IOP Conf. Series: Materials Science and Engineering **428** (2018) 012021 doi:10.1088/1757-899X/428/1/012021

Harnessing Frequency and Language Features for Keyword Extraction on E-commerce Platforms

Chit Khine[1] and K Nongpong[1]

[1] Intelligent Systems Laboratory (ISL), Vincent Mary School of Science and Technology, Assumption University of Thailand, Bangkok 10240, Thailand

E-mail: ck.light@gmail.com; kwan@scitech.au.edu

Abstract. Automatic keyword extraction has become essential with the growing number of internet resources. This method aims to extract quality keywords that are relevant to products in e-commerce platforms. The problem with e-commerce products is that there are wide ranges of categories in which a product can be in it. We propose a technique to improve the extraction of keywords from products information for the e-commerce systems by applying both frequency-based and language-based features. For frequency-based features, we consider the fact that products in the same category may have popular keywords which are different from other frequency-based features. For language-based features, different types of noun phrases are extracted and ranked accordingly. In this work, the proposed category-based document frequency is combined with the traditional TFIDF and noun phrases ranking. The approach is evaluated using product descriptions from Amazon. The results show that our approach performs better than the traditional TFIDF and RAKE by at least 10 percent on various categories of e-commerce products.

1. Introduction

The requirement for retrieving the precise information of text has become the crucial task in information retrieval. Instead of reading a long paragraph or a document, concise keywords can give us the meaning of the whole information. This statement is also true for products where we need to provide the important information for both customers and providers. Many e-commerce businesses need to associate products with keywords to facilitate product search or for product recommendation. However, manual keyword labelling is a tedious and troublesome process. The main challenge of this problem is how to extract quality keywords that best represent and are most relevant to the products themselves.

Regardless of the domain, the automatic keyword extraction can be divided into three important processes: (1) preprocessing, (2) keyword or keyphrase extraction and (3) keyword evaluation. Pre-processing techniques are usually applied to the documents before performing the actual extraction of the keywords to remove the noises from the documents *i.e.* product descriptions in this case. Pre-processing techniques include stop word removal [1-3], stemming or lemmatization [4-6], and n-gram creation techniques [7-8]. Stop word removal is a technique that filtered out the words that are common in the language and provides very little meaning to the document. Stemming or lemmatization generates base form the words using either the linguistical or morphological analysis of words. "N-gram is a set of n following characters extracted from a word. The main idea behind this

approach is that similar words will have a high quantity of *n*-grams in common. For *n* equals to 2 or 3, the words extracted are called digrams (bigrams) or trigrams, respectively" [2-3].

Extraction techniques are usually categorized into supervised [9] and unsupervised techniques [10]. Supervised techniques can be defined as choosing the best keywords from the prepared set of keywords. Unsupervised keywords are the keywords that are extracted from the structure of the document without requiring the training data. Some popular techniques for keyword extraction techniques are TFIDF [11], Part-of-Speech (POS) keyword extraction [12], RAKE [13] and KEA [13], Noun phrases and named entity recognition [14-15]. The details of these extraction algorithms will be explained Section 2.

Evaluation techniques [16-17] are divided into two categories: *manual evaluation* in which the evaluation is performed by human judges whether the keyphrases represent the document's content and *automatic evaluation* which can be categorized into exact matching, morph matching, part of/include matching). These techniques are used to evaluate the performance of extracted keywords.

2. Related works

The extraction of keywords can usually be categorized into supervised and unsupervised keywords where keywords are domain independent or linguistic approach while others can be statistically extracted. There are also approaches which combine the two approaches mentioned before.

Statistical methods are useful for extracting both single or multiple documents. The techniques for extracting keywords in statistical methods include TFIDF [11] and word-occurrence statistical methods TF [17], which stands for term frequency of words that exist in a single document. These term frequencies can be used to determine the popularity of a term in a document. However, popular words such as articles, common nouns can be seen multiple times in many documents. Therefore, the inverse document frequency is used to rationalize the frequency of the term appearing in whole list of documents. TFIDF combines these two values, tf and idf, to generalize the best key terms in documents. Word-occurrence approach is similar to TF-IDF apart from the fact the keywords are made to check the linking connections with other keywords. The number of times a keyword is occurred next to a certain keyword is used as a factor for calculating the relevancy scores of the keywords.

Linguistics approaches consider the linguistic features of the words, sentences and document [18]. Some popular techniques are using part-of-speech hierarchy [12] and noun phrases [14]. Part of speeches are commonly used in linguistical methods and rules are defined to extract different keywords based on part-of-speech hierarchy [12]. Again, 51 of the 56 part-of-speech patterns in English contains noun tags and most popular patterns from documents are made up of noun-phrases [19]. Therefore, many researchers emphasize on using noun phrases for keyword extraction.

Some popular automatic keyword extraction techniques are RAKE and MAUI [13]. These techniques provide extraction frameworks for evaluating keywords. For instance, RAKE extracts keywords by cutting out the stop words (a, an, the, about, etc) and use the remaining keywords. The rest of the extracted keywords are manually assigned to different groups; hence, different values for the keywords' relevancy scores. These can be created by corresponding algorithms using linguistic features, word co-occurrences, statistically assigned values and normally defining the group types for the extracted keywords. MAUI, an extension of KEA is a supervised machine learning method. This technique includes the candidate selection and machine learning based filtering [1]. For candidate selection, they remove stop words, punctuations and create the n-gram keywords which is followed up by defining candidate features with TFIDF, first occurrence and key-phrase-frequency.

3. Framework

Our proposed framework, as shown in Figure 1, can be divided into two main parts which is keyword segmentation where we extract the keywords from the document and keyword analysis where we provide the relevancy scores for the extracted keywords.

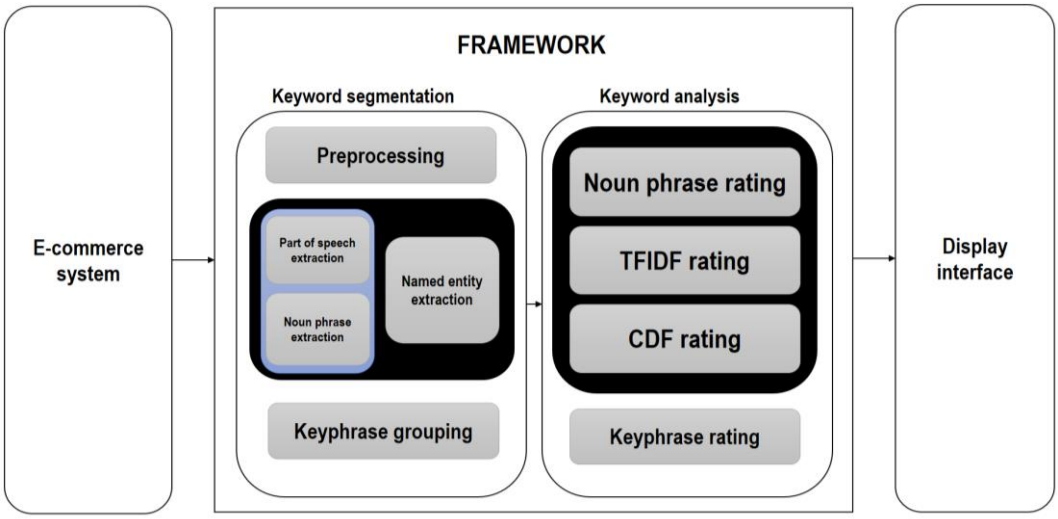

Figure 1. Framework for extracting relevant keywords and keyphrases using noun phrase weighting and frequency based techniques

3.1 Pre-processing
It is preferable to have each document as one long text of one or more sentences. Hence, we perform the formatting of document, line breaks removal and removing noises from the documents such as unnecessary white spaces. Unlike other approaches, standard preprocessing techniques are not performed in this step since we do not want to destroy the structure of sentences before extracting noun phrases. Such operations will be performed after extracting out the keywords. In this work, we assume that the input document must at least contain the product title, product description and the category that such a product belongs to.

3.2 Keyword segmentation
Steps in keyword segmentation includes part-of-speech (POS) extraction, noun phrase extraction, named entity extraction and keyword grouping.

3.2.1 Extraction of part-of-speeches. For part-of-speech extraction, we use the Stanford NLP Parser and POS Tagger to assign part of speech values to each token in the document.

3.2.2 Extraction of noun phrases. Only the noun phrases whose size are not greater than five terms are extracted as keywords. After extracting the noun phrases, we perform the following tasks.

1. Stop word removal – remove stop words which are the same as noun phrases
2. Pronoun removal (e.g. I, we, he, she, they, it)
3. Lemmatization – lemmatize keywords while retaining the part-of-speech of such words.
4. Article removal – remove articles since they have little meaning to the keywords.

3.2.3 Defining named entities. We also consider the fact that noun phrases which include named entities are generally unique in each document; hence, they should be differentiated from non-named entity noun phrases. In this step, Stanford Named Entity Recognizer is used to extract named entities.

3.2.4 Finalize keywords. After performing those three steps above, we attach the properties to the keywords. Such properties are the part-of-speech and a flag for the named entities.

3.3 Keyword analysis
We define two important factors in defining relevance values for our keywords: frequency-based relevancy and language-based relevancy. For keyword analysis, we include three major factors which

are the TFIDF of the keyword, the category-based document frequency of the keyword and the noun phrase rating of the keyword.

3.3.1 Term frequency – inverse document frequency of the keyword. We define two important factors in defining relevance values for our keywords: frequency-based relevancy and language-based relevancy. We calculated the TFIDF of the keyword by using the standard TFIDF formula as shown in equation (1) to get the standard TFIDF score which will get the keywords.

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D) \quad (1)$$

where *tf* = term frequency, *idf* = inverse document frequency, t = term, d = document that the term is inside and D = all documents.

3.3.2 Category-based document frequency of the keyword. We would like to argue that document frequency of a category plays an important role in defining the relevancy values for each keyword. In the normal TFIDF, high document frequency indicates that the token is relatively general in all the documents. However, in this framework, we include the document frequency of a keyword in which the category exists in the document.

$$CDF_{i=0}^{n} = \sum (df(i) / n) \quad (2)$$

Where *df* = document frequency of the category and *n* = the total number of categories

3.3.3 Noun phrase rating of the keyword. Finally, the scores are assigned based on the nature of noun phrases as different types of noun phrases gives different information and impact on the document. For instance, the noun phrase "black-laced Adidas shoe" provides more impact meaning than "black shoe" or "shoe". Table 1 shows the rating assigned to each noun phrase class.

Table 1. Noun phrase ratings

Classes of Noun Phrase	Rating
Noun phrases containing named entities	2
Noun phrases containing adjectives or participles	1.5
Multiple word nouns	1
Single word nouns without proper nouns	0.5

3.3.4 Keyphrase scoring. Keyphrase score is computed by combining all features with the adjustable weights as shown in equation (3).

$$KPR = \beta (TFIDF) + \alpha (CDF / TKC) + \mu (NPR / 2) \quad (3)$$

where KPR = key phrase rating, TFIDF = term frequency – inverse document frequency, CDF = category-based document frequency, TKC = total number of times a keyword is in each category, NPR = noun phrase rating and β, α, μ = weights for each rating (where $\beta + \alpha + \mu = 1$)

The weight used in this work are 0.4 for NPR and 0.3 for both TFIDC and CDF. Though many existing works extracted noun phrases as keywords, this work uses a different approach in keyword rating assignment by taking into consideration of category-based document frequency and noun phrase classes.

4. Evaluation

Keywords extracted from the documents are evaluated using the techniques mentioned in [16]. We have fetched documents from the Amazon dataset [20] which contains product information collected over 11 years. Domain experts are then asked to define golden keywords for each product based on its production description.

4.1 Matching Techniques

We evaluate how effective our proposed technique is by comparing the assigned golden keywords with the keywords extracted by our algorithm using three different matching techniques.

4.1.1 Exact Matching. The extracted keyphrase is considered relevant to the document if it is identical to the golden keyword as shown in equation (4). In other words, "lemon drink" is considered relevant according to exact match if the abstracted keyword is also "lemon drink" without requiring to be case sensitive. Given that

$$\text{Extracted Keyphrase} = e_1\ e_2\ \dots\ e_n \quad (4)$$
$$\text{Golden Keyphrase} = g_1\ g_2\ \dots\ g_n,$$

the extracted keyphrase is considered a match, if $\forall_{1 \le i \le n}, e_i = g_i$.

4.1.2 Morph Matching. The extracted keyphrase is considered relevant to the document if its morphological structure is identical to the golden keyword as shown in equation (5). Both keywords are transformed to their basic forms using lemmatization to check the morph matching. The order of the keywords is important to satisfy this matching. Given that

$$\text{Extracted Keyphrase} = e_1\ e_2\ \dots\ e_n \quad (5)$$
$$\text{Golden Keyphrase} = g_1\ g_2\ \dots\ g_n,$$

the extracted keyphrase is considered a match, if $\forall_{1 \le i \le n}, stemmed(e_i) = stemmed(g_i)$.

4.1.3 Partial Matching. The extracted keyphrase is considered relevant to the document if it is part of the golden keywords as shown in equation (6). Note that stop words are disregarded in partial matching. If a lemmatized form of either golden keyword or extracted keyword is in the keyphrase of the other one, we consider it as partially matched. Given that

$$\text{Extracted Keyphrase} = e_1\ e_2\ \dots\ e_n \quad (6)$$
$$\text{Golden Keyphrase} = g_1\ g_2\ \dots\ g_m,$$

the extracted keyphrase is considered a match, if $\exists_{1 \le i \le n} \exists_{1 \le j \le m}, stemmed(e_i) = stemmed(g_j)$.

From these frequency matching counts, we calculate the precision, recall and f-measures. If the extracted keyword matches with the rule, we consider such a keyword relevant to the document. The precision can be determined by dividing the relevant score by the total extracted keywords while recall can be calculated by dividing the relevant keywords by the total number of keywords.

The evaluation is performed using three different product categories from Amazon products which are Baby, Beauty and Health because each category has different and unique characteristics in the description. This way will show the proposed algorithm's ability to handle various kinds of documents and noises. We decided not to include books in this work because most product information only contain the ISBN instead of a long book description which is hard for the experts to provide golden keywords without going through the actual book contents.

Products in baby and beauty categories have distinct features in which they contain the size/volume/quantity, the year that the product has been created, the materials produced for creating that item, the country that is made from, safety and occasions to be used at. Sometimes, they also include the instructions on how to use the product which contains a lot of unnecessary information for information retrieval. Health products' description contains different kind of information which are symptoms, diagnosis, diseases, experiences, precautions, vitamins and side effects.

4.2 Comparison over baby category

Tables 2, 3 and 4 show the accuracy of the proposed algorithm on the baby category. In this category, our algorithm proves significant advantage in both precision and f-measure in all matching evaluations due to our candidate selection nature between noun phrases and category document frequency. However, the recall was stronger for TFIDF which contains multiple keywords than the other two algorithms.

Table 2. Exact matching in baby category

Algorithm	Precision	Recall	F-measure
Our Algorithm	0.220666	0.323482	0.251575
RAKE	0.127057	0.26284	0.127366
TFIDF	0.033644	0.592813	0.035345

Table 3. Morph matching in baby category

Algorithm	Precision	Recall	F-measure
Our Algorithm	0.232078	0.340088	0.264261
RAKE	0.127366	0.272578	0.165953
TFIDF	0.035345	0.621687	0.065649

Table 4. Partial matching in baby category

Algorithm	Precision	Recall	F-measure
Our Algorithm	0.739721	0.945714	0.814089
RAKE	0.641971	0.986783	0.758592
TFIDF	0.652437	1.000000	0.77257

4.3 Comparison over health category

In Tables 5, 6 and 7, we calculate the results for the health category. In the health category, RAKE gives the best result in exact matching since the RAKE can totally filter out symptoms, diseases without involving the stop words which gives them slight advantage over items which is described using many medical terms. However, in partial matching, our algorithm gives better results than both RAKE and TFIDF since we can generate closer results due to the categorization and noun phrase scoring. Partial matching includes most of the keywords which makes it harder for algorithm to differentiate from one another, but our algorithm shows a slight advantage on precision and f-measure over other algorithms.

Table 5. Exact matching in health category

Algorithm	Precision	Recall	F-measure
Our Algorithm	0.318005	0.3213463	0.319901
RAKE	0.355151	0.344492	0.341751
TFIDF	0.124516	0.33435	0.17625

Table 6. Morph matching in health category

Algorithm	Precision	Recall	F-measure
Our Algorithm	0.831113	0.847931	0.837333
RAKE	0.349222	0.347355	0.341396
TFIDF	0.12242	0.336222	0.174886

Table 7. Exact matching in health category

Algorithm	Precision	Recall	F-measure
Our Algorithm	0.924465	0.938984	0.930288
RAKE	0.88968	0.842659	0.850262
TFIDF	0.86069	0.998042	0.919349

4.4 Comparison over beauty category

The precision, recall and f-measure of beauty category are compared and shown in Tables 8, 9 and 10. Beauty category shows the very similar result as the baby category since the features of the two

categories are very similar. In this category, our algorithm outperforms other techniques in both precision and f-measure.

Table 8. Exact matching in beauty category

Algorithm	Precision	Recall	F-measure
Our Algorithm	0.028766	0.059414	0.037048
RAKE	0.027095	0.058404	0.035414
TFIDF	0.01334	0.087718	0.022605

Table 9. Morph matching in beauty category

Algorithm	Precision	Recall	F-measure
Our Algorithm	0.029197	0.060121	0.037584
RAKE	0.027322	0.059515	0.035792
TFIDF	0.013423	0.088818	0.022758

Table 10. Partial matching in beauty category

Algorithm	Precision	Recall	F-measure
Our Algorithm	0.638428	0.986715	0.754965
RAKE	0.614579	0.985892	0.735427
TFIDF	0.332015	0.99009	0.478469

4.5 Comparison across all categories

The overall performance of automatic keyword extractions for all product categories are compared in Tables 11, 12 and 13. The results show that our algorithm yields better precision and f-measure than other approaches on the exact and partial matchings while outperforming other existing approaches on the morph matching. The evaluation results show that our automatic keyword extraction technique that utilize both frequency-based and language-based features is very promising as it outperforms other techniques in many aspects.

Table 11. Exact matching in all categories

Algorithm	Precision	Recall	F-measure
Our Algorithm	0.189146	0.234747	0.202841
RAKE	0.169768	0.221912	0.168177
TFIDF	0.057167	0.338294	0.078067

Table 12. Morph matching in all categories

Algorithm	Precision	Recall	F-measure
Our Algorithm	0.364129	0.416047	0.379726
RAKE	0.16797	0.226483	0.181047
TFIDF	0.057063	0.348909	0.087764

Table 13. Partial matching in all categories

Algorithm	Precision	Recall	F-measure
Our Algorithm	0.767538	0.957138	0.833114
RAKE	0.71541	0.938445	0.781427
TFIDF	0.615047	0.996044	0.723463

References

[1] Leung A 2016 Evaluating automatic keyword extraction for internet reviews, Master Report, University of Lorraine

[2] Medelyan A 2016 NLP keyword extraction tutorial with RAKE and Maui, [Online]. Available: https://www.airpair.com/nlp/keyword-extraction-tutorial. [Accessed 10 September 2017]
[3] Buckley C 1985 *Implementation of The SMART Information Retrieval System*, Cornell University, Ithaca
[4] Vijayarani S, Ilamathi J, Nithya, Phil M 2015 Preprocessing techniques for text mining - an overview, *Int. J. of Computer Science & Communication Networks*, *5*(3), 7-16
[5] Ramasubramanian C and Ramya R 2013 Effective pre-processing activities in text mining using improved Porter's stemming algorithm, *Int. J. of Advanced Research in Computer and Communication Engineering, 2*(12), ISSN (Online): 2278-1021
[6] Manning C D, Surdeanu M, Bauer J, Finkel J, Bethard S J and McClosky D 2014 The Stanford CoreNLP natural language processing toolkit, *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55-60
[7] Jivani A G et al., A comparative study of stemming algorithms, *Int. J. Comp. Tech. Appl*., 2(6), 1930-1938, ISSN:2229-6093.
[8] Sharma D 2012 Stemming algorithms, a comparative study and their analysis, *Int. J. of Applied Information Systems (IJAIS)* – ISSN: 2249-0868, Foundation of Computer Science FCS, New York, USA, *4*(3), September
[9] Mestrovic A, Beliga, Sanda Martincic-Ipsic, Slobodan 2015 An overview of graph-based keyword extraction methods and approaches, *J. of Information and Organizational Sciences 39*
[10] Sharan A, Siddiqi, Sifatullah 2015 Keyword and key-phrase extraction techniques: A literature review, *Int. J. of Computer Applications 109*
[11] Manning C D, Raghavan P and Schütze H 2009 *An Introduction to Information Retrieval*, Cambridge University, England
[12] Khoury R, Karray F and Kamel M S 2008 Keyword extraction rules based on a part-of-speech hierarchy, *Int. J. of Advanced Media and Communication, 2*(2), ISSN: 1462-4613
[13] Rose S, Engel D, Cramer N and Cowley W 2010 Automatic keyword extraction from individual documents, in Text Mining: Applications and Theory, West Sussex, Wiley, 1-20
[14] Singh Kathait S, Varshney A, Tiwari S, Sharma A 2017 Unsupervised key-phrase extraction using noun phrases, *Int. J. of Computer Applications, 162*(1), 1-5
[15] Lahriri S, Mihalcea R and Lai P H 2015 Keyword extraction from emails, *Int. J. of Natural Language Engineering, 23*(02), 1-23
[16] Zesch T and Gurevych I 2009 Approximate matching for evaluating keyphrase extraction, *Proc. of Recent Advances in Natural Language Processing*, 484-489
[17] Matsuo Y and Ishizuka M 2004 Keyword extraction from a single document using word co-occurrence statistical information, *Int. J. on Artificial Intelligence Tools, 13*(1), 157–169
[18] Dutta A 2016 A novel extension for automatic keyword extraction, *Int. J. of Advanced Research in Computer Science and Software Engineering, 6*(5), ISSN: 2277-128X, 160-163
[19] Hulth A 2003 Improved automatic keyword extraction given more linguistic knowledge, *Proc. of the 2003 conference on Empirical methods in natural language processing*, 216-223
[20] He R and McAuley J 2016 Modeling the visual evolution of fashion trends with one-class collaborative filtering, *Proc. of the 25th Int. Conference on World Wide Web*, 507-517