.cross_decomposition.PLSSVD.html

collection of MS MARCO which consists of approximately 8M passages. Translations were made available into Russian and Chinese as with the TREC Novelty track as an added bonus Farsi was included as well. Our first experiment is to hold back a random part of the training data to see if reducing the query and passage embeddings via `mvlearn` can improve the ability of the LaBSE vector in CLIR.

To this end, we hold back a random 1% of the 800K query-passage pairs and compute a multiview embedding using 5 views: English query, and relevant passages in each of the four languages: English, Farsi, Russian, and Chinese. Based on our results with the TREC Novelty data we considered number of components set around 200 and found at least when computing one joint 5-view embedding a larger number of components gives better results. Here, we see that dimension 200 is still close to the best choice. Note that these results are computed with a joint projection computed based on all 5-views simultaneously. Recall the results presented so far are based on a 2-view model, one for each language.[4] This model is convenient to compute and could be tuned further, but clearly shows that the training data of MS MARCO does improve results of using LaBSE for CLIR in all cases.

Table 5: MS MARCO CLIR document retrieval precision at 1 for the held out 1% of the training data, reg=0.0.

CLIR Task	Baseline	MCCA(200,0)	MCCA(300,0)	MCCA(400,0)
en/en	0.498	0.632	0.638	0.643
en/fa	0.338	0.456	0.462	0.462
en/ru	0.468	0.581	0.584	0.584
en/zh	0.389	0.493	0.496	0.499

5 TREC NeuCLIR 2022 Submission

This year's TREC NeuCLIR track had three tasks. They were:

1. The *Ad Hoc CLIR Task* was the main task in the NeuCLIR track. There is a large document collection in Chinese, Persian, and Russian. There is also a set of English topics. For each topic, the CLIR system produces a ranked list of the 1000 most relevant documents to the topic.

2. The *Reranking Task* is very similar to the ad hoc one. Again, there is a set of English topics, but the system also receives a ranked list of 1000

[4]Preliminary testing seems to indicate that the 5-view models may be superior 2-view vs. 5-view: zh:0.47 vs 0.50, fa:0.44 vs 0.46, ru:0.57 vs 0.58, en:0.63 vs 0.64. Note the 2-view model is shown for dimension 150, which seemed a bit better than dimension 200. It is likely that the extra views need more dimensions, but seem to improve results.

Table 6: MS MARCO CLIR document retrieval precision at 1 for the held out 1% of the GEP training data, reg=1.0 and reg=0.0, signal ranks = dim.

CLIR Task	Baseline	MCCA(200,0)	MCCA(300,0)	MCCA(400,0)
en/en	0.498	0.574	0.580	0.579
en/fa	0.338	0.400	0.401	0.402
en/ru	0.468	0.506	0.511	0.513
en/zh	0.389	0.444	0.447	0.448
en/en	0.498	0.667	0.662	0.651
en/fa	0.338	0.481	0.475	0.472
en/ru	0.468	0.603	0.593	0.595
en/zh	0.389	0.516	0.517	0.513

Table 7: MS MARCO CLIR document retrieval precision at 1 for the held out 1% of the GEP training on varying lengths of data, reg=0.0, signal ranks = dim.

CLIR Task	Train	Baseline	MCCA(200,0)	MCCA(300,0)	MCCA(400,0)
en/en	500K	0.498	0.667	0.662	0.651
en/en	50K	0.498	0.663	0.655	0.638
en/en	5K	0.498	0.623	0.572	0.505
en/fa	500K	0.338	0.481	0.475	0.472
en/fa	50K	0.338	0.476	0.465	0.456
en/fa	5K	0.338	0.439	0.407	0.362
en/ru	500K	0.468	0.603	0.593	0.595
en/ru	50K	0.468	0.599	0.586	0.581
en/ru	5K	0.468	0.562	0.514	0.463
en/zh	500K	0.389	0.516	0.517	0.513
en/zh	50K	0.389	0.518	0.514	0.500
en/zh	5K	0.389	0.482	0.441	0.384

documents. The submission is again a ranked list of the documents returned for each topic ordered from most relevant to least relevant. In this case, however, the only documents allowable in the list are those that were present in the original ranking.

3. There was also a *Monlingual Retrieval* Task. This can often serves as a reference point for CLIR systems. In this track, the monolignual task was very similar to the ad hoc one. The topic (query) files are actually human translations of the English topic files.

We limited our submissions, due to time, to the re-ranking problem and the ad hoc task for the NeuCLIR track i.e., we did not attempt the Monolingual task.

Our overall strategy was very similar to the one described previously. We used a language agnostic sentence representation. Prior experience had success with the LaBSE model, so we continued using that as our foundational model. We combined this with a five-view 200-dimension GCCA model trained on the translated MS MARCO data.

For both the *Ad Hoc* task and the *Reranking* task, the topic/query file contained a `topic_title` and a `topic_description` field. The actual articles contained a `title` and a `text` field. First we would compute the LaBSE representation, $y_{\text{field\_name}}$, of each of these four fields. Our overall score, s, consisted of the maximum of four inner products:

$$\begin{aligned} s = \max(& < x_{\text{topic\_title}}, x_{\text{title}} >, \\ & < x_{\text{topic\_title}}, x_{\text{text}} >, \\ & < x_{\text{topic\_description}}, x_{\text{title}} >, \\ & < x_{\text{topic\_description}}, x_{\text{text}} >) \end{aligned}$$

In general, we will compute $x_{\text{field}} = f(y_{\text{field}})$ i.e, the actual representation used in the inner product of some function of the LaBSE representation of the field's text. We note that from `sentence_transformers` the LaBSE embedding vectors are normalized automatically. They lie on the unit sphere. So, our three scores that we compute are:

1. A *BASELINE* which is simply the identify function $f(x) = x$.

2. A *CCA-unnormalized* score which is $f(x) = CCA(x)$. Here, the appropriate view (language) is applied e.g., we transform NeuCLIR Russian sentences with the MCCA Russian view trained on MS MARCO.

3. A *CCA-normalized* score which is $f(x) = \frac{CCA(x)}{||CCA(x)||}$. Since the output of MCCA is not necessarily on the unit sphere, we project back. Here, again, we use the transformation for the appropriate language.

Since the reranking task included a list of 1000 ranked articles per topic, we opted to compare our results with that. Specifically, we looked at the Jaccard similarity between the articles our ad hoc systems (Baseline, CCA-normalized, and CCA-unnormalized) with the given files.

The tables below show the Jaccard similarities for Russian, Farsi, and Chinese. In all three cases, we can look at the average Jaccard similarity across the 114 topics. In general, the normalized CCA was better than both LaBSE without modification or LaBSE and CCA unnormalized. Unnormalized CCA was better than the baseline in both Russian and Farsi. See, for example, the second (mean) row in tables 8 and 10. In Chinese however, the unnormalized CCA actually under-performed the unmodified LaBSE baseline. See, e.g., Table 9.

Table 8: NeuCLIR Russian.

RUSSIAN		Baseline	CCA-normalized	CCA-unnormalized
	topic id	jaccard1	jaccard2	jaccard3
count	114.000000	114.000000	114.000000	114.000000
mean	64.236842	0.064858	0.069954	0.066649
std	39.928543	0.085305	0.083329	0.080584
min	0.000000	0.001001	0.000500	0.000500
25%	30.250000	0.016260	0.017812	0.016777
50%	60.000000	0.035197	0.040583	0.037883
75%	99.750000	0.079768	0.087548	0.085924
max	136.000000	0.492537	0.512859	0.483680

Table 9: NeuCLIR Chinese.

CHINESE		BASELINE	CCA-normalized	CCA-unnormalized
	topic id	jaccard1	jaccard2	jaccard3
count	114.000000	114.000000	114.000000	114.000000
mean	64.236842	0.067544	0.070705	0.066279
std	39.928543	0.079115	0.082964	0.078340
min	0.000000	0.000000	0.000000	0.000000
25%	30.250000	0.021581	0.020278	0.018849
50%	60.000000	0.047123	0.050144	0.047669
75%	99.750000	0.088139	0.085482	0.079622
max	136.000000	0.598721	0.628664	0.616815

Table 10: NeuCLIR Farsi.

FARSI		BASELINE	CCA-normalized	CCA-unnormalized
	topic id	jaccard1	jaccard2	jaccard2
count	114.000000	114.000000	114.000000	114.000000
mean	64.236842	0.047265	0.052518	0.049575
std	39.928543	0.051507	0.055751	0.053468
min	0.000000	0.001001	0.000500	0.000500
25%	30.250000	0.013300	0.015873	0.016260
50%	60.000000	0.030928	0.033592	0.031726
75%	99.750000	0.059884	0.069662	0.067094
max	136.000000	0.297017	0.331558	0.319261

We do note that, in absolute terms, these Jaccard similarities are not very high. It caused us to question the quality of this system relative to the one given for the reranking task. They are too high to be random, however, so we think our system is picking up on something, and based on the results we can confirm this.

The following plots summarize our submissions for NeuCLIR. In both Figure 1 for Russian and Figure 2 for Farsi, the results mirrored the results in the training data. The GCCA fine-tuning improves the retrieval and the normalized vectors give a further improvement. In Figure 3 for Chinese, the three submissions have precision-recall curves which intertwine, so we do not see any improvement here. In all three languages, performance was boosted in the reranking runs. The evaluation largely validated that lightweight fine-tuning can give significant improvement for CLIR. As the approach is much less expensive than computing the embeddings, it could be employed in a real-time setting using little computational resources to fine-tune results.

6 Conclusions

We demonstrate 200 dimensional embeddings could also be used in the CLIR task effectively. In particular, on one TREC data set a 3-4% improvement in performance was achieved using the CCA vectors. On a second TREC data set, which is believed to be closer to the operational task, a naive Bayes in conjunction with CCA vectors achieved a 5-8% improvement in performance. Of greater promise is that if 200-long vectors are used, the storage requirement would be 74% less for comparable performance.

For the MS MARCO data we built a single model using 800K question-answer tuples, which included the query, and relevant answers in English, as well as translations into Farsi, Russian, and Chinese. This model gave great improvement (as measured by Precision at 1) across all languages.

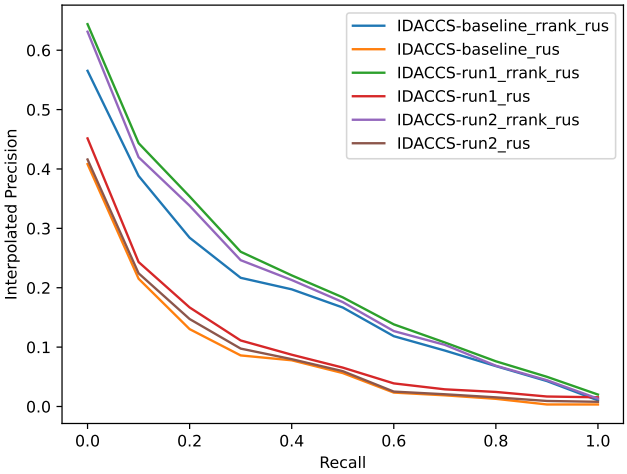

Figure 1: NeuCLIR precision-recall for Russian.

The MS MARCO `MCCA(200)` model, a joint a 200 dimensional subspace of the 768 dimensional LaBSE embeddings was the one used for the TREC submission. The original vectors were also submitted for comparison.

While the approach is not meant to compete with a full neural net fine-tuning, it does provide an extremely cheap alternative to neural net fine-tuning for CLIR or other tasks that depend on computing similarities of sentence representation. It is key to note that the training projection on a CPU is 6 times faster than computing the LaBSE embeddings on a GPU. Such low-cost fine-tuning, which was shown to require as little as 5000 aligned tuples, could allow customization at a user level for IR tasks.

References

[1] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[2] J. R. KETTENRING. Canonical analysis of several sets of variables. *Biometrika*, 58(3):433–451, 12 1971.

[3] Suraj Nair, Eugene Yang, Dawn Lawrie, Kevin Duh, Paul McNamee, Kenton Murray, James Mayfield, and Douglas W. Oard. Transfer learning approaches for building cross-language dense retrieval models. In Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog,

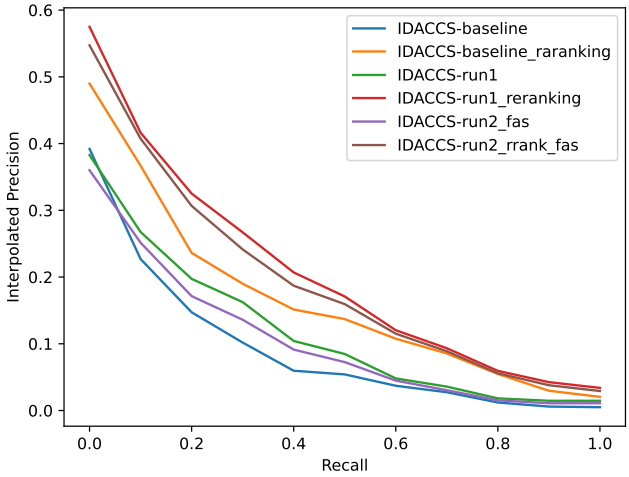

Figure 2: NeuCLIR precision-recall for Farsi.

Kjetil Nørvåg, and Vinay Setty, editors, *Advances in Information Retrieval*, pages 382–396, Cham, 2022. Springer International Publishing.

[4] Ronan Perry, Gavin Mischler, Richard Guo, Theodore Lee, Alexander Chang, Arman Koul, Cameron Franz, Hugo Richard, Iain Carmichael, Pierre Ablin, Alexandre Gramfort, and Joshua T. Vogelstein. mvlearn: Multiview machine learning in python. *Journal of Machine Learning Research*, 22(109):1–7, 2021.

[5] Arthur Tenenhaus and Michel Tenenhaus. Regularized generalized canonical correlation analysis. *Psychometrika*, 76(2):257–284, April 2011.

[6] Brian Thompson and Philipp Koehn. Exploiting sentence order in document alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5997–6007, Online, November 2020. Association for Computational Linguistics.

[7] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12310–12320. PMLR, 18–24 Jul 2021.

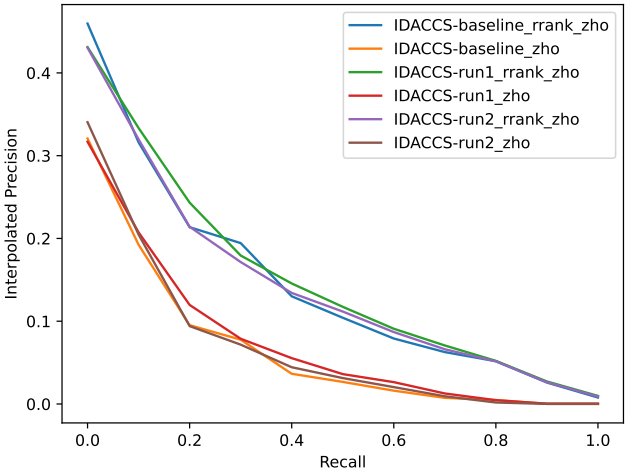

Figure 3: NeuCLIR precision-recall for Chinese.