S^{2}ORM: Exploiting Syntactic and Semantic Information for Opinion Retrieval

Liqiang Guo, Xiaojun Wan
Institute of Computer Science and Technology & The MOE Key Laboratory of Computational Linguistics,
Peking University, Beijing 100871, China
{guoliqiang, wanxiaojun}@pku.edu.cn

ABSTRACT

Opinion retrieval is the task of finding documents that express an opinion about a given query. A key challenge in opinion retrieval is to capture the query-related opinion score of a document. Existing methods rely mainly on the proximity information between the opinion terms and the query terms to address the key challenge. In this study, we propose to incorporate the syntactic and semantic information of terms into a probabilistic language model in order to capture the query-related opinion score more accurately.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

Opinion Retrieval, Syntactic Tree, Semantic Relatedness

1. INTRODUCTION

Blog opinion retrieval is the task of finding blog posts that express an opinion about a given query. In most opinion retrieval systems, two scores are required to be computed for each blog post: the relevance score against the query and the opinion score about the query. The two scores are then combined and the blog posts are finally ranked according to the combined scores. The key challenge of opinion retrieval is how to capture the query-related opinion score. The simplest method is to compute the proportion of the opinion terms or sentences in a document (blog post) as the query-related opinion score, but it is not accurate because the process of calculating the opinion score is independent of the given query. This proximity-based method in [1] assumes that if the distance between an opinion term and a query term is smaller, the opinion term is more likely to be related to this query term. This assumption is not very appropriate, and it may be wrong in several cases. In this study, we propose to make use of the syntactic and semantic information of terms to capture the query-related opinion score more accurately. We propose to incorporate the syntactic and semantic information into a probabilistic opinion retrieval model (S^{2}ORM). The experiment results demonstrate the effectiveness of our S^{2}ORM model, which can outperform the state-of-the-art proximity-based model [1] and other baselines.

2. S^{2}ORM

2.1 Basic Model

In this study, our proposed S^{2}ORM model is based on the following generative model [1]:

$$p(d|o,q) \propto p(d,o,q) = p(d)p(q|d)p(o|q,d)$$

$$\overset{rank}{=} \log(p(d)p(q|d)) + \log\big(p(o|q,d)\big)$$

The first part $p(d)p(q|d)$ is the relevance probability of the document to the query. The second part $p(o|q,d)$ is the query-related opinion probability of the document. Moreover, we use a combination parameter to tune the two parts.

$$\overset{rank}{=} \mu \log(p(d)p(q|d)) + (1-\mu)\log\big(p(o|q,d)\big)$$

The first part can be directly obtained from the TREC baselines. The second part can be further refined by smoothing it with the opinion probability of the document as follows [1]:

$$p_s(o|q,d) = \lambda p(o|q,d) + (1-\lambda)p(o|d)$$

In this study, $p(o|q,d)$ is captured by making use of the syntactic and semantic information. With the assumption that all opinion terms appearing in the document are related to the query, the opinion probability of the document $p(o|d)$ is defined as follows: $p(o|d) = \sum_{t\in d} p(o|t)p(t|d)$

where $p(t|d) = \text{count}(t,d)/|d|$ denotes the relative frequency of term *t* in the document, and $p(o|t)$ is the opinion probability of the term *t*. Usually the opinion probability of a term can be obtained from an opinion lexicon.

2.2 Syntactic and Semantic Information Based Query-Related Opinion Probability

Formally, we regard a document as a sentence set, e.g., $d = \{s_{1,\ldots,}s_{i,\ldots,}s_m\}$. Each sentence can be regarded as a term set, e.g., $s_i = \{t_{1,\ldots,}t_{j,\ldots,}t_n\}$. So we have: $p(o|q,d) = \sum_{i=1}^{m} p(o,s_i|q,d)$, where $p(o,s_i|q,d)$ is the query-related opinion probability of sentence s_i and it can be estimated as: $p(o,s_i|q,d) = \sum_{j=1}^{n} p\big(o,t_j\big|q,d\big)$, where $p\big(o,t_j\big|q,d\big)$ denotes the query-related opinion probability of term t_j in sentence s_i and it can be estimated as follows:

$$p\big(o,t_j\big|q,d\big) = p\big(t_j\big|q,d\big)p\big(o\big|t_j,q,d\big)$$

Here we let $p\big(t_j\big|q,d\big) = p(t_j|d)$ with the assumption that t_j and q are conditionally independent given the document *d*. We assume here that each term appears in every position of all the sentences with equal probability, i.e., $p\big(t_j\big|d\big) = \frac{1}{|d|\times|V|} \propto \frac{1}{|d|}$. $|V|$ is the size of the term lexicon. And, $p\big(o\big|t_j,q,d\big)$ can be estimated as follows:

$$p(o|t_j,q,d)$$
$$= \max_{t\_noun \in s_i} p(o|t_j) p_{modi}(t_j, t\_noun)\text{SR}(t\_noun, q)$$

$p_{modi}(t_j, t\_noun)$ denotes the modifying probability between the term t_j (a potential opinion term) and the noun *t_noun*. SR(t_noun, q) denotes the semantic relatedness between the noun *t_noun* and the query *q*.

2.2.1 *Estimating* $p_{modi}(t_j, t\_noun)$

The modifying probability between an opinion term and a noun in a sentence can be estimated by using the information derived from the syntactic tree structure of the sentence with the tree kernel

Copyright is held by the author/owner(s).
WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

method. Thus, a tree kernel-based SVM classifier [3] (TK_SVM) is used to capture the modifying probability. The feature of TK_SVM is based on a Path-enclosed Tree. It is the smallest common sub-tree including two specified terms (e.g. an opinion term and a query term) in a syntactic tree. In order to train TK_SVM, we annotate some training data manually. Finally, we use the *sigmoid* function to normalize the ranking value into [0, 1].

$$p_{modi}(t_j, t\_noun) = sigmoid(\text{TK\_SVM}(t_j, t\_noun))$$

where $\text{TK\_SVM}(t_j, t\_noun)$ denotes the output value of TK_SVM for the feature tree structure of t_j and *t_noun*.

2.2.2 Estimating SR (t_noun, q)

The semantic relatedness is a score that reflects the semantic relationship between the meanings of two concepts. The semantic relatedness between a noun *t_noun* and a query *q* can be estimated as: $\text{SR}(t\_noun, q) = \text{avg}_{q_{i\in q}}\text{SR}(t\_noun, q_i)$.

The first popular approach is using WordNet for capturing the semantic relatedness between two terms.

1). WordNet-based Semantic Relatedness (**WN_SR**).

$$\text{SR}(t\_noun, q_i) = \begin{cases} \text{WN\_SR}(t\_noun, q_i) & t\_noun \in \text{Top } k \\ 0 & \text{else} \end{cases}$$

We rank all the nouns according to their semantic relatedness values in descending order, and *Top k* denotes the set of the first *k* nouns in the ranking list.

The second popular approach is making use of the probabilistic topic model (PTM) [4]. In PTM, the distribution of a term *t* can be represented as a vector. Thus, the semantic relatedness can be calculated by the Cosine metric (Cos), the Kullback Leibler divergence metric (KL), the Euclidian distance metric (Euc) or a Mark metric which is based on the conditional probability [4].

2).Probabilistic Topic Model-based Semantic Relatedness (**PTM_SR- *x***).

$$\text{SR}(t\_noun, q_i) = \begin{cases} \text{PTM\_SR} - x(t\_noun, q_i) & t\_noun \in \text{Top } k \\ 0 & \text{else} \end{cases}$$

The *x* can be replaced with ‘Cos’, ‘KL’, ‘Euc’ or ‘Mark’.

3. EVALUATION RESULTS

The BLOG06 collection is used for evaluation, and it includes 150 queries in TREC Blog Track 2006~2008.The data of 2006~2007 is used as training data and the data of 2008 is used as test data. In the preprocessing phase, the stop words and link tables [2] are removed. The general opinion lexicon used in [2] is used in this study. We follow the approach in [1] to normalize the relevance scores. For TREC baselines 2~5, we use the very simple linear method *N1* in [1]. *LRLR* in [1] is used on TREC baseline 1 due to the poor performance of *N1* on this baseline.

In Section 2.2.2, we propose two different approaches to capture the semantic relatedness between a noun and a query: WN_SR and PTM_SR-*x*. The MAP results of different semantic relatedness approaches over TREC baselines 1~5 are shown in Table 1. The low performance of WN_SR is attributed to some terms that are not included by WordNet, such as the query term *‘Carmax’*. So, all the following experiments adopt the PTM_SR-*x* approach.

Table 1: Performance of MAP over TREC baselines 1~5 using different semantic relatedness approaches with $\mathbf{k = 1000, \lambda = 1, \mu = 0.5}$

	b1	b2	b3	b4	b5
WN_SR	0.2781	0.2506	0.3637	0.3222	0.2906
PTM_SR-Mark	0.2841	0.2730	0.3743	0.3309	0.2893
PTM_SR-KL	0.2951	0.2797	0.3787	0.3354	0.2959
PTM_SR-Cos	**0.2965**	**0.2852**	**0.3824**	**0.3431**	**0.3056**
PTM_SR-Euc	0.2946	0.2785	0.3805	0.3368	0.2953

We compare our proposed S^2ORM model with the following models over the TREC baselines 1~5:

GORM: This model is based on the classical generative model described in section 2.1 and it adopts the generic query-related opinion probability method.

Laplace: This is a proximity-based model proposed by [1]. It uses the Laplace kernel function to capture the proximity information between the opinion terms and the query terms.

LaplaceInt: The model is also proposed by [1], which achieves the state-of-the-art performance via smoothing the Laplace model with the document’s opinion probability.

The MAP results of Laplace and LaplaceInt shown in Table 2 are also published in [1] with its best parameter values. The MAP results of S^2ORM are under $\lambda = 0.4\sim0.5$, $\mu = 0.5\sim0.8$ and $k = 1000\sim5000$.We find that S^2ORM achieves the best performance over TREC baselines 2~5. On average, S^2ORM gets 2.62% increase in MAP over the LaplaceInt model. In particular, our S^2ORM model performs very well over TREC baselines 2 and 3. The performance is not very high on TREC baseline 1, because the file of TREC baseline 1 is not complete.

Table 2: The MAP performance of different models over TREC baselines 1~5. NoRerank is a baseline without using any re-ranking model.

	b1	b2	b3	b4	b5	avg	ΔMAP
NoRerank	0.3239	0.2639	0.3564	0.3822	0.2988	0.3250	
Laplace	0.3960	0.2881	0.3989	0.4267	0.3188	0.3657	12.52%
LaplaceInt	**0.4020**	0.2886	0.4043	0.4292	0.3223	0.3693	13.63%
GORM	0.3340	0.2811	0.3964	0.3571	0.3149	0.3367	3.60%
S^2ORM	0.3975	**0.3061**	**0.4223**	**0.4300**	**0.3332**	**0.3778**	**16.25%**

4. ACKNOWLEDGMENTS

The work was supported by NSFC (61170166), Beijing Nova Program (2008B03), NCET (NCET-08-0006) and National High-Tech R&D Program (2012AA011101).

5. REFERENCES

[1] S. Gerani , M. J. Carman, and F. Crestani. Proximity-Based Opinion Retrieval. In Proceedings of SIGIR '10, 2010.

[2] L. Guo, F. Zhai, Y. Shao and X. Wan. PKUTM at TREC 2010 Blog Track. In Proceedings of TREC'10, 2010.

[3] A. Moschitti. Making tree kernels practical for natural language learning. In Proceedings of EACL’06, 2006.

[4] M. Steyvers and T. Griffiths. Probabilistic Topic Models. In Landauer, T., McNamara, D., Dennis, S., Kintsch, W., Latent Semantic Analysis: A Road to Meaning. 2006.