



Learning to Rank for Arabic Transcriptions Retrieval

Farida Sabry¹, Mayada Hadhoud², Nevin Darwish³

Computer Engineering Department, Faculty of Engineering, Cairo University, Egypt

¹faridasabry@eng.cu.edu.eg, ²mayadahadhoud@gmail.com, ³ndarwish@ieee.org

Abstract

The amount of spoken documents being shared on the web per minute is increasing dramatically posing a true challenge for any search engine in order to satisfy its customers' queries. With the ingoing improvement in the speech recognizers accuracy, this research addresses the problem of ranking transcriptions that can be obtained by speech recognizers to enhance search engine ranking results. Depending on the title of the video and some of its meta-data only is not sufficient for some queries that have the information need to get relevant spoken segments within audio files. Feature extraction based on both the meta-data of the spoken documents and the timed spoken content transcription for an Arabic audio dataset for Quran is proposed. The results revealed that applying learning to rank techniques are superior to the baseline unsupervised BM25 scoring. In addition, using transcription-based features proved its effectiveness in terms of both the Normalized Discounted Cumulative Gain (NDCG@10) and Expected Reciprocal Rank (ERR@10).

Keywords: spoken content retrieval, learning to rank, ranking.

1. Introduction

The volume of multimedia on the web is increasing dramatically; hundreds of hours of speech content are being shared on the web every minute in the form of video or audio on video sharing websites. Valuable information in this content cannot be effectively browsed and searched without effective retrieval and ranking. Spoken Content Retrieval (SCR) can be defined as the task of returning speech media results that are relevant to the query information need (Larson & Jones 2011). A general architecture for SCR system can be viewed as shown in Figure 1. The automatic speech recognition (ASR) component is used to convert speech in the audio collection of spoken documents into a lattice representation or best transcription together with timing information. The indexing module is then used to index the lattice representation or best transcription together with the metadata extracted for the spoken documents to produce a timed index. The user query terms are then analyzed using the same analyzer used by the indexer, this is represented as a dotted line in Figure 1. The index is then searched and matching audio segments are retrieved and ranked. The retrieved results are ranked mainly based upon the title of the video and its meta-data like description, tags, views, ratings, playlist, shares, comments, age of video, channel views and subscribers inbound links (e.g. links from outside of YouTube pointing to your videos). This way of ranking may result in inaccurate sorted retrieved results because the title of the video and its metadata can be faked by dishonest search engine optimization (SEO) fellows. They rely on keywords known for their high hit rate while the actual content of the video may be irrelevant causing spam documents to be ranked among legitimate ones. The ranking of the retrieved results can be based on one of the information retrieval (IR) models including: boolean model (BM), vector space model (VSM), probabilistic model (PM) or language modeling (LM).

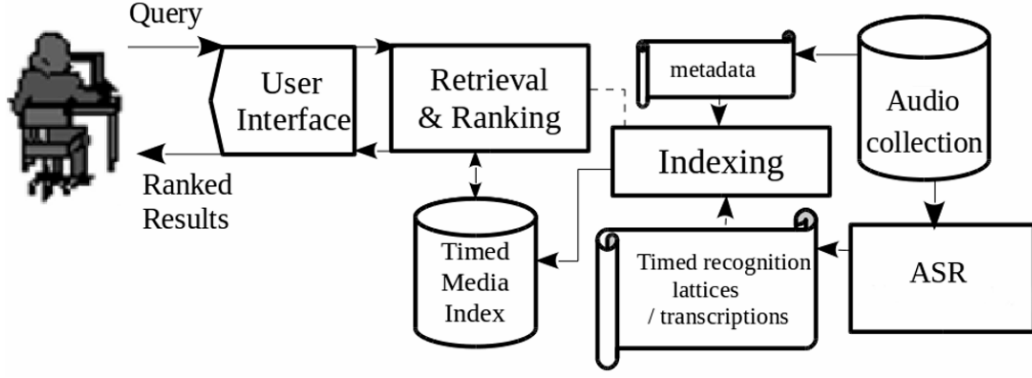


Figure 1. Spoken content retrieval architecture

Learning to rank (L2R) is a research field that started in the last decade aiming to automatically create a ranking model through applying machine learning techniques to features extracted from queries and documents. A ranking model is learned based on training data and then applied to the unseen test data. According to the survey (Liu 2011), the learning-to-rank techniques are classified as either pointwise, pairwise or listwise.

It is noted that learning to rank techniques have been successfully adopted in other tasks different from conventional text IR. Some examples include ranking tweets, searching entities in (Chen et al., 2016) and searching images in (Zhao & Zhang, 2015). To our knowledge, the only other work that applied learning to rank techniques in SCR is (Ma, 2015). They proposed using them in keyword search (KWS) for learning from features like word/morpheme burstiness, rescored confusion network posteriors, acoustic/prosodic qualities and phoneme recognition results.

This paper proposes to use learning to rank techniques to solve the problem of ranking in SCR based on features extracted from both the Arabic spoken content timed transcriptions as well as meta-data as to get the illegitimate matches down in the retrieved result set.

2. Research Method: L2R framework for SCR, dataset and results

In most learning to rank systems for text documents ranking, the features are the scores of common unsupervised ranking algorithms applied to different document representations. The different ranking algorithms and representations provide different views of the relevance of the document to the query. The multiple perspectives represented by these features are the backbone of any L2R system. The same approach can be applied to audio transcriptions search by extracting features for query-transcriptions pairs. With features extracted for spoken documents, all learning to rank techniques developed for ranking text documents can be used.

L2R framework in SCR can be represented as in Figure 2. The query set Q ; is the set of queries used for training and testing $Q = \{q_1, q_1, \dots, q_m\}$. The set of all audio transcriptions is $D = \{D_1, D_2, \dots, D_i, \dots, D_m\}$, $D_i = \{d_{i,1}, d_{i,2}, \dots, d_{i,n_i}\}$ is the set of transcription documents of size n_i retrieved for query q_i and $R_i = \{r_{i,1}, r_{i,2}, \dots, r_{i,n_i}\}$ is the set of labels associated with the query-transcription pairs $(q_i, d_{i,j})$ where labels represent relevance grades. A feature vector $F_{i,j} = \{f^1_{i,j}, f^2_{i,j}, \dots, f^x_{i,j}\}$ of x features is extracted for each query-transcription document pair $(q_i, d_{i,j})$ with some features being query-dependent that they depend on the similarity and matching of

a document to the terms of the query. Other features are query-independent and can be extracted for each document ahead of query-time. A training set $QD_t = \{q_1:(D_1:F_{1,j},R_1), q_2:(D_2:F_{2,j},R_2), \dots, q_t:(D_t:F_{t,j},R_t)\}$ is formed from a subset of Q ; $Q_t = \{q_1, q_1, \dots, q_t\}$ and their corresponding documents features vectors and relevance labels. And a testing set $QD_T = \{q_1:(D_1:F_{1,j},R_1), q_2:(D_2:F_{2,j},R_2), \dots, q_T:(D_T:F_{T,j},R_T)\}$ is formed from a subset of the queries set $Q_T = \{q_1, q_2, \dots, q_T\}$ and their corresponding documents features vectors and relevance labels.

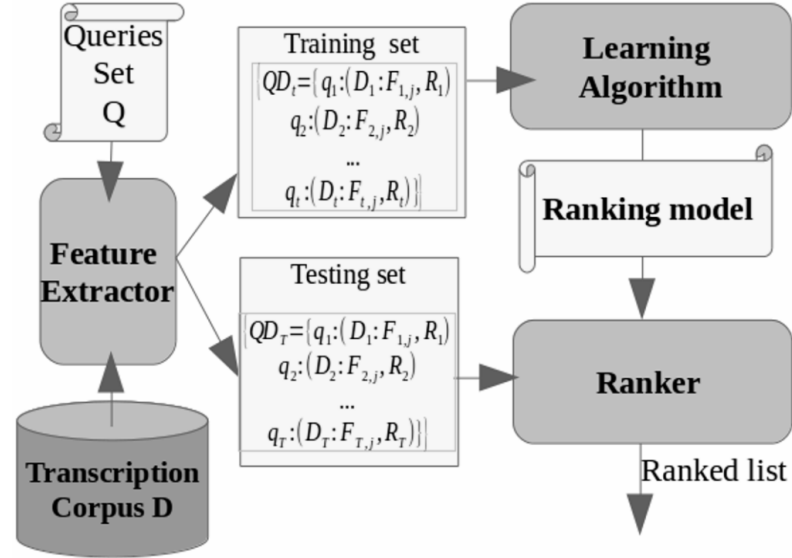


Figure 2. Learning to rank framework for spoken content retrieval

As shown in Figure 2, the feature extractor uses the query set Q to extract features from the transcriptions corpus D . The output of the feature extractor is the query-transcriptions pairs QD with the set of features F which is divided into QD_t used for training and QD_T used for testing. After labeling the query-documents pairs with their graded relevance labels R , the training set QD_t is fed to a learning to rank algorithm that produces a ranking model. This model is used by the ranker to get a ranked list for query-transcriptions pairs in the testing set.

2.1 Feature Extraction

One of the most important tasks of a learning to rank system is the selection of a feature set. For spoken documents, there are a set of meta-data fields used by video sharing websites like title, channel name, description, tags, uploaded time, views count, comments, likes and dislikes. We added features based on the timed transcription for spoken segments. A total of 75 features were extracted, extracted features can be classified as either query-dependent or query-independent features. Table 1 lists the extracted features together with their descriptions.

The query-independent features are based on the meta-data and measures of the importance and freshness of the audio; like uploaded time, views count, likes count, dislikes count, comments count, the stream length of the fields in the transcription file, the duration of the audio file, the number of segments for the transcription of speech. Query-independent features are formatted in *italics* in Table 1.

Query-dependent features are used for measuring the matching of the query terms with text in title, channel name, description, tags, transcribed text and with the whole text. We added a feature that represents the listening time for the first occurrence of a full covered query match to give preference for shorter listening time to first match. This is preferable for low-

bandwidth connections. These query-dependent features can be further classified into: query-terms-matching features, probabilistic-model features like BM25 and language modeling (LM) scores. Language modeling approaches in information retrieval try to estimate a language model for each document -in our case it is a transcription document together with its textual fields (title, description, channel name, tags)- and then rank documents according to the likelihood that the query at hand has been generated from the estimated language model. We also used language modeling smoothing methods (Jelinek Mercer, absolute discounting and using Dirichlet priors) as recommended in (Zhai and Lafferty 2001). The groups of sixes (6:11, 12:17 ...and 66:71) in Table 1 correspond to the features for each of the six textual fields in the transcription document (title, description, channel, tags, transcription segments and the whole document).

Table 1: Extracted features and their descriptions

ID	Feature	Description
1	<i>age of the document</i>	$\frac{currentTime - uploadedTime}{currentTime}$
2	<i>comments count</i>	Number of comments
3	<i>views count</i>	Number of views
4	<i>likes count</i>	Number of likes
5	<i>dislikes count</i>	Number of dislikes
6:11	covered query term number	Number of matching query terms in a field
12:17	covered query term ratio	$\frac{numberOfMatchingQueryTerms}{totalNumberOfQueryTerms}$
18:23	<i>fieldLength</i>	Length of each of the six field in terms of number of tokens
24:29	IDF (Inverse Document Frequency)	$\sum_{q_i \in q \cap d} \log \frac{N}{n_i}$
30:35	TF (Sum of Term Frequency)	$\sum_{q_i \in q \cap d} TF(q_i, d)$
36:41	TF-IDF	$\sum TF.IDF$
42:47	Boolean Model	whether query terms exists in the transcription field or not
48:53	BM25	$\sum_{i=1}^n IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{ D }{avgdl}\right)}$
54:59	LM with Jelinek Mercer Smoothing	$P_\lambda(w \vee d) = (1 - \lambda)P_{ml}(w \vee d) + \lambda P(w \vee C)$
60: 65	LM with Absolute Discounting	$P_s(w \vee d) = \frac{\max(c(w; d) - \delta, 0)}{\sum_{w^* \in V} c(w^*; d)} + \sigma P(w \vee C)$

66:71	LM with Dirichlet	$P_{\mu}(w \vee d) = \frac{c(w; d) + \mu P(w \vee C)}{\sum_{w^* \in V} c(w^*; d) + \mu}$
72	<i>nsegments</i>	Number of segments
73	<i>match_start</i>	Starting time for the first matching segment
74	<i>duration</i>	duration of the audio file
75	<i>relevant_segment_duration</i>	$\frac{\sum_{i=1}^{S_d} rel(s_i) * duration(s_i)}{\sum_{i=1}^{S_d} duration(s_i)}$

2.2 Dataset

The dataset used in this study is based upon the verse by verse Quran dataset (Verse by verse Quran dataset 2016). This dataset was chosen for two reasons; the first is the availability of timing information for verse segments in all chapters of the Quran. The second reason is the availability of text transcription for Quran verses. The corpus has been used to get the timings for each verse in each chapter in the Holy Quran. The corpus was then augmented to create more documents by forming documents of three consecutive verses, five consecutive verses and all verses in each chapter by combining verses' textgrids; the format used by Praat (Weenink & Boersma, 2016). We then used YouTube API with queries for each chapter to get realistic metadata for the documents. A timed transcription document in XML format is then generated as shown in Figure 3.

A total of 30,544 transcription documents were generated of different lengths with consecutive verses and corresponding youtube fetched metadata for each set of documents, with about 4% spam documents having fake titles, descriptions and tags that are relevant to some queries while their actual content is irrelevant. The transcription documents were then indexed by the open source Apache Solr. We added a modified root-based analyzer to be used by lucene analyzer in Solr during indexing and search since the light Arabic stemming algorithm (Larkey et al., 2007) used by Solr lead to low recall. For example, it doesn't retrieve transcriptions containing the word "وجوهكم" (your faces) when querying for "وجوه" (faces).

A set of 340 queries were built based on the knowledge of the domain under test "Quran files". A pooling strategy as adopted in TREC (Text Retrieval Evaluation Conference) has been applied using Solr with applying different similarity measures. Queries vary in their number of terms and generality, a sample set of queries is shown in Table 2 together with their English translation equivalents. We supplied the set of retrieved documents to our feature extractor to extract the 75 features as described before. The relevance of the retrieved documents to the query has been judged by giving one of 5 relevance labels from 0 to 4 as used in LETOR (Liu 2011). The LETOR formatted query-transcription pairs with their corresponding extracted features were then used to train learning to rank models using RankLib implementation (Dang, 2016) for the learning to rank algorithms; it has also been

used in (Chen et al., 2016). In addition, five-fold cross validation was used to overcome overfitting and to assess how the model will generalize to an independent unknown data set.

```
<transcription_doc xmax="46.0235" xmin="0.0">
  <title>114-سورة الناس, الشيخ محمد محمود الطبلأوي</title>
  <description>المصحف المجود</description>
  <tags>
    [[[سورة الناس, القرآن الكريم, الطبلأوي, المصحف المجود],
      Quran, Islam, Koran,
      مصر, الاسلام, سورة الناس, حفص عن عاصم, Sheikh, Al Tablawy]]]
  </tags>
  <channel>مصطفى بكير</channel>
  <uploaded_time>2006-09-27T08:17:02</uploaded_time>
  <duration>46.0235</duration>
  <raw_file_name>114_001_006.xml</raw_file_name>
  <views>3587</views>
  <likes>13</likes>
  <dislikes>0</dislikes>
  <comments>2</comments>
  <url>http://www.youtube.com/watch?v=RhGeA8cio_o</url>
  <tier name="segments">
    <trans xmax="7.923125" xmin="0.0">قُلْ أَعُوذُ بِرَبِّ النَّاسِ</trans>
    <trans xmax="12.633125" xmin="7.923125">مَلِكِ النَّاسِ</trans>
    <trans xmax="17.604375" xmin="12.633125">إِلَهِ النَّاسِ</trans>
    <trans xmax="27.042625" xmin="17.604375">رَبِّ الْوَسْوَاسِ الْخَاسِ</trans>
    <trans xmax="37.5256875" xmin="27.042625">الَّذِي يُوسِّسُ فِي صُدُورِ النَّاسِ</trans>
    <trans xmax="46.0235" xmin="37.5256875">مِنَ الْجَنَّةِ وَالنَّاسِ</trans>
  </tier>
</transcription_doc>
```

Figure 3. Sample transcription file.

Table 2: Sample queries from the query set.

Query	English equivalent
الطعام	The food
الأسواق	The markets
الأمثال	The examples
القرية	The village
الشمس و القمر	The sun and the moon
أولئك هم المفلحون	Those will be the successful
الجوع و الخوف	Hunger and fear
الذين هم عن صلاتهم ساهون	Who are neglectful of their prayers
السمع والأبصار والأفئدة	Hearing and vision and intellect
رحلة الشتاء و الصيف	The journey of winter and summer

2.3 Results

In the first experiment, we compared the results of well-known techniques that can be used for learning to rank (MART (Friedman, 2001), RankNet (Burges et al., 2005), RankBoost

(Freund et al., 2003), AdaRank (Xu et al., 2007), Coordinate Ascent (Donald & Croft, 2007), LambdaMART (Wu et al., 2007), ListNet (Cao et al., 2007) and Random Forests (Breiman 2001)) to the baseline BM25 scoring function.. We used the evaluation metrics used for graded relevance text retrieval like NDCG (Kalervo et al., 2002) and ERR (Chapelle et al., 2009).

Figure 4 shows that models built from learning to rank algorithms outperformed the unsupervised BM25 score ranking in terms of the evaluation measures used: NDCG@10, and ERR@10. The results recorded in Table 3 show that the Coordinate Ascent and tree-based algorithms (MART, LambdaMART, and Random Forests) perform the best for the NDCG@10 and ERR@10 measures.

Table 3: Evaluation measures results for learning to rank algorithms and unsupervised BM25.

Algorithm	NDCG@10	ERR@10
MART	0.6894	0.4536
RankNet	0.5081	0.3864
RankBoost	0.6242	0.4635
AdaRank	0.5486	0.405
Coordinate Ascent	0.7105	0.4758
LambdaMART	0.7082	0.4823
ListNet	0.6713	0.4493
Random Forests	0.6817	0.4489
BM25	0.5069	0.3846

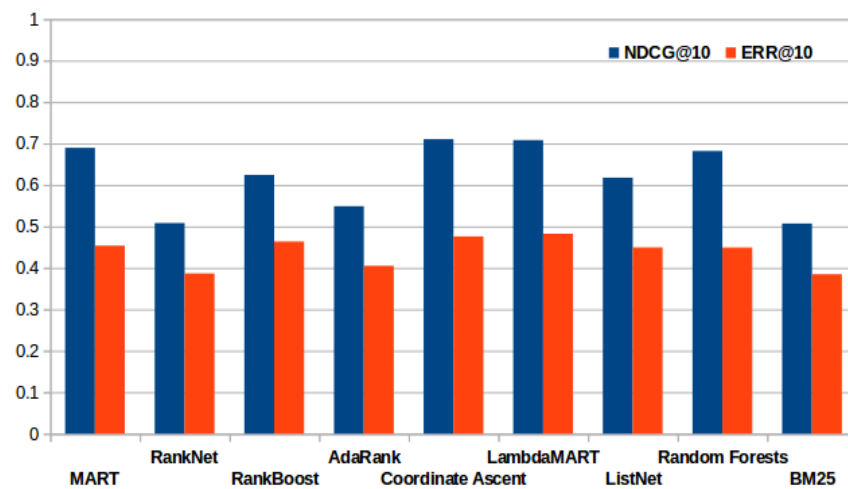


Figure 4. Comparison of evaluation measures for L2R algorithms and unsupervised BM25.

Another experiment was done to assess the performance gain for using the transcription features. The learning to rank algorithms were used to train models with feature vectors of length 50 for only the meta-data features (uploaded time, number of views, comments, likes and dislikes) and text similarity scores described before for the title, description, channel name and tags only. Figure 5 shows the improvement achieved in terms of NDCG@10 for using transcription related features over using the feature vectors of length 50 for training and testing for all algorithms under test. The minimum improvement is achieved in case of RankNet and the maximum is for Coordinate Ascent. Figure 6 shows the improvement achieved in terms of ERR@10 for using transcription related features over not using them for training models for the algorithms under test with a minimum improvement in case of using RankNet and maximum improvement in the case of LambdaMART algorithm.

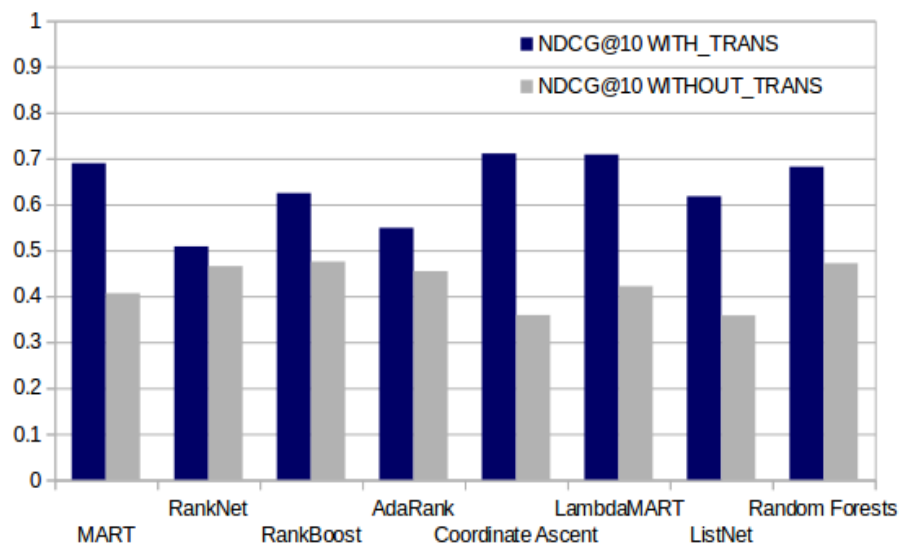


Figure 5. NDCG@10 performance improvement with using transcription features.

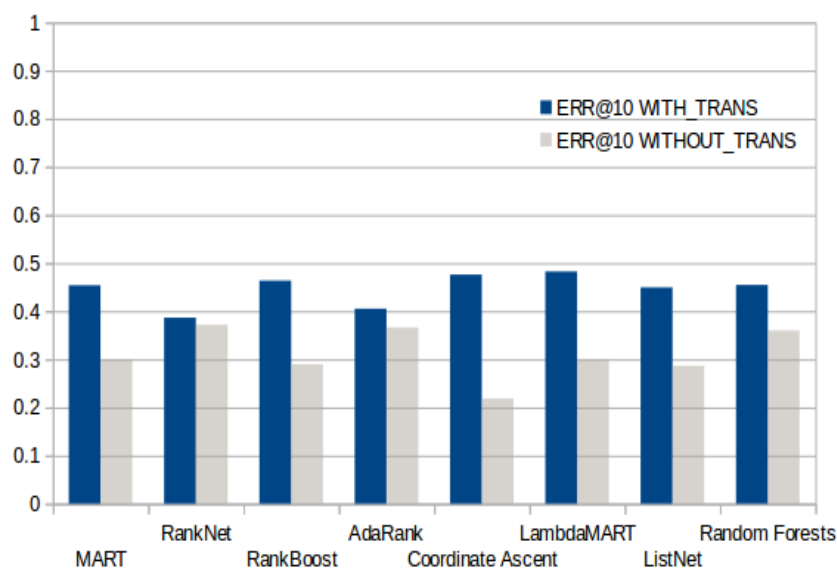


Figure 6. ERR@10 performance improvement with using transcription features.

3. Conclusion




In this work, we formulated the SCR problem as a learning to rank task, showed our proposed extracted features, and explored learning to rank algorithms to learn from these features. A comparison for the state of art learning to rank algorithms performance with respect to NDCG@10, and ERR@10 was done and compared to the baseline unsupervised BM25 score ranking. These comparisons revealed that learning to rank algorithms outperform the BM25 score ranking. In addition, experiments showed the effectiveness of using transcription-based features. This obviously comes at the cost of indexing more information and extracting more features for the retrieved documents for ranking improvement, however this is only performed once for model building. Coordinate Ascent and tree-based algorithms showed the best ranking accuracy in terms of NDCG@10 and ERR@10 for this type of problem. We will further investigate adding acoustic features like loudness, pitch and other frequency-domain features which can also be considered good candidates for exploration to rank spoken documents with better audio quality higher in the retrieved results.

References

- Breiman, L. 2001. Random Forests. *Machine Learning* 45: 5-32.
- Burges, C., T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. 2005. Learning to rank using gradient descent. In Proceedings of the 22nd International Conference on Machine Learning, 89-96. Bonn, Germany.
- Cao, Z., T. Qin, T. Y. Liu, M. F. Tsai, and H. Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In Proceedings of the 24th international conference on Machine learning, 129-136. Oregon, USA.
- Chapelle, O. and D. Metzler, Y. Zhang, and P. Grinspan. 2009. Expected Reciprocal Rank for Graded Relevance. In Proceedings of the 18th ACM conference on Information and knowledge management, 621-630. Hong Kong.
- Chen, J., C. Xiong, and J. Callan. 2016. An Empirical Study of Learning to Rank for Entity Search. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, 737-740. Pisa, Italy.
- Dang, V. The Lemur Project - RankLib. <https://sourceforge.net/p/lemur/wiki/RankLib/> (last accessed November 2018).
- Donald, M, and W. B. Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, June 2007.
- Freund, Y. and R. Iyer, R. Schapire, and Y. Singer. 2003. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research* 4: 933-969.
- Friedman, J. H. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29: 1189-1232.
- Kalervo, J., and J. Kekalainen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20: 422-446.
- Larkey, L. S., L. Ballesteros, and M.E.Connell. 2007. Light Stemming for Arabic Information Retrieval. In Arabic Computational Morphology: Knowledge-based and Empirical Methods, ed. by Abdelhadi Soudi, Antal van den Bosch and Günter Neumann, 221-243. Springer Netherlands.
- Larson, M., and G. J. F. Jones. 2011. Spoken Content Retrieval: A Survey of Techniques and Technologies. *Foundations and Trends in Information Retrieval* 5: 235-422.
- Liu, T. Y. 2011. *Learning to Rank for Information Retrieval*. Springer-Verlag Berlin Heidelberg.
- Ma, M. 2015. Spoken Keyword Rescoring and Document Retrieval for Low-resource Languages. In Proceedings of the 16th Annual Conference of the International Speech Communication Association. Dresden, Germany.

- Rybach, D. and C. Gollan, R. Schluter, and H. Ney. 2009. Audio segmentation for speech recognition using segment features. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Taipei, Taiwan.
- Verse by verse Quran dataset. <http://www.everyayah.com> (last accessed November 2018)
- Weenink, D., P. Boersma. Praat: doing phonetics by computer.
- Wu, Q., C.J.C. Burges, K. Svore, and J. Gao. 2007. Adapting Boosting for Information Retrieval Measures. *Journal of Information Retrieval* 13: 254 - 270.
- Xu, J., and H. Li. 2007. AdaRank: a boosting algorithm for information retrieval. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval: 391-398. Amsterdam, The Netherlands.
- Zhai, C., and J. Lafferty. 2001. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 334-342. New York, USA: ACM Press.
- Zhao, X., X. Li, and Z. Zhang. 2015. Multimedia Retrieval via Deep Learning to Rank. *IEEE Signal Processing Letters* 22: 1487 - 1491.

Biodata

	<p>Farida is a Lecturer in Computer Engineering from Cairo University. She finished her PhD. (2018) and M.Sc. (2009) in Computer Engineering from the same university. Her main research interest is machine intelligence and its application.</p>
	<p>Mayada is an Assistant Professor in Computer Engineering from Cairo University. She obtained PhD. and M.Sc. in Computer Engineering from the same university. Her main research interests are computer vision, machine intelligence applications.</p>
	<p>Emeratus is a Professor at Faculty of Engineering, Cairo University. She was graduated from, Electronics and Communication Engineering Department, Faculty of Engineering, Cairo University, Egypt. She finished her M.Sc. (1978) from McGill University, Canada. She got her Ph.D. (1983) from Cairo University, Egypt. She has many publications in the field of Machine Intelligence and Data Mining.</p>