Potential Good Abandonment Prediction

Aleksandr Chuklin
Yandex & Moscow Institute of Physics and Technology
Moscow, Russia
chuklin@yandex-team.ru

Pavel Serdyukov
Yandex
Moscow, Russia
pavser@yandex-team.ru

ABSTRACT

Abandonment rate is one of the most broadly used online user satisfaction metrics. In this paper we discuss the notion of *potential good abandonment*, i.e. queries that may potentially result in user satisfaction without the need to click on search results (if search engine result page contains enough details to satisfy the user information need). We show, that we can train a classifier which is able to distinguish between potential good and bad abandonments with rather good results compared to our baseline. As a case study we show how to apply these ideas to IR evaluation and introduce a new metric for A/B-testing — *Bad Abandonment Rate*.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Experimentation, Measurement

Keywords

IR system evaluation, search result abandonments

1. INTRODUCTION AND ANALYSIS

Abandonment rate is one of the most broadly used online user satisfaction metrics to evaluate the quality of information retrieval (IR) systems and compare IR systems in A/B-testing experiments (see e.g. [4]). It is generally considered that the lower abandonment rate is the better IR system performs. In [5] it was first introduced and analyzed the notion of *good abandonment*, i.e. situation when the user abandoned search engine result page because her information need has been satisfied[1]. We use the notion of *potential good abandonment* from the same work [5]: we call a query to be a *potential good abandonment* if it can theoretically be answered directly on the SERP (Search Engine Result Page). In [5] they propose an exhaustive analysis of potential good/bad abandonment share in mobile/PC search and in various countries. Another work [1] contains analysis of different types of good abandonment and special SERP elements dedicated to answer user query. In turn we propose a way to **automatically** predict query type (potential good/bad) using machine learning.

We prepared our experimental dataset by extracting a sample of abandoned queries (1500 unique queries) from the query log of Yandex search engine. Then we mark each query as potentially *good*, *bad* or *maybe* good abandonment. Our guidelines were very similar to those used in [5]. For instance, query "NDAQ" should be labeled as *good* because most of the users probably want to see stock quotes (or chart) which could easily be fitted into SERP. On the other hand, query "facebook" could not be satisfied without a click on a search result. Finally, issuing queries like "Anna Kournikova" user may want to find some photos (*bad* abandonment) or may just want to know who this person is (*good* abandonment). Because we are not sure whether some intent is bigger, we put *maybe* label to this query.

It is also worth mentioning that it was not allowed to use any of the existing search engine systems, so some queries may be completely unclear and left unlabeled (about 17%). We exclude such queries from the training process. Overall, the label distribution from the assessment is the following: ***good*: 34%, *bad*: 49%, *maybe*: 16%** . As shown in [5] these proportions may be different across countries.

2. METHOD

After obtaining judgements we built a feature vector for each query. We used 3 different feature sets in our setup:

Topical features (64 binary features). We classified queries into topics, such as blogs, music, medicine, etc. We combined hand-crafted rules with classifiers based on query reformulations (built in a manner similar to [6]). Our motivation was that some query categories may imply query being potentially good and bad abandonments. At the same time we understand that certain query categories, like for example "Questions", may contain both potentially good (simple and specific questions) and bad (difficult and broad questions) abandonments

Linguistic features (11 features): query length in characters/words, inverted length in words ($InvWordCount = 1/QueryWords$), sum of query words' IDFs, *RussianLanguage* (binary feature). We also include pre-retrieval query performance predictors from [2] (*specificity* group: *AvICTF*, *SCS*, *AvIDF*, *DevIDF*, *MaxIDF*, *AvQL*). Our intuition was that the query clarity and specificity influences not only the

[1]In [5] they considered only queries that were not followed by any click or further query within next 24 hours. Instead we consider queries that were not **directly** followed by a click, but may be followed by another query.

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

Table 1: Classification Results

Class	Precision	Recall	F-measure
good	0.57	0.53	0.55
bad	0.66	0.77	0.71
maybe	0.50	0.31	0.38

Table 2: Feature Strength

Feature Set	Accuracy	Average F-measure
baseline	49.3%	0.325 (+0%)
w/o topical	52.3%	0.516 (+59%)
w/o linguistic	60.1%	0.583 (+79%)
w/o history	60.8%	0.592 (+82%)
all features	**61.3%**	**0.601 (+85%)**

query performance, but also its potential to be answered directly on the SERP.

Query history (9 features)[2]: average number of SERP clicks per query, average number of "next" result pages examined for the query, click entropy, presence of navigational intent, geographical region distribution entropy, query specific to morning, day, evening, night (4 binary features). The idea behind this feature set was that previous user interaction with the system may be to some extend motivated by query being potentially good or bad abandonment.

For classification we decided to use `SVM` algorithm with feature normalization and RBF kernel as a state-of-the-art algorithm used for many information retrieval tasks. This algorithm performed not worse than other algorithms available in Weka Machine Learning Library[3].We fitted algorithm parameters using grid search technique proposed in [3]. Results are summarized in Table 1. We used stratified 10-fold cross-validation to evaluate our algorithms.

We then evaluated each set's performance by removing it from the data and repeating the same training procedure. As a baseline we used classifier which marks all queries as *potentially bad* (the approach implied by online retrieval evaluation methods based on abandonment). Results are summarized in Table 2. We can see that topical feature set is essential for our task, while two other sets do improve its performance. We also identified best individual features in each set. For each feature set S and each $f_0 \in S$ we removed all $f_i \in S \setminus \{f_0\}$ and calculated accuracy. Both **linguistic** and **history** feature sets did not have any particular leader. However *InvWordCount*, *RussianLanguage* and *SCS* performed slightly better than other linguistic features. In **topical** feature set many individual features give us significant accuracy gain. Here are TOP-5 query categories: *Shopping*, *Download*, *Local*, *Video*, *Adult*.

Case Study. Now let us discuss how to apply our classifier to refine abandonment rate metric and make it less "noisy". We call such a metric *Bad Abandonment Rate*. The idea is the following: when we compare two IR systems in traditional A/B-testing setup, we should consider only queries that are highly unlikely to be *good abandonments*. We propose to build an automated classifier that exploits various features. For that purpose we merged *good* and *maybe* classes and performed logistical transformation of SVM output to obtain classifier that outputs probability of being labeled as potential *bad* abandonment. We

[2]These features were calculated using 3-month Yandex query log

[3]See http://www.cs.waikato.ac.nz/ml/weka/.

Table 3: Bad Abandonment Rate

Metric	queries used	unclear queries
Baseline	100%	51%
Bad Aband. Rate	7%	<1%

can alter probability threshold and choose one that gives us the best precision with sufficiently good recall[4]. Finally we managed to build a classifier with **Precision = 1** and $Recall = 0.15$. Several other points from precision-recall curve: $(P = 0.95, R = 0.28)$, $(P = 0.9, R = 0.40)$, $(P = 0.8, R = 0.55)$.

Now we define *Bad Abandonment Rate* metric as an abandonment rate calculated **only** for queries classified as *potentially **bad*** with maximal confidence (i.e. we leave $0.15 \cdot 49\% \approx 7\%$ of all queries). All other queries (*good* and *maybe* classes) may result in good abandonment, so they should be considered as unclear for the purpose of IR systems comparison. We use traditional *Abandonment Rate* metric as a baseline (see Table 3). We can see that by decreasing number of queries used for system evaluation we can guarantee that only very small fraction of abandoned SERPs may be actually good abandonments. We believe that *Bad Abandonment Rate* better represents IR system quality than traditional abandonment rate metric.

3. CONCLUSIONS

In this paper we developed a classification framework to automatically predict such query characteristic as potential user satisfaction with SERP itself without need to click. We presented three different feature sets and evaluated each set's performance. We also addressed a problem of currently existing abandonment rate metric and proposed a method called *Bad Abandonment Rate* aimed to decrease a number of unclear queries when comparing two IR systems. As a next step we would like to validate our metric compared to other online user satisfaction metrics.

In this work we discussed only *potential* good abandonments. Another important direction might be studying **actual** good and bad abandonments: for particular user and particular *potentially good* query classify user's abandonment as either good or bad. Of course we need to extend our feature set by features extracted from real user sessions. On the top of such a technique we can develop more precise filtering than *Bad Abandonment Rate* proposed here.

4. REFERENCES

[1] L. Chilton and J. Teevan. Addressing people's information needs directly in a web search result page. In *WWW'11*.

[2] C. Hauff, F. D. Jong, and D. Hiemstra. A Survey of Pre-Retrieval Query Performance Predictors. *CIKM'08*.

[3] C. Hsu, C. Chang, and C. Lin. A practical guide to support vector classification, 2003.

[4] R. Kohavi, R. Henne, and D. Sommerfield. Practical guide to controlled experiments on the web: listen to your customers not to the hippo. In *KDD'07*.

[5] J. Li, S. Huffman, and A. Tokuda. Good abandonment in mobile and PC internet search. *SIGIR'09*.

[6] M. Pasca. Weakly-supervised discovery of named entities using web search queries. *CIKM'07*.

[4]Another possible approach might be to use such probabilities as query weights when calculating abandonment rate.