AN ASR N-BEST TRANSCRIPT NEURAL RANKING MODEL FOR SPOKEN CONTENT RETRIEVAL

Yasufumi Moriya, Gareth. J. F. Jones

ADAPT Centre, School of Computing, Dublin City University, Dublin 9, Ireland

ABSTRACT

Spoken Content Retrieval (SCR) using ASR transcripts is increasingly important for multimedia content archives. However, SCR is often impacted by ASR errors. In recent years novel neural ranking methods for information retrieval (IR) have achieved improved search effectiveness over established methods. In this study, we examine neural ranking methods in SCR. Specifically we introduce two new neural ranking methods designed for use with errorful ASR transcripts. In the first, we train a neural ranking model using both manual transcripts and ASR transcripts or manual transcripts and artificially created ASR transcripts. In the second, we use an ASR N-best extension of two existing neural ranking methods: Deep Relevance Matching Model (DRMM) and Position-Aware Convolutional Recurrent Matching (PACRR). We report two sets of SCR experiments which evaluate our neural ranking methods. In the first, we use ASR transcripts to retrieve documents from an archive of spoken instruction videos. In the second, we examine a known-item search on a collection of user-generated spoken videos. We find that training the neural ranking model on both ASR and manual transcripts improves Mean Reciprocal Rank (MRR). The N-best extension of PACRR is particularly effective in both experiments in comparison to a standard BM25 IR model.

Index Terms— Spoken content retrieval, Neural ranking, Information Retrieval, ASR N-best

1. INTRODUCTION

The ever growing archives of digital multimedia spoken content, such as user-generated videos and podcasts, are increasing the importance of effective spoken content retrieval (SCR) systems. Existing SCR systems generally operate as a simple combination of content transcription using automatic speech recognition (ASR) with information retrieval (IR) methods to return a ranked list of documents potentially relevant to a given user search query. While ASR has been significantly improved for well defined tasks including broadcast speech and audio books [1, 2] other content such as user-generated content is highly varied and word error rates (WERs) for such data can often reach 30-40% which can degrade IR effectiveness [3].

Existing work on SCR has focused on the use of established standard IR models such as BM25 [3, 4]. However, similar to ASR, recent years have seen the introduction of effective neural methods in IR [5]. Such methods, referred to as neural rankers, assign a relevance score to a document for a given search query. Two neural models which have shown superior performance to traditional IR models are a Deep Relevance Matching Model (DRMM) [6] and a Position-Aware Convolutional Recurrent Relevance Matching (PACRR) model [7].

In this paper, we investigate the use of neural ranking models for SCR. We propose two approaches to adapting neural rankers to address the problem of ASR noise in SCR. The first model examines the use of manual transcripts of spoken content, corresponding ASR transcripts and artificially created ASR transcripts for training of the neural models. The second method is an N-best extension of neural models which uses the N-best output of an ASR system to address the problem of correct words missing from a 1-best ASR transcript. In our invesigation we build DRMM and PACRR models for spoken video transcripts for the How2 dataset [8]. Using video titles as search queries, we find that the N-best PACRR model improves mean reciprocal rank (MRR) score from 43.9% to 51.3% over a BM25 baseline. In a second experiment for a known item search task on a collection of user-generated spoken videos using Blip10000 [9], we find that the N-best PACRR model improves MRR from 39.38% to 56.02% on a development query set and from 37.23% to 47.08% on a test set over the BM25 baseline. This latter experiment examines a domain mismatch scenario where the neural ranking models are trained on How2 and evaluated on Blip10000 known item search task.

The remainder of the paper is organised as follows. Section 2 outlines existing relevant work. Section 3 describes our proposed approaches to building noise robust neural rankers for SCR. Section 4 shows experimental results using these models, followed by conclusions in Section 5.

2. RELATED WORK

As described in Section 1, there is currently much interest in neural ranking methods in the field of IR [5]. However, the only previous work that we are aware of applying neural re-ranking in SCR is a submission to the TREC 2020 Podcast track [10]. This used a passage retrieval model to retrieve segments of podcasts with re-ranking using a T5 model. Other recent work on SCR has focused on learning representations from acoustic signals and searching for documents or spoken terms using this representation (query-by-example) [11, 12, 13].

N-best re-ranking is a classic approach for improving WER of ASR transcripts [14]. Our work using N-best hypotheses for neural re-ranking is inspired by [15], where N-best transcripts are used with a spoken language understanding system to overcome the errors in 1-best transcripts. There has been some limited research examining the use of N-best ASR transcripts using non-neural IR models in SCR, but these have shown only limited value benefit to SCR effectiveness, e.g. [16]. We are not aware of any previous work which makes use of N-best ASR transcripts in neural re-ranking for SCR.

3. ASR NOISE ROBUST NEURAL RANKING MODELS FOR SCR

In this section we introduce the DRMM and PACRR neural ranking models and our proposals for using them in SCR with noisy ASR transcripts. The DRMM and PACRR models analyse the interactions between a query-document pair to produce a relevance score for the document [6, 7]. Suppose a query consists of Q terms with $q = w_1^q w_2^q ... w_Q^q$ and a document of its length D consists of $d = w_1^d w_2^d ... w_D^d$, query-document interactions are represented as a matrix of similarity of each query term against each document term. Typically, the query terms and document terms are represented as fixed dimension word embedded vectors [17], and cosine similarity is used to measure similarity, resulting in a similarity matrix of size $S \in [-1,1]^{|Q|\times|D|}$. While PACRR directly applies convolution to the similarity matrix of query and document terms, DRMM transforms each row of the similarity matrix corresponding to interactions of one query term against all document terms into a histogram of cosine similarity values with b bins. The advantage of DRMM is that variable length documents can be represented as vectors of the same size, while PACRR requires padding or cut-off of documents to prepare similarity matrices of the equal size. Another important difference between them is that PACRR can exploit word order preserved in the similarity matrix unlike DRMM.

To train the neural ranking model, a triplet of a query Q, a document relevant to Q and a document non-relevant to Q, (q, d^+, d^-) is taken as its input [6, 7]. The model is trained to predict a relevance score 1 for true query-document pairs and 0 for false query-document pairs as follows:

$$L(q, d^+, d^-; \Theta) = max(0, 1 - s(q, d^+) + s(q, d^-)) \quad (1)$$

where $s(q, d^+)$ and $s(q, d^-)$ are relevance scores from positive and negative query-document pair and Θ is the model parameter. While the model is not directly optimised to increase MRR scores, the progress of model training is monitored using MRR on a separate query-document validation set.

We propose two approaches to building neural ranking models robust to ASR transcription errors. In our first approach we examine ASR transcripts, artificially created ASR transcripts and manual transcripts for d^+ and d^-. We hypothesise that either ASR transcripts or artificially created ASR transcripts can enable the neural ranking system to learn error patterns of ASR transcripts to produce a more robust re-ranker for SCR with ASR transcripts. Our second approach is an N-best extension of the neural ranking model. It is known that re-ranking N-best hypotheses using a strong neural language model can reduce WERs [14]. This indicates that the best transcription of a document is not always its first hypothesis. We hypothesise that using an N-best extended neural ranking model can determine a more reliable relevance score than a single 1-best transcript.

3.1. Adding ASR transcripts to manual transcripts

In our first approach we seek to build neural ranking models robust to ASR noise by using ASR transcripts or ASR transcripts artificially created from manual transcripts. The use of artificially created ASR transcripts enables us to control WERs of transcripts, and to generate diverse ASR error patterns than the single output of an ASR system. In Section 4.2 we examine model training on ASR transcripts and manual transcripts individually, combination of natural and artificial ASR transcripts with manual transcripts. Combination of two is achieved by including them in a training corpus independently for each document ID. We now explain the creation of our artificial ASR transcripts.

Suppose we have a query Q and a manual transcript of a relevant document D for Q, To generate $\tilde{D}$, a corrupted version of D, we compute a probability of substituting a term w_i for w_j from an actual collection of ASR transcripts using

$$P(w_i \mapsto w_j) = \frac{c(w_i \mapsto w_j)}{\sum_{k=1}^{K} c(w_i \mapsto w_k)}$$

where the numerator is the number of times w_i is substituted for w_j and the denominator is the total number of w_i substituted for another word. Terms within Q are selected randomly, with each one having a selection probability p_q, to construct a subset of the query terms. Further, the terms within D are chosen randomly with probability p_d, with each selected term being corrupted. One of the three ASR error types, deletion, insertion or substitution is applied to each selected term with an equal likelihood. Deletion removes the a selected term from D. A term to insert or substitute for w_j is sampled randomly from the probability distribution of $P(w_i \mapsto w_j)$. Insertion inserts w_j after w_i, while substitution replaces w_i with w_j.

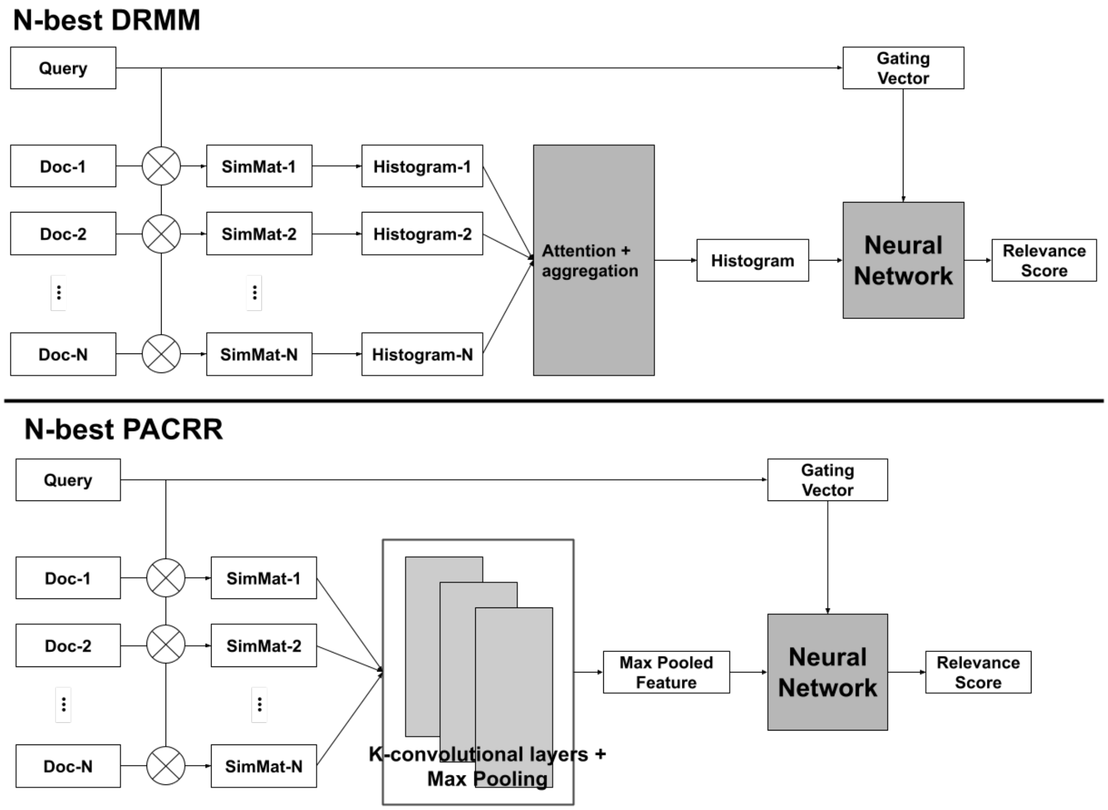

Fig. 1. *Top* N-best version of DRMM, *Bottom* N-best version of PACRR. Grey squares indicate these are learnable paramters of the model. The crossing of Query and Doc is the interaction of query terms and document terms. SimMat is a similarity matrix of query terms and document terms.

3.2. N-best extension of DRMM

The upper part of Figure 1 shows our N-best extension of DRMM. The standard DRMM model takes a single document D as input along with a query Q to produce a relevance score. In our N-best configuration, N transcripts for each spoken document D are used. The core idea of our N-best DRMM is to apply attention weights to N matching histograms, and to aggregate them to form a single matching histogram as an input of a feed-forward network. This enables the ranking model to produce a relevance score which takes account of all N hypotheses of document D. The aggregated output of the feed-forward network is a relevance score from the DRMM.

In the N-best settings, each of the N-best transcripts of D is transformed into N similarity matrices S_1, S_2 $, ..., S_N$. As mentioned earlier, the input of a DRMM is a fixed length matching histogram. Therefore, each row of the similarity matrices is transformed into a histogram of b bins. This produces N matching histograms $M_1, M_2, ..., M_N$ with size $M_i^{|Q|\times|b|}$. Since the value of cosine similarity ranges in $[-1, 1]$, for example, when the number of bins is set to 5, elements of a similarity matrix will be sorted according to [1.0, -0.6], (-0.6, -0.2), [-0.2, 0.2), [0.2, 0.6), [0.6, 1.0]. Following [6], a logarithm is applied to each frequency bin of the matching histogram.

The goal here is to apply attention mechanism and aggregation to N matching histograms and form a single matching histogram M_D in which information from N documents is gathered. This can be expressed as follows,

$$a_{ij} = \frac{exp(\mathbf{w}_a \mathbf{m}_{ij})}{\sum_{i=1}^{N} exp(\mathbf{w}_a \mathbf{m}_{ij})} \tag{2}$$

$$\mathbf{m}_{Dj} = \sum_{i=1}^{N} a_{ij} \mathbf{m}_{ij} \tag{3}$$

where $\mathbf{m}_{ij}$ is a vector of the ith document of the jth query's matching histogram (jth row of M_i), $\mathbf{w}_a$ is an attention wegith vector, and a_{ij} is a weight vector for $\mathbf{m}_{ij}$. The aggregated matching histogram of N-best documents $\mathbf{M}_D$ is input to a neural network. The application of the gating mechanism is described in the original DRMM paper [6]. Following [18], however, we use a concatenation of inverse document frequency (IDF) and word embedding of query term w_i^q for the gating vector. If a feed forward network with Z layers is applied to the aggregated matching histogram of N-best,

$$\mathbf{m}_j^{(0)} = \mathbf{m}_{Dj} \tag{4}$$

$$\mathbf{m}_j^{(z)} = tanh(\mathbf{W}^{(z)}\mathbf{m}_j^{(z-1)} + \mathbf{b}^{(z)}) \tag{5}$$

$$o = \sum_{j=1}^{Q} g_j \mathbf{m}_j^{(Z)} \tag{6}$$

where $\mathbf{m}_j^{(z)}$ is a hidden representation of the jth row of a matching histogram after the zth layer, $\mathbf{W}^{(z)}$ the weight matrix of the zth layer, $\mathbf{b}^{(z)}$ the bias term of the zth layer, g_j the gating weight for a query term j and o is the final relevance score. The gating weight g_j can be computed using the attention mechanism similar to Eq 2, except using a different weight vector, and $\mathbf{m_{ij}}$ is replaced for concatenation of IDF and word embedding of query term j.

3.3. N-best extension of PACRR

The lower part of Figure 1 shows our N-best extension of PACRR. Similar to N-best DRMM, each of the N-best transcripts is transformed into a similarity matrix $S_1, S_2, ..., S_N$. Unlike DRMM, however, PACRR directly takes as input similarity matrices and a length of document is defined as a hyper-parameter [7]. Therefore, the size of a similarity matrix is $S \in [-1,1]^{|Q| \times |l_d|}$, where l_d is the maximum length of document set as a hyper-parameter. Padding is applied to documents below this length and documents exceeding this length are cut-off to contain l_d terms.

The core concept of PACRR is to apply $l_g - 1$ convolutional layers to similarity matrices with kernel sizes of $2 \times 2, 3 \times 3, ..., l_g \times l_g$. For instance, the kernel size of 2×2 corresponds to bi-gram similarity of query terms and document terms. While vanilla PACRR applies the convolutional layers to a single similarity matrix of query and document terms, N-best PACRR applies convolution to N similarity matrices at one time. This is analogous to image recognition where an input image typically consists of red, green, blue (RGB) channels; hence three channels. Instead of three, the number of input channels of convolutional layers for N-best PACRR is set to N. This input is formed by stacking N similarity matrices $S_1, ..., S_N$ and creating a 3D tensor S. A hyper-parameter for convolutional layers is the number of filters l_f. This parameter controls the number of output values for each convolutional operation. The stride size is set to (1, 1), which proceeds convolutional operations 1 step at a time. To retain the identical size of output of convolutional layers to that of the similarity matrices, padding is applied to similarity matrices with the size of kernel size $k - 1$ leading to a matrix of size $|Q + k - 1| \times |l_d + k - 1|$. The convolutional operation with the $k \times k$ kernel size can be expressed as:

$$C_{l_f}^k = Conv^k(S) \tag{7}$$

The output of a convolutional layer with the kernel size k is a tensor of size $Q \times l_d \times l_f$. Note that uni-gram similarity of query terms and document terms is the original input similarity matrix $C_N^1 = S$. The convolutional operations therefore lead to $C_N^1, C_{l_f}^2, ..., C_{l_f}^{l_g}$.

The convolutional layers are followed by two max pooling layers. The first retains the most salient values over the N-best dimension for C_N^1 and over the filter dimension for $C_{l_f}^2, ..., C_{l_f}^{l_g}$. This produces l_g 3D tensors of size $Q \times l_d \times 1$. Another hyper-parameter l_s is required for the second l_s max pooling layer which retains l_s salient values over the query dimension of each l_g-gram tensor, followed by concatenation of these tensors to form a matrix of the size $Q \times (l_g \times l_s)$. This is an input matrix of the feed-forward network.

Although the original paper and its follow-up work [7, 18] propose to convert the gating vector into gating weights, and concatenate the gating weights with this input matrix, we found that it was empirically better to apply the feed-forward network to the feature after max pooling, to multiply the gating weights by output of the feed-forward network, and to aggregate this output for a relevance score. This is essentially how a relevance score is computed in DRMM using the weights from the gating vector shown in Eq 4-6, except that the non-linear function used is ReLU instead of the hyperbolic tangent as suggested in [7].

4. EXPERIMENTS

4.1. Experimental setup

Training Dataset We use the How2 dataset for development of our neural ranking systems [8]. The How2 dataset consists of 19,770 instruction videos, manual transcripts and their titles[1]. We split the dataset into 14,770 videos for training of the neural ranking models and 5,000 videos for evaluation. The 5,000 evaluation videos consist of the 391 videos of the official dev and test sets, and 4,609 videos randomly selected from the remaining videos. The video titles and corresponding manual or ASR transcripts were used as true query-document pairs to train the model. ASR transcripts of the How2 data were generated using a hybrid HMM-DNN ASR system trained on 960 hours of LibriSpeech data [19]. The architecture of the ASR system used 17 time-delayed layers using lattice-free maximum mutual information [20]. The WER of the ASR How2 transcripts for the whole dataset including train, dev and test was 31.1%. We generated {5,10,20}-best transcripts for experiments on the N-best neural ranking models. To obtain a gating vector, we computed IDF of terms from the manual transcripts of training set. Each of the query terms was a concatenation of its IDF with its pre-trained Glove word embedding with 300 dimensions [17].

Model Architecture The size of histogram bins for DRMM

[1] Some of the videos may no longer be available on YouTube.

was set to 5, since this was empirically better than 3,10. The DRMM architecture is as follows: an input layer takes a matching histogram of 5 dimensions and produces hidden representation of 30 dimensions, followed by two hidden layers of size 5 and 1 respectively. These are the same as those described in the original paper [6]. After each of these layers, a non-linear hyperbolic tangent function was applied. For PACRR, the maximum number of document terms l_d was set to 700. The number of filters of the convolutional layers l_f was set to 16. The number of convolutional layers l_g was 3; hence bi-gram convolution and tri-gram convolution. These hyper-parameters for PACRR were empirically chosen after testing several values. For a 1-best input document, the number of salient values to be kept at second max pooling l_s was set to 3, while when an input document was 10-best, this parameter was set to 5. Empirically, setting l_s to 3 10-best transcripts led to worse results and vice versa. This indicates that N-best transcripts contain more useful information than 1-best transcripts. l_s was set to 5 when using 5-best transcripts and to 10 when using 20-best transcripts.

Other Hyper-parameters The initial learning rate of the DRMM was set to 0.001 and that of the PACRR was 0.0005. These learning rates ensured model learning to converge within 30 epochs and to produce the optimal results. The Adagrad optimiser, which adjusts parameter updates given input, was used following [6]. The mini-batch size was 100 to train a model faster while data can be loaded in a memory. For each How2 video title (query) of the training data, we ran BM25 to obtain a ranked list of documents and used this list to choose 5 negative samples, which could reasonably be confused with actual correct documents given their titles. To monitor performance of the neural ranking models, 100 pairs of video titles and their corresponding document were randomly chosen from 14,770 title-document pairs and kept for validation. For each validation sample, the model produced relevance scores for 500 documents from the list of BM25 outputs, for which we computed mean reciprocal rank (MRR). We kept the version of the model which produced the highest MRR score from 30 epochs. When adding artificially created ASR transcripts to the training data, one of these was generated using $p_q = 0.1$ and $p_d = 0.1$.

Evaluation on How2 The 5,000 evaluation video titles were used as queries and the corresponding 5,000 documents ranked for each query. The 5,000 evaluation documents were either ASR transcripts or N-best ASR transcripts, manual transcripts were not used for search experiments. BM25 was first applied to rank the documents for each title. The neural ranking models were used to re-rank the top 1,000 documents returned by BM25. Documents were represented as $\{5,10,20\}$-best ASR transcripts when using the N-best neural ranking model.

Table 1. MRR results for neural re-ranking models using ASR transcripts and artificially created ASR transcripts.

model	train_data	MRR
BM25	N/A	43.92
DRMM	asr	41.3
DRMM	manual	42.9
DRMM	manual+asr	43.64
DRMM	manual+asr_artificial	41.52
PACRR	asr	**47.67****
PACRR	manual	46.97*
PACRR	manual+asr	**48.33****
PACRR	manual+asr_artificial	46.56

Evaluation on Blip10000 The second experiments were carried out on the Blip10000 dataset of semi-professional user-generated online video content [9]. The goal of this experiment was to evaluate the neural ranking system in a domain-mismatch scenario for a known item search task. We created manual transcripts for 670 videos from the dev set (20 hours) and 566 videos of test set (15 hours) to evaluate the quality of ASR transcripts. We developed another ASR system on manual transcripts using the How2 dataset and the untranscribed dev set of Blip10000. The system was trained using a semi-supervised technique when dealing with the untranscribed dev set of Blip10000. This system generated ASR transcripts of WER 26.82% on the dev subset and 39.21% on the test subset. The same system was used to generate ASR transcripts of the whole Blip10000 corpus. We created 15 known item queries for dev and 35 queries for test using Amazon Mechanical Turk. These 50 videos were selected randomly from transcribed parts of the dataset. Similar to the first experiments on How2, for each query, the neural ranking models were used to re-rank top 1,000 videos returned by the BM25 model. For dev queries, 3,061 videos were searched and for test queries, 3,371 videos were searched.

4.2. Using ASR transcripts for neural ranking model training

Table 1 shows results of retrieving How2 videos for title queries. Higher MRR scores indicate that the user are more likely to find the target document at higher rank (i.e., sooner). As can be seen in the table, PACRR was more successful in retrieving documents than the BM25 baseline and DRMM. The best MRR score observed was MRR 48.33 using PACRR trained on a combination of manual transcripts and ASR transcripts. None of the DRMM models produced a better MRR score than BM25. Asterisks (*) and (**) denote MRR results which are statistically significantly better than the baseline with $p < 0.05$ and $p < 0.01$ respectively.

Regarding the effectiveness of training a model on manual, ASR, manual and ASR, and manual and artificially created ASR transcripts, we can conclude that training a model on manual and ASR transcripts is the best on How2. Surprisingly, PACRR trained on ASR transcripts produced better

Table 2. MRR results for N-best neural re-ranking models using ASR transcripts and artificially created ASR transcripts.

model	train_data	MRR
BM25-1best	N/A	43.92
BM25-nbest	N/A	43.55
DRMM-n10	asr	44.27
DRMM-n10	manual+asr	44.83
DRMM-n10	manual+asr_artificial	44.85
PACRR-n10	asr	**50.97****
PACRR-n10	manual+asr	**51.27****
PACRR-n10	manual+asr_artificial	49.61**
DRMM-n5	manual+asr	42.56
DRMM-n20	manual+asr	44.83
PACRR-n5	manual+asr	48.00**
PACRR-n20	manual+asr	50.43**

MRR than PACRR trained on manual transcripts. The reason for this is likely to be that ASR transcripts of the training set and the evaluation set were created using the same ASR system. These results indicate that the neural ranking systems can learn error patterns of ASR transcripts produced by this system, and effectively apply this knowledge to ranking documents. The artificially created ASR transcripts were not found to be effective as ASR transcripts, indicating that these do not resemble transcripts created by the ASR system.

4.3. N-best extension of DRMM and PACRR

Table 2 summarises MRR results of neural ranking models trained on N-best transcripts. The result of BM25-nbest was produced by keeping 1-best transcripts as base transcripts and adding unique words which are not present in the 1-best transcripts to these base transcripts. As can be seen in Table 2, the standard BM25-1best was still better than BM25-nbest. When training an N-best neural ranking system, training can be done by only using N-best ASR transcripts (asr), using manual transcripts as 1-best and $N-1$ ASR transcripts (manual+asr), and using manual transcripts as 1-best and $N-1$ artificially created ASR transcripts (manual+asr_artificial).

In this experiment, PACRR again achieved better results than DRMM with 51.27 MRR when the model was trained on manual transcripts and $N-1$ ASR transcripts. While DRMM produced better results than BM25, unlike in Table 1, its results are still lower than those for PACRR. This may arise from PACRR exploiting word order in the documents. There was no difference between N-best DRMM trained on 10-best and 20-best transcripts, although training on 10-best was better using 5-best or 20-best. This indicates that all the information useful for improve N-best retrieval effectiveness for the How2 dataset is contained in the 10-best lists.

Table 3. MRR results for neural ranking models and BM25 on known item queries for Blip10000. The rightmost column shows relative changes to MRR score of the BM25 baseline.

model	train_data	dev	test	+/-BM25
BM25	N/A	39.38	37.23	0.0
DRMM	manual+asr	22.75	31.97	-10.95
DRMM-n10	manual+asr	21.29	28.73	-13.30
PACRR	manual+asr	40.5	36.24	+0.07
PACRR-n10	manual+asr	55.83	46.90	+13.06**
DRMM	asr	19.26	27.97	-14.69
DRMM-n10	asr	21.25	29.53	-12.92
PACRR	asr	41.07	36.72	+0.59
PACRR-n10	asr	**56.02**	**47.08**	**+13.25****

4.4. Known item search on Blip10000

Table 3 summarises results for MRR on our known-item search task for Blip10000 using the neural rankers trained using the How2 data. Unlike the results in Table 2, the N-best version of the ASR PACRR is slightly better than the combined manual and ASR model. This means that the most effective re-ranker can be trained without use of manual transcripts. DRMM consistently performed poorly, with interestingly, the N-best DRMM performing worse than vanilla DRMM. This could be accounted for by the difference in the lengths of transcripts between How2 and Blip10000. The average length of How2 documents was 251.68 words with standard deviation 109.02, while that of Blip10000 was 1458.46 with standard deviation 2621.16. Since DRMM exploits frequency of query-document similarities, the varied document lengths in Blip10000 could have a negative impact on the model. On the other hand, PACRR was robust to the varied document lengths observed in Blip10000.

5. CONCLUSION

In this work, we proposed two approaches to the application of neural ranking models in SCR. The first combines real or artificial ASR transcripts with manual transcripts to train a neural ranking system. The second extends a neural ranking system to make use of N-best ASR transcripts. Our experimental results show that use of both manual and ASR transcripts improves PACRR model. The PACRR model also benefits more from the N-best extension than DRMM. Overall, our experiments show very good effectiveness for the PACRR model. Particularly encouraging is the result using only ASR transcripts, showing that these techniques can be applied for SCR without the need for large amounts of in domain manually transcribed content.

Acknowledgement: This work was supported by Science Foundation Ireland as part of the ADAPT Centre (Grant 13/RC/2106) at Dublin City University.

6. REFERENCES

[1] Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur, "End-to-end speech recognition using lattice-free mmi," in *Proceedings of Interspeech*, 2018, pp. 12–16.

[2] Christoph Lüscher, Eugen Beck, Kazuki Irie, Markus Kitza, Wilfried Michel, Albert Zeyer, Ralf Schlüter, and Hermann Ney, "RWTH ASR Systems for LibriSpeech: Hybrid vs Attention," in *Proceedings of Interspeech*, 2019, pp. 231–235.

[3] M. Larson and G. J. F. Jones, "Spoken Content Retrieval: A survey of techniques and technologies," *Foundations and Trends in Information Retrieval*, vol. 4, no. 4-5, pp. 235–422, 2012.

[4] S. E. Robertson and S. Walker, "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval," in *Proceedings of ACM SIGIR Conference*, 1994, SIGIR '94, p. 232–241.

[5] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng, "A deep look into neural ranking models for information retrieval," *Information Processing & Management*, vol. 57, no. 6, pp. 102067, 2020.

[6] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft, "A deep relevance matching model for ad-hoc retrieval," in *Proceedings of the International on Conference on Information and Knowledge Management*. 2016, p. 55–64, Association for Computing Machinery.

[7] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo, "PACRR: A position-aware neural IR model for relevance matching," in *Proceedings of EMNLP*. 2017, pp. 1049–1058, Association for Computational Linguistics.

[8] R. Sanabria, O. Caglayan, S. Palaskar, D. Elliott, L. Barrault, L. Specia, and F. Metze, "How2: a large-scale dataset for multimodal language understanding," in *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. NeurIPS, 2018.

[9] Sebastian Schmiedeke, Peng Xu, Isabelle Ferrané, Maria Eskevich, Christoph Kofler, A. Martha Larson, Yannick Estève, Lori Lamel, Gareth J. F. Jones, and Thomas Sikora, "Blip10000: a social video dataset containing spug content for tagging and retrieval," in *Proceedings of ACM Multimedia System Conference*, 2013.

[10] Petra Galuscáková, Suraj Nair, and Douglas W. Oard, "Combine and re-rank: The university of maryland at the TREC 2020 podcasts track," in *Proceedings of Text REtrieval Conference, TREC*. 2020, vol. 1266 of *NIST Special Publication*, National Institute of Standards and Technology (NIST).

[11] Yi-Chen Chen, Sung-Feng Huang, Chia-Hao Shen, Hung-yi Lee, and Lin-shan Lee, "Phonetic-and-semantic embedding of spoken words with applications in spoken content retrieval," in *Proceedings of Spoken Language Technology Workshop (SLT)*, 2018, pp. 941–948.

[12] Shane Settle, Keith Levin, Herman Kamper, and Karen Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," in *Proceedings of Interspeech*, 2017, pp. 2874–2878.

[13] Yougen Yuan, Cheung-Chi Leung, Lei Xie, Hongjie Chen, and Bin Ma, "Query-by-example speech search using recurrent neural acoustic word embeddings with temporal context," *IEEE Access*, vol. 7, pp. 67656–67665, 2019.

[14] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Proceedings of Interspeech*, 2010, pp. 1045–1048.

[15] Xinyue Liu, Mingda Li, Luoxin Chen, Prashan Wanigasekara, Weitong Ruan, Haidar Khan, Wael Hamza, and Chengwei Su, "Asr n-best fusion nets," in *Proceedings of ICASSP*, 2021, pp. 7618–7622.

[16] Kenney Ng, *Subword-based Approaches for Spoken Document Retrieval*, Ph.D. thesis, Massachusetts Institute of Technology, 2000.

[17] Jeffrey Pennington, Richard Socher, and Christopher Manning, "GloVe: Global vectors for word representation," in *Proceedings of EMNLP*. 2014, pp. 1532–1543, Association for Computational Linguistics.

[18] Ryan McDonald, George Brokos, and Ion Androutsopoulos, "Deep relevance ranking using enhanced document-query interactions," in *Proceedings of EMNLP*, 2018, pp. 1849–1860.

[19] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proceedings of ICASSP*, 2015, pp. 5206–5210.

[20] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," in *Proceedings of Interspeech*, 2016, pp. 2751–2755.