# Mining New Business Opportunities: Identifying Trend related Products by Leveraging Commercial Intents from Microblogs

**Jinpeng Wang**[1], **Wayne Xin Zhao**[1], **Haitian Wei**[2], **Hongfei Yan**[1] **and Xiaoming Li**[1]
[1]School of Electronic Engineering and Computer Science, Peking University, China
[2]Daton Securities Co., Ltd., No.93 Jianguo Rd, Chaoyang District, Beijing, China
JooPoo@pku.edu.cn, {batmanfly,haataa.wei,yhf1029}@gmail.com, lxm@pku.edu.cn

## Abstract

Hot trends are likely to bring new business opportunities. For example, "Air Pollution" might lead to a significant increase of the sales of related products, e.g., mouth mask. For e-commerce companies, it is very important to make rapid and correct response to these hot trends in order to improve product sales. In this paper, we take the initiative to study the task of how to identify trend related products. The major novelty of our work is that we automatically learn commercial intents revealed from microblogs. We carefully construct a data collection for this task and present quite a few insightful findings. In order to solve this problem, we further propose a graph based method, which jointly models relevance and associativity. We perform extensive experiments and the results showed that our methods are very effective.

## 1 Introduction

A trend is a hot topic (e.g., the release of a popular movie) which is being widely discussed by the public. Hot trends usually attract much attention from the public, and they are likely to bring new business opportunities. Consider the following e-shopping scenario. A user in Beijing would like to buy something to reduce the health impacts of Beijing air pollution[1]. Different from traditional e-shopping stories, in this case the user may not have a clear idea of what she should buy, and cannot even formulate the purchase needs into a clear query. Faced with trend-driven business opportunities, e-commerce companies typically ask workers to manually identify related products and make heuristic rules to match user queries (e.g., incorporating trending keywords into related product titles).

To improve trend-driven e-commerce, in this paper, we propose to study the novel task of *automatically identifying trend related products*. Why is it compelling to understand and study trend-driven product purchase? Because hot trends are closely related to business opportunities directly or indirectly. As a case of direct causal relationship, the worldwide popularity of the movie series "Harry Potter" created the great success of the original novels of "Harry Potter". As a case of indirect causal relationship, the stock rise or salary increase might exert positive effects on product sale. Based on our empirical analysis (See Section 3), a considerable proportion, i.e. 50%, of hot trends discussed on the largest Chinese microblog (i.e. Sina Weibo) indeed have corresponding product entries in the largest Chinese *C2C* e-commerce website (i.e. Taobao), which indicates a strong correlation between hot trends and product sale.

Although the task is important and emergent, it has at least two major challenges. First of all, how to infer users' trend-driven purchase intents promptly. A trend usually happens unexpectedly. Without prior knowledge and experiences, it is particularly difficult to make rapid response to relate the trend to candidate products. Our solution is to *leverage trend-related commercial intents from microblogs* by mining users' real-time response to a trending topic. We adopt the solution based on two key considerations: (1) Microblogs are fast. As previous studies showed, the first story of a trending topic in-

[1]See http://www.nytimes.com/2013/04/04/world/asia/two-major-air-pollutants-increase-in-china.html to find more details about the trending topic "Beijing Air Pollution".

deed was usually reported in microblogs rather than traditional news media (Sakaki et al., 2010; Kwak et al., 2010; Leskovec et al., 2009). (2) Microblogs contain users' commercial intents. The microblogging service has become one of the most popular social network platforms, where users may tweet about their needs and desires (Hollerit et al., 2013). E.g., a microblog user may complain about the air quality and evince the desire to buy a mouth mask in a tweet. The example indicates we can make use of tweet-level relatedness to capture the correlation between *trends* and *products*.

Second, how to achieve a comprehensive coverage of related products but not hurting precision. The above solution will miss the related products which have not been discussed in microblogs. Our idea is to take the associativity between products into consideration. Our definition about associativity is very general and can have different instantiations in specific settings. For example, we can define product associativity to be the similarity between product descriptions, or the ratio of historical purchase records in e-commerce companies. However, one-step associativity may not fully discover the underlying relatedness between products due to the fact that the product associativity is indeed transitive. Thus, a transitable associativity model is needed.

To address these two challenges, we propose a unified graph based ranking algorithm which jointly models the above two aspects, i.e., relevance and associativity. Given a trend, the algorithm runs in an iterative way and seeks a trade-off between relevance and associativity by propagating the scores on the product graph. Our contribution can be summarized as follows: (1) we introduce the novel task of identifying trend related products, most of all, we propose to leverage trend-related commercial intents from microblogs; (2) we present insightful empirical analysis to illustrate the correlation between hot trends and product sale (See Section 3); (3) we propose a novel graph based ranking algorithm which jointly considers relevance and associativity; (4) we carefully construct the test collection based on real data of the largest microblog and the largest *C2C* e-commerce website in China. (5) we perform extensive experiments and present some important implications for practice.

To the best of our knowledge, our work was the first to consider identifying trend related products by leveraging commercial intents from microblogs. We believe the current work will have important impact on industry and inspire more follow-up research studies. The rest of this paper is organized as follows. We present the data collection and empirical analysis of the impact of hot trends on product sale in Section 3. We present a novel graph-based method in Section 4. Experimental setup and results are discussed in Section 5 and Section 6. Finally, the related work is discussed in Section 7. And the conclusions and future work are given in Section 8.

## 2 Problem Definition

A *trend* is a hot topic widely discussed by the public, e.g., the release of a hot movie. Usually, a trend $e$ can be described by a small set of keywords denoted by $\mathcal{K}_e$ and a corresponding time span $\mathcal{T}_e$.

**Trend-related Products Identification**: Given a trend $e$, we assume that the following inputs are available: 1) tweets that contain trend keywords $\mathcal{K}_e$ and 2) a product database which provides a set of candidate products $\mathcal{P} = \{p_1, p_2, ..., p_n\}$ with necessary detailed information, e.g., titles and descriptions. The objective of trend-related products identification is to identify products in $\mathcal{P}$ that are related to trend $e$ within the time span $\mathcal{T}_e$, denoted by $\mathcal{P}_R$. For convenience, we will not explicitly mention the time span unless needed.

To better understand the problem, we first present an illustrative example in Table 1, which will be discussed as the running case throughout the paper. In this example, we can see that a few users tweet their product needs related to the trend "Air Pollution". We take Taobao as the product database and present a few related products in it.

Table 1: An illustrative example for the studied task.

| |
|---|
| **Trend keywords**: Air Pollution |
| **Tweets**: |
| What bad air! We need to buy masks ASAP!!! |
| I am planing to buy an air purifier to defend air pollution. |
| #air_pollution I recommend to keep some houseplants at home. |
| **Product database**: Taobao[2] |
| **Related products**: Mouth Mask, Air Purifier, Houseplant |

[2]The biggest C2C e-commerce site in China, similar to eBay.

## 3 Data and Observations

As discussed earlier, hot trends may exert positive effects on the sale of related products. In this section, we will construct a deep analysis on this point by presenting quantitative answers to the following two problems:

- $Q_1$: What is the proportion of hot trends that potentially lead to business opportunities, and how is their impact on related products?
- $Q_2$: How is the associativity between related products?

These findings are key and fundamental to develop our models.

### 3.1 Data Collection

To perform the above analysis, the key is how to construct an experimental data collection which relates hot trends to corresponding related products. We jointly consider microblogs and e-commerce platforms: we obtain hot trends in microblogs and manually identify trend-related products in e-commerce websites. In this paper, we adopt Sina Weibo[3] as the microbloging platform and Taobao[4] as the e-commerce platform, which are the biggest microblogging service and the largest C2C company in China respectively. The analysis method is general and can equally apply to other platforms. For both two data signals, we consider a two-month time span, i.e. from May 2013 to June 2013.

*Trend detection.* Since trend detection is not our focus, we directly obtained trends from "trending topics" provided by the microblog platform. Our work can be easily extended to incorporate a trend detection component. Similar to "trending topics" in Twitter, Sina Weibo provides a public list of top searched keywords which can be obtained by the Weibo search API[5]. In the list, top 50 keywords are presented and ordered by the number of being searched. Weibo classifies these keywords into five categories: *China*, *movie*, *business*, *person* and *sports*. We consider these keywords to be trend keywords. These keywords are dynamically updated and we monitor the trend lists in the considered time span. We define the start and end time of a trend to be the first day and the last day on the trend list respectively, which spans the active interval of a trend. We only keep the trend which has an active interval with more than one day. For each trend, we use the trend keywords to retrieve all related tweets in the active interval, and use the pattern based method in (Hollerit et al., 2013) to extract all mentioned product keywords. We present a few example patterns used for extracting product keywords in Table 2. After that we can obtain a set of product keywords for each trend.

Table 2: Example patterns for extracting product keywords.

| Patterns | Example segments of tweets |
|---|---|
| 买*(buy)* | 买了飞利浦剃须刀送父亲<br>*bought father a Philips PT720 (Electric Razor)* |
| 使用*(use)* | 使用N95口罩降低污染<br>*use N95 (mouth mask) to reduce impact of bad air* |
| 推荐<br>*(recommend)* | 推荐Galaxy S4<br>*recommend Galaxy S4 (cell phone)* |

*Related product identification and annotation.* For each trend, we have the product keyword set together with the trend keywords as described above. We use these keywords to retrieve candidate products in the product search engine of Taobao within the active interval of the trend. For each candidate product, we further crawl its product page and obtain corresponding related products suggested by Taobao, which are treated as candidate, too. We invite two senior post-graduate students major in economics as human judges. The judge is required to make a binary decision whether a product is related to a trend by following a detailed guideline compiled by a senior officer of an e-commerce company in Beijing. For each trend, we provide the trend keywords, product keywords in tweets, related tweets, related news articles from China Daily[6]. Web access is available during the annotation process. Due to space limit, we do not present the annotation guideline here. We use Cohen's kappa to measure the agreement of these two judges, which has a high value of 0.75. To speed up the work, we further group all products which have the same

[3] http://www.weibo.com/

[4] http://www.taobao.com/

[5] http://s.weibo.com/top/summary

[6] http://www.chinadaily.com.cn

lowest categorial label (e.g., leaf label) [7], and we will treat a group as a product in later experiments. We only keep the products with the same judgments and the trends with at least one related product. We present the statistics of data set in Table 3 [8]. Since current e-commerce search engines mainly adopt keyword matching based retrieval method, we further examine the performance of simply using trend keywords as queries. We compute the percentage of related/unrelated products with at least one trend keyword in their description. We can see that only 29.7% related products can be found on average. These statistics indicate that more effective methods are needed for the current task.

Table 3: Statitics of the data set.

| | |
|---|---|
| # business-related trends | 113 |
| average candidate products per trend | 55.1 |
| average related products per trend | 7.3 |
| average perc. of rel. prod. with trend keywords | 29.7% |
| average perc. of unrel. prod. with trend keywords | 6.3% |

### 3.2 Observations

Now we analyze the data collection and present our observations.

$A_1$: First of all, it is important to find out the proportion of hot trends that potentially leads to business opportunities. Recall that each trend has a category label and possibly a set of related products identified by the judges. We refer to a trend with related products as a business-related trend. We present the statistics in Fig. 1. We can see that about 36% of all trends have corresponding related products in Taobao, which indicates that these trends highly relate to business. *Movies* and *Sports* have higher proportions of business-related trends, i.e. 81% and 52% respectively, while the other categories have lower proportions but still with a substantial number of business related trends. It is noteworthy that *Business* has the lowest proportion, the major reason is that trends in *Business* are usually general events, i.e., the release of new economic policy, which do not directly correspond to related products. As we discussed earlier, these trends may have indirect impact on product sales. Currently, we only focus on direct impact, and indirect impacts will be considered in future work.

Next we continue to examine the impact of hot trends on the sale of related products. We obtain product sales from Taobao product pages. As we can see in Fig. 2, the average sale of related products in all categories gradually increased with trends going on. Interestingly, we can see that categories *Movies* and *China* achieved very significant increase. Products related to *Movies* trends are usually related to the movie itself, e.g., movie tickets; while products related to *China* tend to be commodities (e.g., the mouth masks for the trend of "Air Pollution") or trending products (e.g., Shenzhou-10 Spacecraft Model for the trend of "the launch of Shenzhou-10").

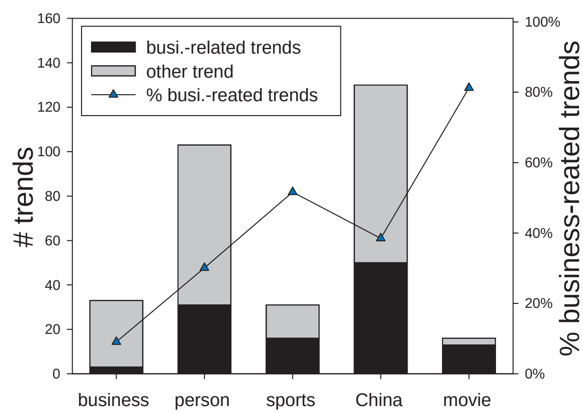



Figure 1: The proportion and volume of business-related trends in five categories.

$A_2$: Recall we have discussed that product associativity is useful for improving the coverage of related products. Here we would like to quantitatively examine the associativity between related products given a trend. For a trend, we first compute the average pairwise similarity between *related* products in terms of their descriptive texts (e.g. title and description). Since there are more unrelated products, we randomly sample an equal number of unrelated products from the candidate products we previously generated. Then we compute the average similarity between a related product and an unrelated product. We further average these values

[7]Taobao has provided a category tree for products: http://list.taobao.com/browse/cat-0.htm).

[8]The data set can be downloaded at `http://sewm.pku.edu.cn/~wjp`.

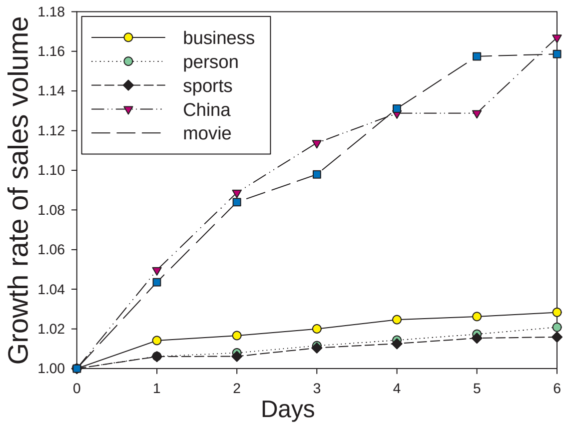



Figure 2: An illustrative analysis of the impact on related products in five categories. We measure the impact by computing the average growth ratio of sale in Taobao.

over all the trends of each category. The average similarity of related-related product pairs is 0.112, while the average similarities of unrelated-unrelated and related-unrelated product pairs are 0.039 and 0.058 respectively.[9]

In summary, $A_1$ indicates that a large proportion of hot trends are potentially related to products and will exert positive effects on product sale; $A_2$ indicates that there is a strong associativity between related products, which can be utilized to improve both precision and recall of the algorithm.

## 4 The Proposed Method

In this section, we present a graph based ranking algorithm jointly models the relevance of a product and the associativity between products. Recall that we have collected a set of product keywords and a set of candidate products for each trend. Our aim is to re-rank these candidate products to obtain a better ranking of related products. We adopt a biased random walk algorithm: 1) relevance is modeled as biased restart probability and 2) associativity is modeled through random walk on the product graph.

### 4.1 Modeling the Product Relevance

Recall that in Section 2 we use the pattern based method to extract product keywords from tweets related to a trend. However, to stimulate the real scenario that we want to identify the related products at the beginning of a trend, we only keep the keywords which were contained in tweets published in the first three days when a trend began. These extracted product keywords directly reveal users' commercial intents on the trends. Instead of modeling personalized intents, we consider learning a unified trend-driven intent by representing the intent as a weighted vector over these product keywords. And the key is how to set the keyword weight.

**Keyword weighting.** A good weighting method should be able to leverage commercial interests/intents of users well and emphasize the keywords users really focus on. Thus we consider making use of the *retweeting* (a.k.a. *forwarding*) mechanism in microblogs. Retweet links are shown to be better in revealing relevance and interests (Welch et al., 2011). Formally, we use the following weighting formula for a keyword $k$:

$$\text{Weight}(k) = \sum_{t \in \mathcal{C}_k} \log_{10}(\#rt_t + 1), \quad (1)$$

where $\mathcal{C}_k$ is the set of all originally-written tweets (i.e., not a retweet) that contain the keyword $k$ in the considered time span, and $\#rt_t$ is the retweet number of a tweet $t$. We further normalize and build the weight vector over all the considered keywords, called as *intent vector*. We denote the intent vector of a trend $e$ by $\vec{e}$.

**Product relevance.** Having the intent vector, now we discuss about how to define the product relevance. Given a product $p$, we extract all the words in the title and description parts of a product. We represent it as a vector using the widely *tf-idf* weighting method. We denote the weight vector of product $p$ by $\vec{p}$. We measure the product relevance between $e$ and $p$ as $\text{rel}(e,p) = \frac{\vec{e}\cdot\vec{p}}{|\vec{e}||\vec{p}|}$.

### 4.2 Modeling the Associativity between Products

To start this part, we first present an illustrative example in Fig. 3. We can see there are four related products for the trend "Air Pollution". We assume that only "mouth mask" was mentioned in microblogs. Now we expect to mine more related products with "mouth mask" as a known related product. We can compute the similarity between

[9]The difference was tested to be statistically significant.

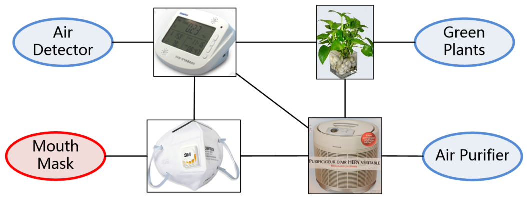



Figure 3: An example to illustrate the importance of associativity. Products which were mentioned in tweets related to "Air Pollution" are marked in red circles while the others are marked in blue circles. The link between products indicate the similarity between two products. Links with weights lower than a predefined threshold are not considered. Although "green plants" is related to this trend, it was not mentioned in tweets and did not have a direct link to "mouth mask".

a pair of products. Intuitively, if the similarity between a candidate product and "mouth mask" is higher than a predefined threshold, we can consider it to be related, too. In this example, "air detector" and "air purifier" are similar to "mouth mask" in terms of product descriptions and considered to be related, while "Green plants" is determined to be unrelated since it has very little overlap words with "mouth mask" in the description. It indicates one-step similarity method is not able to fully capture the real associativity between products.

Thus, we propose to use the random walk method to propagate the relatedness score on the product graph. Let $P$ denote the number of all the candidate products, and $\boldsymbol{r}_{P\times 1}$ denote the relatedness score vector where $r_i$ denote the relatedness score of product $p_i$.

We first construct the product graph. We represent each candidate product as a vertex in the graph and built the link with the cosine similarity between the descriptive texts of two products as the link weight.[10] We denote the similarity matrix by $\mathbf{M}_{P\times P}$ and $M_{i,j}$ denotes the similarity between products $p_i$ and $p_j$. Formally, we formulate the problem in a standard PageRank form

$$\boldsymbol{r}^{(n+1)} = \mu \cdot \boldsymbol{r}^{(n)} \cdot \mathbf{M} + (1-\mu)\cdot \boldsymbol{y}, \qquad (2)$$

where $\boldsymbol{y}$ is the restart probability vector usually set to be uniform. With this method, it is easy to see that relatedness score can be propagated on the product graph, which better captures underlying associativity between products.

[10]Other similarity methods can be used, e.g., co-purchase history record.

### 4.3 Jointly Modeling Relevance and Associativity

Having discussed about how to model both relevance and associativity, now we are ready to present a joint model to capture these two factors. By following (Zhao et al., 2013), the main idea is that instead of using a uniform restart distribution $\boldsymbol{y}$, we use an relevance biased restart distribution in Eq. 2. We set the restart probability of a product to its corresponding relevance. Formally, we set $y_i = \text{rel}(e, p_i)$. Let us further explain the idea. At the beginning of each iteration, each product is first assigned to its relevance score: the more relevant it is, the larger score it has. During the iteration, each product begins to collect relevance evidence from its neighbors on the product graph: the more relevant neighbors it has, the larger score it obtains. And the final score is indeed a trade-off between its own relevance score and neighboring relevance scores it receives. In order to obtain an ergodic walk, we add a small value, i.e. $10^{-4}$, to each entry of $\boldsymbol{y}$ and then normalize this vector. We denote our algorithm as JMRA (Jointly Modeling Relevance and Associativity).

To have an intuitive understanding of our algorithm, let us turn to the example in Fig. 3 again. At the beginning, only "mouth mask" has a large relevance score, with the iteration going on, the relatedness score will be propagated between products on the graph. Although "green plants" has not a direct link with "mouth mask", it can obtain relatedness score from its neighbors, i.e. "air detector" and "air purifier". JMRA is able to discover such latent associativity between products.

## 5 Experimental Setup

We use the test collection which have been described in Section 3. The statistics of the data set is shown in Table 3.

### 5.1 Evaluation Metrics

For a real product search engine, top results are particularly important, thus we adopt *precision@5* and *precision@10* as the evaluation metrics. Similar to Information Retrieval, we also consider using Mean

Average Precision (MAP) as metrics to measure the overall quality of retrieved products.

### 5.2 Methods to Compare

We compare the following methods for inferring relating products:

*SALES*: we rank the candidate products by their historical sales volume descendingly.

*TREND*: we use trend keywords as queries and rank the products by their relevance.

*TREND+fb*: based on *TREND*, we further incorporate pseudo-relevance feedback (Salton, 1971; Salton and Buckley, 1997). After some tuning (See Section 6.5), top 3 search results were used to update the query.

*JMRA*$_r$: it is our method which only considers product relevance in Section 4.1.

*JMRA*$_r + fb$: we further apply pseudo-relevance feedback to *JMRA*$_r$.

*JMRA*$_{r+a}$: it is our method which considers both relevance and associativity in Section 4.3.

*JMRA*$_{r+a} + fb$: we further apply pseudo-relevance feedback to *JMRA*$_{r+a}$.

## 6 Experimental Results and Analysis

In this section, we first evaluate the performance of the proposed approach and the comparison methods. Next, we analyze a problem in real e-commerce search engines, i.e., the cold start. Then, we give a qualitative case study to further demonstrate the effectiveness of the proposed approach. Finally, we examine the parameter sensitivity to the performance.

### 6.1 Comparison of Performance

We present the results of various methods in Table 4. We first examine the performance of baselines *SALES*, *TREND* and *TREND+fb*. First, *SALES* has the worst performance due to the fact that a trend usually happen unexpectedly and historical records may not predict it well. The second observation is that the improvement of *TREND+fb* over *TREND* is little. This is mainly because that very few related products can be identified only based on trend keywords so feedback method does not work very well on it.

Then we compare our relevance based methods with the above three baselines. Note that the major difference between *JMRA*$_r$ and *TREND* is that *JMRA*$_r$ makes uses of both trend keywords and product keywords extracted from microblogs. We can see that *JMRA*$_r$ performs better than all three baselines. It proves the effectiveness of leveraging commercial intents from microblogs. Another interesting point is that the relative improvement *JMRA*$_r$*+fb* over *JMRA*$_r$ is larger than that *TREND+fb* over *TREND*. The reason is that pseudo relevance feedback relies highly on top results and a system with better search quality will benefit more from it.

Finally, we consider evaluating our full models which jointly consider relevance and associativity. It is easy to see *JMRA*$_{r+a}$ yields a significant improvement over *JMRA*$_r$ and even outperforms *JMRA*$_r$*+fb*. This observation supports our assumption that product associativity is very important in this task. Again, pseudo relevance feedback has also improved *JMRA*$_{r+a}$.

In summary, our results have shown some important implications for trend-related product retrieval on e-commerce search engines: 1) microblogs are very good signals to learn users' commercial intents; 2) product associativity is particularly important; 3) other advanced retrieval methods are potentially useful, e.g., pseudo relevance feedback.

Table 4: The overall performance of all the methods.

| Models | P@5 | P@10 | MAP |
|---|---|---|---|
| *SALES* | 0.345 | 0.379 | 0.225 |
| *TREND* | 0.543 | 0.325 | 0.327 |
| *TREND+fb* | 0.550 | 0.325 | 0.328 |
| *JMRA*$_r$ | 0.611 | 0.527 | 0.336 |
| *JMRA*$_r$*+fb* | 0.661 | 0.552 | 0.348 |
| *JMRA*$_{r+a}$ | 0.733 | 0.609 | 0.392 |
| *JMRA*$_{r+a}$*+fb* | **0.734** | **0.624** | **0.404** |

### 6.2 Cold Start

It is noteworthy that we have considered all the candidate products within the entire active interval of a trend when constructing the test collection. This is mainly to obtain a good coverage of related products since some e-commerce companies might release new products as the response to a trend. During the active interval of a trend, the e-commerce companies may make some heuristic rules to enhance the retrieval of related products, e.g., incorporating trend keywords into product titles and descriptions. In the real application scenario, an effective method is expected to identify related products at the beginning of a trend when the e-commerce workers may not make any response to the trend. How would it be if we do not have the manually generated trend keywords from workers in product titles and descriptions?

To answer the question, in this part, we continue to examine the impact of cold start on different methods. We select three methods as comparisons, i.e., *TREND+fb*, *JMRA*$_r$ $+ fb$ and *JMRA*$_{r+a}$ $+ fb$. We first use keyword matching methods to obtain all the products that related to trend keywords. The descriptive text (i.e., title and description) of these products has been refined to match trend queries by sellers in e-commerce websites. We further removed all the trend keywords in the desriptive text of these products, and gradually add the trend keywords back to original products. In such a process, we would like to examine how cold start affects the performance of different methods.

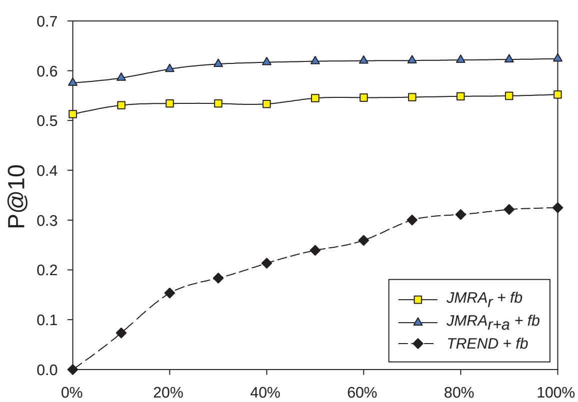



Figure 4: **The impact of cold start for different methods.**

We present the results in Fig. 4. First, performance of all these methods improve with the increase of products with trend keywords. Second, cold start does not affect the relative performance order of different methods, i.e., *TREND+fb* $<$ *JMRA*$_r$ $+ fb$ $<$ *JMRA*$_{r+a}$ $+ fb$. Finally, all the methods display similar impact patterns: "significantli increasing" $\rightarrow$ "stable". An interesting observation is that *JMRA*$_r$ $+ fb$ and *JMRA*$_{r+a}$ $+ fb$ have much more stable performance compared to *TREND+fb*. It indicates that our methods are very robust to the cold start, and potentially applicable in real e-commerce search engines.

### 6.3 Case Study

In order to have a intuitive understanding of how different method perform, we present a case study in this part. We select *JMRA*$_r$, *JMRA*$_r$*+fb* and *JMRA*$_{r+a}$*+fb* as comparisons. The results are shown in Table 5. We can see that *JMRA*$_{r+a}$*+fb* have identified the most related products. We further analyze the contribution of different factors. Compared with *JMRA*$_r$, *JMRA*$_r$*+fb* has got one more related product "Houseplant" due to the reason pseudo feedback can make use of top related search results to improve the queries. In this case, "Houseplant" is identified to be related because it is very similar to "Air Detector" (We have presented the corresponding most associative products in brackets in Table 5). Similarly, the comparison between *JMRA*$_r$*+fb* and *JMRA*$_{r+a}$*+fb* shows the effectiveness of product associativity.

### 6.4 Error Analysis

To further understand the shortcomings of the proposed methods, we use the example in Table 5 for error analysis. Based on our manual inspection, errors may arise from two major sources for our method:

- Product keyword extraction errors: we use a pattern-based product keyword extraction method, and it tends to incorporate some irrelevant words. For example, given the topic "air pollution", users would talk about the impact of "car exhaust" on air quality and advocate to reduce automobile usage and sale. The current keyword extraction method might mistake the word "car" for a product related keyword.

- Search engine retrieval errors: in this paper, we rely on the Taobao product search engine for candidate product generation. It is highly

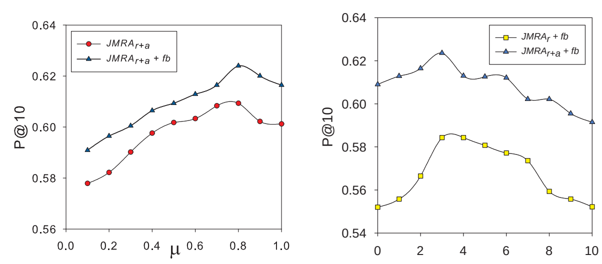
Table 5: A qualitative comparison of three methods on the topic of "Air Pollution". We mark related products in bold.

**Sample keywords learnt from microblogs**:
air pollution, mouth mask, air, air purifier, respirator, house, mask, warm, bus, car, purified water

| *JMRA*$_r$ | *JMRA*$_r$+*fb* | *JMRA*$_{r+a}$+*fb* |
|---|---|---|
| **Mouth Mask** | **Mouth Mask** | **Mouth Mask** |
| **Air Detector** | **Air Detector** | **Air Purifier** |
| **Air Purifier** | **Air Purifier** | **Air Detector** |
| Toy House | Humidifier | **Respirator** |
| **Respirator** | **Respirator** | **Oxygen Bag (*Mouth Mask*)** |
| Toy Car | Party Mask | Humidifier |
| Environment-friendly Bags | Toy Car | **Houseplant (*Air Detector*)** |
| Humidifier | **Houseplant (*Air Detector*)** | **Anti-pollution Medicine (*Oxygen Bag*)** |
| Purified Water | Environment-friendly bags | Purified Water |
| Warmer | Purified Water | Party Mask |

based on surface-form matching to retrieve related products. Therefore, given a query "mouth mask", it might return some irrelevant products, e.g., "party mask". Clearly, pseudo-relevance feedback will also bring additional irrelevant products if top search results contain irrelevant ones.

To solve these problems, one promising way is to leverage more context information about the candidate products and construct deep semantic analysis. We will leave it as future work.




(a) (b)

Figure 5: Parameter sensitivity. a) The impact of the damping factor $\mu$ and b) the impact of the number of top products used for pseudo feedback.

### 6.5 Parameter Sensitivity

The only parameter for *JMRA* is the damping factor in the random walk model, i.e., $\mu$. Intuitively, a larger value of $\mu$ emphasize the associativity more while a smaller value emphasize the relevance more. We tune this parameter at a step of 0.1 and present the results in Fig. 5(a). We can see the performance of *JMRA*$_{r+a}$+*fb* is consistently better than that of *JMRA*$_r$+*fb* and peaks at around "0.8". It indicates the robustness of *JMRA*$_{r+a}$+*fb* and the importance of product associtivity.

We further examine the impact of the number of top products used for pseudo feedback for *JMRA*$_r$+*fb* and *JMRA*$_{r+a}$+*fb*. In Fig. 5(b), we can see that both *JMRA*$_{r+a}$+*fb* and *JMRA*$_r$+*fb* achieved their best at "3". It indicates that we only need to consider very top results for pseudo feedback.

### 6.6 Title or Description?

In previous experiments, for each product, we used the descriptive text in both title and description. In this part, we consider examining the individual effect of *title* and *description*. We use *JMRA*$_{r+a}$+*fb* as the examined method since both relevance and associativity relies on the text information.

Table 6: Evaluating the performance of *JMRA*$_{r+a}$+*fb* with different text sources.

| sources | P@5 | P@10 | MAP |
|---|---|---|---|
| title | 0.690 | 0.591 | 0.364 |
| description | 0.711 | 0.602 | 0.387 |
| title+description | 0.734 | 0.624 | 0.404 |

As shown in Table 6, we can see that the performance of only using *description* is better than that of only using *title* and a combination of both parts achieve the best. *title* is usually carefully compiled by e-commerce sellers, thus it reveals the most highlights of the products but very short; while *description* contains more informative text but tends to incorporate noise. In future work, we will consider a more principled way to combine *title* and *description*, e.g., weighted combination.

### 6.7 Efficiency

Finally, we present a few discussions about the issue of efficiency. All codes were implemented in Python 2.7, and all experiments were performed on a PC with *Intel(R) Core(TM)i5 CPU 760 @ 2.8GHz* and 8GB memory.

Sicne we group products by the categorial label, the number of candidate products is usually very small. Thus, our method *JMRA* runs very efficiently. Even on an extremely large set of candidate products, the iterative random walk algorithm can be easily implemented in a distributed way (Bahmani et al., 2011) and would have very good efficiency.

## 7 Related Work

Our work is mainly related to the following lines:

**Mining the microblogs.** Microblogs have been one of the most popular social networking platforms, and they have recently attracted much attention from research communities. The studies on trend (or event) detection (Benson et al., 2011; Weng and Lee, 2011; Sakaki et al., 2010; Zhao et al., 2012) tried to make use of the rapid response of microblogs users as the signal to automatically identify external events. Another important aspect is the content analysis of tweets, including the recommendation of real-time topical news (Phelan et al., 2009), sentiment or opinions analysis (Meng et al., 2012), event summarization using tweets (Chakrabarti and Punera, 2011; Lin et al., 2012; Zhao et al., 2013), etc. Our work do not explicitly incorporate a trend detection component, instead we make use of the trending topics provided by the microblogs platforms. It will be easy to incorporate other trend detection methods as our input.

**Identifying online users' commercial intents.** The identification of online users' commercial intents has been quite an important research problem in the past. Most researches focus on capturing commercial intention from search queries (Dai et al., 2006; Strohmaier and Kröll, 2012), click-through behaviors (Ashkan and Clarke, 2009), users' mouse movements or scrolling behaviors (Guo and Agichtein, 2010) and search logs (Strohmaier and Kröll, 2012). The most related to our work is the work in (Hollerit et al., 2013), which attempts to detect commercial intent on twitter. But we have very different focus. They aim to identify tweet-level commercial intents while ours aim to identify trend-driven commercial intents. In addition, we also present how to make use of these identified intents and our paper focuses on how to identify trend related products for e-commerce companies to improve service when faced with hot trends.

## 8 Conclusions

In this paper, we make the first attempt to identify trend related products by leveraging commercial intents from microblogs. We propose a way to construct the evaluation set for this task and present some insightful findings. We propose a graph based method to joint model relevance and associativity. We perform extensive experiments, including quantitative and qualitative analysis.

Currently, our approach is indeed a framework to solve this task, and we may consider improving the individual components in it, e.g. consider non-product keywords in tweets. For future work, we will consider incorporating a trend detection component into our method, which can be more flexible to adapt to various trend signals. We can also refine the method of the product keyword extraction by using more principled solutions.

## Acknowledgments

We thank the anonymous reviewers for the constructive comments. The work was partially supported by NSFC Grant 60933004, 61073082 and 61272340. Jinpeng Wang was supported by the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive & Digital Media Programme Office, Media Development Authority. We thank Taobao for the access to the product data and all Taobao data in this paper will be only used for research purpose.

## References


Azin Ashkan and Charles LA Clarke. 2009. Term-based commercial intent analysis. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 800–801. ACM.

Bahman Bahmani, Kaushik Chakrabarti, and Dong Xin. 2011. Fast personalized pagerank on mapreduce. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, SIGMOD '11, pages 973–984, New York, NY, USA. ACM.

Edward Benson, Aria Haghighi, and Regina Barzilay. 2011. Event discovery in social media feeds. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 389–398. Association for Computational Linguistics.

Deepayan Chakrabarti and Kunal Punera. 2011. Event summarization using tweets. In *Proceedings of the 5th Int'l AAAI Conference on Weblogs and Social Media (ICWSM)*, July.

Honghua Kathy Dai, Lingzhi Zhao, Zaiqing Nie, Ji-Rong Wen, Lee Wang, and Ying Li. 2006. Detecting online commercial intention (oci). In *Proceedings of the 15th international conference on World Wide Web*, pages 829–837. ACM.

Qi Guo and Eugene Agichtein. 2010. Ready to buy or just browsing?: detecting web searcher goals from interaction data. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 130–137. ACM.

Bernd Hollerit, Mark Kröll, and Markus Strohmaier. 2013. Towards linking buyers and sellers: detecting commercial intent on twitter. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 629–632. International World Wide Web Conferences Steering Committee.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a social network or a news media? In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM.

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 497–506, New York, NY, USA. ACM.

Chen Lin, Chun Lin, Jingxuan Li, Dingding Wang, Yang Chen, and Tao Li. 2012. Generating event storylines from microblogs. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 175–184. ACM.

Xinfan Meng, Furu Wei, Xiaohua Liu, Ming Zhou, Sujian Li, and Houfeng Wang. 2012. Entity-centric topic-oriented opinion summarization in twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 379–387. ACM.

Owen Phelan, Kevin McCarthy, and Barry Smyth. 2009. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, pages 385–388. ACM.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 851–860, New York, NY, USA. ACM.

Gerard Salton and Chris Buckley. 1997. Readings in information retrieval. chapter Improving retrieval performance by relevance feedback, pages 355–364. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Gerard Salton, editor. 1971. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, Englewood, Cliffs, New Jersey.

Markus Strohmaier and Mark Kröll. 2012. Acquiring knowledge about human goals from search query logs. *Information Processing & Management*, 48(1):63–82.

Michael J. Welch, Uri Schonfeld, Dan He, and Junghoo Cho. 2011. Topical semantics of twitter links. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 327–336, New York, NY, USA. ACM.

Jianshu Weng and Bu-Sung Lee. 2011. Event detection in Twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media (ICWSM)*, Menlo Park, CA, USA. AAAI.

Wayne Xin Zhao, Baihan Shu, Jing Jiang, Yang Song, Hongfei Yan, and Xiaoming Li. 2012. Identifying event-related bursts via social media activities. In *EMNLP-CoNLL*, pages 1466–1477.

Wayne Xin Zhao, Yanwei Guo, Rui Yan, Yulan He, and Xiaoming Li. 2013. Timeline generation with social attention. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 1061–1064.