

# Semi-supervised Category-specific Review Tagging on Indonesian E-Commerce Product Reviews

**Meng Sun**

Tokopedia, Singapore  
daisy.meng@tokopedia.com

**Marie Stephen Leo**

Tokopedia, Singapore  
marie.leo@tokopedia.com

**Eram Munawwar**

Tokopedia, Singapore  
eram.munawwar@tokopedia.com

**Seong Per Lee**

Tokopedia, Singapore  
seong.lee@tokopedia.com

**Albert Hidayat**

Tokopedia, Jakarta  
albert.hidayat@tokopedia.com

**Muhamad Danang Kerianto**

Tokopedia, Jakarta

**Paul C. Condylis**

Tokopedia, Singapore  
paul.condylis@tokopedia.com

**Sheng-yi Kong**

Tokopedia, Singapore  
angus.kong@tokopedia.com

## Abstract

Product reviews are a huge source of natural language data in e-commerce applications. Several millions of customers write reviews regarding a variety of topics. We categorize these topics into two groups as either “category-specific” topics or as “generic” topics that span multiple product categories. While we can use a supervised learning approach to tag review text for generic topics, it is impossible to use supervised approaches to tag category-specific topics due to the sheer number of possible topics for each category. In this paper, we present an approach to tag each review with several product category-specific tags on Indonesian language product reviews using a semi-supervised approach. We show that our proposed method can work at scale on real product reviews at Tokopedia<sup>1</sup>, a major e-commerce platform in Indonesia. Manual evaluation shows that the proposed method can efficiently generate category-specific product tags.

## 1 Introduction

E-commerce product reviews are a rich source of direct feedback from the customers. Written in free text natural language, product reviews contain a significant amount of information regarding a variety of topics that are important to prospective buyers.

Tokopedia conducted customer survey research to understand the sources of information that potential buyers assess while making a purchase decision. This internal research shows that around 15% customers consider product reviews as the most important source of information and it is the third

highest among all 20 possible information sources. Internal analysis of the “click rate” of various components on the platform’s product listing page also shows that components related to product reviews have the second highest click rate which further emphasises the importance of product reviews for prospective buyers.

Although reviews are important information sources, manually filtering relevant information is a cumbersome process for a buyer when making a purchase decision. Tokopedia has several hundreds of millions of customer reviews, generated by millions of users over the years. Therefore, extracting relevant tags for each product so that prospective buyers can quickly filter the most relevant reviews based on their topic of interest becomes important to make a quick purchase decision and improve buyer engagement on the platform.

We categorize topics in reviews into two types. The first type of topics are the generic topics that exist in reviews of products from any category, and they are about the generic information that customers care about. In the e-commerce platform, for example, the generic topics are “customer service”, “delivery”, “packaging quality”, “price”, and so on. The second type of topics are the category-specific topics. These topics are detailed description of the product specific attributes. Since different products have different attributes, the category-specific topics are very different for products from different categories. For example, for products in Phone Case category, the category-specific topic could be “cable hole”, while for products in Herbal Medicine category, the category-specific topic would be “ingredients”. The focus of this paper is to generate tags of category-specific topics for products across different categories.

<sup>1</sup>[www.tokopedia.com](http://www.tokopedia.com)

There are several challenges for this work. Firstly, the category-specific topics are widely different among products of different categories. Therefore, it’s impossible to get labeled data to apply supervised methods which are normally used when generating tags. Secondly, we work on informal Indonesian language. Though Indonesian language shares the same alphabet with English, Indonesian language differs from English in certain significant ways such as different sentence structure, prefix and suffix modifiers and slang spellings. Also since we work on reviews, the texts are informal, and contain a mixture of Indonesian, English, abbreviations and slang, which further increases the difficulty.

The focus of this work is to address the above mentioned challenges. We proposed a semi-supervised method, and successfully applied it to product reviews from different categories in the e-commerce platform. We also evaluated our results with manually labeled data.

The rest of this paper is organized as follows. We describe related work in the literature in Section 2. We then describe our approach to extract category-specific tags from Indonesian language review text in Section 3. Experiments and results are discussed in Section 4.

## 2 Related Work

While we can use a supervised learning approach to get generic topics from product reviews, it is impossible to use supervised approaches with “category-specific” topics due to the sheer number of possible topics for each product category. Therefore, we use an unsupervised method to extract topics from product reviews in this paper.

One of the earliest unsupervised method to extract keywords from text is the statistics based method. Frequency or Term Frequency - Inverse Document Frequency (TF-IDF) score is calculated on the n-grams of all the reviews. The n-grams with higher score will be extracted as tags. Graph-based methods (Mihalcea and Tarau, 2004; Altuncu et al., 2019) can also used to extract keywords, where each token is a vertex and an edge is defined when two tokens are in the same context window. Both methods however, fail to group n-grams of similar meaning together.

Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and it’s variants (Yan et al., 2013; Xiong and Guo, 2019) are popular methods to group words

into topics. However LDA processes a document as a bag of words with the assumption that each word is independent of each other. Therefore this method loses valuable occurrence information. Clustering method like k-means, DBSCAN can group similar words based on word embedding. However, word embedding is high dimensional data and clustering fails to work well on it due to the curse of dimensionality.

A neural network model was proposed by He et al. (2017) to group phrases into topics. It overcomes the drawbacks of LDA and clustering methods by utilizing the embedding information with attention mechanism to attend to important tokens in the sentence. We use this model in this paper.

## 3 Category-specific Tag Generation Approach

In this section we describe how the category-specific topic and the product tags are generated. The pipeline is shown in Figure. 1.

### 3.1 Phrase Extraction

We extract phrases from each text review using Stanford NLP’s dependency parser (Manning et al., 2014). Among all the extracted dependencies (Nivre et al., 2016), we choose three kinds as shown in Table 1. These dependencies are about nouns, as the phrases extracted by them are more likely to be about the products. Examples of dependencies that are not selected such as verb, adverb and so on is shown in Table 2.

UDP	meaning	example
<i>amod</i>	adjectival modifier	Sam eats red meat
<i>nsubj</i>	nominal subject	The car is red.
<i>compound</i>	compound	Phone book

Table 1: Universal Dependency Relations (UDP) chosen to extract phrases from product reviews. (<https://universaldependencies.org/u/dep/>) (We show examples in English.)

We further drop phrases which contain stop words derived from NLTK Indonesian stop word list (<https://www.nltk.org/>), and a list that is manually labeled by an internal product team. We only remove stopwords after phrase extraction, since phrase extraction needs the complete sentence input to extract phrases more accurately.

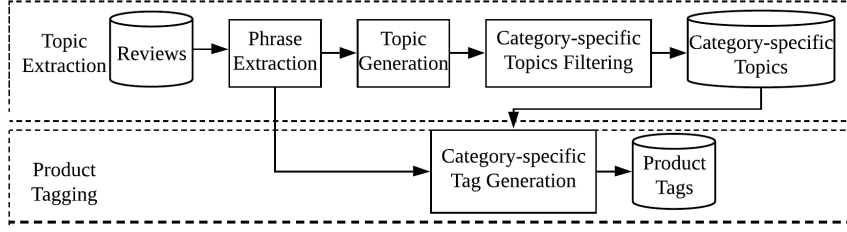


Figure 1: Category-specific topic extraction and product tagging pipelines

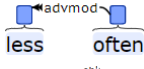
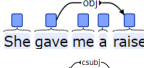
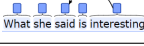
UDP	meaning	example
<i>advmod</i>	adverb modifier	
<i>obj</i>	object	
<i>csubj</i>	casual object	

Table 2: Examples of the dependencies not selected for phrase extraction. (We show examples in English.)

### 3.2 Topic Generation

A topic is a group of phrases sharing a similar concept. Different topics, on the other hand, are separate groups of phrases of different concepts. On the phrases from each product category, we apply the Unsupervised Aspect Extraction (UAE) model (He et al., 2017) to extract topics. The UAE model generates topics by first learning  $K$  topic embeddings, the number of topics  $K$  is predefined. Phrases within a product category are then grouped to the topic that is closest in embedding.

As shown in Figure 2, the model has three layers: the embedding layer, the attention layer and the auto-encoder layer. We concatenate the review

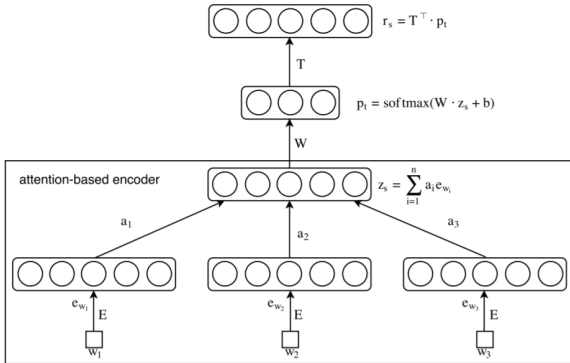


Figure 2: UAE Model Structure

phrases from one product as the input to the embedding layer. The embedding layer is initialized with a word2vec embedding of dimension  $d$ , that is trained on all the reviews of this category. Since

StanfordNLP dependency parser generates phrases with two tokens, concatenating the embeddings of each token in the phrase gives us a phrase embedding of dimension  $2d$ .

The attention layer takes these phrase embeddings, and calculates a weighted sum of the phrases, as  $\mathbf{z}_s = \sum_{i=1}^n a_i \mathbf{e}_{w_i}$ , where  $\mathbf{e}_{w_i} \in \mathbb{R}^{1 \times 2d}$  is the embedding for the  $i^{\text{th}}$  input phrase, and  $a_i$  is the weight computed by the attention layer based on both the relevance of the filtered phrase to the  $K$  aspects and the relevance to the whole sentence which is trained with the following formulas.

$$a_i = \frac{\exp(d_i)}{\sum_{j=1}^n \exp(d_j)}$$

$$d_i = \mathbf{e}_{w_i}^T \cdot \mathbf{M} \cdot \mathbf{y}_s$$

$$\mathbf{y}_s = \frac{1}{n} \sum_{i=1}^n \mathbf{e}_{w_i}$$

In the auto-encoder layer, the encoder compresses  $\mathbf{z}_s$  to a vector of probabilities  $\mathbf{p}_t$  with  $\mathbf{p}_t = \text{softmax}(\mathbf{W} \cdot \mathbf{z}_s + \mathbf{b})$  and the decoder reconstructs a sentence embedding with  $\mathbf{r}_s = \mathbf{T}^T \cdot \mathbf{p}_t$ . Here  $\mathbf{T} \in \mathbb{R}^{K \cdot 2d}$  is the learned aspect embedding matrix, which is in the same embedding space as the phrase embedding.

The loss function of the model is defined as  $L(\theta) = J(\theta) + \lambda U(\theta)$ , where  $\theta$  represents the model parameter,  $J(\theta)$  is proportional to the hinge loss between  $\mathbf{r}_s$  and  $\mathbf{z}_s$ , and  $U(\theta)$  is the regularization term which encourages orthogonality among the rows in the aspect embedding  $T$ .

### 3.3 Category-specific Topic Filtering

Category-specific topics are unique to each product category and not generic. To sift out the general topics from all the generated topics, we use a supervised method.

As the generic topics are similar across all product categories, we made a general word list which contains the frequent words in general phrases. Examples from the general word list are *berfungsi*,

*semoga, bonus, sis, kualitas, oke, super, boss.* (The English translations are *function, hopefully, bonuses, sis, quality, okay, super, boss.*)

A phrase is considered a general phrase if both words in the phrase are in the general word list. If more than a certain percentage  $\eta$  of all the phrases in one topic are general phrases, the topic is considered a general topic, otherwise the topic is a generated category-specific topic, which will be used in the next step.

After supervised filtering, manual labeling is applied to each phrase on the generated category-specific topics. Since we’ve already applied topic extraction and supervised filtering, the number of phrases to be manually labeled is reduced dramatically. For each phrase, we label it either as generic, incoherent or category-specific. Generic phrases are those phrases about general aspects, including delivery, fits description, packing quality, customer service, price. General descriptions about the product quality are also general phrases, these phrases can be used to describe products from most of other categories as well, such as *produk bagus (good product)*. Incoherent phrases are those that are not about the same concept as the majority of the other phrases in the same topic. And category-specific phrases are the phrase about the category-specific aspects of the category, and they are coherent with the majority of the phrases in the same topic.

The category-specific phrases in each topic will be used for tag generation as will be described in Section 3.4. And the frequent words in the generic phrases will be added to the general word list for use in supervised filtering of future topics.

### 3.4 Category-specific Tag Generation

With the filtered category-specific topics, we generate the category-specific tags.

For each product, we group the review phrases to corresponding topics as discussed in Section 3.1 and Section 3.2. We use supervised method shown in Section 3.3 to filter category-specific topics from all the generated topics. Then, we rank the phrases in each topic according to the frequency of phrases in the reviews of this product and choose the one with highest ranking as the tag of this topic for this product. The results are uploaded to a data warehouse.

## 4 Experimental Setup

In this section, we apply our proposed method to product reviews from Tokopedia. We demonstrate the experimental results, and show the evaluation results of the generated category-specific topics.

### 4.1 Data

We use reviews from 89.5 Million products across 18 product categories as the dataset. The average number of reviews in each category, and the average string length of reviews in shown in Table 3 (column: “#reviews” and “average length”).

Category English	# reviews	Average length	# topics	average p@100	topic rate
Handphone Charger	518890	58.03	5	69	100%
Men Sneakers	474767	55.28	3	63	50%
Men Analogue Clock	461819	58.96	5	64	83%
Plant seeds	309374	58.47	7	78	88%

Table 3: Product review statistics and evaluation results for 4 sample categories.

### 4.2 Model Result

After doing phrase extraction, we applied UAE model for topic extraction. We performed the same preprocessing as He et al. (2017) and used word2vec to train the word embeddings with dimension  $d = 200$ . We modified the model structure to accept phrase input as described in Section 3.2, and we shared the same parameter settings as He et al. (2017). We apply our method to each category separately, and we set the number of topics as  $K = 14$  for topic generation. Then, we apply category-specific filter on the extracted topics for all categories with  $\eta = 40\%$ . The general word list we used contains 127 words.

The average time to get generated category-specific topics on extracted phrases is 2 hours per category with around 0.5M reviews. On average, we generate 5 category-specific topics for each category. We show the number of generated category-specific topics for each category in Table 3 (column: “#topics”). We show some of these generated category-specific topics in Table 4.

### 4.3 Evaluation

The most essential part of this work is the automatic generation of category-specific topics. In this section, we show the evaluation results for the quality of the category-specific topic generation.



Category	English	Topic Example
Handphone	Id	android hp,sony hp,lenovo hp,mini ipad
	En	android hp,sony hp,lenovo hp,mini ipad
Charger	Id	sesuai model,sesuai size,sesuai bentuk
	En	fit models, fit sizes, fit shapes
Men Sneakers	Id	automatic jam, pria jam,jutaan jamm
	En	automatic clocks,men clocks,millions of hours
Plant Seeds	Id	semi tumbuh,bismillah tumbuh,daya tumbuh
	En	spring grows, bismillah grows, power grows

Table 4: Example of generated category-specific topics in Indonesian language (Id) for four selected categories and their English (En) translations.

### 4.3.1 Evaluation Metric

An internal product team labeled the results from supervised filtering. They label each phrase as category-specific, general or incoherent as described in Section 3.3. On average, it took one person 3 minutes to label all the phrases of one topic. We apply the evaluation metrics used in He et al. (2017) and Chen et al. (2014). Following their setting, we get the score  $precision@n$  ( $p@n$ ) for each generated category-specific topic, as the number of category-specific phrases among the top  $n$  phrases. We show the average  $p@100$  for sample categories in Table 3 (column: “average  $p@100$ ”). From the result, we can see the majority of the phrases in the generated topics are category-specific in meaning.

We define any topic with  $p@n > 60$  as a category-specific topic, and we define topic rate as

$$\text{topic rate} = \frac{\#\text{category-specific topics}}{\#\text{generated category-specific topics}}$$

We show the topic rate for selected category in Table 3 (column: “topic rate”). We can see more than half of the generated category-specific topics will be selected after manual filtering, thus, human labeling will be very efficient on the automatically generated category-specific topics.

## 5 Conclusion

In this paper, we described a pipeline for category-specific review tagging using phrase extraction, topic generation, category-specific topic filtering and tag generation. Given the product reviews, the pipeline generates the category-specific tags for each product and customers can filter product reviews with these tags. The pipeline is being implemented on product reviews at Tokopedia, and proved to be successful when scaled to large number of reviews. We also evaluated the quality of

the generated category-specific topics with manual labeling and results show that the pipeline can generate coherent category-specific topics.

## References

- M Tarik Altuncu, Eloise Sorin, Joshua D Symons, Erik Mayer, Sophia N Yaliraki, Francesca Toni, and Mauricio Barahona. 2019. Extracting information from free text through unsupervised graph-based clustering: an application to patient incident records. *arXiv preprint arXiv:1909.00183*.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect extraction with automated prior knowledge learning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 347–358, Baltimore, Maryland.
- Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 388–397, Vancouver, Canada.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. *The Stanford CoreNLP natural language processing toolkit*. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain.
- Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC’16)*, pages 1659–1666.
- Ao Xiong and Qing Guo. 2019. Chinese news keyword extraction algorithm based on textrank and topic model. In *International Conference on Artificial Intelligence for Communications and Networks*, pages 334–341. Springer.
- Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1445–1456, Rio de Janeiro, Brazil.