Unsupervised tag recommendation for popular and cold products

Article *in* Journal of Intelligent Information Systems · August 2019

DOI: 10.1007/s10844-019-00574-9

Anand Konjengbam
Indian Institute of Technology Hyderabad

Nagendra Kumar
Agency for Science, Technology and Research (A*STAR), Singapore

Noname manuscript No.
(will be inserted by the editor)

Unsupervised Tag Recommendation for Popular and Cold Products

Anand Konjengbam · Nagendra Kumar · Manish Singh

Abstract The rapid expansion of the Internet and its connectivity has given tremendous growth to e-commerce sites. Product reviews form an indispensable part of e-commerce sites. However, it is challenging and laborious to go through hundreds of reviews. In this paper, we address the problem of summarizing reviews by means of informative and readable tags. We present a novel unsupervised method of generating tags and rank them based on relevance. We refine the generated tags using NLP syntactic rules to make them more informative. Our proposed Tagging Product Review (TPR) system takes into consideration the opinions expressed on the product or its aspects. We also address the problem of tag generation for cold products, which have only a limited number of reviews and that too, with very short content. We use transfer learning to build a tag cloud from popular product reviews and use it to identify good tags from cold product reviews. We evaluate our proposed system using online reviews of twelve products of varying popularity, collected from Amazon.com. Our result demonstrates the effectiveness of our approach at generating relevant tags compared to three popular baseline methods. Our proposed approach gives an average tag relevance score (NDCG) of around 79% for popular products and 85% for cold products. Our approach also gives an average precision of 89% for identifying correct tags. The results suggest that our TPR system successfully summarize reviews by means of tags.

Keywords Electronic Commerce · Tagging Review · Information Search · Review Summarization · Opinion Mining · User Feedback

1 Introduction

Web 2.0 is marked by a tremendous growth of social media and user-generated content, such as texts, images and videos. Tags have lately surfaced as a convenient way of organizing and summarizing such user-generated contents due to their simplicity in indexing and ease of user participation. A tag is a keyword that is used to describe the object of content, and it facilitates keyword-based classification and information

Indian Institute of Technology Hyderabad, India - 502285
Email: {cs14resch11004, cs14resch11005, msingh}@iith.ac.in

search. The unique ability of tags to group and share information has changed how people consume information. Recently researchers [4, 6] have shown that tags are one of the most reliable sources for many Information Retrieval (IR) services, such as content classification, searching, and ranking of posts. Tags have also been used for expert profiling [33] and document summarization [40].

Tags are used in Social Networking sites, such as Facebook, Twitter, and YouTube for seeking information, increasing visibility [7] and connecting people with a common interest. They are also employed in Community Question Answering (CQA) sites, such as Yahoo! Answers, Stack Exchange and Quora for indexing question-answers [47] and for routing questions [46] to get answers from experts.

Tag recommendation is useful both for customers as well as manufacturers. Tag recommendation would play a vital role in providing a bird's eye view of the reviews and influence the purchasing decision of customers. Users can provide their aspect preference and personalized tags relevant to the selected tags may be shown to the users. Such a tag-based summary would also provide market intelligence to manufacturers. It would provide valuable feedback to manufacturers about their products and tells which components they need to pay attention to.

Although social networking sites allow users to create their own tags while posting content, CQA sites require users to select a set of tags from the collection of tags that are created by privileged users. For quality control, CQA sites allow only users with at least some minimum reputation score to create new tags. The visibility of content depends primarily on the post content and the tags associated with it. Most of the existing tagging schemes rely mainly on their users to manually annotate content with tags.

Manually assigning tags is laborious and is often difficult due to the enormous number of possible tags. For example, the CQA site StackOverflow has around 50K tags, and a user has to select between 1-5 tags that best describe their question. Few of these sites use tag recommendation systems [11, 13, 39] to help the user select good tags. Tag recommendation systems can only help with tag annotation, but not with tag creation. In this paper, we present a tag generation system for opinionated contents, such as reviews and debates. Although the problem of sentiment analysis has been widely studied for opinionated contents, to the extent of our knowledge, there is no existing work that has addressed the problem of automatically tagging such contents. We propose the use of tags as a mechanism to summarize product reviews.

A recent trend of research on tag generation [33, 34, 38] has focused on providing useful and relevant tag suggestion to users. Most of the existing approaches result in unigram tags, which are sometimes not sufficiently informative. In the case of product reviews, the generated tags should be rich enough to eliminate the need for the customer's prior insight into the product. To illustrate, Fig. 1 shows the tags generated from reviews of a camera product by Amazon.com.

It can be noticed that almost half of the tags are unigram such as 'canon', and 'digital', which do not give useful information about the product. Also, tags such as 'nikon coolpix', 'pictures also', etc. do not give useful information about the product. Instead, providing phrase as the whole conveys deeper understanding than individual words. To better understand the problem, consider the following example of a camera review:

Fig. 1: Amazon generated tags for a camera from reviews (accessed on 25.12.2018)

Example 1 “Best picture quality for any point and shoot I have used. The lens is excellent and the camera case is good. It has an amazing flash for the price! If you’re just starting out, I would recommend starting out with this flash so that you have something good while saving up for top-notch lighting equipment. It is easy to use even for an amateur such as myself.”

Here, summarizing the review with the unigram tags such as ‘flash’ is not useful to understand the quality of the product or its aspect ‘flash’. But, summarizing it with the phrase tag like ‘amazing flash’ makes it easy to understand the customer’s sentiment associated with the product aspect ‘flash’. We extend conventional tag generation approaches to generate meaningful tag phrases from the opinionated text and use the tag phrases to summarize customer reviews.

As products generally have a considerable number of customer reviews, computing a set of meaningful tags for review summarization is a challenging task. A product may have many aspects and some aspects are more popular than others, thereby having a significant impact on consumers’ decision. In Example 1, the aspect ‘picture quality’ would have more impact than the aspect ‘camera case’. ‘Picture quality’ is a primary aspect for buying a camera while ‘camera case’ is an accessory and it is not related to the performance of the camera. We address this task as a tag ranking problem.

Often, there are cases where the grammatical structure of the reviews does not give sufficient information for generating useful tags. In Example 1, the opinion ‘excellent’ is associated with the aspect ‘lens’ although they do not appear adjacent. Forming the tag phrase ‘excellent lens’ makes more sense than the tag ‘lens’. We introduce five Natural Language Processing (NLP) rules based on Part-of-Speech (POS) structure that helps in making the tags meaningful. Given a product, we build a system that assists users by providing a set of meaningful tags from the product reviews. Our system achieves promising results when evaluated on 12 real-world product review datasets from Amazon.com.

Following are some of the significant observations: The tag phrases generated by our unsupervised approach are related to the products and are found helpful to understand the product reviews. Popular (high-rated) products have a higher number of reviews than cold (low-rated) products, suggesting that people are mainly interested in popular products. Also, the average length of a review is much shorter for cold products compared to popular products, suggesting that people tend to talk about their good experiences much more than bad experiences. So, cold products suffer from the problem of generating noisy tags. We show how tags from popular product reviews can be leveraged to find relevant tags in cold product reviews.

Our key contributions are as follows:

- We address the problem of customer reviews summarization. We present the problem as the tag generation problem.
- We propose a novel approach to find useful tags by ranking and refining the generated tags.
- We investigate the performance of tag generation on reviews of 12 products.
- We categorize products into popular products and cold products and investigate the tag generation performance.

The rest of the paper is structured as follows. A brief review of the related work is presented in Section 2. Section 3 details the proposed approach for unsupervised tag generation from reviews. We present the evaluation and conclusion in Section 4 and Section 5 respectively.

2 Related Works

Researchers have studied the problem of tag recommendation for various applications. Existing approaches of tag recommendation can be broadly classified into two categories: supervised [28, 35, 44] and unsupervised [31, 37, 42]. Majority of the work falls into the supervised approach where the given object or item already has some tags, and the task is to expand the tags or predict tags of similar items. Supervised approaches may be further classified into three sub-categories based on how they work: (1) co-occurrence based approaches, (2) Matrix-based approaches, and (3) graph-based approaches.

On the other hand, unsupervised approaches do not require any tag to work. They extract information from the content and its associated features to recommend tags. In our paper, we work on extracting tags from contents that do not have tags associated with them. Since we give more focus on extracting tags that also has sentiment, we also present how our work is related to existing research on sentiment analysis.

2.1 Supervised Tag Recommendation

Supervised approaches can be used where some tagged data is available in the form of training data. For recommendations systems, such data form ternary relations between users, items and tags. For news posts, blogs posts, etc., it is in the form of binary relation between items, and tags. Works on various supervised approaches are discussed below:

Co-occurrence based approaches utilize tags assigned to a collection of objects to recommend tags of a new object using association rules. Given some existing tags associated with a target object, this approach expands the tag collection based on the co-occurrence pattern of the candidate tag with the initial tags [12, 14, 21, 26, 36, 44]. [12] propose an interactive way of recommending relevant tags whenever a user selects a new tag by considering the user's tag usage history. The co-occurrence based approaches are dependent on the availability of some initial tags assigned to the target objects. These approaches require a lot of tagged data as they use tag co-occurrence to recommend tags for the target object.

Matrix factorization approaches work well for data that have information on users, tags, and items. They represent the tag assignment to items in the form of a matrix and factorize the matrix into smaller ones to discover latent features between different entities. Reducing the dimension of the matrix makes it easier to find the relationships between users, objects, and tags. [32] used matrix factorization and TensorFlow to model the pairwise relationship between users, tags, and items. Although this exploits the benefits of dimensionality reduction, matrix factorization work well only for dense data and involves expensive operations that are not so scalable [2].

Graph-based approaches follow a graph representation of the system, where the nodes are the objects, and the edges denote the similarity relationship between the objects. They exploit the textual features and neighborhood similarity information to find candidate tags [16, 18, 25]. The social connection between users is exploited in [24] to build a graph-based tag recommendation system. Tag propagation between objects with similar textual content is proposed in [25, 48]. [10] modeled the folksonomy as a heterogeneous graph containing tags, users, and objects as nodes. Graph-based approaches do not perform well when the data has noisy content like user reviews.

2.2 Unsupervised Tag Generation

Unsupervised methods are suitable for items where only the content is available without any tags. They are generally used to find relevant topics from documents such as social media posts [31], community question answering sites [22] and scientific publications [33]. They leverage contextual information such as patterns, similarities and syntactic rules to generate topical tags. Topic modeling has been used to extract topical words from texts and to suggest relevant tags to characterize the contents [31]. [37] followed a graph-partitioning method to find the most representative topics from a document. The performance of the aforementioned methods highly relies on the assumption of a dense set of data upon which the model is built. [33] use content-based unsupervised tag recommendations as a mechanism for expert profiling in the scientific domain. They showed that keywords are the most effective content to generate tags compared to titles and abstracts.

Our approach differs from the existing methods in that (1) We focus on tagging opinionated documents, such as product reviews and debates, (2) We focus on generating tags that capture the semantic objects from reviews. Existing works on unsupervised tag recommendation focus on extracting topical tags that have objective information. Extracting only topical tags are not helpful to understand product reviews. Reviews contain lots of subjective information from the reviewer. We incorporate NLP rules to associate semantic words to the tags. (3) Product reviews are harder to deal with compared to the scientific documents. Reviews are relatively short compared to other documents and do not have well-defined components, such as title, abstract and keywords, that complement tag generation. While scientific documents are well structured, reviews are generally unstructured, which makes it hard to process.

2.3 Sentiment Analysis

Sentiment analysis is generally used to find the polarity of a text content such as tweets, product reviews, and social media post. Sentiment analysis of user content has been widely studied [9, 23]. Existing works on sentiment analysis focus on classifying the sentiment of a content [23], which could be carried out at the content level, sentence level, or aspect level [15, 41, 45]. Our work is related to aspect level sentiment analysis. Aspects are the objects associated with the sentiment. NLP syntactic rules are generally used to extract aspects and sentiments. Hu and Liu [15] used association mining to extract frequent nouns as aspects. Wu et al. [45] utilized dependency parser to extract noun phrases and verb phrases from reviews as candidate aspects. Unlike existing works that focus on extracting aspects and polarity, we use sentiment analysis to improve the tag generation by adding prominent sentiment words related to the tags. Adding dominant sentiment words make the tags more informative. We combine the contextual and syntactic information from the reviews to generate tags that help understand product quality.

3 Tagging Product Reviews

In this section, we present our unsupervised approach to summarize product reviews by generating top-k tags. We give the problem definition in Section 3.1 and our solution in the remaining three subsections.

3.1 Problem Definition

There are many factors that determine the nature of reviews that a product gets. For example, a popular product, which is available since a significant time, may have many reviews, some of which could also be very detailed. On the other hand, a new product or a low rated product may have only few reviews, most of which will be very short. Although for popular products, the reviews may explain the pros-and-cons of different aspects of the product; for low rated products, most of the reviews will be short with an overall negative sentiment. To tag these different classes of product reviews, we propose the following two tag generation problems.

Problem 1 Tagging Popular Products: Let P be a product and $R_P = \{r_1, r_2, r_3, ..., r_n\}$ be the set of customer reviews of the product P, where each review r_i consists of a customer's feedback and an overall rating on the product P. Our task is to generate top-k tags $T = \{t_1, t_2, ..., t_k\}$ for the review set R_P.

Popular products have many reviews. Many of these reviews have sizable content and has a high rating for the product. They contain experience of users about the product and its comparison with other products. In general, a popular product has hundreds of reviews and the average length of each review is around 1500 characters.

Since popular products have many reviews and lot of content, if we use existing topic labeling methods, such as LDA [3], TNG [43], etc., to generate the tags, then we will get many topic labels, all of which we cannot show to users. Moreover, many of these labels may not be suitable for human reading. For example, they would generate tags such as ‘years ago’, ‘level camera’, ‘camera takes’, etc. To reduce the number of tags, we select top-k tags that summarizes all the reviews. To make it more presentable and informative to users, we process the top-k tags using NLP syntactic rules, as described in Problem 3.

Problem 2 Tagging Cold Products: Let P be a product that has very few reviews or has mostly low ratings, and R_P=$\{r_1, r_2, r_3, ..., r_n\}$ be the set of customer reviews of the product P, where each review r_i consists of a customer's feedback and an overall rating on the product P. Our task is to generate top-k tags $T = \{t_1, t_2, ..., t_k\}$ for the review set R_P.

In contrast to popular products, there are many cold products, such as low rated products and new products, which have less than 15 reviews with an average length of 500 characters per review. These products suffer from the problem of ‘cold start’ due to lack of review content. In this paper, we are not targeting new products as many of them do not have reviews or rating information. We are targeting low rated products that have few reviews, which are mostly negative. Low rated products are not good quality products; thus most of the reviews contain customers' discontent with the product. Such reviews are generally short and may not even contain much information about the product. Due to these reasons, it is hard to generate tags for them.

Existing topic-generation methods do not work well when we have such a small amount of data. For example, consider a low-rated product review: ‘It is complete waste of time and money’. Since the review is short and does not even contain much of product-related information, it is hard to generate candidate tags. We propose a tag generation method for such products by using tags from popular products.

Problem 3 Tag Refinement: Given the top-k generated tags for a product P, parse the reviews and use NLP syntactic rules based on the POS patterns to make the tags more readable and informative through word transformation and by including popular customer sentiment.

Topic modeling algorithms are effective in finding noun forms of tags. However, such tags may not capture the sentiment terms associated with the tags and thus may not give sufficient semantic information. For example, the tag ‘lens cap’ represents an aspect only, but does not provide the opinion of customers associated with the aspect. Providing the associated sentiment ‘loose’ to form the tag ‘loose lens cap’ makes the tag more meaningful. Sometimes, the grammatical structure of the tags makes it difficult to understand. For example, the tag ‘focusing low light’ does not make much sense. We can make it meaningful by rewriting as ‘low light focusing’. Occasionally, the tags may fail to capture the right context. For instance, the tags ‘good picture’ and ‘big lens’ are completely different from the tags ‘not good picture’ and ‘overly big lens’ respectively. We make the tags more readable by adding popular sentiment words from the reviews and changing the grammatical structure of the tags based on NLP syntactic rules.

3.2 System architecture

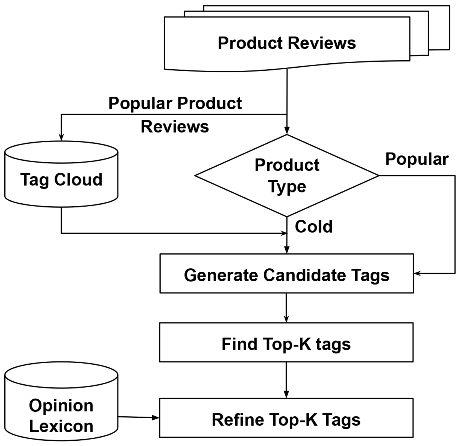

Fig. 2: System architecture

Fig. 2 shows the architectural overview of our proposed review tagging system. Given a product, the system takes user reviews as input and generates top-k product relevant tags as output. The reviews undergo the following steps: (1) generating candidate tags, (2) ranking the candidate tags, and (3) refining the top-k tags using syntactic rules. Unlike existing supervised tag recommendation systems [1, 5], this paper presents an unsupervised method to generate tags for customer reviews. The details are explained in the following subsections.

3.3 Generating Candidate Tags

As grammatical rules are generally not followed while writing reviews, reviews contain many noisy texts such as stopwords, repeated sequence characters, acronyms, slangs and spelling mistakes. Such faulty words may generate unwanted tags and we may also miss good tags. Since we do not want such distorted texts to hinder the tag generation process, we clean review texts using the following pre-processing steps:

Stop word removal: Stop words are non-semantic words that frequently occur in all texts (e.g., 'about' and 'while'). Tags generated with those frequent words will not have any semantic relevance to the product. We remove stop words using the NLTK's stop word corpus [30].

Removal of repeated character sequence: Repeated sequence of a character is often used in customer reviews to emphasize customer sentiment. For example, Nikon is the *bestttt*. Our work does not focus on finding sentiment strength of such words. Moreover, such words may cause variations of the same tag such as *best* and *bestttt*. We want to maintain consistency and capture the correct word as tags. So, we discard any extra ending characters that occur more than once.

URL filtering: Users frequently provide URLs (e.g., www.nikon.com) to some other pages for reference or more detailed information. Such URLs are only pointers and do not contribute to the tag generation process. So, URLs are identified and removed from the reviews.

Acronym expansion: Customers use acronyms and slangs while writing to save keystrokes. Understanding the meaning of these words are important for generating the tags. We identify and replace the acronyms and slangs with their actual words using an acronym dictionary.

Spelling correction: Spelling mistake is one of the most common problems in customer reviews. For instance, a customer may misspell the word *excellent* as *excelent*. This may result in different variations of the same tag. In order to correct the spelling mistakes, we employ an NLP based spelling corrector [29].

After refining the reviews of noisy text, the next step is to generate candidate tags. Customers normally do not provide any topic marker while writing reviews. Topic modeling methods based on bag-of-words representations are commonly used to identify key terms from text documents [8, 13, 40, 43]. We consider each review as a document and adopt a bag-of-words representation of the reviews by extracting n-grams with varying sizes (n=1, 2, 3) from the corpus. Using the n-grams, we extract candidate tags by applying three popular topic modeling techniques, namely popularity, TF-IDF and Topical n-gram (TNG) [43]. The techniques are explained as follows:

Popularity: Popular tags are the n-grams that frequently appear in customer reviews. We use term frequency based popularity algorithm [40] to find candidate tags. Popularity is defined as follows:

$$Pop(q) = \frac{f_q}{max_k f_k} \tag{1}$$

where q is the given tag, f_q is the number of times the tag q appears in all reviews, and $max_k f_k$ is the count of the most popular tag across all reviews. A high popularity score indicates that the tag is highly relevant as many times it has been mentioned in the reviews.

TF-IDF: To reduce the significance of commonly appearing tags in favor of more unique ones, we use TF-IDF score of tags, which is defined as:

$$TF\text{-}IDF(q) = Pop(q) * \log\left(\frac{N}{n_q}\right) \tag{2}$$

where N is the total number of reviews of a product, n_q is the number of reviews with tag q in it. TF-IDF score has two terms: the first term represents the term frequency (TF) and the second term represents the Inverse Term Frequency (IDF).

Topical n-gram (TNG): TNG [43] is a probabilistic topic modelling algorithm that extends Latent Dirichlet Allocation (LDA) [3] to model topics from documents. Unlike LDA, TNG can model words as well as phrases. TNG model is able to decide whether a bigram is a phrase or not according to context. For example, the bigram 'good picture' carries a special meaning with respect to the topic camera', whereas the phrase 'good news' does not. TNG does this by adopting a joint distribution

P(t,b|x) and has the distributional presumptions:

$$x_i|x_{i-1}, t_i, b_i = 1, \theta \sim Discrete(\theta_{t_i}) \tag{3}$$

$$x_i|x_{i-1}, t_i, b_i = 0, \phi \sim Discrete(\phi_{t_i}, x_{i-1}) \tag{4}$$

$$b_i|x_{i-1}, t_{i-1}, \gamma \sim Bernoulli(\pi_{t_{i-1}}, x_{i-1}) \tag{5}$$

where each document d is expressed as a mixture of latent topics t_i's. Each topic t is expressed as a mixture of terms x's determined by a multinomial distribution θ. b_i is a binary variable that indicates whether the term x is affected by the preceding term or is the beginning of a new n-gram starting at the i^{th} position. If $b_i = 1$, then the term x_i is not affected by the previous word and is considered to be the start of a new n-gram. b_i divides a collection into sequential non-intersecting n-grams of different sizes. $\gamma_{t,x}$ and $\phi_{t,x}$ are distributions equipped with conjugate prior distributions:

$$\gamma_{t,x} \sim Beta(\lambda) \tag{6}$$

$$\phi_{t,x} \sim Dirichlet(\delta) \tag{7}$$

where γ and δ are some hyperparameters.

The above tag generation algorithms favor frequently occurring n-grams as candidate tags. Candidate tags generated by popularity and TF-IDF are mostly unigrams as they occur more frequently than phrases. Unlike popularity and TF-IDF, TNG takes the sequential nature of text into consideration to generate words and phrases as topics. As phrases bear more meaningful information than words, TNG gives better tags compared to popularity and TF-IDF. We consider the tags generated from TNG as candidate tags for further processing.

Due to the limited availability of reviews in cold products, TNG is not able to generate good key terms as tags. Many of the irrelevant words get generated as tags. We do not want to show irrelevant words as tags. To find the useful tags from cold product reviews, we leverage popular product reviews having a similar domain. We approach this as a transfer learning problem.

Candidate tags of popular products contain useful tags that are commonly used by customers while writing reviews such as 'picture quality', 'shutter speed', 'battery life', and 'touch screen'. So, for products having a similar domain, the same set of frequent aspects would be present in popular as well as cold products. Although such useful tags are also present in cold products, TNG is not able to detect such tags due to the sparsity of reviews. We leverage the candidate tags from reviews of popular products to enhance the tag generation of cold products by discovering useful tags.

After applying TNG, we collect the top 50 candidate tags containing aspect information from each popular product belonging to similar domain and use it to form a tag cloud (C). From C, we use the aspect information to find tags information for the cold products. For a cold product, we parse through each review and find the tags that are present in C. C is also used to find the review-tags dictionary (RT) for each review, which stores the tags present in each review. RT is in turn used to obtain the top-k most useful tag set.

3.4 Ranking Candidate Tags

Some reviewers write long reviews while others write short reviews. TNG gives more importance to long reviews as they often have repeated key terms. But, in short reviews, key terms generally appear only once and related terms are also absent. Although useful tags may be present in short reviews, frequent tags from long reviews tend to dominate the result of TNG. To overcome this bias, we take into consideration the number of reviews where a tag is present and focus on finding the useful tags instead of frequent ones. A tag that is present in many reviews is considered more useful than a tag that is present in few reviews.

We propose a greedy approach to rank the candidate tags obtained using TNG to find the tags that are preferred by many users. We use coverage to find the useful tags, i.e., tags that cover maximum number of reviews. Coverage of a tag t is defined as:

$$Cov(t) = \frac{n_t}{N} \tag{8}$$

where n_t is the number of reviews that contain the tag t and N is the total number of reviews. The tag ranking algorithm is given in Algorithm 1.

Algorithm 1 Tag Usefulness Ranking TUR(P,P_P,R_P)

Input: P: Product
P_p: Set of popular products belonging to domain $D = \{p_1, p_2, ..., p_l\}$
R_P: $\{r_1, r_2, ..., a_n\}$ = Customer reviews of product P
T_C: $\{tc_1, tc_2, ..., tc_n\}$ = Candidate tags for popular products P_p
Output: $C = \{c_1, c_2, ..., c_m\}$ = Tag cloud = set of candidate tags of popular products
T: $\{t_1, t_2, ...t_k\}$ = Top-k relevant tags generated from the reviews
Method:

1: $C \leftarrow \{\}, T_C \leftarrow []$, Max coverage tags, $T \leftarrow \phi$, Review-tags dictionary, $RT \leftarrow []$
2: **for all** $p \in P_p$ **do**
3: $\quad T_C \leftarrow$ RankCandidateTags(p)
4: $\quad$ **for all** $c_i \in T_C$ **do**
5: $\quad\quad$ **if** $c_i \notin C$ **then**
6: $\quad\quad\quad C$.Add(c_i)
7: $\quad\quad$ **end if**
8: $\quad$ **end for**
9: **end for**
10: **if** $P \in P_p$ **then**
11: $\quad$ **for all** $r_i \in R_p$ **do**
12: $\quad\quad RT[r_i] \leftarrow$ ExtractTags(r_i, T_C)
13: $\quad$ **end for**
14: **else**
15: $\quad$ **for all** $r_i \in R_p$ **do**
16: $\quad\quad RT[r_i] \leftarrow$ ExtractTags(r_i, C)
17: $\quad$ **end for**
18: **end if**
19: **while** $len(T) \leq k$ or $R_p \neq \Phi$ **do**
20: $\quad t \leftarrow \arg\max_{c_i}\{|cov(c_i, RT)|\}$
21: $\quad T$.Append(t)
22: $\quad R_t \leftarrow$ AssociationSet(t, RT)
23: $\quad T_C \leftarrow T_C - t$
24: $\quad R_p \leftarrow R_p - R_t$
25: **end while**
26: return T

Given a product P and a set of reviews R_P, TUR follows a greedy approach to find the most useful tags. Step 1 initializes the tag cloud C, the ranked list of candidate tags T_C, the useful tag set T and the review-tags dictionary RT. Steps 2–9 uses the candidate tags of popular products to get the tag cloud C. For each product p, step 3 finds the top-50 tags that have the highest TNG topic relevance score and store it in T_C. Then, for each tag c_i in T_C, steps 4–8 adds c_i to the tag cloud C if it is already not present. We leverage C to find tags from cold product reviews. Step 10 checks if the product P is present in the popular product set P_p. P_p contains a collection of popular products obtained after analyzing the average product rating of each product. For each review r_i, steps 11–13 use the candidate tags T_C to extract the relevant tags present in r_i and store the tags in the Review-Tag dictionary RT. While for a cold product review, step 15–17 uses the tag cloud C to extract review specific tags RT. Using the relevant tags present in RT, steps 19–25 iterate until we get either the top-k useful tags or all the reviews are covered by the tags. At each iteration, it finds and appends the most helpful tag from T_C to T, which covers the maximum number of uncovered reviews at that moment. Step 20 finds the tag t that has maximum coverage and adds it to the tag set T in step 21. Step 22 computes the association set (R_t) of t to find the reviews that are covered by it. Steps 23–24 remove t and the associated reviews (R_t) from T_C and R_P respectively. Finally, step 26 returns the list of top-k most useful tags (T) after greedily finding the tags.

3.5 Refining Top-K Tags using Syntactic Rules

The previous step generates a list of core phrase tags from the reviews. Although the generated top-k tags are useful, some of the generated tags do not give sufficient semantic information due to missing sentiment terms. In general, popular products are associated with positive sentiment words, while many of the cold products are associated with neutral or negative sentiment words. To find the sentiment terms, we analyze the Part-of-Speech (POS) patterns that are generally associated with phrases of opinionated text using the Stanford NLP dependency parser and develop five syntactic rules to enhance the usefulness of the tags. We apply these syntactic rules after parsing the reviews and also finding the top-k useful tags T. The tag phrases obtained after applying the syntactic rules are found to be more readable and meaningful. Table 1 presents the syntactic rules.

3.5.1 Opinion Rule

Opinion words are generally adjectives and they provide useful subjective information. So, whenever an adjective is present before a noun tag phrase (NP), we check if it belongs to an opinion word by using an opinion lexicon [20] and note its polarity. When an opinion word (JJ) is present before a noun phrase tag (NP) in reviews, the opinion word is considered as a part of the tag phrase. However, different customers may use different opinion words of varying polarity for the same tag. Consider an example, where multiple customers associate the tag 'picture quality' with different opinion words such as 'amazing', 'excellent', 'good', 'poor', 'bad', etc. For a given tag, generating multiple tag phrases with varying opinion words would lead to several ambiguous or redundant tags such as 'amazing picture quality', 'good picture quality', and 'poor picture quality'. We need to find the right polarity and the accurate opinion word associated with the tag. To this end, we count the number of positive and negative opinion words associated with the tag and select the polarity that has maximum count as the polarity of the tag. Among the same polarity opinion words,

Rule	POS Pattern	Description	Example
Opinion rule	JJ+NP	Opinion word (JJ) is present before tag phrase (NP) in reviews	amazing picture quality, good battery life, loose lens cap
Negation rule	NG+NP	Negation word (NG) is present before tag phrase (TP)	not good picture, not auto focus
Intensifier rule	RB+JP	Adverb intensifier (RB) is present before adjective tag phrase (JP)	very long zoom, moderately bright flash
Gerund transformation rule	VBG+JP → JP+ VBG	If a verb (VBG) is present before an adjective tag phrase(JP), then put it at the end of the tag	processing raw image → raw image processing
Opinion inclusion rule	NP/JP+VBZ+JJ → JJ+NP	If an opinion word (JJ) is present at the end of a noun/adjective phrase tag with a verb (VBZ) in between, place the opinion word (JJ) at the beginning of the tag and remove the verb (VBZ)	lens cap is loose → loose lens cap; optical zoom works great → great optical zoom

Table 1: POS rules used to enhance the meaning of tags

we choose the most frequent word as the opinion word associated with the tag. In the above example, the tag ‘picture quality’ is associated with more positive opinions. Among the positive opinion words, considering ‘amazing’ appears in more number of reviews compared to ‘excellent’ and ‘good’, we refine the tag as ‘amazing picture quality’.

3.5.2 Negation Rule

Sometimes, tags are associated with negation words, thereby reversing the meaning of the tags. For instance, the meaning of the tag phrase ‘good picture’ is reversed when the negation word ‘not’ is present before it, forming the tag phrase ‘not good picture’. Negation rule is used to handle such cases. We add negation word to a tag phrase only if the majority of the customer talking about the tag phrase are negative about it. If a negation word appears before a tag phrase, we count the number of times the tag phrase is associated with negation word in the reviews. If it is more than half of the number of occurrence of the tag phrase, we associate a negation word ‘not’ with the corresponding tag phrase.

3.5.3 Intensifier Rule

Adverbs (RB) intensifies an adjective phrase (JP) when present before the phrase. So, whenever an adverb is present before an adjective phrase tag, intensifier rule is used to add adverbs (RB) present before an adjective phrase tag (JP). Doing so complements the sentiment associated with the phrase tag. For example, the tags ‘overly big lens’, and ‘fast shutter speed’ conveys more information than the tags ‘big lens’, and ‘shutter speed’ respectively.

3.5.4 Gerund Transformation Rule

Gerunds in some of the tag phrases appear in distorted forms such as ‘processing raw image’, ‘focusing low light’, etc. These phrases have a gerund (VBG) followed by an adjective phrase (JP). We observe that if the gerund appears after the adjective phrase (JP + VBG), the phrases are more readable. Gerund transformation rule is used to find the tag phrases that contain gerunds followed by adjective phrase (VBG + JP). Whenever such a phrase occurs, the gerund is shuffled at the end of the phrase (JP + VBG) as shown in Table 1.

3.5.5 Opinion Inclusion Rule

In some cases, we observe that the opinion words (JJ) appear after tag phrases (NP/JP) with a verb (VBZ) in between (NP/JP + VBZ + JJ) such as ‘lens cap is loose’, ‘optical zoom works great’, etc. Opinion inclusion rule adds opinions words having such a pattern to the beginning of the phrase tag and discard the verb (VBZ). For example, the phrase ‘lens cap is loose’ is transformed into ‘loose lens cap’.

4 Evaluation

In this section, we present the evaluation of our proposed approach on tag recommendation for product reviews. We first describe the dataset and experimental setup. Then, we present the evaluations and interpretations of our observations.

4.1 Experimental Setup

4.1.1 Dataset

To perform our experiments, we consider product reviews from Amazon. We use the Amazon review dataset[1], which is one of the most popular datasets in review analysis research [19, 27, 49]. The dataset comprises of consumer experiences, such as review, rating, and helpfulness votes on products along with product information, such as descriptions, brand, price, etc., on 24 product domains for the period May 1996 - July 2014. Every product domain has multiple sub-domains. We choose the ‘Camera and its accessories’ domain and conduct experiments on the consumer reviews of 12 electronics products from 5 sub-domains, namely 8 digital cameras, 1 digital frame, 1 router, 1 speedlight and 1 wireless trigger.

4.1.2 Product Review Selection

We observe that the nature of reviews varies from popular products to cold products. Popular products have significantly more reviews, which are also quite detailed, compared to the cold products. From the dataset, we identify popular products and cold products based on the average overall rating of the product and the number of reviews. For the experiment, a product with an average overall rating of four or

[1] http://jmcauley.ucsd.edu/data/amazon/

five and a huge number of reviews is considered as a popular product. On the other hand, a product with an average overall rating of one or two and with less number of reviews is considered a cold product. To ensure that the reviews are useful, we select only those products that have reviews with average sentence length greater than 5 and at least two helpfulness votes. We choose the top six popular products and the top six cold products with the highest number of reviews. The entire dataset contains 13218 sentences. The statistics of the dataset is shown in Table 2.

Product name	**No. of reviews**	**Product type**
Digital Camera 1	137	P
Digital Camera 2	137	P
Digital Camera 3	134	P
Camera Flash	403	P
Digital Frame	110	P
Router	205	P
Digital Camera 4	17	C
Digital Camera 5	15	C
Digital Camera 6	18	C
Digital Camera 7	8	C
Digital Camera 8	10	C
Wireless Trigger	14	C

Table 2: Dataset Statistics

From Table 2, we can infer that the number of reviews for popular products is much more compared to cold products. This is because customers do not prefer to buy low rated products and also do not spend much time to describe their bad experiences. Also, the average number of sentences for high-rated products is much more than the low-rated products. This indicates that while writing reviews, customers tend to describe more about their likings of high-rated products and very little about low-rated products.

4.1.3 Tag Relevance Assessment

To determine the relevance of the generated tags, we asked ten Ph.D. students to label the generated tags. All ten students have proper knowledge of electronic products. We select a set of reviews that cover the generated tags and are provided to the students for reference. Then, the scholars are asked to grade each of the generated tag based on the following five-point relevance scale:

– **non-relevant (score 0)**: The tag is distorted and has no association with the product (e.g., view, stars, ghost hunting, awkward place).
– **ordinary (score 1)**: Although the tag is well-formed, the association with the product is ambiguous (e.g., accessory genie, canon line) or partial (e.g., frame, shutter).
– **marginally relevant (score 2)**: The tag is well-formed and is a secondary product aspect without sentiment (e.g., camera settings, extra battery, wireless feature).

- **relevant (score 3)**: The tag is well-formed and is a primary product aspect without sentiment (e.g., white balance, viewfinder, power button, touch screen).
- **highly relevant (score 4)**: The tag is well-formed and is either a primary or secondary feature with sentiment (e.g., awesome video, compact design, fast focus, flash works great).

After the students came up with their own relevance scale, any ambiguity in tag scoring was resolved together through discussion and mutual agreement. We observe that around 31% of the tags are non-relevant or ordinary, 25% of the tags are either relevant or highly relevant, and 44% of the tags are marginally relevant. The high percentage of non-relevant and ordinary tags are expected as the baseline algorithms generate mostly bad tags and tags at the lower position of the rankings are not much relevant to the products. The scores of the tag relevance assessment are considered as ground truth labels for evaluation of tag relevance and tag ranking. We observe that, for all the products, the proportion of non-relevant and ordinary tags is lower than that of relevant and highly relevant tags. This supports the suitability of the chosen approaches for generating useful tags.

4.1.4 Baseline Methods

We compare the performance of the proposed Tag Usefulness Ranking (TUR) algorithm used in Tagging Product Review (TPR) system with the three topic modeling techniques described in Section 3, namely, popularity, TF-IDF, and TNG, taken as baselines. TUR follows an iterative and greedy approach for selecting the top k tags by choosing the tags that are present in most yet uncovered reviews in each iteration. The tags are selected such so as to maximize the reach of reviews.

4.2 Evaluation Metric

4.2.1 Effectiveness of Tag Ranking

We evaluate the effectiveness of tag ranking using the evaluation metric Normalized Discounted Cumulative Gain at top-k (NDCG@k) [17]. NDCG@k of a ranked list of k tags is defined as follows:

$$NDCG@k = \frac{1}{iDCG} \sum_{i=1}^{k} \frac{2^{t(i)} - 1}{log(1+i)} \tag{9}$$

where $t(i)$ is the scoring function that indicates the score assigned to the tag at the i^{th} location, iDCG is the DCG value of the top-k tags obtained from an ideal ranking. The use of iDCG normalizes the value of NDCG@k within a range of $[0, 1]$. The value of the scoring function $t(i)$ of a tag i is obtained from the tag relevance assessment score of the corresponding tag. NDCG@k favors the ranking that ranks the most relevant tags at the top. In our experiments, we consider different values of k ranging between 1 to 50.

4.2.2 Accuracy of Tag Generation

The accuracy of tag refinement is measured using precision. Precision gives the percentage of correct tags present in the generated tag list after refinement. Precision is defined as follows:

$$P = \frac{\#\ correctly\ generated\ tag}{\#\ generated\ tags} \tag{10}$$

We use the tag relevance scores for ground truth labels. Tags with a relevance score of 4 or 5 are considered as correct tags while the rest are considered as incorrect tags.

4.3 Results

4.3.1 Evaluation of Tagging Popular Products

This section we present the evaluation of popular products. For cold products, it is presented in the following subsection.

Fig. 3 shows the comparison between tag ranking algorithms for popular products using NDCG. The NDCG score of popularity and TF-IDF is comparable, with each having an average NDCG score of 46.16% and 50.42% respectively. Compared to them, TNG gives a higher NDCG score, with an average of 64.62%, improving the other two baselines by over 18% and 14% respectively.

For 'On Camera Flash' and 'Router', the NDCG score of popularity and TF-IDF methods are comparatively lower than the rest of the products. As these two products have significantly more number of reviews compared to the other four products, the presence of noisy tags gets amplified. However, the presence of large corpus complements our TUR approach in finding better coverage, giving a relatively high NDCG score compared to other products. Digital frame has the least number of reviews among popular products. Hence the performance of popularity, TF-IDF and TNG are comparable due to the lesser amount of noisy text.

For two of the products, namely 'Router' and 'Digital Frame', there is a sharp rise in NDCG score of popularity and TF-IDF methods when the value of 'k' increases initially. This is because they largely rely on the frequency of tags in the corpus. As words occur more frequently than phrases, most of the top-k tags for popularity and TF-IDF methods are trivial words. Moreover, they segregate the corpus into small independent components, such as unigrams, bigrams, and trigrams, and in the process, the contextual information of the text is lost. This causes frequently occurring unrelated words like 'time', and 'money' to appear as top tags. As a result, these two methods give very low NDCG score when the value of 'k' is small. As 'k' increases, relevant tags appear in the list, and hence NDCG score rises. Among the three baselines, TNG gives the highest average NDCG because it uses the contextual information of the text in the corpus to generate meaningful words and phrases as tags. TNG's ability to generate phrases as topics also improves its NDCG score.

Although TNG gives meaningful tags, many of the tags are irrelevant to the product, such as 'poor weather', 'tech savvy', 'bottom line', 'bright light', etc. The presence of such tags at the top lowers the relevance score. TUR improves the ranking of TNG by filtering out such irrelevant tags. TUR follows an iterative and greedy approach to find tags that cover the most number of reviews instead of depending solely on the tag frequency in the corpus. TUR improves the relevance of generated tags by finding tags that are most talked about by customers. Since most of the tags by TUR are relevant, the value of NDCG does not fluctuate much with varying values of 'k'. Applying syntactic rules to the generated tags further improves the readability

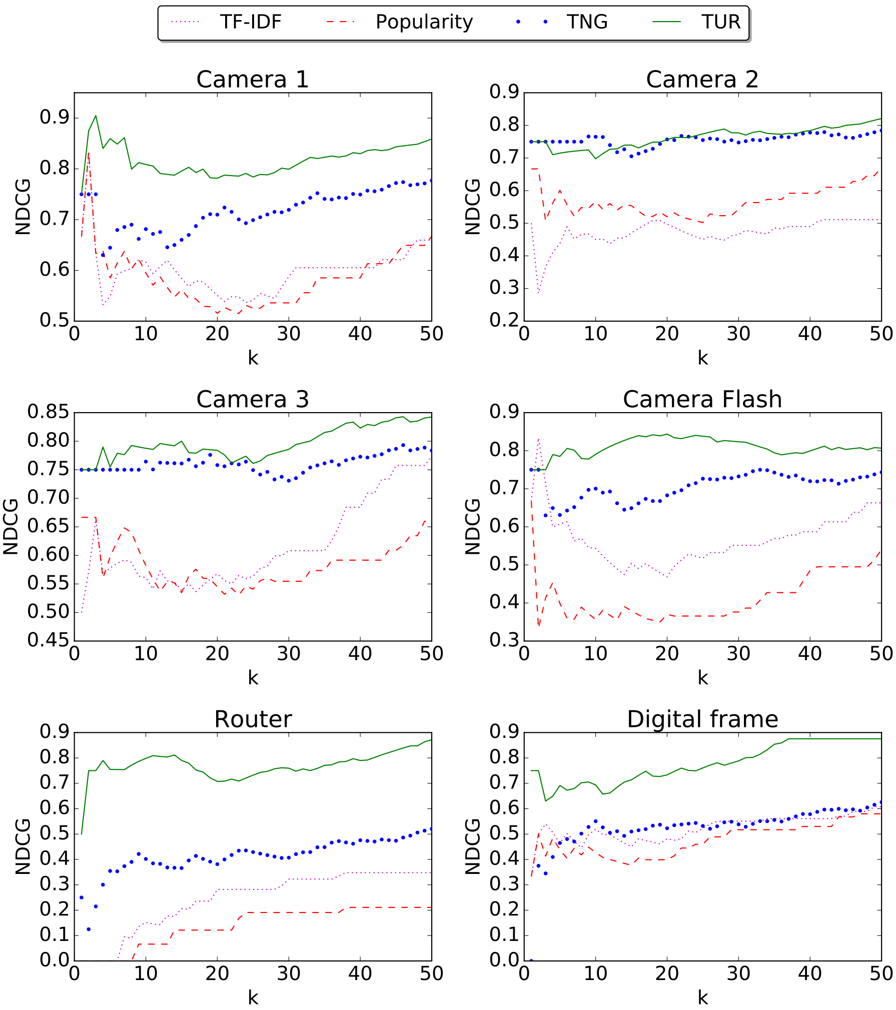

Fig. 3: Evaluation of top-k tags for popular products using the metric NDCG@k

of the tags. This, in turn, increases the tag relevance score, thereby improving the NDCG score. For all popular products, on average TUR outperforms the popularity, TF-IDF and TNG based approaches by over 32%, 28%, and 14% respectively. Thus from the results, we can infer that our proposed TUR algorithm can better identify important opinionated aspects from customer reviews compared to standard topic modeling algorithms.

4.3.2 Evaluation of Tagging Cold Products

Fig. 4 shows the result of tag relevance evaluation for cold products. For cold products, our proposed TUR approach outperforms the three baseline approaches by a

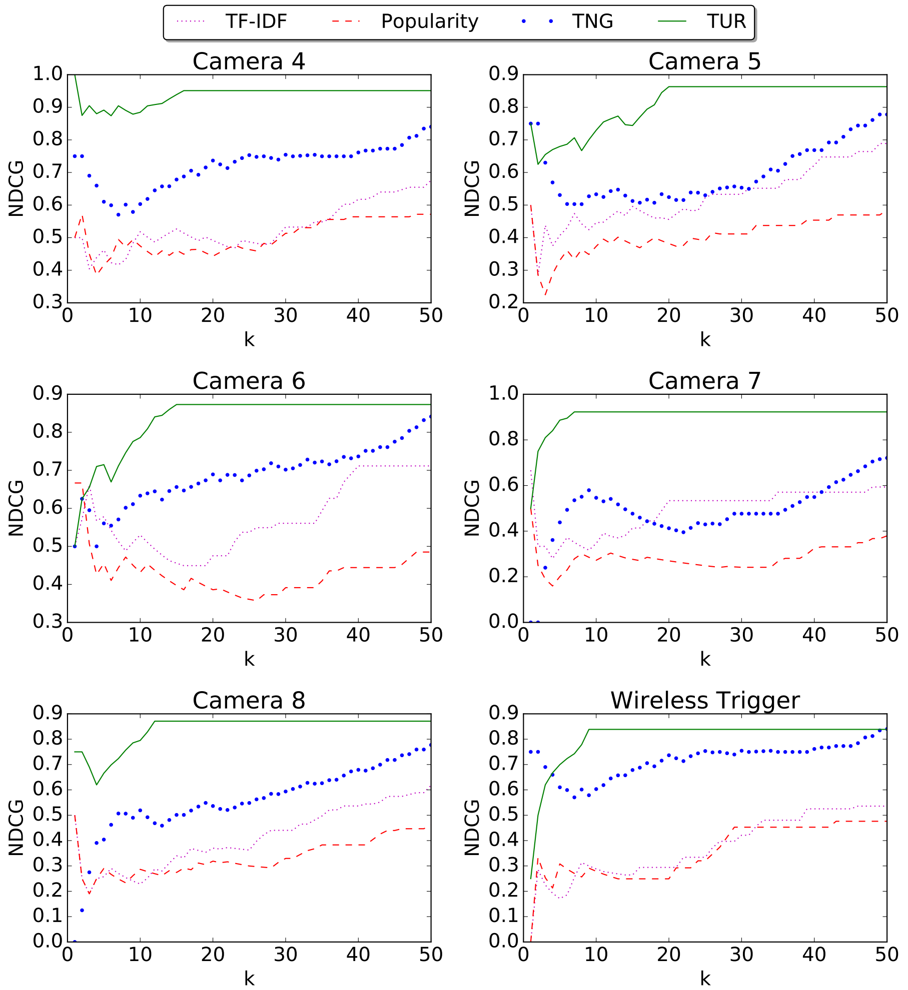

Fig. 4: Evaluation of top-k tags for cold products using the metric NDCG@k

huge margin. For the six cold products, the average NDCG of TUR is 85.56% compared to the average NDCG of 38.57%, 48.57%, and 62.90% for popularity, TF-IDF and TNG respectively. In other words, our approach outperforms the baselines by over 47%, 37%, and 23% respectively. Our approach is very helpful compared to baseline methods because majority of the e-commerce products have few reviews, comparable to cold products.

The big difference in NDCG for cold products is due to the use of tag cloud-based transfer learning. Since cold products have small review corpus, the frequency of key terms is very less. Hence popularity and TF-IDF methods perform poorly by picking up frequently occurring irrelevant terms such as ‘buy’, ‘design’, and ‘money’ as tags. The lack of review text also affects the performance of TNG in generating key topics. Due to the absence of sufficient contextual information, the tags generated by

TNG are less relevant (e.g., 'worth noting', 'bottom line', etc.). Compared to popular products, the relevance score of TNG drops considerably for low values of 'k'.

Our proposed approach overcomes the lack of contextual information by leveraging transfer learning to identify good tags. The use of tag cloud from popular products to identify tags from cold product reviews compensates the lack of contextual information. Due to the lack of review text, only a few tags are present in cold product reviews. Therefore, the relevance curve of TUR increases at the beginning and continues as a horizontal line with increasing k.

4.3.3 Evaluation of Tag Refinement

Product name	Product type	Popularity	TF-IDF	TNG	TUR
Camera 1	P	0.32	0.36	0.84	0.96
Camera 2	P	0.34	0.32	0.84	0.90
Camera 3	P	0.34	0.42	0.82	0.90
Camera flash	P	0.24	0.30	0.82	0.90
Digital frame	P	0.18	0.24	0.66	0.84
Router	P	0.06	0.14	0.46	0.82
Camera 4	C	0.34	0.40	0.40	0.88
Camera 5	C	0.30	0.38	0.50	0.88
Camera 6	C	0.26	0.28	0.74	0.98
Camera 7	C	0.28	0.22	0.50	1.00
Camera 8	C	0.36	0.38	0.66	0.88
Wireless trigger	C	0.20	0.22	0.50	0.88

Table 3: Evaluation of tag refinement using the precision of top 50 tags

We use precision metric to evaluate the effect of tag refinement. The effect of tag refinement is shown in Table 3. From Table 3, it can be observed that our proposed approach significantly outperforms the other baseline approaches both for popular and cold products. For both the type of products, popularity and TF-IDF methods give low accuracy of 26.83% and 30.50% respectively. TNG gives an average precision of 64.50%, which is better than that of popularity and TF-IDF. TNG performs better for popular products compared to cold products. On average, TNG gives 25% higher precision for popular products than cold products. This is because TNG depends on the size of the corpus and the number of reviews for popular products is much more than cold products. On the other hand, our proposed approach (TUR) gives a consistently good result with an average precision of 90.16%, which is 25.66% higher than TNG. By ranking the tags based on usefulness and refining the tags based on syntactic rules, the average precision improves for both popular and cold products. In the case of cold products, we observe that TUR gives very high precision. This is because only a limited number of tags are present in cold product reviews. For instance, even though 'Camera 7' has only 8 reviews, our approach can find 7 relevant tags giving a precision of 100%.

5 Conclusion

This paper presents a novel unsupervised approach for automatic tag generation from product reviews. We construct a Tagging Product Review (TPR) system using topic modeling and NLP syntactic rules. We use TNG to find and rank tag phrases and then apply our TUR algorithm to get the important tags that cover most of the reviews. The use of topic modeling enables us to generate tags without using external data. We also introduce five syntactic rules to enhance the usefulness of the tags. Generating tags for cold products is challenging due to the lack of contextual information. We overcome this shortcoming by using tag clouds that are generated from popular product reviews. Our evaluation of twelve datasets shows that our approach can generate and rank tags significantly better than existing approaches, whose tags give either partial or no information about the product. Also, we develop a greedy approach to rank and refine candidate tags. Our approach is applicable to any domain which has textual content such as movie reviews, community forums, social media sites, and debate posts as it does not require any pre-trained information. In our future work, we plan to apply our approach to social media sites for summarizing user discussions.

References

1. Belém F, Santos R, Almeida J, Gonçalves M (2013) Topic diversity in tag recommendation. In: Proceedings of the 7th ACM conference on Recommender systems, ACM, pp 141–148
2. Belém FM, Martins EF, Almeida JM, Gonçalves MA (2014) Personalized and object-centered tag recommendation methods for web 2.0 applications. Information Processing & Management 50(4):524–553
3. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. Journal of machine Learning research 3(Jan):993–1022
4. Bullock BN, Hotho A, Stumme G (2018) Accessing information with tags: Search and ranking. In: Social Information Access, Springer, pp 310–343
5. Cao H, Xie M, Xue L, Liu C, Teng F, Huang Y (2009) Social tag prediction base on supervised ranking model. In: Proceeding of ECML/PKDD 2009 Discovery Challenge Workshop, pp 35–48
6. Chen YL, Chang CL, Yeh CS (2017) Emotion classification of youtube videos. Decision Support Systems 101:40–50
7. Cook J, Kenthapadi K, Mishra N (2013) Group chats on twitter. In: Proceedings of the 22nd international conference on World Wide Web, ACM, pp 225–236
8. El-Kishky A, Song Y, Wang C, Voss CR, Han J (2014) Scalable topical phrase mining from text corpora. Proceedings of the VLDB Endowment 8(3):305–316
9. Fang X, Zhan J (2015) Sentiment analysis using product review data. Journal of Big Data 1(2):1–14
10. Feng W, Wang J (2012) Incorporating heterogeneous information for personalized tag recommendation in social tagging systems. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 1276–1284

11. Ferragina P, Piccinno F, Santoro R (2015) On analyzing hashtags in twitter. In: International Conference on Web and Social Media (ICWSM), AAAI Press, pp 110–119
12. Garg N, Weber I (2008) Personalized, interactive tag recommendation for flickr. In: Proceedings of the 2008 ACM conference on Recommender systems, ACM, pp 67–74
13. Godin F, Slavkovikj V, De Neve W, Schrauwen B, Van de Walle R (2013) Using topic models for twitter hashtag recommendation. In: Proceedings of the 22nd International Conference on World Wide Web, ACM, pp 593–596
14. Heymann P, Ramage D, Garcia-Molina H (2008) Social tag prediction. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp 531–538
15. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 168–177
16. Hu M, Lim EP, Jiang J (2010) A probabilistic approach to personalized tag recommendation. In: 2010 IEEE Second International Conference on Social Computing, IEEE, pp 33–40
17. Järvelin K, Kekäläinen J (2002) Cumulated gain-based evaluation of ir techniques. ACM Trans Inf Syst 20(4):422–446, DOI 10.1145/582415.582418, URL http://doi.acm.org/10.1145/582415.582418
18. Jäschke R, Marinho L, Hotho A, Schmidt-Thieme L, Stumme G (2007) Tag recommendations in folksonomies. In: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, pp 506–514
19. Johnson R, Zhang T (2014) Effective use of word order for text categorization with convolutional neural networks. arXiv preprint arXiv:14121058
20. Konjengbam A, Ghosh S, Kumar N, Singh M (2018) Debate stance classification using word embeddings. In: International Conference on Big Data Analytics and Knowledge Discovery, Springer, pp 382–395
21. Krestel R, Fankhauser P, Nejdl W (2009) Latent dirichlet allocation for tag recommendation. In: Proceedings of the third ACM conference on Recommender systems, ACM, pp 61–68
22. Kumar N, Mudda KY, Trishal G, Konjengbam A, Singh M, et al. (2019) Where to post: Routing questions to right community in community question answering systems. In: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, ACM, pp 136–142
23. Liu B (2012) Sentiment analysis and opinion mining. Synthesis lectures on human language technologies 5(1):1–167
24. Liu K, Fang B, Zhang W (2010) Speak the same language with your friends: augmenting tag recommenders with social relations. In: Proceedings of the 21st ACM conference on Hypertext and hypermedia, ACM, pp 45–50
25. Lu YT, Yu SI, Chang TC, Hsu JYj (2009) A content-based method to enhance tag recommendation. In: Twenty-First International Joint Conference on Artificial Intelligence
26. Menezes GV, Almeida JM, Belém F, Gonçalves MA, Lacerda A, De Moura ES, Pappa GL, Veloso A, Ziviani N (2010) Demand-driven tag recommendation. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, pp 402–417

27. Meng X, Bradley J, Yavuz B, Sparks E, Venkataraman S, Liu D, Freeman J, Tsai D, Amde M, Owen S, et al. (2016) Mllib: Machine learning in apache spark. The Journal of Machine Learning Research 17(1):1235–1241
28. Mishne G (2006) Autotag: a collaborative approach to automated tag assignment for weblog posts. In: Proceedings of the 15th international conference on World Wide Web, ACM, pp 953–954
29. Norvig P (2007) How to write a spelling corrector. De: http://norvig com/spell-correct html
30. Perkins J (2010) Python text processing with NLTK 2.0 cookbook. Packt Publishing Ltd
31. Ramage D, Dumais S, Liebling D (2010) Characterizing microblogs with topic models. In: Fourth International AAAI Conference on Weblogs and Social Media
32. Rendle S, Schmidt-Thieme L (2010) Pairwise interaction tensor factorization for personalized tag recommendation. In: Proceedings of the third ACM international conference on Web search and data mining, ACM, pp 81–90
33. Ribeiro IS, Santos RL, Gonçalves MA, Laender AH (2015) On tag recommendation for expertise profiling: A case study in the scientific domain. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, ACM, pp 189–198
34. Shen Z, Arslan Ay S, Kim SH, Zimmermann R (2011) Automatic tag generation and ranking for sensor-rich outdoor videos. In: Proceedings of the 19th ACM international conference on Multimedia, ACM, pp 93–102
35. Si X, Sun M (2008) Tag-lda for scalable real-time tag recommendation. JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE 6(2):1009–1016
36. Sigurbjörnsson B, Van Zwol R (2008) Flickr tag recommendation based on collective knowledge. In: Proceedings of the 17th international conference on World Wide Web, ACM, pp 327–336
37. Song Y, Zhang L, Giles CL (2011) Automatic tag recommendation algorithms for social recommender systems. ACM Transactions on the Web (TWEB) 5(1):4
38. Sood S, Owsley S, Hammond KJ, Birnbaum L (2007) Tagassist: Automatic tag suggestion for blog posts. In: ICWSM
39. Toba H, Ming ZY, Adriani M, Chua TS (2014) Discovering high quality answers in community question answering archives using a hierarchy of classifiers. Information Sciences 261:101–115
40. Venetis P, Koutrika G, Garcia-Molina H (2011) On the selection of tags for tag clouds. In: Proceedings of the fourth ACM international conference on Web search and data mining, ACM, pp 835–844
41. Wang H, Lu Y, Zhai C (2010) Latent aspect rating analysis on review text data: a rating regression approach. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 783–792
42. Wang J, Hong L, Davison BD (2009) Rsdc'09: Tag recommendation using keywords and association rules. In: DC@ PKDD/ECML
43. Wang X, McCallum A, Wei X (2007) Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In: icdm, IEEE, pp 697–702
44. Wu L, Yang L, Yu N, Hua XS (2009) Learning to tag. In: Proceedings of the 18th international conference on World wide web, ACM, pp 361–370
45. Wu Y, Zhang Q, Huang X, Wu L (2009) Phrase dependency parsing for opinion mining. In: Proceedings of the 2009 Conference on Empirical Methods in Nat-

ural Language Processing: Volume 3-Volume 3, Association for Computational Linguistics, pp 1533–1541
46. Xia X, Lo D, Wang X, Zhou B (2013) Tag recommendation in software information sites. In: Proceedings of the 10th Working Conference on Mining Software Repositories, IEEE Press, pp 287–296
47. Yang B, Manandhar S (2014) Tag-based expert recommendation in community question answering. In: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), IEEE, pp 960–963
48. Zhang N, Zhang Y, Tang J (2009) A tag recommendation system based on contents. ECML PKDD Discovery Challenge 2009 (DC09) p 285
49. Zhang X, Zhao J, LeCun Y (2015) Character-level convolutional networks for text classification. In: Advances in neural information processing systems, pp 649–657