A Formal Product Search Model with Ensembled Proximity

Zepeng Fang[1], Chen Lin[1,2,*], Yun Liang[3]

[1] Department of Computer Science, Xiamen University, Xiamen, China, 361005
zpfang@stu.xmu.edu.cn,
[2] Shenzhen Research Institute of Xiamen University, Shenzhen, China, 518057
chenlin@xmu.edu.cn,
[3] College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China, 510642
sdliangyun@163.com

Abstract. In this paper we study the problem of product search, where products are retrieved and ranked based on how their reviews match the query. Current product search systems suffer from the incapability to measure correspondence between a product feature and its desired property. A proximity language model is presented to embed textual adjacency in the frequency based estimation framework. To tailor for product search problem, we explore strategies for distinguishing product feature and desired property, quantifying pair-wise proximity based on conditional probability, and aggregating review opinions at product level. Experiments on a real data set demonstrate good performances of our model.

Keywords: Information retrieval · Proximity model · Product search

1 Introduction

Recently, there has been increasing interest in the problem of product search, due to the abundance of online reviews. Product review sites, such as TripAdvisor[1], Yelp[2], have attracted numerous users, and thus have generated an incredible volume of comments. Unfortunately, it is impossible for users to absorb all the information for every candidate product. Product search is then considered to be a prominent tool to explore online reviews and to make smart consumptions.

Product search queries usually consists of consumption preferences on multiple product features. The goal of product search engine is to locate the right product reviews, and rank them based on how they meet people's demands. For example, the query in Fig.1 seeks for a restaurant with nice decor that serves hot pot. Product B is relevant, since the second review is a supporting evidence which explicitly states that the two features of restaurant satisfy the query need.

[1] http://www.tripadvisor.com/
[2] http://www.yelp.com/

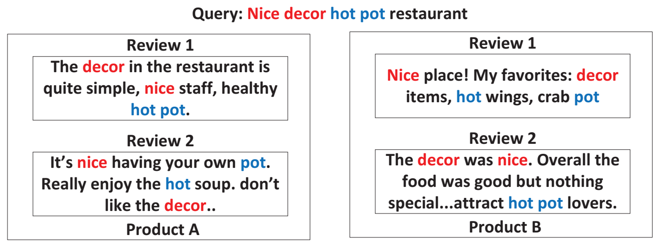


Fig. 1. An illustration of a product search problem

To retrieve preferred products from online reviews, several approaches [1, 2] have been proposed, most of which are based upon a probabilistic model that measures query-review relevance. They consider a review as supporting evidence, if all the query keywords appear in the review. This type of models is limited, as it treats the query as a plain, unstructured "bag of words", and does not distinguish the pair-wise correspondence between preferences and features. As illustrated in Fig. 1, the first review of product A contains all query keywords. Nevertheless, it is not a supporting evidence. Because the preferred opinion "nice" does not correspond to "decor", instead it is used to describe the feature "staff".

The key issue in product search systems is to quantify the relevance between the desired property and the corresponding feature in the reviews. One may easily recognize that relevance is reflected by textual adjacency. Information Retrieval (IR) models incorporating term proximity like PLM [3] and BM25P [4] have been shown to significantly enhance the performance of IR systems. However, directly applying them for product search is problematic. On one hand, these models hold a constraint of closeness to all query terms, which might be too strict to harm the accuracy of product ranking. For example, in Fig. 1, the average proximity scores for all query terms in review 2 for product B is in fact the largest of four reviews. But since both the preferred opinion keywords are close to the feature keywords ("nice" to "decor", "hot" to "pot"), this review is a supporting evidence. On the other hand, given a pair of product feature and preference, traditional IR models are only capable of capturing their association at document level, while product search is implemented at entity level. Therefore, it is necessary to quantify the overall degree of association.

In this paper, we present a formal model to address the above two problems. Following the classic framework of language modeling, we compute the likelihood of observing the query for each product. The query likelihood is factorized to conditional probability of preferred opinion given the product feature. We study the estimation of conditional probability and present several strategies to embed proximity in estimation. Furthermore, we study the effect of aggregated proximity from the review corpus.

2 Related Work

The great potential of entity search [5] has been acknowledged in recent years. Typical entity search paradigms include expert search [6], query driven product retrieval [1, 2, 7]. Product retrieval are either built upon a keyword search framework [2], or a probabilistic language model framework [1]. Sources for product search are mainly product profiles and online reviews. When retrieving product from reviews, a critical property is that reviews are opinionated on different product aspects, and thus demands special treatment. In [1], relevance is evaluated by aggregating over all query aspects and achieve a good effect. To measure relevance of product aspect and opinion, a new indexing unit, Maximal Coherent Semantic Unit is defined and employed in the ranking process [7].

Language modeling approaches have been extensively studied in IR community, e.g. query likelihood, divergence and relevance and so on [8]. Beyond general frameworks for computing unigram relevance, one may also want to reward documents in which query terms appear close to each other. To exploit such proximity heuristics, some researchers attempt to capture word dependency by utilizing a larger matching unit, e.g. bigram [9]. To avoid making the indexing space too sparse, Markov Random Field model [10] is presented to collectively score unigram, bigram and textual unit within a certain window. Other researchers incorporate query term proximity into an existing retrieval model either directly or indirectly. Directly applying term proximity usually involves defining a combination of relevance score from existing retrieval models and adjusting scores from proximity heuristics [11]. Indirect methods embed proximity measures and term frequencies in a unified model. In [3], a language model for each position of a document that takes into account propagation of word count from other places. In [4], it extends the well-established BM25 [12] model by taking a linear combination of Ngram proximity based BM25 models for different N. This line of researches also include CRTER [13], which introduces a pseudo term that is the combination of the individual terms, and is weighted by the intersection of impact propagated from each individual at different positions.

3 Model

Let $D = \{d_1, d_2, \cdots, d_N\}$ be the product universe and $C = \{R_d\}$ is the collection of all reviews, where R_d is the set of review documents for product d. The query consists of several preference phrases on multiple product features, $q = \{(o, f)\}$ in which o denotes the preferred opinion terms and f represents the corresponding feature keywords. Our goal is to estimate the likelihood of generating query from the hidden product model.

$$p(q|d) = \Pi_{(f,o)} p(f, o|d) = \Pi_{(f,o)} p(f|d)p(o|f, d) \tag{1}$$

The first part $p(f|d)$ is the probability of selecting feature f from the product, which can be estimated by Dirichlet Prior smoothing with parameter μ.

The second part $p(o|f,d)$ defines the relevance between an opinion o and a feature f in d's reviews. In the next subsection, we will elaborate how to incorporate term proximity between opinion and feature keywords into its estimation.

3.1 Conditional Probability Estimation

Proximity Parameterized The first strategy **PP** is to represent $p(o|f,d)$ as the probability density function with respect to term proximity $d(o,f,R_d)$. We assume that given the feature f, the author will select opinion o according to a Gaussian distribution $p(o|f,d) \sim N(0,\sigma^2)$. Note that the probability for a Gaussian achieves its maximum at its mean, and decreases as the value is distant from the the mean. Therefore, if the distance between opinion term o and the feature word f is smallest, then we will get the maximum $p(o|f,d)$. The above observations lead to the following functional form

$$p(o|f,d) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{d(o,f,R_d)^2}{2\sigma^2}) \tag{2}$$

Proximity Adjusted Another strategy **PA** is to first compute the probability $p(o|f,d)$, then modify it by the proximity. With Jelinek-Merccer smoothing, we have $p(o|f,d) = (1-\lambda)\frac{c(o,f,R_d)}{c(f,R_d)} + \lambda\frac{c(o,f,C)}{c(f,C)}$, where λ is the parameter.

However, the above definition depends only on the co-occurrences, thus ignores the impact of proximity. We employ an exponential weighting scheme to simulate the negative correlation between terms proximity and conditional probability, so that the confidence of a frequency-based estimation $c(o,f,R_d)$ decreases as the absolute distance increases. In order to guarantee the adjusted function is a probability, i.e. $\Sigma_o p(o|f,d) = 1$, the dominator $c(f,R_d)$ should be regularized accordingly. Note that, $\int_{-\infty}^{+\infty} \exp -x^2 dx = \sqrt{\pi}$. Therefore, we have

$$p(o|f,d) = (1-\lambda)\frac{c(o,f,R_d)\exp\left(-d(o,f,R_d)^2\right)}{c(f,R_d)\sqrt{\pi}} + \lambda\frac{c(o,f,C)}{c(f,C)} \tag{3}$$

Proximity Censored Finally we consider probability estimation by directly manipulating the event space. As in the **PA** strategy, $p(o|f,d)$ is proportional to the co-occurrence frequency of the opinion term o and feature keyword f. But such a relatedness is actually invalid if the two words are far away. Therefore we define the event of observing terms o,f in a text window of size ϵ as $c_\epsilon(o,f,R_d)$. The probability, according to strategy **PC**, is defined as

$$p(o|f,d) = \frac{c_\epsilon(o,f,R_d)}{c(f,R_d)} \tag{4}$$

where $c_\epsilon(o,f,R_d) = |\{d(o,f,R_d) < \epsilon\}|$, and obviously $\Sigma_o p(o|f,d) = 1$

3.2 Proximity Aggregation

For a document, the proximity $d(o,f)$ is the absolute difference of the positions of terms o and f. Because both terms o,f might appear at multiple positions in the product reviews R_d, we need to study the aggregation of term proximity.

Min strategy returns the minimal proximity between the opinion and feature in a product. Intuitively, the **Min** strategy suggests that the most significant evidence is adopted, i.e. if only one consumer gives positive feedback, the product will be regarded as relevant.

Avg strategy measures average proximity of two terms in the product reviews. This strategy assumes that all reviews matter, i.e. the product is deemed relevant when the overall feedbacks are good.

Max strategy calculates the maximal proximity between an opinion and its neighboring feature in the product specific reviews. Intuitively, the **Max** strategy only considers the weakest evidence, i.e. the product is relevant if the most critical consumer speaks highly of it.

The above three heuristics are simple and intuitive. The **Min** strategy and **Max** strategy are based on single evidence instead of the collective opinions. The **Avg** strategy is a global measurement, but it is still sensitive to outliers. A typical problem for mining social documents is that it usually involves diverse social behaviors. In the setting of product search, reviews for a specific product generally contain quite distinct or even opposite opinions. As a result, we may need a more accurate measurement to reflect the collected opinions on the required product features. Thus, we present a **ClusterMin** strategy as follows.

First we represent each review $r \in R_d$ as multiple $V-$dim vectors r^o, each for a product feature in the query $o \in q$. The $v-$th element of the feature specific review vector is the minimal distance $r_v^o = \min_{v \in r} d(o, v)$ between the given product feature o and the $v-$th word of the opinion lexicon in the review. We adopt K-means algorithm to cluster all reviews $r \in R_d$ for product d. The centroid of each cluster is represented by multiple feature specific vectors. In the assigning step, calculate the distance between a review and a centroid s^k in cluster k by the combination of feature specific Euclidean distance $\|r - s^k\|_2 = \Sigma_o \|r^o - s^{o,k}\|_2$. As opinions are naturally divided into three categories: positive, neutral and negative, we set $K = 3$ for clustering. When the clustering converges, we choose the cluster centroid which has the nearest query specific feature-opinion distance $s^i = s^k | \min_{k=1 \to 3} \Sigma_{<o,f> \in q} s_f^{o,k}$, and set the aggregated distance as the corresponding element defined by the cluster centroid $d(o, f) = s_f^{o,i}$.

4 Experiment

We evaluate our model with the open benchmark [1]. The data set consists of reviews for hotels in different cities and cars of various years. For the purpose of this study, only queries which contain both opinions and associated product features are remained. Statistics of experimental data set is shown as Tab. 1.

The proposed retrieval model is implemented on the Terrier[14] platform. To enhance efficiency, reviews for a single product are merged to a unified document unit. Term distance across reviews is assigned a large value 400. In preprocessing, stop words are not removed, and porter stemmer is adopted.

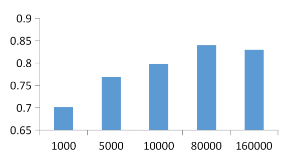
We adopt NDCG@10 (Normalized Discounted Cumulative Gain) as evaluation metric in the following experiments.

Table 1. Statistics of data set

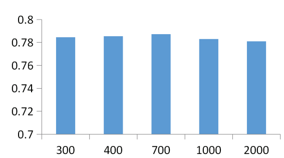
Hotels		Cars	
No. Cities	5	No. Years	3
Avg. No. Hotels	143.2	Avg. No. Cars	199.3
Avg. No. reviews per hotel	60	Avg. No. reviews per car	67.7
Avg. document length	1219.4	Avg. document length	1097.3
Avg. No. Queries	5	Avg. No. Queries	5

4.1 Conditional Probability Strategy

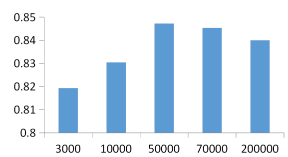
We first tune the smoothing parameter μ for each strategy. The proximity aggregation strategy is fixed to be **Min**. As shown in Fig.2(a),2(b),2(c), the smoothing parameter μ for good performance tends to be large. A larger μ indicates that the feature probability relies more on the global feature probability $p(f|C)$. It is reasonable since the product specific reviews are associated with a limited number of features. Also, we observe that the effect of decaying confidence in the **PA** significantly shrinks the smoothing parameter μ as shown in Fig.2(b).



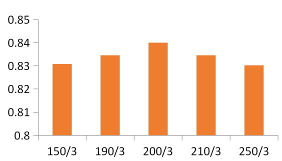
(a) tune μ with $\sigma = 200/3$



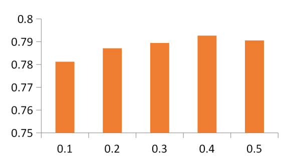
(b) tune μ with $\lambda = 0.2$



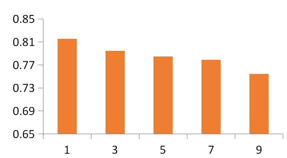
(c) tune μ with $\epsilon = 1$



(d) tune σ with $\mu = 80000$



(e) tune λ with $\mu = 700$



(f) tune ϵ with $\mu = 1000$

Fig. 2. Parameter tuning for **PP**, **PA** and **PC** strategy

Next we report the effect of strategy specific parameters in Fig. 2(d),2(e),2(f). Best performance for **PP** is achieved when σ is around 200/3. For a Gaussian distribution with zero mean, 99.7% of the data points are in the range of $[-3\sigma, +3\sigma]$, which is in line with our assumption that each product feature should at least appear in the same passage with the corresponding opinion (passage length = 400 as mentioned above). Best performance for **PA** is obtained when $\lambda = 0.4$, which suggests the confidence plays a more significant effect than the global statistic. For **PC** strategy , $\epsilon = 1$ performs best, which means when an opinion is directly adjacent to it associating feature, it is the most relevant.

4.2 Proximity Aggregation Strategy

We next study the performance of four aggregation strategies. The comparison is carried in the hotel data set, Beijing category. Parameters $\mu = 80000, \sigma = 200/3$ for **PP**, $\mu = 700, \lambda = 0.2$ for **PA**. As shown in Tab.2, **Min** and **ClusterMin** both achieve satisfying results, which verifies the most significant evidence in the reviews play a great role in the calculation of relevance.

Table 2. The performance of four aggregation strategies

	Min	Avg	Max	ClusterMin
PP	0.8400	0.5201	0.5285	0.9086
PA	0.7871	0.4803	0.5274	0.7514

4.3 Comparative Study

We finally analyze the performances of our work, compared with state-of-the-art systems including (1) traditional IR models BM25 [12], PL2 [14]; (2) positional models BM25P [4], CRTER [13], MRF [10] and PLM [3]. These models are not dedicated to entity search scenarios, thus we first rank the reviews, and then re-rank the products by counting the number of reviews for each product in the top 100 search results; (3) product search framework, i.e. OpinionRank [1]. Parameters $\mu = 80000, \sigma = 200/3$ for **PP**, $\mu = 700, \lambda = 0.4$ for **PA**, $\mu = 50000, \epsilon = 1$ for **PC**. And we use **Min** aggregation for all strategies. From Tab. 3, we have the following conclusions. (1) In general, positional models outperform traditional IR models, which highlight the importance of proximity constraints in IR system. (2) A unified model designed for product search performs significantly better than the re-ranking scheme based retrieval. (3) Our models are comparable to OpinionRank. Among the three paradigms, **PP** is most stable and generally obtains best results, which verifies the competency of our contribution.

Table 3. Performance of various product search systems

models	BM25	PL2	rrBM25P	rrCRTER	rrMRF	rrPLM	OpinionRank	PP	PA	PC
hotels										
beijing	0.5179	0.5234	0.7753	0.7620	0.7685	0.8276	**0.8521**	0.8346	0.7927	0.8472
dubai	0.6160	0.6228	0.6450	0.7066	0.6400	0.8106	0.8401	**0.8579**	0.7149	0.8246
new-delhi	0.4323	0.4360	0.6319	0.6532	0.6576	0.7462	**0.8130**	0.8045	0.6820	0.7345
san-francisco	0.4551	0.4619	0.7839	0.6532	0.7603	0.8227	0.8130	**0.8702**	0.8328	0.8274
shanghai	0.5097	0.5206	0.7249	0.6865	0.7603	0.8178	0.8239	**0.8276**	0.7460	0.7849
Average	0.5062	0.5129	0.7122	0.6923	0.7173	0.8050	0.8284	**0.8389**	0.7537	0.8037
cars										
2007	0.8908	0.8900	0.9133	0.9259	0.9152	0.9349	**0.9458**	0.9443	0.9369	0.9198
2008	0.8781	0.8788	0.9174	0.9167	0.9257	0.9308	0.9347	**0.9376**	0.9248	0.9179
2009	0.9176	0.9163	0.9129	0.9256	0.9257	0.9186	0.9494	**0.9526**	0.9430	0.9320
Average	0.8955	0.8950	0.9145	0.9227	0.9222	0.9281	0.9429	**0.9453**	0.9349	0.9233

5 Conclusion

In this paper we present a positional language model for product search problems. Our contributions are two-fold: (1) we incorporate pairwise proximity into the estimation of conditional probability of generating an opinion given a product feature; (2) we explore the aggregation strategies to ensemble review evidences to evaluate the relevance of a product. Experiments on real data set verify the competence of the presented framework. In the future, we plan to extend the model to tolerate noisy query segmentation. Also, clustering on the fly is a potential direction to speed up the computation for **ClusterMin** aggregation.

Acknowledgment. Chen Lin is partially supported by China Natural Science Foundation under Grant No. NSFC61472335, Base Research Project of Shenzhen Bureau of Science,Technology, and Information under Grand No. JCYJ20120618155655087, Baidu Open Research Grant No.Z153283. Yun Liang is supported by the Science and Technology Planning Project of Guangdong Province (2016A050502050), and the NSFC-Guangdong Joint Fund (U1301253).

References

1. Ganesan, K., Zhai, C.: Opinion-based entity ranking, Information retrieval. 15 (2), 116–150(2012)
2. Duan, H., Zhai, C., Cheng, J., et al.: Supporting keyword search in product database: A probabilistic approach. Proc. VLDB Endow. 6 (14), 1786–1797(2013)
3. Lv, Y., Zhai, C.: Positional language models for information retrieval. In: SIGIR, pp. 299–306(2009)
4. He, B., Huang, J.X., Zhou, X.: Modeling term proximity for probabilistic information retrieval models. Information Sciences. 8 (14), 3017–3031(2011)
5. Yao, T., Liu, Y., Ngo, C.W., et al.: Unified entity search in social media community. In: WWW, pp. 1457–1466(2013)
6. Balog, K., Azzopardi, L., de Rijke, M.: Formal models for expert finding in enterprise corpora. In: SIGIR, pp. 43–50(2006)
7. Choi, J., Kim, D., Kim, S., et al.: Consento: A new framework for opinion based entity search and summarization. In: CIKM, pp. 1935–1939(2012)
8. Zhai, C.: Statistical language models for information retrieval a critical review. Foundations and Trends in Information Retrieval. 2 (3), 137–213(2008)
9. Srikanth, M., Srihari, R.: Incorporating query term dependencies in language models for document retrieval. In: SIGIR, pp. 405–406(2003)
10. Metzler, D., Croft, W.B.: A markov random field model for term dependencies. In: SIGIR, pp. 472–479(2005)
11. Büttcher, S., Clarke, C.L.A., Lushman, B.: Term proximity scoring for ad-hoc retrieval on very large text collections. In: SIGIR, pp. 621–622(2006)
12. Robertson, S.E., Walker, S., Hancock-Beaulieu, M., et al.: Okapi at TREC-4. NIST SPECIAL PUBLICATION SP. 73-96(1996)
13. Zhao, J., Huang, J.X., Ye, Z.: Modeling term associations for probabilistic information retrieval. ACM Trans. Inf. Syst. 32 (2),7:1–7:47(2014)
14. Ounis, I., Amati, G., Plachouras, V., et al.: Terrier: A High Performance and Scalable Information Retrieval Platform. In: SIGIR Workshop on Open Source Information Retrieval. (2006)