EXTENDING ENGLISH IR METHODS TO MULTI-LINGUAL IR

**Carlos Lassance**
Naver Labs Europe

France carlos.lassance@naverlabs.com

**ABSTRACT**

This paper describes our participation in the 2023 WSDM CUP - MIRACL challenge. Via a combination of i) document translation; ii) multilingual SPLADE and Contriever; and iii) multilingual RankT5 and many other models, we were able to achieve first place in both the known and surprise languages tracks. Our strategy mostly revolved around getting the most diverse runs for the first stage and then throwing all possible reranking techniques at them. While this was not a first for many techniques, we believe that we had some novel approaches. For example, we trained the first SPLADE model that works in more than 10 languages. However, a more careful study of the results is needed in order to verify if we were able to achieve first place just due to brute force or if the hybrids we developed really brought improvements over the other teams’ solutions.

**1 Introduction**

In this paper, we detail our WSDM CUP 2023 - MIRACL submissions. Due to the time limitations that we were subject to during the competition, this paper will differ vastly from a traditional research paper where models are trained and evaluated to be comparable. Instead,this will mostly read as an incremental improvement report, detailing the iterative steps that we took while.developing for the competition. We believe that the three main pillars that differentiated our approach from other teams were: i) the use of NLLB-200 for the translation of the corpuses back to English, allowing us to use the original SPLADE models, ii) the training of a novel SPLADE model capable of working in the original 16 languages of the MIRACL dataset, and iii) Training the first multilingual RankT5 [16]. We aim to make all of these models available and easy to reproduce for the community, but unfortunately these will not be available for quite some time.

We follow the strategy we used on our latest TREC notebooks, in that we strive for making this more streamlined than a normal research paper would be. We will now present a list of the papers that better introduce and detail the models we used here and refer the reader to them for a more detailed explanation, as we are mainly focused here on how to apply their insights to MIRACL: i) Training non English SPLADE models [11], ii) The SPLADE model [5, 10], iii) The Contriever model and its pretraining [8], iv) The RankT5 reranker [16], v) MonoT5 [13], vi) The LCE loss [6], vii) ColBERT [9], and viii) For our ensembling we use Ranx [1] and their min-max normalized sum ensembling.

**1.1 The MIRACL dataset**

The MIRACL dataset [15] is composed of 16 languages and relies on monolingual retrieval. The majority of our time was focused on these languages, and we will detail our approaches in Sections 2 and 3.*NLE participation at MIRACL*

In addition, two languages were added at the end, without training data, as “surprise” languages. Section 4 details how we generated our runs for these languages. Finally, we do a quick analysis on queries and labelling in Section 5.

**2 Training the first stage**

For the first stage of retrieval, we report step by step how our models evolved during the two months of preparing for the MIRACL competition. We detail our results in the dev set, with incrementing tables of results, leading to the final hybrid that will be used for reranking.

**2.1 The initial baseline**

We start with the baseline provided by the organizers which include an mDPR model trained on mMARCO and BM25. We ensemble these models and generate *HYBRID 0*, which we present in Table 1. We already start noticing that ensembling is strong in this dataset, as the ensemble has almost 1*.*4*×* the effectiveness (nDCG@10) of mDPR, which was the stronger model in the ensemble, and the Recall@100 is already close to 90%.

Table 1: Baselines provided by the organizers. Just a simple hybrid is a good baseline.

# model nDCG@10 Recall@20 Recall@100 Models

a BM25 39.3 60.9% 78.7%
b mDPR **41.5 62.8% 78.8%**

Hybrids
0 a+b 57.8 83.0% 93.7%

**2.2 Going back to English leads to improvement**

The next step was to translate the corpora into English. This is the simplest way of working in any language, as English translators are generally fairly strong and there are a multitude of retrievers and rerankers available off-the-shelf for English data. Moreover, this also adds the knowledge of the translator into the mix. Ideally, we would have used something like Google Translate or DeepL, which are are already effective and resilient to problems, but lack a free-to-use API1. Thus we used the NLLB200 model [4], which i) generated a lot of straing results that were filtered manually (things like ”I don’t know it” or ”I won’t translate this” came out)., ii) was too expensive to run with the largest version, thus for the small corpora we used the large NLLB (3B params), and for the corpora larger than 4M documents we used the distilled version, and iii) Due to the cost of the models we only translated the first 128 tokens. Initial experiments showed a very sharp drop in performance when using the distilled version. However, it would be too costly to use the full model for the largest datasets.

Now that every corpus is translated to English, we took one of the SPLADE++ [5] models and fine- tuned 16 different versions – one on each translation (for the English model we just used the MIRACL corpus). This led to what we call T-SPLADE, which added to the BM25 and mDPR leads to “HYBRID 1”. T-SPLADE provided a large boost in accuracy over BM25 and mDPR, and when ensembled with HYBRID 0, the Recall@20 and Recall@100 surpassed 90% and 97%, respectively. The implication of this is that it would be difficult to improve over this fairly basic approach.

**2.3 Adding monolingual SPLADE for non-English languages**

Due to a recent submission [11] and the TREC-NeuCLIR 2022, we already had 5 pretrained SPLADE models for in Arabic, Chinese, Japanese, Persian, and Russian. Of those we noticed that the Arabic and Chinese

1Some solutions exist, but we were not sure if we could actually use those as they mostly break the terms of

the APIs.

2

*NLE participation at MIRACL*

Table 2: Result comparison adding translated-SPLADE.

# model nDCG@10 Recall@20 Recall@100 Models

a BM25 39.3 60.9% 78.7%
b mDPR 41.5 62.8% 78.8%
c T-SPLADE **54.5 71.3% 83.3%**

Hybrids
0 a+b 57.8 83.0% 93.7%

1 a+b+c 70.0 90.4% 97.2%

did not generalize well to MIRACL, so we just used the other three. Adding these models to the fold increased effectiveness by a small margin, but not enough to be considered a new baseline. In fact, contrary to what we had seen in other datasets, training monolingual SPLADE models did not work well in MIRACL as compared to multilingual, which we confirm in the following paragraphs.

**2.4 The introduction of multi-lingual SPLADE - you have to fail first to make it work**

Now to the only possibly novel part of our submission, multi-lingual SPLADE models (mSPLADE). This has already somewhat been tried with SPLADE-X, which was not multi-lingual but cross-lingual, where they had difficulties: not only with the tokenizer size (which vastly increases memory use), but also with the pretrained models. For the pre-trained models our solution was to pre-train from scratch on MIRACL following [11]. However, we had no solution for the amount of memory required for the larger tokenizer size. But, we were fortunate to have access to a few A100’s with 80GB of RAM, which is the only hardware that could train models of this size. Unfortunately, we were limited to training with at most 128 tokens and indexing with at most 256.

For the tokenizer, we initially used the tokenizer from XLM-Roberta, which has 240k+ tokens, and then pretrained with MLM+FLOPS on the 15 non-English corpora of MIRACL. We then finetuned on the languages of mMARCO that are present in the 15 (Arabic, Chinese, French, Hindi, Indonesian, Japanese, Russian, Spanish), and finally finetuned on MIRACL using two different val- ues of lambda (mSPLADE-small and mSPLADE-large for the length of the documents). However, while the effectiveness was better than the translated T-SPLADE models we were not satisfied.Note that as these models are not trained for English, we consider that they have the same effectiveness as the best of mDPR and BM25 when averaging.

Thus, we went back and recreated a tokenizer from scratch with just 120k tokens, but based on the statistics of the 15 non-English MIRACL datasets. We then pre-trained with MLM+FLOPS on MIRACL, finetuned on mMARCO and finally finetuned on MIRACL to get a new model that we call mSPLADE-sTok. Note that we still used the same max sequence length of 128 for training, but it was much faster to redo all steps for this new model due to the smaller tokenizer. With more patience (i.e. increasing pretraining steps and the max sequence lengths), these models would probably be better. mSPLADE-sTok is not only faster to run, but it is also much more stable and effective compared to mSPLADE-small and mSPLADE-large. Finally, as this was a competition we added all three models to the ensemble, creating “HYBRID 2”, which improves upon the previous one, achieving a Recall@20 that is almost as good as the Recall@100 of our first hybrid. We note that there could probably be even more gains by properly tuning the hyperparameters (sTok has smallish docs and large queries, which is not ideal for SPLADE) and by letting it train for more time – especially the MLM+FLOPS pretraining part, which uses only 5 epochs.

**2.5 Proper dense multilingual retrieval - mColbert and mContriever enter the fold**

In the previous paragraphs, we did our best to improve sparse multilingual retrieval, but we have no base for comparing it to dense models trained on MIRACL. Thus, we now introduced mContriever and mContriever-ColBERT. The idea is simple: we take the mContriever finetuned on mMARCO and further finetune it on MIRACL. To generate mColBERT, we take the mContriever we just finetuned and further train it on MIRACL, but this time under the ColBERT-brute force framework. If we

3

*NLE participation at MIRACL*

Table 3: Result comparison by adding mSPLADE models.

# model nDCG@10 Recall@20 Recall@100 Models

a BM25 39.3 60.9% 78.7%
b mDPR 41.5 62.8% 78.8%
c T-SPLADE 54.5 71.3% 83.3%
d SPLADE-mono Only available in 3 languages

e mSPLADE-small 57.9 76.2% 88.7%
e mSPLADE-large 60.2 78.6% 90.6%
f mSPLADE-sTok **63.9 81.6% 92.4%**

Hybrids
0 a+b 57.8 83.0% 93.7%

1 a+b+c 70.0 90.4% 97.2%
2 a+b+c+d+e+f 74.2 92.1% 98.3%

Disregard English (for which mSPLADE is not trained) mSPLADE-sTok has similar effectiveness to both, and by adding all models together for “HYBRID 3” we get a very impressive first stage ranker. However, at this point we were starting to get diminishing returns, especially considering that the models we are adding are much better than the initial ones.

**2.6 Learning to ensemble**

Considering the diminishing returns we had with adding the models, we consider two more hybrids:

• “HYBRID BEST” is learned by testing all the ensembling possibilities and keeping the one

with the best nDCG@10 on the dev set
• “HYBRID SIMPLE” tries to simplify, keeping only one model per category, thus: i) lexi-

cal retrieval: BM25, ii) translated retrieval: T-SPLADE, iii) sparse multilingual retrieval: mSPLADE-sTok, and iv) dense multilingual retrieval: mContriever. The goal of this hybrid is to check how much we lose by not going brute-force with all models.

We display the final results in Table 4 and note that by learning the best hybrid, we can get a little bit more improvement, achieving a better Recall@20 than the baseline hybrid achieved with Recall@100. The SIMPLE hybrid with just 4 models is capable of almost the same performance as hybrid 2 which uses 7 different methods.Note that we still have to verify if everything we did in the dev set generalizes to one of the test sets.

**3 Training the rerankers**

Now that we finished with the first stage retrievers is time for the last effectiveness increment: reranking. We note that due to the already high effectiveness of the first stage, it is not clear which top*k* we should rerank, as even if Recall@k+i is larger than Recall@k, adding more documents always adds noise to reranking, so we always test @*{*10,20,100*}* and present the best of the three. All of our rerankers are trained using a set of negatives coming from the top1k of our best model at the time (which makes it harder to reproduce). We aim to make these models available as we have done for TREC-DL and TREC-neuCLIR, but it could take a while to actually have them online.

**3.1 monoMT5 and RankT5**

To set up our main rerankers, we start from the mono MT5-13b trained on MMARCO that got impressive results on TREC neuCLIR 22 by the Unicamp’s team2. We then first finetune that model following the monoT5 recipe, with only the negatives from the official MIRACL qrels. We then also

2available at: unicamp-dl/mt5-13b-mmarco-100k

4

*NLE participation at MIRACL*

Table 4: Comparison of all our first stage models on the dev set.

# model nDCG@10 Recall@20 Recall@100

Models
a BM25 39.3 60.9% 78.7%
b mDPR 41.5 62.8% 78.8%
c T-SPLADE 54.5 71.3% 83.3%

d SPLADE-mono Only available in 3 languages
e mSPLADE-small 57.9 76.2% 88.7%
f mSPLADE-large 60.2 78.6% 90.6%
g mSPLADE-sTok 63.9 81.6% 92.4%
h mContriever 64.6 82.7% 93.2%
i mColBERT **65.5 83.5% 93.6%**

Hybrids
0 a+b 57.8 83.0% 93.7%
1 a+b+c 70.0 90.4% 97.2%
2 a+b+c+d+e+f+g 74.2 92.1% 98.3%
3 a+b+c+d+e+f+g+h+i 75.9 93.1% 98.8%

Best max(a+b+c+d+e+f+g+h+i) **77.2 94.0% 98.9%** Simple a+c+g+h 74.0 91.8% 98.1%

finetune it as a RankT5-Encoder only [16]3, this time using the negatives from the best model we had at the time.

We present the results of reranking “Hybrid best” in Table 5. We notice that mono MT5 does not improve over the first stage hybrid, showing that we probably could have trained it better. However, RankMT5 actually improves slightly over the first stage hybrid, and combining the first stage hybrid with the two rerankers (Simple Reranking Hybrid) allows us to reach an nDCG@10 of 80.4, which would be the best score in the public dev leaderboard.4

Table 5: Reranking with mT5-13b.

# model nDCG@10 # in simple reranking hybrid

First stage hybrids
Best 77.2 16

Rerankers
*α* mono mT5-13b 76.9 14

*β* Rank mT5-13b 78.6 15

Reranking hybrids
Simple reranker hybrid max(Best first stage + *α* + *β*) 80.4 n/a

**3.2 Throwing everything but the kitchen sink**

We were surprised by the lack of improvement from the rerankers we used, so we started testing a many different rerankers. Finally, we ended up with 6 rerankers: the 2 we presented in the previous subsection and an extra 4 RankT5-Encoder-only models, all trained in a similar fashion, but starting

3We use RankT5-Encoder only as it has very similar results to full RankT5, but it is much less costly as it

removes the decoder
4We note that a lot of teams stopped submitting to the public dev, using the leaderboard solely for test-a,

thus making it hard to compare on the dev only.

5

*NLE participation at MIRACL*

from different pretrained language models, namely: Bloomz-3b [12]5, mT0(xl6 and xxl7) [12] and a flan-T5-xl [3]8 trained solely in English that we run over the translated corpora.

**English special case** Considering that we already had a bunch of rerankers finetuned for English due to our participation in TREC-DL 2022, we tested a few of them and actually noticed a slight im- provement (72.9 nDCG@10 improves to 73.5 nDCG@10) when we added the monoT0pp reranker to the hybrid. In other words, the actual reranker ensemble considers 6 rerankers for all languages, but on English, we actually consider 7.

**Results on evaluation set** We present the results of all rerankers over “Hybrid best” in Table 6. We notice that none of our rerankers outperformed the first stage by more than 1.5 nDCG@10 and that combining these 4 rerankers improves our hybrid from 80.4 to 81.0 which was the best result on the dev set. However, that is not that far from what we could achieve with just our first stage retrievers (77.2). In other words, we probably need to get a better way of training rerankers (same conclusion as our TREC-DL paper). We also note that the first stage was the most important part, appearing in all languages, while BloomZ was actually only useful for Arabic and Bengali.

Table 6: Full reranking results over “Best” first stage hybrid.

# model nDCG@10 # in simple rr hybrid # in best rr hybrid

First stage hybrids
F Best 77.2 16 16

Rerankers
*α* mono mT5-13b 76.9 14 12

*β* Rank mT5-13b 78.6 15 13
*γ* Rank mT0-xl 77.8 15 10
*δ* Rank mT0-xxl 78.2 15 8
*ϵ* Rank Bloomz 71.2 15 2
*ζ* Translate + Rank T5 70.0 15 4

Reranking hybrids
Simple max(F + *α* + *β*) 80.4 -

Best max(F + *α* + *β* + *γ* + *δ* + *ϵ* + *ζ*) 81.0 -

**Results on the test sets** Now that we have described our approach and how it evolved over the development set, we can finally talk about the leaderboard and the competition itself, including t*he* two test sets, A and B. Test set A was released first, with a 12-language subset (removing 4 languages, Spanish, Farsi, French, and Chinese). Test set B was the final evaluation set and was released in the final days of the challenge. Unfortunately, we did not keep track of our different hybrids and rerankers during the challenge period, thus we can only evaluate the final submission.

We present the results for the top 3 teams on test-A in Table 7 and for test-B in Table 8.

Table 7: Test-a leaderboard results.

# team Average ar bn en fi id ja ko ru sw te th 1 nle **(ours) 81.2 84.2** 91 **64.5 78.6 62.9 83.4** 77.7 **83 86.6 93.8** 87.6 2 ew 81 83.1 **92.2** 66.3 76.9 61.2 83 **80.1** 81.7 85.3 93.1 **87.9** 3 vector 80.7 83.1 91.6 62.4 77.3 **62.9** 82.2 78.1 82.4 86.8 93.6 87.5

Our massive first-stage hybrid and then reranking hybrid has allowed us to achieve the best result in Test-a by a very slim margin of 0.2 (81.2 vs 81.0) and a slightly larger margin on Test-b (1.5, 73.3 vs 71.8). In fact, our margin on the Test-b is larger than the margin between the 2nd and 11th places,

5bigscience/bloomz-3b
[6https://huggingface.co/bigscience/mt0-xl](http://6https://huggingface.co/bigscience/mt0-xl)
[7https://huggingface.co/bigscience/mt0-xxl](http://7https://huggingface.co/bigscience/mt0-xxl)
[8https://huggingface.co/google/flan-t5-xl](http://8https://huggingface.co/google/flan-t5-xl)

6

*NLE participation at MIRACL*

Table 8: Test-b leaderboard results.

Average ar bn en es fa fi fr hi id ja ko ru sw te th zh 1 nle **(ours) 73.3 72.8** 81.6 **83.1** 74.6 **72.1 69.2 71.8 79.7** 46.1 **75.9 72.1 77.5 76.6** 54.3 **86.1 78.8**

2 NOT CIIR 71.8 70.7 80 81 **75.8** 71.2 68.3 68.1 77.5 **47.5** 72.3 72 74.1 72.8 53.5 85.3 78.7 3 bott 71.6 70.3 82 79.8 70.4 70.6 67.5 69 76.5 44.4 75.2 71 75.7 75.6 **55.5** 85.2 77.5 3 mia 71.6 70 **82.1** 79.8 70.3 70.4 67.2 68.9 76.5 44.4 75.1 71 75.8 75.5 55.3 85.3 77.6 3 ew 71.6 70 82 79.9 70.3 70.4 67.2 68.9 76.5 44.4 75.1 71 75.7 75.6 55.2 85.3 77.6

which means that the competition was quite close with many teams clustered tightly together. Overall, there is not one specific language where we improve a lot, our largest margins are French (+2.7) and Hindi (+2.2). However, there are no languages that we perform badly, as we are always no farther than 1.4 points (also smaller than our “victory margin”) away from the best of that language (Indonesian).

**4 The surprise languages**

The second track of MIRACL was the surprise languages track. The idea was that 2 new languages would be added to the overall dataset, but this time without any training data and without havingtime to prepare for them. In other words, the goal was to really test the capabilities of the systems to adapt to other languages. These two languages were Yoruba and German, with Yoruba having the fewest documents of all languages, while German has one of the largest corpora of the entire 18 languages.

**4.1 First stage**

Given all the models that we had already trained, we wanted to avoid training new models. However, it was kind of a given to at least try for a monolingual SPLADE in German as there was data available in mMARCO [2]. Results for the development set are made available in Table 9. Our mSPLADE models did a very poor job on German, which was somewhat expected as the models with the larger tokenizer did not learn well and the one with the sTok did not know about words in the surprise languages. Nevertheless, mSPLADE-sTok was a part of our final ensemble for Yoruba, which was quite surprising. For German, mContriever had the better performance, while for Yoruba what really helped us was translating to English and then searching in English, with an nDCG@10 of 82.7. Hybrids were penalized a lot because of the poor performance of the mSPLADE models in German, but the “Best” hybrid still got an advantage over all the others. Finally, we got a 100% Recall@10 for Yoruba, which is something that we had not seen in any of the other languages.

**4.2 Rerankers**

For the reranking we used a few different rerankers compared to the known languages track. 9 For Yoruba, we add three models a) the monoT0pp over the translated corpus we had used in English on the known languages track; b) RankT5 using ByT5 [14]10 as its pretrained language model and trained solely on mMARCO; and c) A cross encoding reranker using Deberta [7]11 as its pretrained language model. Results are presented in Table 10. The first thing we notice is that for the surprise languages, the rerankers improved more than for the known languages, showing that they are more robust to the change of language.

**4.3 Comparison against other teams**

**Dev Leaderboard** So the first thing that we want to discuss when comparing to other teams is the leaderboard during the dev phase. We got very nervous when we saw that a lot of other teams were overtaking us, and we started adding a lot of models (as seen in the previous subsection). Final leaderboard results are made available in Table 11. Our main analysis a-posteriori is that as there

9This was mostly due to desperation when we saw everyone overtaking us on the dev set. These rerankers

were then tested on known languages and they did not improve the results.

[10https://huggingface.co/google/byt5-xl](http://10https://huggingface.co/google/byt5-xl)
[11https://huggingface.co/microsoft/mdeberta-v3-base](http://11https://huggingface.co/microsoft/mdeberta-v3-base)

7

*NLE participation at MIRACL*

Table 9: Comparison of all our first stage models on the surprise dev set.

# model German Yoruba Average

Models
a BM25 22.6 40.6 31.6

b mDPR 48.9 44.4 46.7
c T-SPLADE 43.0 **82.7** 62.9
d SPLADE-mono n/a 39.8 n/a
e mSPLADE-small n/a n/a n/a
f mSPLADE-large 9.4 47.5 28.5
g mSPLADE-sTok 9.7 51.3 30.5
h mContriever **51.8** 55.1 53.5
i mColBERT 49.7 36.6 43.2

Hybrids
0 a+b 54.9 61.6 58.3

1 a+b+c 64.8 84.5 74.7
2 a+b+c+d+e+f+g 61.2 84.6 72.9
3 a+b+c+d+e+f+g+h+i 63.7 84.7 74.2

Best max(a+b+c+d+e+f+g+h+i) **67.4 88.3 77.9** Simple a+c+f+g 56.1 86.1 71.1

Table 10: Full reranking results over “Best” first stage hybrid on the surprise languages.

# model German Yoruba Average

First stage hybrids
F Best 67.4 88.3 77.9

Rerankers
*α* mono mT5-13b 70.6 **94.3 82.5**

*β* Rank mT5-13b **71.5** 92.7 82.1
*γ* Rank mT0-xl 70.1 93.1 81.6
*δ* Rank mT0-xxl 69.8 91.2 80.5
*ϵ* Rank Bloomz 55.3 76.5 65.9
*ζ* Translate + Rank T5 69.1 90.1 79.6
*η* Translate + mono T0pp 65.7 90.6 78.2
*θ* mono ByT5 68.5 92.9 80.7
*λ* DeBerta 65.2 80.7 73.0

Reranking hybrids
Simple max(F + *α* + *β*) 74.2 93.6 83.9

Best max(F + *α* + *β* + *γ* + *δ* + *ϵ* + *ζ*+ *η* + *θ* + *λ*) **74.4 94.8 84.6**

was no training data for both languages, it led to other teams overfitting to the dev set, but we don’t have any idea if that is actually the case.

**Test-B Leaderboard** Much to our surprise the results of Test-B were completely different from what we had seen on the development set and are presented in Table 12. Our reranking ensemble which was the weakest of the top 10 on the development was actually the strongest on the Test-b, while we still kept our advantage in the Yoruba language. We probably got lucky that our models worked pretty well without needing to retrain specifically on the surprise languages and thus were more robust to the change from the dev to the Test-b set on the Surprise languages.

8

*NLE participation at MIRACL*

Table 11: Leaderboard for the development set on the surprise languages.

# team German Yoruba Average Models

1 dm **91.2** 85.3 **88.3**
2 yesbody 78.0 94.5 86.2
3 bott 78.6 93.2 85.9
4 eternal 78.4 93.1 85.8
5 mia 78.4 92.8 85.6
6 ew 78.4 92.8 85.6
7 NOT CIIR 77.7 93.3 85.5
8 ale 78.4 92.0 85.2
9 nle **(ours)** 74.4 **94.8** 84.6

Table 12: Leaderboard for the test-b set on the surprise languages.

# (# dev) team German Yoruba Average Models

1 (9) nle **(ours) 76.6 95.0 85.8**
2 (3) bott 73.7 93.2 83.5
3 (5) mia 73.6 93.2 83.4
4 (6) ew 73.5 93.2 83.4

**5 Query analysis and labeling**

Looking back, we may have made things actually harder for ourselves. If one considers that MIRACL labeling was generated mostly based on BM25/mDPR/mColBERT12, which are weaker than the ones considered in this work, we could be just increasing the amount of false negatives. This is what we aim to study in this section. Note that we do not intend to bash the annotation of MIRACL, a dataset with so many queries and so much annotation is not only essential to the community but very costly (both in monetary value and amount of work).Our goal here is to look into if we made it harder for ourselves or if spending so much time on first stage retrievers was worth it.

Just as a preliminary analysis, let us consider the Judged@ metric. As there’s an average of 10 documents judged per query, one would expect that Judged@10 increases a lot when nDCG@10 increases. However, this is not actually what we see. For example, Hybrid 0 (nDCG@10 57.8) has an average Judged@10 of 50.1, while our best run actually reduces it to 48.5%. While this is probably due to the fact that we learn to not retrieve the known negatives, it is still counter-intuitive to what one would expect (for example Recall@20 increased from 83.0% to 94.0%). This could mean that there are potential false negatives that we found, that would not have been found with BM25 or mDPR, making it harder for the rerankers as not only do they need to learn to properly rank, but also to avoid the labeling bias.

However, actually looking at the data from French13 showed the opposite effect: i.e. There are more false negatives on our top1 that were also present on the top100 of Hybrid 0 than the opposite. The French dev set is composed of 343 queries, of which 217 are judged as positive for our submitted model, 93 are known negatives and only 33 (slightly less than 10%) are unjudged. Of those 33, 5 were not on the top200 of Hybrid 0, and of those 5 only 2 (40%) are actually a false positive and 1 of the others we do not agree with the positive passage. On the other hand, out of the other 28, 15 (53%) are false negatives14 and another 4 (making 19/28 or around 67%) would actually be positives

12which is different from our mColBERT
13We could also have looked into English, but much more time would be needed as there are more than 100

queries where the top1 document is not judged.
14Note that this is actually 4% of the total queries of the devset and thus not that relevant

9

*NLE participation at MIRACL*

if the date/important information was not removed in the crawling process For example, consider the following query and its translation15:

1260128#0 — Quand les premiers Oscars ont-ils eu lieu?” — ’When did the first Academy Award (Oscars) ceremony happened?’

The document found by our system was (Title - Document):

11044#19 — Oscars du cin´

ema - Initialement organis´

ee sous forme de banquet, la

ee par Douglas Fairbanks, a lieu le a` ...— Academy
Awards (Oscars) - The first ceremony, initially organized as a private dinner func-
tion and presided by Douglas Fairbanks, happened on at...

Note that the actual dates and place are missing. What we believe happened is that since some dates are actually links, they were removed by the crawling process.

premi`

ere c´

er´

emonie, pr´

esid´

**Summary of analysis:** Our analysis actually shows that the labeling of the dataset is a smaller problem than we suspected at first, but has led us to find some possible concerns on the Wikipedia crawl missing important parts of the documents. It also shows that the models tend to highly rank a document for a “date” based question, even if no date is actually present in the document, making it so that the model sort of knows the format of a response for a date based question, but does not care about the answer itself, more about its format.

**6 Conclusion**

For the WSDM 23 CUP - MIRACL, we used all the knowledge we had from previous participations at TREC in order to generate a large ensemble of models. While there’s probably not much differ- ence between the full ensemble and a smaller, smarter subset of models, we were actually able to draw from the knowledge embedded in each model, leading to winning both tracks (known languages and surprise languages). As a first start for the MIRACL dataset, we found that this competition was really fruitful and we hope to get a better understanding of what made our runs work so that we can make them more efficient and need less computation in the future. We also hope that this experi- ment report is useful for the readers and are more than happy to further discuss the whole process of transferring the English-based methods to a multilingual setting.

**Acknowledgement**

We would like to thank the organizers of the WSDM 23 CUP - MIRACL for all their hard work in making the competition not only take place, but also run smoothly. They did a mostly thankless job of getting more than 700k dedicated annotations for the multilingual dataset used in this competition, which will for sure foster development in this much-needed subject.

**References**

[1] E. Bassani and L. Romelli. ranx.fuse: A python library for metasearch. In M. A. Hasan and L. Xiong, editors, *Proceedings of the 31st ACM International Conference on Information &* *Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 4808–4812. ACM, 2022.

[2] L. Bonifacio, V. Jeronymo, H. Q. Abonizio, I. Campiotti, M. Fadaee, R. Lotufo, and R. Nogueira. mmarco: A multilingual version of the ms marco passage ranking dataset. *arXiv* *preprint arXiv:2108.13897*, 2021.

[3] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei. Scaling instruction-finetuned language models, 2022.

15translation made by a fluent but non-native speaker of both languages

10

*NLE participation at MIRACL*

a, J. Cross, O. C¸ elebi, M. Elbayad, K. Heafield, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

[4] M. R. Costa-juss`

[5] T. Formal, C. Lassance, B. Piwowarski, and S. Clinchant. From distillation to hard negative sampling: Making sparse neural ir models more effective. In *Proceedings of the 45th Interna-* *tional ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2353–2359, 2022.

[6] L. Gao, Z. Dai, and J. Callan. Rethink training of bert rerankers in multi-stage retrieval pipeline. In *Advances in Information Retrieval: 43rd European Conference on IR Research,* *ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*, pages 280–286. Springer, 2021.

[7] P. He, J. Gao, and W. Chen. Debertav3: Improving deberta using electra-style pre-training

with gradient-disentangled embedding sharing, 2021.
[8] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave. To-

wards unsupervised dense information retrieval with contrastive learning. *arXiv preprint* *arXiv:2112.09118*, 2021.

[9] O. Khattab and M. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on* *research and development in Information Retrieval*, pages 39–48, 2020.

[10] C. Lassance and S. Clinchant. An efficiency study for splade models. In *Proceedings of* *the 45th International ACM SIGIR Conference on Research and Development in Information* *Retrieval*, pages 2220–2226, 2022.

[11] C. Lassance, H. D´

ejean, and S. Clinchant. An experimental study on pretraining transformers
from scratch for ir. *arXiv preprint arXiv:2301.10444*, 2023.

[12] N. Muennighoff, T. Wang, L. Sutawika, A. Roberts, S. Biderman, T. L. Scao, M. S. Bari,
S. Shen, Z.-X. Yong, H. Schoelkopf, et al. Crosslingual generalization through multitask fine-
tuning. *arXiv preprint arXiv:2211.01786*, 2022.

[13] R. Nogueira, Z. Jiang, and J. Lin. Document ranking with a pretrained sequence-to-sequence

model. *arXiv preprint arXiv:2003.06713*, 2020.
[14] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel.

Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the*
*Association for Computational Linguistics*, 10:291–306, 2022.

[15] X. Zhang, N. Thakur, O. Ogundepo, E. Kamalloo, D. Alfonso-Hermelo, X. Li, Q. Liu,
M. Rezagholizadeh, and J. Lin. Making a miracl: Multilingual information retrieval across
a continuum of languages. *arXiv preprint arXiv:2210.09984*, 2022.

[16] H. Zhuang, Z. Qin, R. Jagerman, K. Hui, J. Ma, J. Lu, J. Ni, X. Wang, and M. Bendersky.
Rankt5: Fine-tuning t5 for text ranking with ranking losses. *arXiv preprint arXiv:2210.10634*,
2022.

11