International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Semantic Based Topic Generation from Search Term Reports

Manjima Raj[1], Chithra Rani P. R.[2]

[1]M.Tech Student, Department of Computer Science and Engineering, Ilahia College of Engineering and Technology, Muvattupuzha, Kerala, India

[2]Assistant Professor, Department of Information Technology, Ilahia College of Engineering and Technology, Muvattupuzha, Kerala, India

Abstract: ***Internet has strengthened the ability to search the content people produce on the internet and find those pages that are relevant to a given query. Marketing agencies are created out of this and manages thousands of keywords in order to reach the right customer. Advertisers can use Search term reports to see the performance of their ads on search queries. Given large keyword portfolio, Identifying new search terms with high potential from search term reports can become a burden on even experienced advertisers. Advertisers have to determine the purchase intents of users. Once the target intents are determined, advertisers can target those users with relevant keywords. In order to scale the keyword management, Semantic topic generator is proposed where we learn the hidden topics in available in search term reports.***

Keywords: Latent Semantic Indexing, Search Engine Marketing, Latent Dirichlet Allocation**,** Search term reports

1. Introduction

Lucrative markets are created out of the information-seeking behavior of billions of people traversing the web [1]. Consider a do-it-yourself (DIY) social network creation platform that sells subscription based services. There are two options for the company to market their services:
1. Launch a marketing campaign for displaying ads to the full set of Internet users, or
2. Employ what is known as precision targeting as in "show my ad to any user who enters the query create my own social networking site".

The key insight in the second option is that a user query may indicate what is known as purchase intent. If a user types into a web search engine "create my own social networking site", then we may infer that this user wants to create a social network, and is potentially willing to pay for the provided service. There could be many such users browsing the web.
An advertiser can create an online advertisement that speaks exactly to the market segment containing these users or in other words create an advertisement to the specific purchase intent in question. This new type of advertising is attractive to both advertisers and customers at the same time. The query "create my own social networking site is called a search keyword, or keyword for short, and the type of advertising that revolved around this notion is called keyword-based advertising.

A 'search term' is the exact word or set of words a customer enters when searching on Google.com or one of our Search Network sites. A 'keyword' is the word or set of words advertisers create for a given ad group to target ads to customers. SEM agencies and practitioners manage thousands of search keywords on behalf of their clients. Any increase in the number of products being offered results in an increase in the number of search keywords being managed. The campaign management dashboards provided by Google Adwords, Google Adwords Editor for bulk edits, Microsoft adCenter or Bing Ads have interfaces to change bid, scope, budget, and many other attributes per keyword or per groups of key words. In addition, the management dashboards have features for advertisers to annotate their various bid choices, keyword ideas, and advertisement text options, and later on to conduct controlled experiments (A/B tests) on attribute variants [2]. The ability to run controlled experiments enables advertisers to test and select the best performing variants for their online marketing campaigns.

However, given a large keyword portfolio and many variants to consider, the campaign management can easily become a burden on even experienced advertisers. Before creating a search campaign, an advertiser first needs to identify purchase intents, expressible as search keywords [4]. These information needs are succinct descriptions of what each client, product or service is offering to its users. It is very difficult to know what the hidden information need is from a few words in the query. The management dashboards fall short of providing a solution to formulate the hidden needs.

2. Related Work

Probabilistic topic modeling is used to discover and annotate large archives of documents with thematic information [5]. With an increasing number of news, scientific articles, books, blogs, and web pages, it gets more difficult to categorize and search among the wealth of information. The idea behind LDA is to model documents as being generated from multiple topics (e.g., K topics) such that each of these topics is a probability distribution over a pre-defined vocabulary. Each document in the corpus exhibits topics in different proportions. In order to learn word distributions per topic and topic proportions per document, posterior probabilistic inference is used. With the observed documents in the corpus as output, hidden topical structure can be inferred by a deterministic variational method. In the variational inference procedure, a simpler distribution that contains free variational parameters is used to approximate the true

posterior distribution. There are three sets of hidden variables each of which is governed by a different variational parameter: (1) a word distribution per topic, (2) a topic distribution per document, and (3) word-to-topic assignment per document. All variables are assumed to be independent of each other. The variational parameters that maximize the log likelihood of the observations under the model are computed iteratively by continuous optimization using coordinate ascent. The presence of one latent topic may be correlated with the presence of another. Since there is no notion of time in the model, documents in the corpus are exchangeable. However this assumption is in appropriate for many corpora including search terms reports. Search terms submitted by users reflect evolving content.

3. Proposed System

In order to match the users in need of a particular service with the client, which provides that service, advertisers have to determine the purchase intents or information needs of target users. Once the target intents are determined, advertisers can target those users with relevant search keywords. In order to compile a relevant set of search keywords, advertisers analyze search terms reports, search query logs, and trend reports provided by ad-brokers. In these reports, advertisers are exposed to how their target users express their hidden information need. Reviewing how users express their intent is very crucial as there is an impedance mismatch between how an advertiser describes an information need versus how a user expresses that need [9]. After reviewing reports, the advertiser may decide to add new search keywords to the portfolio. As a consequence, the size of the keyword portfolio keeps increasing over time. The portfolio may be pruned by deleting or pausing keywords that perform poorly [3], but the pruning has the risk of failing to capture the target information needs. Given a large number of keywords, it is difficult to maintain keyword coherence within a campaign, and even more difficult to manage multiple campaigns consisting of a wide array of continuously evolving sets of keywords. Even though, management dashboards provide many features to slice and dice these keywords, they do not provide a semantic overlay or a topical structure on top of the existing campaigns. In order to scale SEM with an increasing number of product offerings while at the same time optimizing for conversions, we propose Semantic Topic Generator where we learn the latent topics hidden in the available SEM search terms data. Our foundational hypothesis is that there are a set of information needs hidden in and behind the search terms as a collection; the latent topics that are discovered through probabilistic inference over this collection correspond to these information needs. The topical structure stands as a viable tool to manage SEM campaigns with precise targeting of users in terms of relevance and to optimize for conversions. Semantic Topic Generator uses an LDA-based topic model. Since information needs may change over time or drift in concept, we learn dynamic topic models by sequentially chaining model parameters in a Gaussian process across a well-defined epoch. We assessed the quality of the models learned in Semantic Topic Generator by showing the predictive power of the framework. Since Semantic Topic Generator‟s internal model can be used to reduce dimensions of the search term space; we foresee that advertisers can scale their campaign management to thousands of keywords comfortably with the use of Semantic Topic Generator.

4. System Design

A search term also known as a search keyword is what a web user types in for querying a search engine. An advertisement keyword is what an advertiser uses for targeting search engine users. Keywords consist of tokens, each of which is a word of a natural language. A topic is a multinomial distribution over a chosen vocabulary of words found in the search terms. For example, networking sites topic has words about social networking with high probability, while private networks topic has words about privacy issues with high probability. The assumption here is that these topics have been specified before the data arose. Assume that some number of “topics”, which are distributions over words, exist for the whole search terms collection. Each search term is assumed to be generated as follows: First choose a distribution over the topics, then for each word, choose a topic assignment and choose the word from the corresponding topic. In order to incorporate the temporal correlation between latent structures, DTM is proposed where each topic evolves from its predecessor smoothly. Latent semantic indexing is used to find more keywords with same semantic sense as those find from LDA.

4.1 Latent Dirichlet Allocation

The intuition behind LDA [10] is that search terms encompass multiple information needs. For example, consider the search term “create a private social community for communication and collaboration”. In this case, the user would like to create a social community, which should be private, and furthermore wants it to be used for enhancing community communication and collaboration. The words about creating online social communities, such as “create”, “Social”, and “community” is highlighted in yellow and marked with number 1 to indicate topic 1; words about privacy in online social networks, such as “private”, are highlighted in pink and marked with number 4 to indicate topic 4; and words about social networking sites for collaboration, such as “communication” and “collaboration”, are highlighted in green and marked with number 3 to indicate topic 3. With this insight, we can see that this search term blends private networks, social communities, and networking sites. LDA assumes that there is a generative process that gave rise to the observed search terms. And the probabilistic inference is used to find out the hidden parameters that characterize the generation. A topic is a multinomial distribution over a chosen vocabulary of words found in the search terms. The goal of LDA [1] is to find out the topic structure given the observed search terms. The topic structure consists of the topics themselves, per-document topic distributions, and per document per word topic assignments.

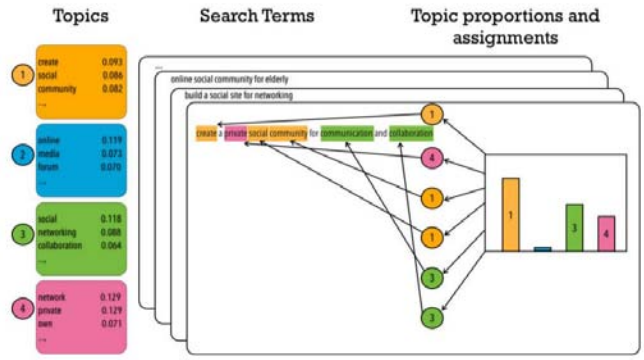

Figure 1: Latent Dirichlet Allocation

The topics and topic assignments are depicted in Figure 1.Assume that some number of "topics", which are distributions over words, exist for the whole search terms collection (far left). Each search term is assumed to be generated as follows: First choose a distribution over the topics (the histogram at far right); then for each word, choose a topic assignment (the colored circles) and choose the word from the corresponding topic. With the topics specified, the algorithm for generating a search term is given in Algorithm 1. Since the algorithm itself is self-explanatory, textual description is omitted.

Algorithm 1: GenerateSearchTerm

1: Input: Topics (there are K topics)
2: Output: topic assignments to words, topic proportions
3: N ← lengthofsearchterm.
4: topicProportions ← topic1 : 0,topic2 : 0,...,topick : 0.
5: topicAssignments ← 1 : None,2 : None,...,N : None.
6: topicDistrandomlychooseadistributionoverTopics.
7: for i in range(N) do
8: topic ← sampleatopicfromtopicDist.
9: wordDist ← retrieveworddistributionfortopic.
10: word ← randomlychooseawordfromwordDist.
11: topicProportions[topic] ← topicProportions[topic] + 1.
12: topicAssignments[i] ← topic.
13: end for
14: return topicProportions, topicAssignments.

4.2 Selection of Words for Vocabulary V

As each topic is a distribution over a fixed vocabulary V, the selection of words for the vocabulary affects the quality of the model fit. The vocabulary can be determined using various methods such as TF-IDF, Cosine based Similarity. Semantic Topic Generator is depicted in Figure 2.

Cosine Similarity based selection

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0 is 1, and it is less than 1 for any other angle. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors at 90 have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0, 1]. Note that these bounds apply for any number of dimensions, and cosine similarity is most commonly used in high dimensional positive spaces. For example, in information retrieval and text mining, each term is notionally assigned a different dimension and a document is characterized by a vector where the value of each dimension corresponds to the number of times that term appears in the document. Cosine similarity then gives a useful measure of how similar two documents are likely to be in terms of their subject matter. The technique is also used to measure cohesion within clusters in the field of data mining. Cosine distance is a term often used for the complement in positive space, that is:

$$D_C(A, B) = 1 - S_C(A, B).$$

It is important to note, however, that this is not a proper distance metric as it does not have the triangle inequality property and it violates the coincidence axiom; to repair the triangle inequality property while maintaining the same ordering, it is necessary to convert to angular distance. One of the reasons for the popularity of cosine similarity is that it is very efficient to evaluate, especially for sparse vectors, as only the non-zero dimensions need to be considered.

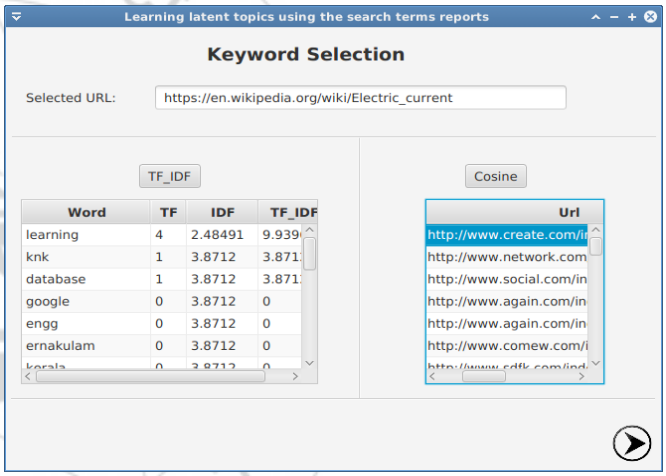

Figure 2: Semantic Topic Generator

4.3 Pre-Processing of the Data

Given a random permutation of the vocabulary, each word is first assigned to an index value equal to the words rank in the permutation. Since there are $|V|$ words in V, word indices run from 1 to $|V|$. Each search term $d \in t$ is for matted according to what is known as the LDA-C format [10]. The given search term $d = w_1 w_2 ... w_N (\forall w\, n \in V)$ is represented as

$$N\, i(w_1) : c(w_1)\ i(w_2) : c(w_2) \ldots i(w_N) : c(w_N),$$

Where N counts the total number of unique words of V in d, the function $i(w)$ return the index of w in V, and the function $c(w)$ returns the number of occurrences of w in d. For d ="create a private social community for communication and collaboration", the words "a", "for", and "and" are not included in V as they are common stop words. Then, d ="create private social community communication collaboration" can be formatted as

6 3: 1 51: 1 ... 234: 1,

Where the index of word in V "create" is taken as 3, the index of word "private" is taken as 51, and the index of word "collaboration" is set to 234 for illustration.

4.4 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a method for discovering hidden concepts in document data. Each document and term (word) is then expressed as a vector with elements corresponding to these concepts. Each element in a vector gives the degree of participation of the document or term in the corresponding concept. The goal is not to describe the concepts verbally, but to be able to represent the documents and terms in a unified way for exposing document-document, document-term, and term- term similarities or semantic relationship which is otherwise hidden. An Example Suppose we have the following set of five documents
d_1: Romeo and Juliet.
d_2: Juliet: O happy dagger!
d_3: Romeo died by dagger.
d_4: Live free or die, that‟s the New-Hampshire‟s motto.
d_5: Did you know, New-Hampshire is in New-England.
and a search query: dies, dagger. Clearly, d_3 should be ranked top of the list since it contains both dies, dagger. Then, d_2 and d_4 should follow, each containing a word of the query. However, what about d_1 and d_5? Should they be returned as possibly interesting results to this query? As humans we know that d_1 is quite related to the query. On the other hand, d_5 is not so much related to the query. Thus, we would like d_1 but not d_5, or differently said, we want d_1 to be ranked higher than d_5. The question is: Can the machine deduce this? The answer is yes, LSI does exactly that. In this example, LSI will be able to see that term dagger is related to d_1 because it occurs together with the d_1‟s terms Romeo and Juliet, in d_2 and d_3, respectively. Also, term dies is related to d_1 and d_5 because it occurs together with the d_1‟s term Romeo and d_5‟s term New-Hampshire in d_3 and d_4, respectively. LSI will also weigh properly the discovered connections; d_1 more is related to the query than d_5 since d_1 is doubly connected to dagger through Romeo and Juliet, and also connected to die through Romeo, whereas d_5 has only a single connection to the query through New-Hampshire.

4.5 Singular Value Decomposition

Singular Value Decomposition is a particular decomposition in Matrix Analysis. It depends upon the fact that if a matrix A of size m by n is transposed to form A_t (which will be of size n by m) such that $A_t[j, i] = A[i, j]$ for all i and j, the product $A_t \times A$ is square and will have Eigen Values which are either positive or zero. The positive square roots of these values are the singular values of the original matrix A. Associated with the singular values are two matrices U and V which form the remainder of the Singular Value Decomposition, $A = U \times s \times V$ where S is a diagonal matrix of the singular values. The U and V matrices each have orthonormal columns, defining directions in the spaces.

5. Conclusion

Semantic topic generator is proposed where hidden topics from search term reports are found. The topics are extremely useful to manage SEM campaigns to give users what they need. Search term reports contains all that information regarding the intent of users, topics is made discoverable by analyzing those search term reports. Semantic topic generator uses a semantic based topic model. Cosine based similarity method is applied to select the most comparative information. Since information needs may change over time, dynamic topic models are introduced across a well-defined epoch. LDA is combined with latent semantic indexing to capture all those topics which have the same semantic sense as those found using LDA from the search term reports. Since LSI based semantic creations describes classes and relationships, it helps to scale campaign management to thousands of keywords comfortably.

References

[1] Ahmet Bulut, Member, IEEE, “TopicMachine: Conversion Prediction in Search Advertising Using Latent Topic Models IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 11, NOVEMBER 2014
[2] R. Kohavi, R. Longbotham, D. Sommerfield, and R. M. Henne, “Controlled experiments on the web: Survey and practical guide, Data Mining Knowl. Discovery, vol. 18, no. 1, pp. 140181, 2009
[3] P. Rusmevichientong and D. P. Williamson, “An adaptive algorithm for selecting profitable keywords for search based advertising services, in Proc. 7th ACM Conf. Electron. Commerce, Jun. 2006, pp. 260269.
[4] C. D. Manning, P. Raghavan, and H. Schutze, Introduction to Information Retrieval. New York, NY, USA: Cambridge Univ. Press, 2008.
[5] D. M. Blei, “Probabilistic topic models, Commun. ACM, vol. 55, no. 4, pp. 7784, 2012.
[6] J. Chang and D. M. Blei, “Hierarchical relational models for document networks, The Ann. Appl. Statist., vol. 4, no. 1, pp. 124150, 2010.
[7] D. M. Blei and J. D. Lafferty, “Dynamic topic models, in Proc. 23rd ACM Int.Conf. Mach. Learning, 2006, pp. 113120.
[8] B. Ribeiro-Neto, M. Cristo, P. B. Golgher, and E. S. de Moura, “Impedance coupling in content-targeted advertising, in Proc. 28th ACM Int. Conf. Res. Develop. Inform. Retrieval, Aug. 2005, pp. 496503.
[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation, J. Mach. Learning Res., vol. 3, pp. 9931022, Mar. 2003.
[10] R. Tibshirani,”Re Regression shrinkage and selection via the lasso”, J. Royal Statist. Soc. Series B, vol. 58, no. 1, pp. 67288, 1996.

Author Profile

Manjima Raj received the Bachelor of Technology degree in Information Technology from Mahatma Gandhi University, Kerala. She is currently doing Master of Technology degree in Computer Science and Engineering with Specialization in Information Systems from Mahatma Gandhi University, Kerala.

Chithra Rani P.R is an assistant professor at Information Technology Department, ICET, Muvattupuzha, Kerala.