ISI's SEARCHER II System for TREC's 2023 NeuCLIR Track

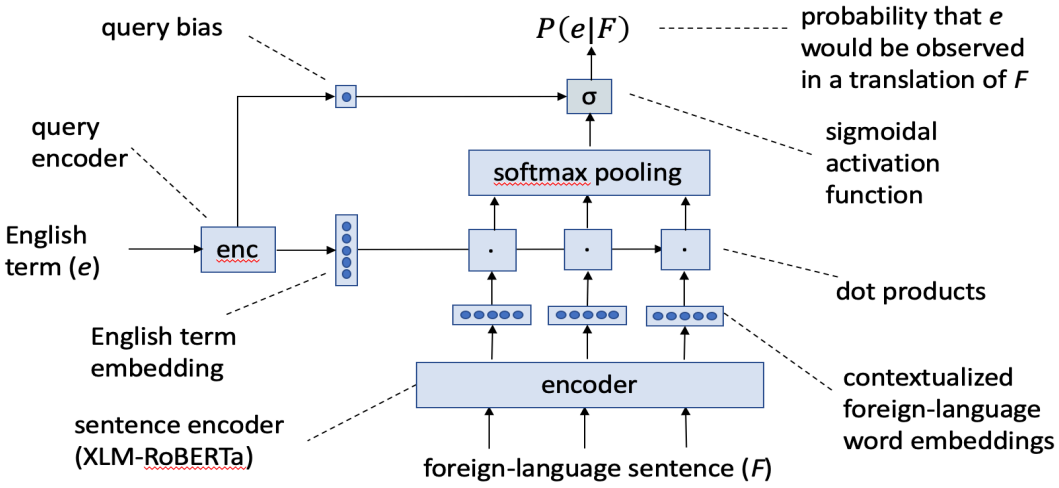
Scott Miller (smiller@isi.edu), Shantanu Agarwal (shantanu@isi.edu), Joel Barry (joelb@isi.edu)

Summary

ISI's submission to TREC's 2023 NeuCLIR track is a system called SEARCHER II (Shared Embedding Architecture for Effective Retrieval). It is a two-stage system in which both stages are neural based. The first stage produces an initial ranking over the entire collection, while the second re-ranks the top 1000 candidates returned by the first stage. Rankings of both stages are combined to produce a final ranked list. The system operates on English queries and native language documents with no translations of either.

Stage 1 Neural Network

The core neural network of SEARCHER II's first stage [Barry et al.] was developed under IARPA's MATERIAL program. It is trained to predict the probability of an English query term appearing in a possible translation of a native-language sentence. More specifically, it attempts to match a *contextualized* embedding of a foreign word with the embedding of an English query word. A diagram of the network is shown below.



Stage 1 Training

For NeuCLIR, the network was trained on 2M parallel Chinese/English sentence pairs from publicly available sources (e.g., ParaCrawl, News Commentary). Also included in the training are bilingual dictionary entries, again from publicly available sources (e.g., PanLex, GeoNames). Training was done for 10 epochs on a single 2080ti gpu. Training took roughly 4 days. The encoder was XLM-RoBERTa-base [Conneau et al.].

Stage 1 Ranking Function

Stage 1's ranking function follows a similar approach to PSQ [Darwish and Oard] where the BM25 algorithm is adapted to utilize probabilistic evidence and to enable cross-lingual retrieval.

Specifically, we replace the core statistics of BM25 (term frequencies and document frequencies) with expected values derived from SEARCHER's probability estimates:

$$\mathbb{E}(tf(e,D)) = \sum_{S \in d} p(e|S)$$

$$p(e \in D) = 1 - \prod_{S \in d} 1 - p(e|S)$$

$$\mathbb{E}(df(e) = \sum_{d \in C} p(e \in D)$$

Here,

- $\mathbb{E}(tf(e,D))$ is the expected number of times an English word *e* appears in a translation of foreign-language document *D*.
- $p(e|S)$ is SEARCHER's estimate of the probability of *e* occurring in a translation of a foreign-language sentence *S*.
- $\mathbb{E}(df(e)$ is the expected number of documents whose translation contains *e*.

Replacing BM25's frequency statistics with these terms yields a ranking score for cross-lingual queries.

$$score(D,Q) = \sum_{q \in Q} IDF(q) \frac{\mathbb{E}(tf(q,D)) \cdot (k_1 + 1)}{\mathbb{E}(tf(q,D)) + k_1(1 - b + b\frac{|D|}{avg\_dl}}$$

$$IDF(q) = ln\left(\frac{N - \mathbb{E}(df(e) + 0.5}{\mathbb{E}(df(e) + 0.5} + 1\right)$$

Where *D* is a foreign-language document; *Q* is an English query; |*D*| is the length of *D* (in words); *avg_dl* is the average document length for all documents in the collection; *N* is the number of documents in the collection; and k_1 and *b* are the usual BM25 free parameters.

Stage 1 Sparse Representation

SEARCHER's Stage 1 neural network encodes **query terms** using a simple embedding matrix, resulting in *uncontextualized* embeddings where the embedding for a query term depends only on itself and not any other terms in the query. This characteristic enables us to precompute a sparse representation for native language documents. More specifically, at indexing time, the probability of every English word in the vocabulary is evaluated (in parallel on a GPU) for each native language sentence and the results placed in an inverted index, e.g. {*English term*: (*document_id*, *probability*), …, (*document_id*, *probability*)}. Pruning eliminates very low probability terms, resulting in indexes containing 5-10 English terms for each native language term.

Because indexes are constructed by comparing terms in a continuous embedding space, alternative translations and closely related terms are included. Taking an example from Swahili (one of the original MATERIAL languages), '*kulipita **gari** jingine*' translates to '*passing another **car***'. In this case, both *car* and *vehicle* are placed in the index with high probability, while more distantly related terms, e.g., *SUV* or *truck*, are also identified but assigned lower probabilities.

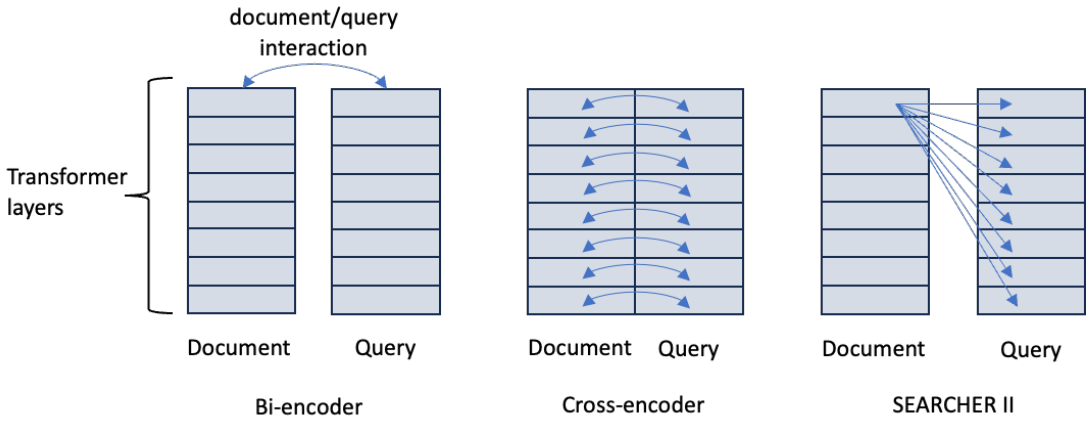
Unlike query terms, **document terms** are *contextualized*, providing for differentiation between polysemous word senses. Taking another Swahili example, the term *nyanya* may alternatively mean *grandmother* or *tomatoes*, e.g., '*babu na **nyanya***' (*grandfather and **grandmother***) or '*vitunguu na **nyanya***' (*onions and **tomatoes***). Because document embeddings depend on context, *nyanya* more closely matches *grandmother* in the former case and *tomatoes* in the latter case.

The sparse inverted index constructed by Stage 1 thus includes multiple translations and related alternatives for each document term, as well as taking account of polysemous senses.

Stage 2 Neural Network

Neural approaches for IR frequently involve one of two architectures: bi-encoders or cross-encoders. SEARCHER II takes an intermediate approach, aiming to derive advantages of each.

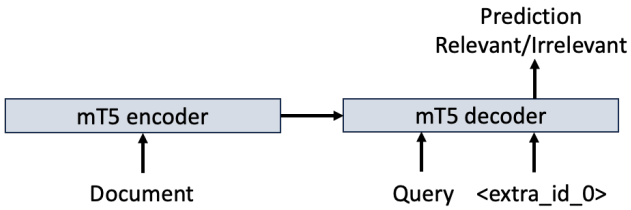
- In bi-encoders, such as ColBERT [Khattab and Zaharia], queries and documents interact only after each is encoded. Because queries and documents are encoded independently of each other, document encoding can be performed at indexing time, enabling queries to be processed relatively quickly.
- In cross-encoders, queries and documents are jointly encoded and interact at every transformer layer. Thus, documents can be encoded only after the query is known, resulting in slower query processing. However, the greater number of interactions between documents and queries can result in better IR accuracy.
- In SEARCHER II, each layer of the query encoder interacts with the document's representation, as in cross-encoders. However, interactions are limited to only the top layer of the document encoder. Thus, like bi-encoders, documents can be pre-encoded at indexing time.



Because queries are typically much shorter than documents, and because transformers have quadratic time complexity, the time required to process each query is significantly less than for a cross-encoder. For example, given a 250-word document and an 8-word query, a cross-encoder's query time complexity is $(250 + 8)^2$, whereas for SEARCHER II it is $250 * 8^2$, or approximately one fourth the time.

SEARCHER II was inspired by the encoder-decoder architecture used for machine translation and similar tasks. Our implementation is based on mT5 [Xue et al.] where documents are

presented to mT5's encoder and queries are presented to its decoder. The model is trained on a forced-choice task of predicting one of two values for the final token (either yes or no, indicating relevant or irrelevant). The loss function is pointwise cross entropy. Other than a custom loss function, we did not modify mT5. The decoder's causal mask was left in place, limiting query attention only to previous, not subsequent, tokens.



Stage 2 Training

We initially attempted to simply train mT5 on MSMARCO triples [Bajaj et al.] where the passages were translated into Chinese by Google Translate [Bonifacio et al.] and the queries left in English. However, results were disappointing; learning was slow and uneven, and the model reached only a moderate level of accuracy. Instead, we found a three-step training strategy to be more effective.

- We first pretrained the model to learn the correspondence between Chinese and English. To do so, we trained mT5-large to an MT objective on four million parallel Chinese/English sentences from publicly available sources (a superset of the data used to train the Stage 1 model). Training required approximately four days on a single node with four RTX-8000 GPUs.
- Next, we trained the model to learn (monolingual) information retrieval. In this step, we continued training the model for a single epoch on one million MSMARCO English triples to a pointwise IR objective. This step required around one day using four RTX-8000 GPUs on a single node.
- Finally, we fine-tuned the model to perform Chinese-English CLIR. Here, we trained on only 84,000 translated MSMARCO triples (English queries, Chinese passages). This step required around two hours, again on four RTX-8000 GPUs in a single node. We found that fine tuning on the CLIR task for any longer degraded accuracy.

Final Ranking and Results

Both Stage 1 and Stage 2 contribute usefully to the CLIR task. Because it is not restricted to training on MSMARCO (or another IR collection), the Stage 1 model can incorporate broader lexical coverage. Meanwhile, by training on MSMARCO, the Stage 2 model learns broader semantic associations useful for IR. We combine rankings of the two stages using reciprocal rank fusion [Cormack et al.]. Results on TREC's 2023 Chinese NeuCLIR task are shown below.

Recall@100	0.749556
Recall@1000	0.897792
nDCG@20	0.492215

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina

`Stoica, Saurabh Tiwary, Tong Wang: *MS MARCO*: *A Human Generated MAchine Reading COmprehension Dataset*, https://arxiv.org/abs/1611.09268

Joel Barry, Elizabeth Boschee, Marjorie Freedman, Scott Miller: *SEARCHER*: *Shared Embedding Architecture for Effective Retrieval*, CLSSTS@LREC 2020: 22-25

Luiz Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, Rodrigo Nogueira: *mMARCO: A Multilingual Version of the MS MARCO Passage Ranking Dataset*, https://arxiv.org/abs/2108.13897

G.V. Cormack, C.L.A. Clarke, Stefan Buttcher: *Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods*, SIGIR'09, July 19–23, 2009, Boston, Massachusetts

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzman, Edouard Grave, Myle Ott, Luke Zettlemoyer, Veselin Stoyanov: *Unsupervised Cross-lingual Representation Learning at Scale*, https://arxiv.org/abs/1911.02116

Kareem Darwish, Doug Oard: *Probabilistic structured query methods*, Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 338–344 (2003)

Omar Khattab, Matei Zaharia: *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*, https://arxiv.org/abs/2004.12832

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, Colin Raffel: *mT5: A massively multilingual pre-trained text-to-text transformer*, https://arxiv.org/abs/2010.11934

Acknowledgements

This research is based in part upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract # FA8650-17-C9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

SEARCHER II is a collaborative effort with the JHU HLTCOE. The authors wish to express their thanks for many helpful discussions with Jim Mayfield, Dawn Lawrie, and Eugene Yang of the HLTCOE and Doug Oard of the University of Maryland.