NiCT at TREC 2009: Employing Three Models for Entity Ranking Track

Youzheng Wu, Hideki Kashioka
Spoken Language Communication Group, MASTAR Project
National Institute of Information and Communications Technology (NiCT)
2-2-2 Hikaridai, Keihanna Science City, Kyoto 619-0288, Japan
{youzheng.wu, hideki.kashioka}@nict.go.jp

Abstract

This paper describes experiments carried out at NiCT for the TREC 2009 Entity Ranking track. Our main study is to develop an effective approach to rank entities via measuring the "similarities" between supporting snippets of entities and input query. Three models are implemented to this end. 1) The DLM regards entity ranking as a task of calculating the probabilities of generating input query given supporting snippets of entities via language model. 2) The RSVM ranks entities via a supervised Ranking SVM. 3) The CSVM, an unsupervised model, ranks entities according to the probabilities of input query belonging to topics represented by entities and their supporting snippets via SVM classifier. The evaluation shows that the DLM is the best on P@10, while the RSVM outperforms the others on nDCG.

1 Introduction

The first year of the TREC 2009 Entity Ranking track aims to investigate the problem of related entity finding, which is defined as follows:

> *Given an input entity, by its name and homepage, the type of the target entity, as well as the nature of their relation, described in free text, find related entities that are of target type, standing in the required relation to the input entity.*

About the detail of the task, please refer to the overview paper of the track. An example of the input entities is shown in Figure 1. For convenience of the writing, we rename *input entity* to *input query* labeled as Q, use Q_t to denote the *entity_name* and Q_n to denote the narrative field.

Inspired by the approaches used in TREC Expert Search track (in that person names are required to return, http://www.ins.cwi.nl/projects/trec-ent/wiki/index.php/Main_Page), we regard entity ranking as a task of calculating the "similarities" between input query and supporting snippets of entities. In this guiding idea, our study mainly focuses on investigating how effectively using supporting snippets of entities to rank them. To this end, three models are employed in this year's participation.

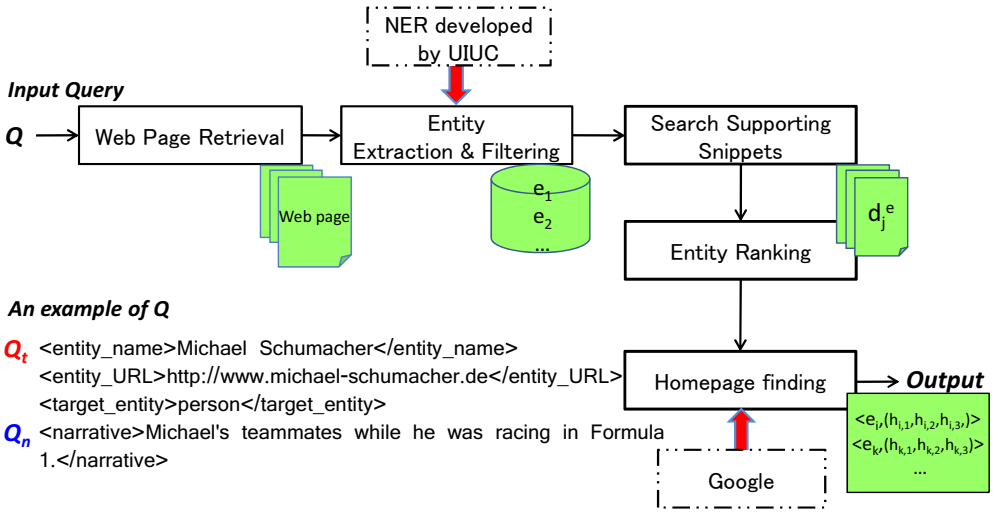

Figure 1: Architecture of Our Entity Ranking System

2 System Architecture

The architecture of NiCT's participant system, demonstrated in Figure 1, is a cascade of the following five components.

☞ The *Web Page Retrieval* extracts keywords from Q_t and Q_n to retrieve some related Web pages or documents. We compare two retrieval strategies: INDRI search engine (`http://www.lemurproject.org/`) retrieving documents from the ClueWeb09_English_1 corpus and Google search engine retrieving web pages from the Internet.

☞ The *Entity Extraction & Filtering* extracts the related entities from the retrieved pages that match the type of the target entity. The extraction is supported by a named entity recognition tool developed by the Cognitive Computation Group at UIUC (`http://l2r.cs.uiuc.edu/~cogcomp`). For example, phrases/words tagged with *PER*, *ORG* and *MISC* are extracted when target entities are person, organization, and product, respectively.

To filter out noises in the extracted entities, we rank the entities according to the scores $\pi_3(e)$ calculated by,

$$\pi_3(e) = \begin{cases} 2 & if \text{ the } W_e \text{ has a hyperlink to and a hyperlink from the } W_{Q_t}, \\ 1 & else\ if \text{ the } W_e \text{ has either a hyperlink to or a hyperlink from the } W_{Q_t}, \\ 0.5 & else\ if \text{ the } W_e \text{ has a hyperlink to or a hyperlink from the } W_x \text{ that contains some words of } Q_t, \\ 0 & otherwise \end{cases} \quad (1)$$

where, W_e and W_{Q_t} denote the Wikipedia page of entity e and Q_t, respectively. W_x denotes any Wikipedia page.

At last, we select some of the extracted entities as the input of the following components using the criterion: If the number of the entities which scores are larger than 0 is less than 100, the top 100 entities are selected; otherwise, all of the entities which scores are larger than 0 are selected. To simplify the writing, we use **e** and e (or e_i) to represent the

set of the related entities, and one of entities in **e**, respectively.

☞ For each entity e, the *Search Supporting Snippets* creates a query by combining entity e and the keywords from Q_t and Q_n, submits the query to a search engine, and retains the snippets returned by search engine as the supporting snippets of entities e. Similarly, we compare the supporting snippets retrieved by INDRI from the ClueWeb09_English_1 corpus and that retrieved by Google from the Web.

☞ The *Entity Ranking* is the kernel of the system, which ranks related entities by calculating "similarities" between input query Q and supporting snippets of related entities. In our participation, we employ three models, i.e., Document Language Model (abbreviated to DLM, as described in Section 3), Supervised Ranking Support Vector Machine Model (abbreviated to RSVM, as described in Section 4), and Unsupervised Classification SVM (abbreviated to CSVM, as described in Section 5).

☞ The *Homepage Finding* first submits the entity e_i to Google, and then the first three pages that can be found in the ClueWeb09_English_1 corpus are regarded as its homepages $h_{(i,1)}, h_{(i,2)}, h_{(i,3)}$. Note that we have no module of identifying homepages for entities currently, therefore, Google is applied. In future work, we will work on it.

3 DLM

The DLM regards the TREC 2009 Entity Ranking track as a problem of estimating the probability $p(e|Q)$ of generating a related entity e given input query Q. In implementation, we estimate this probability by using supporting snippets of entity e to connect e and input query Q, which is expressed using,

$$p(e|Q) = \sum_{d^e} p(e, d^e|Q) \quad (2)$$

$$= \sum_{d^e} p(d^e|Q) * p(e|d^e, Q) = \sum_{d^e} p(d^e|Q) * p(e|d^e) \quad (3)$$

$$\approx \sum_{d^e} p(Q|d^e) * p(e|d^e) \quad (4)$$

where, d^e is a supporting snippet of entity e, $p(Q|d^e)$ denotes the probability that input query is generated by a supporting snippet, $p(e|d^e)$ allows us to model the probability that a supporting snippet mentions entity e. Both $p(Q|d^e)$ and $p(e|d^e)$ can be estimated by any state-of-the-art IR formulas. Note that Equation (3) is obtained by assuming that Q and e are independent given supporting snippet d^e, Equation (4) is obtained by assuming that probability $p(d^e)$ is uniform.

Actually, the above DLM has been widely used in the TREC expert search tracks [1]. However, the independence between Q and e is a very strong assumption, which ignores the relationship between Q and e.

Inspired by the proximity measure for IR [2] and the Wikipedia link information for the INEX Entity Ranking task [3], we incorporate the proximity measure and the Wikipedia link information among entities into the above DLM. Our proposed DLM can be expressed by Equation (5).

$$p(e|Q) \propto \pi_3(e) \times \sum_{d^e} p(d^e|Q) \times p(e|d^e, Q) \quad (5)$$

where, π_3 means the Wikipedia link information, which is calculated using Equation (1), $p(e|d^e, Q)$ is calculated by,

$$p(e|d^e, Q) = p(e|d^e) + \pi_1(e, Q_t; d^e) + \pi_2(e, Q_n; d^e) \quad (6)$$

feature	value	description
MATCH	$\sum_{d^e} overlap(d^e, Q)$	Word overlap between supporting snippets and input query
MISMATCH	$\sum_{d^e} mismatch(d^e, Q)$	Word mismatch
COS	$\sum_{d^e} cosine(Q, d^e)$	Cosine similarity
DIST1	$\sum_{d^e} \pi_1(e, Q_t; d^e)$	Proximity similarity between e and *entity name*
DIST2	$\sum_{d^e} \pi_2(e, Q_n; d^e)$	Proximity similarity between e and *narrative*
FREQ	$cnt(d^e)/\sum_e cnt(d^e)$	Normalized frequency
ILINK	Whether the W_e has or does not have an income link from W_{Q_t}	
OLINK	Whether the W_e has or does not have an outcome link from W_{Q_t}	

Table 1: Features used in the RSVM.

where, π_1 and π_2 denote the proximity-based similarity between e and Q_t in the supporting snippet d^e, and the proximity-based similarity between e and Q_n in d^e, respectively. π_1 and π_2 are calculated using Equation (6) and (7).

$$\pi_1(e, Q_t; d^e) = log(\varphi + e^{-\delta(e,Q_t;d^e)}) \quad (7)$$

$$\pi_2(e, Q_n; d^e) = log(\varphi + e^{-\delta(e,Q_n;d^e)}) \quad (8)$$

where, φ is a parameter to allow for certain variations, $\delta(e, Q_t; d^e)$ (or $\delta(e, Q_n; d^e)$) is minimum distance between e and Q_t (or Q_n) in d^e, which is defined as the smallest distance value of all pairs of unique matched words. For example,

$$\delta(e, Q_t; d^e) = min_{q \in Q_t \cap d^e} Dis(e, q; d^e) \quad (9)$$

where, $Dis(e, q; d^e)$ is the minimum number of words between entity e and keyword q of Q_t in d^e. The minimum distance is used because [2] proved that the minimum outperformed the maximum and average distances.

In summary, the main idea of Equation (6) lies in: small distance between entity e and Q_t, as well as small distance between e and Q_n, imply their strong semantic relation, thus we reward cases where they are really close to each other, the distance contribution becomes nearly constant as the distance becomes larger.

4 RSVM

Learning to rank is a new area in statistical learning, in parallel with learning for classification, regression, etc. Ranking SVM, a hot research topic in IR [4], is a typical method of learning to rank, which is different from SVM in terms that the training data in ranking is relative ordering or partial orders.

Our RSVM is concerned with applying Ranking SVM for the TREC Entity Ranking task. About the theory of the Ranking SVM, please refer to [4]. The features of entities used in the RSVM are extracted from their corresponding supporting snippets, as shown in Table 1.

In Table 1, $overlap(d^e, Q)$ and $mismatch(d^e, Q)$ are calculated by Equation (10) and (11), respectively. $cnt(d^e)$ is the number of supporting snippets of entity e.

$$overlap(d^e, Q) = \frac{\sum_{q \in Q} \delta(q, d^e)}{|Q|} \quad (10)$$

Related entity	Sample of the supporting snippets
Rubens Barrichello	Michael Schumacher - Wikipedia, the free encyclopedia - In 2007, in recognition of his contribution to Formula One racing, ... At the 2002 Austrian Grand Prix, Schumacher's teammate, Rubens Barrichello, ...
	Rubens Barrichello - Wikipedia, the free encyclopedia ... Barrichello drove for Ferrari from 2000 to 2005, as Michael Schumacher's teammate, ... In the 2006 Formula One season, his new teammate Jenson Button gave Barrichello the ...
	Rubens Barrichello Profile - Honda Formula 1 Driver Rubens Barrichello Photo (c) Honda Racing F1 Team ... Joining Ferrari as Michael Schumacher's teammate in 2000, he finally had a car capable of winning. ...
	Rubens Barrichello Memorabilia With 11 victories and a podium finish in every race Michael Schumacher and Ferrari ... Rubens Barrichello Formula 1 Motor Racing Print - Sport ... He regularly outpaced his more experienced teammates. ...
	Rubens Barrichello — Formula One Drivers — All Time — F1 Pulse Compare Rubens Barrichello's performance in F1 to other drivers and get all the ... early stages of his racing career before taking a step towards Formula One, ... classifying second behind race winner and teammate Michael Schumacher. ...
Jacques Villeneuve	Michael Schumacher and Jacques Villeneuve vied for the title in 1997. ..... In 2007, in recognition of his contribution to Formula One racing, the Nrburgring racing track .... At the 2002 Austrian Grand Prix, Schumacher's teammate, ...
	5.1 Racing record; 5.2 Complete Champ Car results; 5.3 Complete Formula One results.... Jacques Villeneuve driving for the Williams Formula One team at the 1996... despite coming under pressure from the Ferrari of Michael Schumacher. ... Button would prove to become the second of Villeneuve's teammates to ...
	jacques villeneuve $\ll$ F1 Blog A website by people with an incurable obsession with Formula One Racing ... Rubens Barrichello - if he hadn't become Michael Schumacher's teammate, ...
	Michael Schumacher was soon making a name for himself and in 1984 he won the ... when Jordan's Formula One driver Bertrand Gachot found himself in jail and Schumacher ... where he qualified 7th ahead of his more experienced teammate. ... poor start to 1997 Schumacher clawed back Jacques Villeneuve's advantage until ...
	Jacques Villeneuve BMW Sauber formula 1 profile and photo gallery ... +Michael Schumacher F1 +Michael Schumacher ... He moved swiftly to Indy Car racing, and was Rookie of the Year in 1994. ... almost winning his first race, after qualifying in pole, but teammate Damon Hill took the victory. ...

Table 2: A sample of the supporting snippets.

$$mismatch(d^e, Q) \quad = \quad \frac{\sum_{q\in} 1 - \delta(q, d^e)}{|Q|} \tag{11}$$

In implementation, the development data is used to train an RSVM. The ranking SVM tool is provided by `http://svmlight.joachims.org/`.

5 CSVM

Either the DLM or the RSVM is trying to measure the "simlarity" between the input query Q and a related entity e via the supporting snippets of e. Their difference lies in the approaches of using supporting snippets. In this section, we present a novel algorithm of using supporting snippets and illustrate the proposed algorithm named to the CSVM with an example.

Suppose that we are asked to rank two related entities, i.e., *Rubens Barrichello*, *Jacques Villeneuve*, to the input query Q shown in Figure 1. Table 2 shows a sample of the supporting snippets for each entity.

From Table 2, we find that most of the supporting snippets of entity *Rubens Barrichello* express the meaning of *Rubens Barrichello is Michael Schumacher's teammate*, while most of the supporting snippets of entity *Jacques Villeneuve* roughly include the meaning of *Jacques Villeneuve is Michael Schumacher's competitor*. Therefore, it is reasonable to assume that each entity together with its supporting snippets consist of a topic. Entities represent the topic signatures, while the supporting snippets are regarded as the instances of the topics.

Consequently, the CSVM regards the entity ranking task as a kind of classification task, which can be formalized by,

- ✍ Using topics represented by entities and their supporting snippets as training instances to train an SVM classifier; For the example in Table 2, the topic represented by entity *Jacques Villeneuve* and the topic represented by entity *Rubens Barrichello* have 93 and 96 training instances, respectively.

- ✍ Using the trained SVM classifier to estimate the probabilities of input query Q belonging to the topics. For the same example, the probabilities of the input query belonging to the topic represented by *Jacques Villeneuve* and the topic represented by *Rubens Barrichello* are 0.427384 and 0.572616, respectively.

- ✍ Outputting the entities according to the probabilities in descending order. For the example, the output is,

$$< \textit{Rubens Barrichello}, 0.572616 >$$
$$< \textit{Jacques Villeneuve}, 0.427384 >$$

In summary, the schematic diagram of the *Entity Ranking* module in the CSVM is shown in Figure 2.

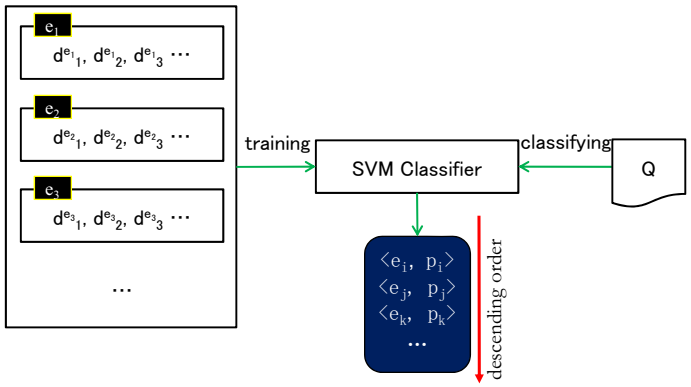

Figure 2: Schematic Diagram of the *Entity Ranking*

In implementation, the LIBSVM tool (`http://www.csie.ntu.edu.tw/~cjlin/libsvm`) is employed. Usually, SVM just predicts class label but not probability information. To extend SVM for probability estimates, the approach proposed in [5] is adopted in the LIBSVM.

The MATCH, MISMATCH, COS, DIST1, DIST2, ILINK, OLINK features in the RSVM are also used in the CSVM. However, the values of the first five features in the CSVM are $overlap(d^e, Q)$, $mismatch(d^e, Q)$, $cosine(Q, d^e)$, $\pi_1(e, Qt; d^e)$, and $\pi_2(e, Qn; d^e)$, respectively. Note that the values of these features are different from those in the RSVM because the values in the RSVM are sum of these values. The values of the ILINK and

OLINK features are 1 or 0 that are same as the RSVM. Besides these features, each word q in input query Q is also extracted as classification features, which values are set to $TF(q) \times IDF(q)$.

Therefore, the number of the features is $|Q| + 7$, $|Q|$ denotes the number of unique words in the input query Q.

6 Comparison

The common ground among the three models lies in: ranking entities via measuring "similarity" between supporting snippets of entities and input query. Table 3 compares their differences.

	Model of measuring similarity	Idea of Ranking Entities via	Training data	Speed
DLM	Language Model	Estimating probability of generating input query given related entity connecting by supporting snippets	Not required	Fast
RSVM	Ranking SVM	A typical of learning to rank formalized as a problem of binary classification on instance pairs, and then to solve the problem using SVM	Required	Fast
CSVM	Classification SVM	Estimating probability of input query belonging to topics represented by related entities and their supporting snippets	Not required	Relatively Slow

Table 3: Differences among the Models.

7 Experiments

This section lists our submitted runs and the configuration used for each, and reports on the results of our submissions. The metrics used for measuring performance are nDCG and P@10.

7.1 Submitted Runs

Four runs are submitted for the TREC official evaluation.

- RUN-1, RUN-2 and RUN-3: are the CSVM, the DLM and the RSVM, respectively. They use the same configurations: Google is used in the *Web Page Retrieval* and *Search Supporting Snippets* components.
- RUN-4: is the DLM that uses INDRI search engine in the *Web Page Retrieval* and *Search Supporting Snippets* components.

7.2 Finding Supporting ClueWeb09_English_1 Documents

The RUN-1, RUN-2 and RUN-3 just use the Web as their source of information. However, the TREC requires us to return not only answers but also supporting documents of answers

from the ClueWeb09_English_1 corpus. Hence, we have to map answers found on the Web to a ClueWeb09_English_1 document. To realize this, the following two steps are conducted for each answer.

- Creating a query that consists keywords from the answer and the input query.
- Employing INDRI engine to search the ClueWeb09_English_1 corpus, the first three documents that contain the exact answer and Q_t are retained as the supporting documents.

7.3 Results

Table 4 lists the nDCG scores, *Best* and *Median* mean the best and the median scores among all participants' systems, respectively, and p value implies significant level of the RUN-1 against the RUN-4 obtained through a two-tailed paired t-test.

Topic	RUN-1	RUN-2	RUN-3	RUN-4	*Best*	*Median*
1	.1349	.1398	.1592	.0576	.2992	.0597
2	.308	.2723	.3079	.0326	.4262	.1012
3	0	0	0	0	.6388	0
4	.2336	.2417	.25	.0417	.2982	.0417
5	.1022	.0645	.0645	.1457	.3697	.1119
6	.1792	.1357	.1527	.2138	.2844	.1168
7	.288	.2955	.2943	.2722	.2955	.0661
8	.0216	.0352	.0279	.0279	.4838	.0559
9	.1742	.1569	.1674	.1428	.3728	.1602
10	.253	.2356	.2518	.1328	.4596	.0598
11	.1898	.1898	.1898	.0572	.3668	.0499
12	.3207	.3417	.3663	.2197	.3663	.0469
13	.0884	.0884	.0884	.0884	.2815	.0884
14	.217	.4355	.404	.185	.6842	.0772
15	.3402	.3096	.3097	.2794	.5796	.0714
16	.0479	.0494	.0479	.0559	.4319	0
17	.233	.2425	.2373	.2284	.3379	.0816
18	.1987	.2002	.1987	.1323	.4312	.1414
19	.2081	.1824	.1669	.0559	.3647	0
20	.1225	.1076	.1288	.1911	.4243	.1725
$ave.$	$.183_{p=(0.01)}$	.1862	.1907	.128	-	-

Table 4: nDCG scores

This table indicates that: 1). In terms of nDCG measurement, the ranking of the implemented models is: the RSVM (RUN-3) $>$ the DLM (RUN-2) $>$ the CSVM (RUN-1). However, the improvements among them are not statistically significant. 2). The improvement of the RUN-1 over the RUN-4 is significant, which means that the *Web Page Retrieval* and the *Search Supporting Snippets* modules play very improvement roles in overall performance. 3). The differences of our runs against the *Best* and *Median* are statistically significant.

The P@10 scores of the runs are reported in Table 5. This table shows that the DLM is slightly better than the others in terms of P@10 measurement. Similarly, the differences are not significant.

Table 5: P@10 scores

	RUN-1	RUN-2	RUN-3	RUN-4	Best	Median
1	0.2	0.2	0.2	0.1	0.2	0
2	0.1	0.1	0.1	0	0.1	0
3	0	0	0	0	0.1	0
4	0.1	0.1	0.1	0	0.3	0
5	0.1	0.1	0.1	0.1	0.4	0.1
6	0.1	0.1	0.2	0.1	0.2	0
7	0.5	0.8	0.5	0.3	0.8	0
8	0	0	0	0	0.5	0
9	0.1	0.1	0.1	0.1	0.2	0
10	0.5	0.5	0.5	0.1	0.8	0
11	0.1	0.1	0.1	0.1	0.3	0
12	0.2	0.2	0.2	0.2	0.3	0
13	0	0	0	0	0.1	0
14	0.1	0.3	0.3	0	0.4	0
15	0.3	0.2	0.1	0.2	0.6	0
16	0	0	0	0	0.6	0
17	0.3	0.5	0.4	0.4	0.5	0
18	0	0	0	0	0.1	0
19	0.1	0.1	0.1	0	0.2	0
20	0.1	0.1	0.1	0.2	0.3	0
$ave.$	0.145	0.175	0.155	0.095	-	-

8 Conclusion

In this paper, we describe NiCT's participant system for the first year of TREC Entity Ranking track. Given entities and their supporting snippets, we mainly focus on developing an effective framework to model entity ranking task. The official evaluation results indicate that our implemented ranking approaches just achieve the above average performance. We must point out that: 1). The experiments are conducted on a small set of testing data, specifically, 20 test queries. 2). Direct comparison among the ranking methods (like the comparison among the RUN-1, RUN-2, and RUN-3) may be better than the comparison among systems (like the comparison between our runs and the *Best*). This is because the *Entity Extraction & Filtering* modules used in systems to extract entities are different, which play very important roles. In future study, we aim at: improving recall of entity extraction in the *Entity Extraction & Filtering*; improving precision of entities that match types of target entities in the *Entity Extraction & Filtering*; and entity homepage finding in the *Homepage Finding*.

References

[1] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. Formal Models for Expert Finding in Enterprise Corpora*In Proc. of SIGIR-2006*, Washington, USA, 2006.

[2] Tao Tao, and ChengXiang Zhai. An Exploration of Proximity Measures in Information Retrieval. *In Proc. of SIGIR-2007*, Amsterdam, The Netherlands, 2007.

[3] Jovan Pehcevski, Anne-Marie Vercoustre, and James A. Thom. Exploiting Locality of Wikipedia Links in Entity Ranking. *In Proc. of ECIR-2008*.

[4] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, et al. Adapting Ranking SVM to Document Retrieval. *In Proc. of SIGIR-2006*.

[5] Ting-Fan Wu, Chih-Jen Lin, Ruby C. Weng. Probability Estimates for Multi-class Classification by Pairwise Coupling. *In Journal of Machine Learning Research 5 (2004) 975-1005*.