YJST at the NTCIR-12 MobileClick-2 Task

Yuya Ozawa
Yahoo Japan Corporation
yozawa@yahoo-corp.jp

Taichi Yatsuka*
Yahoo Japan Corporation
tyatsuka@yahoo-corp.jp

Sumio Fujita
Yahoo Japan Corporation
sufujita@yahoo-corp.jp

ABSTRACT

Yahoo Japan Search Technology(YJST) team participated in the Japanese iUnit Ranking and Summarization subtasks of NTCIR-12 MobileClick-2. For the iUnit Ranking subtask, we adopted LM-based approach, which is implemented on the basis of organizers' baseline system. We examined language model based iUnit ranking using both KL-divergence and negative cross entropy with several model smoothing methods such as Bayesian smoothing with Dirichlet priors which commonly used in the document ranking in language modeling IR, or comparatively new Pitman-Yor process smoothing. Our system achieved 0.807 as Q-measure against the Japanese ranking test set. For the iUnit Summarization task, we used the organizer's LM-based two-layer iUnit summarization baseline system but the ranking module is replaced by aforementioned our extended system. Due to word based matching, the baseline intent identification for the second layer allocation fails to identify any intent when no common word is found between iUnit and Intent. We introduced context based word embedding representation of both iUnit and Intent to identify the intent of iUnits which do not contain any explicit intent words. Finally our system achieved 25.8498 in M-measure against the Japanese summarization test set.

Team Name

YJST

Subtasks

MobileClick iUnit Summarization (Japanese)

Keywords

Language model, Dirichlet prior smoothing, Word embedding

1. INTRODUCTION

Recently, due to the wide spreading mobile devices throughout our daily lives, user interfaces of our web search services should be optimized to the mobile environments, where completely different presentation strategies are required given limited display space of the devices. As part of such efforts which hopefully lead us to completely different user experiences on mobile searches, we participated in the MoblieClick-2 task[4], where we engaged in two subtasks, namely iUnit Ranking and Summarization Subtasks in Japanese.

Our motivations of participating in the subtasks are in two tiers, for the first tier, we intended to verify the effectiveness of the language model based text matching techniques known to be quite effective to ad hoc text search situations[9] [1], but not yet intensively evaluated on a context of text summarization applications. The second tier research question is that such text matching effectiveness can be improved by applying semantic dimension reduction approaches including distributional word representation, i.e. word embedding representations of words.

For the iUnit Ranking subtask, we adopted LM-based approach, which is implemented on the basis of organizers' baseline system. We examined language model based iUnit ranking using both KL-divergence and negative cross entropy with several model smoothing methods such as Bayesian smoothing with Dirichlet priors which commonly used in the document ranking in language modeling IR, or comparatively new Pitman-Yor process smoothing.

For the iUnit Summarization task, we used the organizer's LM-based two-layer iUnit summarization baseline system but the ranking module is replaced by aforementioned our extended system. Due to word based matching, the baseline intent identification for the second layer allocation fails to identify any intent when no common word is found between iUnit and Intent. We introduced context based word embedding representation of both iUnit and Intent to identify the intent of iUnits which do not contain any explicit intent words.

The remainder of this paper is organized ad follows. Section 2 explains related work. Section 3 and 4 describe our approaches to iUnit ranking and summarization subtasks respectively. In Section 5, we explain our evaluation experiments and discuss the results. Section 6 presents our future plans and Section 7 concludes the work.

2. RELATED WORK

The overview of the NTCIR-12 MobileClick-2 task is described in [4], where we used NTCIR-10 1CLICK-2 data for training purpose[3].

There are various approaches of Language Modeling in information retrieval. Inspired by emerging studies in speech recognition community, the language models are the techniques to model a probabilistic distribution that captures statistical regularities of language generation, specifically to predict the next word given previous word sequences. In document retrieval, they treat information items such as

*Both the first and the second authors play the equally important role in this work.

documents as a probabilistic language model of their constituent indexing terms without sequence information. Then the documents are ranked according to the likelihood of generating query based on each document model[8]. We adopted a smoothing method in the literature which commonly used in state-of-the-art information retrieval studies such as Dirichlet priors smoothing[9]. Pitman-Yor smoothing is an extension of Dirichlet priors smoothing; where the absolute discounting method is combined on top of Dirichlet priors[7].

In the iUnit summarization subtask, a word embedding representation of intent text is used in view of intent matching. Mikolov et al.[6] proposed two novel model architectures for computing continuous vector representations of words from very large data sets, namely continuous bag-of-words model and continuous skip-gram model. They evaluated the quality of these representations in a word similarity task and they reported significant improvements in accuracy at much lower computational cost over the state-of-the-art techniques. From such work, they claims that it is possible to train high quality word vectors using very simple model architectures.[1]

3. IUNIT RANKING SUBTASK

In the Ranking subtask, we adopted LM-based approach, where the score of each iUnit against the given query is calculated as the probability of generating iUnit text given the language model of the query text.

Each query has a set of relevant documents, $D = \{d_1, d_2, .., d_n\}$. Each document is represented by the set of words appearing in it, $W = \{w_1, w_2, ..., w_m\}$. Our language models consist of the probability of generating a word given the query model, $P(w|q)$ and of a word given the background model, $P(w|o)$, where parameters are estimated based on the word count in the text of query relevant documents for $P(w|q)$ and non relevant documents for $P(w|o)$, using the title and body fields of provided index data.

The probability of generating each iUnit text is computed based on the probability of each word appearing in the iUnit.

3.1 Baseline Language Modeling Method

This is the baseline language model based iUnit ranking provided by the organizers. query language model $P(w|q)$ is estimated as follows:

$$P(w|q) = \frac{N_{D_q,w}}{N_{D_q}} \tag{1}$$

where $N_{D_q,w}$ is the count of word w in the set of query relevant documents D_q and N_{D_q} is the count of all word positions in documents D_q, set of documents relevant to query q.

Then the score of each iUnit is computed as the summation of the Log Odds Ratio of two language models through each word in the iUnit:

$$score(u, q) = \sum_{w \in W_u} \ln \frac{P(w|q)}{P(w|o)} \tag{2}$$

$P(w|o)$ is a background language model, where D_o, set of non relevant documents is used instead of D_q.

[1]The implementation of their method is publicly available from *Google Code*[5].

3.2 Baseline Vector Space Methods

We tried another baseline approach using a vectorial bag of word representations of documents.

Both iUnit and query, of which the surrogate is a set of relevant documents as aforementioned language modeling baseline, are represented by the bag of words representation of word terms. Each element of vectors is weighted either by boolean variables indicating the appearance of the word or by term frequencies. Thus iUnits are ranked according to the cosine similarity between vectorial representations of iUnit and the query.

$$sim(u, q) = \cos(\vec{u}, \vec{q}) = \frac{\vec{u} \cdot \vec{q}}{|\vec{u}| \cdot |\vec{q}|} \tag{3}$$

As this ranking function does not take word discriminative feature such as IDF into consideration, we introduced a *background discount* into the similarity measure by subtracting a background similarity $\cos(\vec{u}, \vec{o})$.

$$sim(u, q) = \cos(\vec{u}, \vec{q}) - \cos(\vec{u}, \vec{o}) \tag{4}$$

3.3 Dirichlet Prior Smoothing Methods

We used the Dirichlet prior smoothing method as an extension to aforementioned baseline language modeling approach, which was successfully applied to ad hoc document search tasks. As in the baseline language model method, estimated query language models are used to predict the probability of generating iUnit texts. Unlike the baseline method, the background language model $P(w|o)$ is used implicitly for smoothing the query language model. Figure 1 illustrates the process of our approach.

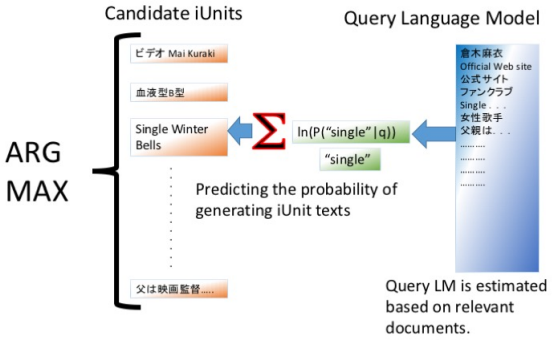

Figure 1: Our Language Model Approach for iUnit Ranking

3.3.1 *Uni-gram Dirichlet prior smoothing*

In this approach we calculate $P(w|q)$ as follows:

$$P(w|q) = \frac{N_{D_q,w} + \mu P(w|o)}{N_{D_q} + \mu} \tag{5}$$

where μ is the hyper parameter for adjusting smoothing impact. The iUnit score against the given query q is calculated as:

$$score(u, q) = \sum_{w \in W_u} \ln P(w|q) \tag{6}$$

Normalized by the iUnit length, this is essentially a negative cross entropy of iUnit and query language models, which is equivalent to KL-Divergence when ranking iUnit. Since the iUnit length normalization turned out not to improve the effectiveness, we stick to the above formula. Thus we compute all the iUnit scores of a query q, and rank them according to the scores.

3.3.2 Bi-gram Dirichlet prior smoothing

We extend the Dirichlet prior smoothing method to the word bi-gram language models as follows:

$$score(u,q) = \sum_{w_i, w_{i+1} \in W_u} \ln P_{bi}(w_{i,i+1}|q) \tag{7}$$

where $w_{i,i+1}$ is the word sequence appearing in the text of either iUnit u or D_q and the bigram language model $P(w_{i,i+1}|q)$ is :

$$P_{bi}(w_{i,i+1}|q) = \begin{cases} \frac{N_{D_q,w_{i,i+1}} + \mu P_{bi}(w_{i,i+1}|o)}{N_{D_q}+\mu} & (w_{i,i+1} \in D_q) \\ P(w_i|q) & (otherwise) \end{cases} \tag{8}$$

where $N_{D_q,w_{i,i+1}}$ is the count of word sequence w_i, w_{i+1} in the set of query relevant documents D_q and N_{D_q} is the count of all word positions in documents D_q.

3.3.3 Mixtured Language Modeling Methods

We combine uni-gram and bi-gram models of the Dirichlet prior smoothing method described above with the mixture parameter α as follows:

$$\begin{aligned} score(u,q) = \alpha \sum_{w_i \in W_u} \ln P(w_i|q) + \\ (1-\alpha) \sum_{w_i, w_{i+1} \in W_u} \ln P_{bi}(w_{i,i+1}|q) \end{aligned} \tag{9}$$

where α is calibrated empirically.

3.4 Other Language Modeling Methods

We experimented on several variations of the language modeling based iUnit ranking; hereafter we focus on the uni-gram language model.

3.4.1 KL Divergence

This approach computes KL Divergence between a query language model and a background language model instead of Log Odds Ratio as follows:

$$score(u,q) = D(P(w|q)||P(w|o)) = \sum_{w \in W_u} P(w|q) \ln \frac{P(w|q)}{P(w|o)} \tag{10}$$

This method is very similar to organizer's baseline Log Odds Ratio method but here each log odds ratio is weighted by $P(w|q)$, hence each ratio is averaged through all words instead of being summed up. Therefore KL-Divergence or relative entropy of two distributions acts as "averaged" divergence of two distributions.

3.4.2 Pitman-Yor smoothing

We applied comparatively new Pitman-Yor smoothing method [7] as follows:

$$\begin{aligned} P(w|q) &= \frac{N_{D_q,w} - \delta + (\mu + \delta V_{D_q}) P(w|o)}{N_{D_q} + \mu} \\ score(u,q) &= \sum_{w \in W_u} \ln P(w|q) \end{aligned} \tag{11}$$

where V_{D_q} is vocabulary size in D_q and δ is discounting parameter. Pitman-Yor process is a non parametric Bayesian model where the discount parameter δ makes the Pitman-Yor process more suitable for modeling power-law tail behavior of the word frequency distribution than the Dirichlet process. Giving 0 to δ reduces this to the Dirichlet smoothing and giving 0 to μ, the absolute discount smoothing, both of which are proposed in [9].

3.4.3 iUnit Language Modeling Method

So far, our language modeling approach used the query language model $P(w|q)$ to estimate the probability of generating the iUnit to rank. This approach is inconvenient when trying to incorporate "query expansion techniques", which intensively applied to document search tasks, since the expansion of neither iUnit representation nor query representation would change the ranking. In order to solve the problem, we introduce the iUnit language model to estimate the probability of generating the query, where iUnit plays the role of "documents" and query of "query" in the ordinary document search setting.

3.4.4 iUnit Language Modeling with Query Expansion Methods

By introducing iUnit language model approach, we expanded queries by using various external resources such as sets of co-topic, co-click and co-session queries[2] from Yahoo! JAPAN search logs, question-answer pairs from Japanese counterpart of Yahoo! Answers, Yahoo! JAPAN chiebukuro.

4. IUNIT SUMMARIZATION SUBTASK

We adopted the LM-based two-layer iUnit summarization baseline which the task organizer provided. It adopts the strategy of allocating query relevant iUnits in the first layer and intent matching iUnits in the second layer.

4.1 LM-based Two-layer iUnit Summarization Baseline

First, we explain the baseline method. iUnits are ranked based on the language modeling methods described in Section 3. Then, the top-ranked iUnits are put into the first layer until the length limit and lower-ranked iUnits are matched against each intent and put into the second layer of the matched intent according to the matching score. The first layer and second layer are filled until the text span exceeds the pre-defined length limit.

We generalize the computation of the iUnit score against each second layer intent as follows:

$$Score(u,i) \quad = \quad R(u) \cdot Sim(u,i) \tag{12}$$

where u and i represent iUnit and intent respectively. $R(u)$ is the iUnit ranking score from the ranking method described in Section 3. Sim(u, i) is the score of intent matching, which we define in the following Subsections. iUnits that are not

used in the first layer are put into the second layer based on $Score(u, i)$.

4.2 Set based Intent Matching

This is an asymmetric similarity function between u and i implemented in organizer's baseline two layer summarization system.

$$Sim_{set}(u, i) = \frac{|W_u \cap W_i|}{|W_i|}$$

where W_x is the set of words contained in x. Notice that this $Sim_{set}(u, i)$ becomes 0 when there is no common word between u and i, where a small amount of similarity is given as a smoothing factor.

4.3 Word Embedding based Intent Matching

In order to address aforementioned shortcomings of the set based intent matching, we propose a method using the similarity between embedding representations of iUnit and intent instead of bag-of-word representations. We represents iUnit embedding Emb_u by the sum of embeddings of including words as follows:

$$Emb_u = \sum_{w_u \in W_u} Emb_{w_u}$$

where Emb_{w_u} is embedding of the word w_u.

In the same way, Intent Embedding Emb_i is represented as follows:

$$Emb_i = \sum_{w_i \in W_i} Emb_{w_i}$$

where Emb_{w_i} is embedding of the word w_i.

Then, we calculate similarity $Sim_{emb}(u, i)$ as follows:

$$Sim_{emb}(u, i) = \cos(Emb_u, Emb_i)$$

where $\cos(X, Y)$ is cosine similarity. We also tried another similarity measure based on the Euclidean distance between vectors in additional experiments.

5. EXPERIMENTS

5.1 Japanese iUnit Ranking Experiments

We used the title and body fields of provided "INDEX" documents against each topic query to train our probabilistic and vector space models.

5.1.1 Training Run Results

Table 1 shows the results of training runs of the methods described in Section 3.

We tried the vector space baseline first but it performed poorly and did not reach organizer's baseline. Then, we tried the KL-divergence method that can be easily implemented on top of the organizer baseline system, and achieved 0.8108 in Q-measure. As we noticed the importance of background information, we extended the vector space baseline by subtracting the similarity to the background vector from that of the foreground vectors; this approach achieved 0.8003 in Q-measure, which is fairly good but did not reach the KL-divergence run. Hereafter, we focus on the language modeling approach and adopted Dirichlet prior smoothing with uni-gram, bi-gram and mixture language models. These

Run description	Run detail	Q-Measure
Random ranking (ORG-R)	–	0.7201
Log Odds Ratio (ORG-L)	Laplace smth	0.7901
Vector Space Cosine	term freq	0.7715
Vector Space Cosine	Boolean	0.78
Vector Space+Background	Boolean	0.8003
Uni-gram Dirichlet priors	$\mu = 1, \alpha = 1$	0.8347
Uni-gram Dirichlet priors	$\mu = 0.5, \alpha = 1$	0.8352
Bi-gram Dirichlet priors	$\mu = 1, \alpha = 0$	**0.8399**
Mixture Dirichlet priors	$\mu = 1, \alpha = 0.5$	0.8375
KL-Divergence	Laplace smth	0.8108
Pitman-Yor	$\mu = 1, \delta = 0.1$	0.8321
iUnit LM	Dir prior $\mu = 1$	0.8258
iUnit LM+cotopic	Dir prior $\mu = 1$	0.8343
iUnit LM+coclick	Dir prior $\mu = 1$	0.8339
iUnit LM+cosession	Dir prior $\mu = 1$	0.8329
iUnit LM+chie	Dir prior $\mu = 1$	0.8345

Table 1: Japanese iUnit Ranking Training Run Results

runs achieved 0.8347 – 0.8399 in Q-measure with training set, and the improvement over the organizer baseline is statistically significant with $\alpha = 0.01$. Pitman-Yor smoothing method performed slightly inferior than Dirichlet prior runs presumably due to inappropriate parameter setting. Although experiments through various TREC and NTCIR test collections indicate that giving the value more than 100 to the parameter μ leads to the best effectiveness on ad hoc document search tasks[1], we obtain the best Q-measure when giving 0.5 to μ in this task. We used the parameter $\mu = 1$ as well which is equivalent to Laplace smoothing. We implemented the iUnit language model method in view of model expansion. We tried the iUnit model expansion by using external resources such as sets of co-topic, co-click and co-session queries or question-answer pairs from Yahoo! JAPAN chiebukuro data. The results are fairly good but they did not reach our best performing runs.

5.1.2 Test Run Results

Our official and additional test runs are shown in Table 2.

Run description	Run detail	Q-Measure
Random Ranking (ORG-R)	–	0.7411
Log Odds Ratio (ORG-L)	Laplace smth	0.7269
Uni-gram Dirichlet priors	$\mu = 10, \alpha = 1$	**0.8072**
Bi-gram Dirichlet priors	$\mu = 1, \alpha = 0$	0.7965
Mixture Dirichlet priors	$\mu = 1, \alpha = 0.5$	0.8029
Uni-gram Dirichlet priors	$\mu = 0.5, \alpha = 1$	**0.8081**

Table 2: Japanese iUnit Ranking Test Run Results; official runs above the double line and additional runs under the double line.

We submitted the runs of Dirichlet smoothing methods with uni-gram, bi-gram and mixture language models. Unlike training run results, the uni-gram model performed the best in Q-measure. There might be room for improving our bi-gram model especially in smoothing that we put aside for the future plans. Another surprising thing in test results is that the random ranking performed even better than the

Submit #	Run type	Ranking	Intent Matching	Limit	M-measure
123	ORG-T	Log Odds Ratio LM	Set based	280	17.4376
437	Addition	Log Odds Ratio LM	Emb+Cos	280	19.094
131	Official	KL-Div LM	Set based	280	21.0259
173	Official	Dir priors LM	Emb+Cos	280	25.8498
231	Official	Dir priors LM	Emb+Cos	0	13.9927
324	Official	Dir priors LM	Emb+Cos	252	25.6084
419	Addition	Dir priors LM	Set based	280	26.7036
442	Addition	Dir priors LM	Emb+Euclidean	280	26.6096

Table 3: Japanese iUnit Summarization Run Results; Limit indicates the first layer length limit.

log odds ratio baseline run. The situation seems to be completely different in the English iUnit ranking subtask, where the log odds ratio baseline performs much better than our language modeling runs[4]. The reasons of poor performance of our methods against English data are also to be investigated in future work.

5.2 Japanese iUnit Summarization Experiments

We trained word embedding vector representations by using publicly available word2vec implementation[5] from <body> elements of given HTML documents. We indicated the parameters as follows: vector size is 200, model is "continuous bag-of-words" and window size is 5.

Table 3 shows our Japanese iUnit summarization run results.

The baseline effectiveness is improved either by introducing word embedding intent matching or our iUnit ranking methods described in Section 3. Combining word embedding based intent matching and iUnit ranking method based on Dirichlet prior smoothing, we achieved 25.8498 in M-measure (#173). It seems the iUnit ranking method largely affects the M-measure. In additional runs, we tried set based intent matching with the same iUnit ranking method and it achieved even higher M-measure as 26.7036 (#419). We compared these two results by query basis; embedding intent matching performed better on 23 queries, performed equally on 4 queries, whereas a set based method performed better on other 73 queries. There are no specific tendency of queries observed, on which either method performed better.

We also tried another similarity function of embedding based intent matching instead of cosine similarity, namely Euclidean distance based similarity, which achieved 26.6096 in M-measure (#442). By comparing above Euclidean run (#442) with the set based run (#419) on query basis, Euclidean performed better on 44 queries, performed equally on 13 queries, and set based method performed better on other 43 queries. The vector similarity measure greatly affects the effectiveness of intent matching of word embedding based. This suggests that the better usage of word embedding representation leads to more effective intent matching solutions.

Furthermore, we investigated the influence of the two layer strategy by adjusting the length limit of the first layer allocation. By default, the limit is set to 280, i.e. same as **X**, the maximum length to be read in an layer and the half of the patience parameter **L** of M-measure. We change this to 0 and 252 (90% of the default limit), of which the results are shown in #231 and #324 of Table 3. From these results, it seems that the default limit is near optimum. Reducing the first layer allocation leads to the degradation in M-measure.

6. FUTURE WORK

For iUnit ranking, we used the ranking only by the divergence between iUnit and background language models. In future work, we use ensemble learning to rank important iUnits using several features including both textual and non-textual features.

For iUnit Summarization, we used word embedding vectors and cosine similarity for the second layer allocation. Our future plans include examining better word embedding representations in view of Intent matching, as well as examining other similarity measures to vectorial matching such as KL-divergence, Jaccard coefficient and so on. Finally, while we adopted the two layer strategy of the organizers' baseline system, optimizing the strategy in view of M-measure makes another challenging research topic, which we tackle in the next step.

7. CONCLUSIONS

In this paper we reported the work carried out by the YJST team in MobileClick-2. We participated in both two subtasks: iUnit Ranking and iUnit Summarization.

For the iUnit ranking subtask, we used Dirichlet prior smoothing in the LM-based iUnit ranking approach. We carried out several experiments examining Uni-gram/Bi-gram iUnit/query models, smoothing methods, ranking functions and so on. As a result, we achieved Q-score of 0.807 in a test run using a Uni-gram model.

For iUnit Summarization, we adopted a new intent matching method using word embedding representations of iUnits and Intents. Using this for iUnit / Intent semantic matching leads to a finer allocation of relevant iUnits to subtly related intents in the second layer. We achieved M-measure of 25.8498, and which is the best of official runs of the Japanese iUnit Summarization Subtask. Moreover, additional experiments suggest the possibility of further improvements on the results with more effective similarity matching.

8. REFERENCES

[1] S. Fujita. Revisiting document length hypotheses: A comparative study of japanese newspaper and patent retrieval. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(2):207–235, 2005.

[2] S. Fujita and Y. Ozawa. Specifications of Related Queries Extracted from Yahoo Search Logs. included in data deliverables.

[3] M. P. Kato, M. Ekstrand-Abueg, V. Pavlu, T. Sakai, T. Yamamoto, and M. Iwata. Overview of the ntcir-10 1click-2 task. In *Proceedings of NTCIR-10*, pages 243–249, 2013.

[4] M. P. Kato, T. Sakai, T. Yamamoto, V. Pavlu, H. Morita, and S. Fujita. Overview of the ntcir-12 mobileclick-2 task. In *Proceedings of NTCIR-12*, 2016.

[5] T. Mikolov. https://code.google.com/archive/p/word2vec/. Google Code | Archive word2vec.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013.

[7] S. Momtazi and D. Klakow. Hierarchical pitman-yor language model for information retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. ACM*, 2010.

[8] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, 1998.

[9] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. In *ACM Transactions on Information Systems, Vol. 22, No. 2*, pages 179–214, 2004.