Automatic Keywords Generation for Contextual Advertising

Pengqi Liu
Carnegie Mellon University
Pittsburgh, PA, USA
pengqil@andrew.cmu.edu

Javad Azimi
Microsoft Inc.
Sunnyvale, CA, USA
jaazimi@microsoft.com

Ruofei Zhang
Microsoft Inc.
Sunnyvale, CA, USA
bzhang@microsoft.com

ABSTRACT

Contextual Advertising (CA) is an important area in the industry of online advertising. Typically, CA algorithms return a set of related ads based on some keywords *extracted* from the content of webpages. Therefore, extracting the best set of representative keywords from a given webpage is the key to the success of CA. In this paper, we introduce a new keywords *generation* approach that uses some novel NLP features including POS and named-entities tagging. Unlike most of the existing keyword extraction algorithms, our proposed framework is able to generate some related keywords which do not exist in the webpage. A monetization parameter, predicted from historical search keyword performance, is also used to rank potential keywords in order to balance the RPM (Revenue Per 1000 Matches) and relevance. Experimental results over a very large real-world data set shows that the proposed approach can outperform the state-of-the-art system in both relevance and monetization metrics.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Machine Learning**]: Metrics—*Performance Measures*

General Terms

Algorithm, Design.

Keywords

Contextual Advertising, Keyword Generation.

1. INTRODUCTION

Contextual Advertising (CA) is one of the main revenue sources of online advertising companies [5] where the goal is displaying relevant ads for a given webpage. Since the content of webpages usually include lot of texts, it is very useful to *extract* or *generate* some representative keywords from the webpages. The keywords are then used to select a set of relevant ads from the ad corpus. Therefore, selecting a set of keywords that can grab the core idea of the webpage is a very important task in CA algorithms. Typical keywords extraction algorithms consist of four steps: 1)*Content Extraction* which extracts the main content of an HTML page [5], 2)*Candidate Keywords Selection* which returns a set of candidate keywords from the content [2], 3)*Feature Extraction* which converts the candidate keywords to a set of features using some rules, and 4)*Predictive Model* which classifies the candidate keywords into relevant and irrelevant keywords based on the extracted features [5].

In this work, we present a novel approach which *generates* a set of representative keywords from a given webpage. We first use N-Gram method [3] to extract a set of candidate keywords. Next, a set of Part-Of-Speech (POS) patterns are introduced in order to filter out the noisy candidate keywords [2]. Then, a set of feature extraction rules are presented to create a set of features for each keyword. Finally, a logistic regression classifier is used to determine the relevant keywords. Once a set of keywords are selected, we use query expansion techniques [4] to generate more related keywords which do not exist in the content. We also introduce a novel ranking scheme in order to *re-rank* the generated keywords based on their relevance and expected RPM. Our experimental results show that the proposed algorithm is able to generate representative keywords given a webpage.

2. KEYWORD GENERATION ALGORITHM

In this section, we introduce our proposed algorithm which consists of six different steps, *Content Extraction*, *Candidate*, *Feature Extraction*, *Prediction Model*, *Keywords Generation* and *Ranking Scheme*.

2.1 Content Extraction

Given a webpage, we first extract the main text along with it's URL and title [5]. Then, we tag the main block with POS and named-entity taggers.

2.2 Candidates

We first create all possible N-Grams where $N \leq 6$. Then, some POS heuristic rules, based on experimental observations, are introduced to filter out the noisy and useless candidate keywords. This can also significantly reduce the number of candidate keywords. The proposed POS rules are as follows: 1) $(Noun)^+$: N-grams containing only one or more noun tags, 2) $(Adjective) + (Noun)^+$: N-grams starting with one adjective and following by one or more nouns, 3)$(Adverb) + (Adjective) + (Noun)^+$: N-grams starting with one adverb following by one adjectives, and following by one or more nouns, and 4)$(Noun)^+ + [CD]$: N-grams starting with one or more nouns and followed by a number, e.g. Xbox 360. We select all of the candidate keywords which follows one of the above rules and the rest of them are discarded.

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577361.

2.3 Feature Extraction

In this section, we propose our feature extraction rules in order to convert each candidate keyword to a set of features. The proposed features are listed as follows:

- *Information Retrieval Features*: 1)Term Frequency (TF), 2)Document Frequency (DF), 3)log of Inverse DF (IDF), and 4)Product of TF and IDF (TF-IDF).
- *Keyword Position Features*: *IsURL* and *IsTitle* which are are binary features indicating whether the keyword appears in the URL/title or not.
- *Candidates Related Feature*: 1)Document Position (DocPos) which is the number of words before the first occurrence of the keyword, and 2)Length which is the number of characters in the keyword.
- *Searched Query Feature*: for each keyword, we count the portion of queries (6 months of Bing search log) which include the candidate keyword as part of the query. This feature is called *Query-Log* which represents the popularity level of the keyword.
- *Named-Entity Feature*: to capture the representativeness of each candidate keywords X, an *Entity-Score* feature is calculated as the average pointwise mutual information [1] of X and extracted *named-entity* terms.

2.4 Learning Model

Once a set of feature $\bar{x}$ is created for a given keyword X, we use Logistic Regression (LR) to calculate its probability of being a good keyword: $p(Y = 1|X = \bar{x}) = \frac{\exp(\bar{x}.\bar{w}^T)}{1+\exp(\bar{x}.\bar{w}^T)}$, where the weight vector $\bar{w}$ is learned based on the training data. If a keyword has a probability of greater than 0.5, it would be selected as a representative keyword.

2.5 Keywords Generation

Once a set of keywords are selected, we use the query expansion techniques [4] including co-bid/co-click keywords and query relationship learning to find related search keywords based on Bing search logs. This can help us to create some relevant keywords that do not *exactly* exist in the content. For example, for the selected keyword *SF Giants*, the keywords *"San Francisco Giants ticket"*, *"SF Giants ticket"*, *"SF Giants jersey"* are added to the set of selected keywords.

2.6 Ranking Scheme

We introduce a heuristic to estimate the monetization value, Expected Revenue (ER) of each generated keyword. Let define $r(X) = CTR(X).CPC(X)$ as the ER of keyword X, where $CTR(X)$ and $CPC(X)$ are the Click Through Rate and Cost Per Click of keyword X based on historical performance on Bing search engine. Then, given the logistic regression score $p(X)$, we calculate the ranking score, $m(X)$, as follows: $m(X) = p(X).\frac{\alpha+r(X)}{\alpha}$, where $\alpha = \frac{100}{l}$ and l is the monetization lever; the lower the monetization lever, the more important the ER in the ranking scheme.

3. EXPERIMENTS

3.1 Data Sets

We used 6 months search queries in Bing with corresponding clicked URLs. First, for each URL, we extract the candidate keywords. Then the keywords appeared in the searched query are labeled as relevant and the rest as irrelevant. The advertiser bidded keywords of clicked ads along with the search result pages are also used as relevant labels. As the result, a large number of samples, 307,829 webpages, are generated for training and evaluation.

3.2 Results

The data is splitted into test (30%) and training (70%). The LR parameters are leaned using the training data and the proposed approach is evaluated over the test data. Precision, Recall, F-Score and Area Under Curve (AUC) are used as evaluation metrics. The results are presented in Table 1. The first row is the results using all of the features and the other rows except for the last one, are the results using all of the features excluding the one listed at the first column. For example, the second row is the result over all features except for Named-Entity Score. The results show the efficiency of our approach in generating relevant keywords from webpages. We also compare our algorithm with KEX [3] which is one of the state-of-the-art keywords extraction algorithms. The results, the last row of Table 1, show that except for precision, our algorithm is able to outperform KEX in all other metrics. The results, the last row of Table 1, show that our algorithm is able to outperform KEX in all metrics except for precision. Our further investigations show that some mislabeling cases caused by the automatic data generation hurt our approach more than KEX, which may explain its lower precision than KEX.

3.3 Ranking Scheme

We also evaluate the effect of the monetization, $l = 10$, in our experiment by measuring the average expected RPM ($CTR*CPC$) of top 5 returned keywords as evaluation metrics. As a result, the proposed algorithm was able to achieve 10% improvement in expected RPM comparing to KEX.

Table 1: The proposed approach results.

Features	Precision	Recall	F-Score	AUC
All Features	0.86	0.68	0.76	0.86
-Entity-Score	0.91	0.58	0.71	0.84
-Query-Log	0.85	0.66	0.74	0.84
-TF-IDF	0.85	0.68	0.75	0.85
KEX	0.91	0.58	0.71	0.84

4. CONCLUSION AND FUTURE WORK

In this paper, we introduced an approach which generates representative keywords from a given webpage using some novel POS patterns and features. Unlike most of the previous work in CA, our proposed approach is able to generate keywords which do not exactly exist in the webpage. A ranking scheme is also introduced to combine the keyword's relevance measurement and monetization value predicted from historical search keyword performance. The experimental results over a very large real world data set demonstrate the effectiveness of proposed approach in generating more relevant and more monetizable keywords comparing to the state-of-the-art system.

5. REFERENCES

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. 1991.

[2] K. S. Dave and V. Varma. Pattern based keyword extraction for contextual advertising. In *CIKM*, 2010.

[3] E. Frank, G. W. Paynter, I. H. Witten, and et al. Domain-specific keyphrase extraction. In *IJCAI*, 1999.

[4] J. Gao, G. Xu, and J. Xu. Query expansion using path-constrained random walks. In *SIGIR*, 2013.

[5] W.-t. Yih, J. Goodman, and V. R. Carvalho. Finding advertising keywords on web pages. In *WWW*, 2006.