Expert Systems With Applications 80 (2017) 136–146

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# Enriched LDA (ELDA): Combination of latent Dirichlet allocation with word co-occurrence analysis for aspect extraction

Mohammadreza Shams, Ahmad Baraani-Dastjerdi*

*Department of Software Engineering, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran*

ARTICLE INFO

*Article history:*
Received 6 October 2016
Revised 23 February 2017
Accepted 24 February 2017
Available online 27 February 2017

*Keywords:*
Aspect extraction
Topic modeling
Sentiment analysis
Latent Dirichlet Allocation (LDA)
Co-occurrence relations

ABSTRACT

Aspect extraction is one of the fundamental steps in analyzing the characteristics of opinions, feelings and emotions expressed in textual data provided for a certain topic. Current aspect extraction techniques are mostly based on topic models; however, employing only topic models causes incoherent aspects to be generated. Therefore, this paper aims to discover more precise aspects by incorporating co-occurrence relations as prior domain knowledge into the Latent Dirichlet Allocation (LDA) topic model. In the proposed method, first, the preliminary aspects are generated based on LDA. Then, in an iterative manner, the prior knowledge is extracted automatically from co-occurrence relations and similar aspects of relevant topics. Finally, the extracted knowledge is incorporated into the LDA model. The iterations improve the quality of the extracted aspects.

The competence of the proposed ELDA for the aspect extraction task is evaluated through experiments on two datasets in the English and Persian languages. The experimental results indicate that ELDA not only outperforms the state-of-the-art alternatives in terms of topic coherence and precision, but also has no particular dependency on the written language and can be applied to all languages with reasonable accuracy. Thus, ELDA can impact natural language processing applications, particularly in languages with limited linguistic resources.



## 1. Introduction

The rapidly increasing and tremendous growth in Internet usage as a tool for daily economic activities leads companies towards establishing online shopping websites and utilizing them in order to increase daily sales rates. The huge amount of customer opinions about different products in online shopping solutions can help managers and decision makers to realize costumers' preferences as well as the rates and directions of change in their interests. The process of analyzing customer opinions in order to extract information pertaining to their interests and preferences is known as sentiment analysis (Liu, 2012). Since people may have varying opinions about each product aspect (property), extracting the correct aspect is one of the important challenges in sentiment analysis. Aspect extraction refers to the process of collecting informative words belonging to each subtopic/property of a topic.

There are many studies on aspect extraction, the most promising of which are based on topic models, e.g. latent Dirichlet allocation (Blei, Ng, & Jordan, 2003) and latent semantic indexing (Hofmann, 1999). In these methods, each topic or aspect is defined as a probabilistic distribution of words. Using only topic models produces incoherent topics and aspects that are incompatible with human judgments (Chang, Gerrish, Wang, & Blei, 2009). To address this issue, several knowledge-based topic models have been proposed. They use prior domain knowledge to make the detected aspects more accurate. This prior knowledge can be extracted in either of two ways: semi-automatic or automatic. Semi-automatic extraction makes use of user assistance by defining two sets: must-set, i.e. words that must exist in the aspect; and cannot-set, i.e. words that must not exist in the aspect (Andrzejewski, Zhu, & Craven, 2009; Mukherjee & Liu, 2012). On the other hand, in automatic extraction, the aspect is generated by executing topic models in two steps. First, topic models are employed on the extensive set of opinions about different domains. Then, the output of the first step is used as knowledge to improve the aspects (Abbasi Moghaddam, 2013; Chen, Mukherjee, Hsu, & Castellanos, 2013).

However, in previous studies, prior knowledge is extracted based solely on the distribution of the words while co-occurrence relations are not explicitly taken into account, which leads to the detection of incomplete and incoherent knowledge. Moreover, in most studies, the step of incorporating prior knowledge in the model is as straightforward as the addition of perfect knowledge. However, the possibility of incorrect knowledge is a key challenge in this regard.

* Corresponding author.
*E-mail addresses:* m.shams@eng.ui.ac.ir (M. Shams), ahmadb@eng.ui.ac.ir (A. Baraani-Dastjerdi).

http://dx.doi.org/10.1016/j.eswa.2017.02.038
0957-4174/

Hence, in this paper, an iterative method based on Dirichlet statistical distribution is proposed, which utilizes the advantage of topic models and co-occurrence of words, simultaneously. In the proposed method, the LDA method is applied in order to extract the preliminary aspects of each topic. Then, for each aspect, two types of word are specified, bridge and hub. The hub and bridge extraction is inspired by the idea proposed in Zhang, Wang, Cao, Wang, and Xu (2016).

**Bridges** are words that have the closest relationship with all aspects of a topic. They are general words that can be seen in any kind of aspect. Therefore, these words should be discarded, since they do not represent any specific aspect in the topic. **Hubs** are words that are strongly related to an aspect. In other words, hubs are the best indicators of a specific aspect, since they frequently co-occur with other words relating to an aspect.

After determining the bridge and hub words for each aspect, the most similar aspects in other topics are selected and the prior domain knowledge is generated based on hub words and similar aspects. Finally, the generated knowledge is added to the model in a weighted manner by means of a Bernoulli distribution to reduce the effect of incorrect knowledge. The weight assignment step, as well as the detection of hub and bridge words, is based on the co-occurrence of the words. The proposed method is conducted iteratively to improve the quality of the extracted aspects after each iteration.

Since the LDA model has been enriched with prior domain knowledge, we call the proposed method *Enriched Latent Dirichlet Allocation* (*ELDA*). The advantages and novelty of the proposed ELDA can be listed as follows:

- Language independence: The proposed methodology is language-independent, allowing it to be applied to various languages; unlike most of the works in the natural language processing domain, our proposed method does not rely on any linguistic resources (e.g. ontology or lexicon). Therefore, it is by no means dependent on a specific language. This means that the proposed method can be applied to any language easily. To prove this, in addition to an English dataset, a dataset consisting of more than 100,000 documents about various products in the Persian language is extracted and the proposed method is applied to both datasets and evaluated afterwards.
- Employing the advantages of both the topic model and the co-occurrence of the words simultaneously: each method, topic models and co-occurrence of the words, has its own advantages. Therefore, their combination can improve the aspect extraction results.
- Knowledge validation: in addition to utilizing the topic model and word co-occurrence, knowledge validation is also considered to exclude incorrect and general knowledge.
- Improving aspect extraction for small corpora: the proposed method involves relevant topics to enhance knowledge extraction. Moreover, some filters are utilized to diminish the impact of incorrect knowledge. These make our method suitable in domains with a small number of documents.
- Automatic knowledge generation: unlike most related methods that generate prior knowledge in manual or semi-automatic fashion, the proposed method is fully automatic.

The rest of paper is organized as follows: the second section reviews the previous studies in the aspect extraction domain. Then, Section 3 describes the proposed method and its constituent steps in detail. Section 4 introduces the datasets that are used in evaluation and represents the results of the proposed method compared with the most well-known baselines in the aspect extraction domain. Finally, Section 5 summarizes the paper and outlines areas of interest for future works.

## 2. Previous works

Aspect extraction aims to collect informative words belonging to each subtopic/property of a topic. Recently, this issue has become one of the most attractive research fields in sentiment analysis and other various domains such as information retrieval, natural language processing and text mining (Lin & He, 2009; Marrese-Taylor, Velásquez, & Bravo-Marquez, 2014; Mei, Ling, Wondra, Su, & Zhai, 2007; Rana & Cheah, 2016; Wang, Wang, & Song, 2017; Yuan, Cong, Ma, Sun, & Thalmann, 2013).

Most recent studies in the aspect extraction domain focus on strengthening extracted aspects by incorporating external knowledge. These studies can be divided and evaluated in three categories: topic models, graphs and lexicons. In the following, the three categories are described and their differences from the proposed approach are discussed.

### 2.1. Topic model-based methods

In this category, the methods try to improve the extracted aspects by integrating prior knowledge with the most common topic models such as latent Dirichlet allocation (Blei et al., 2003) and latent semantic indexing (Hofmann, 1999). Initial studies in this regard integrate the knowledge with the topic models in the must-set and cannot be set in manual or semi-automatic ways (Andrzejewski et al., 2009; Mukherjee & Liu, 2012).

However, preparing the so-called knowledge for different topics and datasets using the manual method is quite sophisticated and costly. Recent studies have tried to automatically mine the knowledge based on related datasets in a specific domain (Abbasi Moghaddam, 2013; Chen et al., 2013; Yang, Kotov, Mohan, & Lu, 2015). All of these methods assume that the prior knowledge is flawless. However, in fact, neglecting knowledge validation can result in incoherent aspects.

The methods proposed in Chen and Liu (2014a, b) and Wang, Chen, and Liu (2016) can be considered as the studies most relevant to this work. In these methods, similar aspects are determined by clustering different domains. Then, frequent item sets are extracted from each cluster to be added to the LDA model as prior knowledge. In these methods, simple metrics such as PMI are used to verify the correctness of the knowledge and prevent invalid knowledge. The most important difference between our proposed ELDA and methods in this category is that ELDA utilizes words' co-occurrence to mine prior knowledge. This issue not only causes detection of more aspects but also improves the quality of the aspects. Furthermore, in ELDA, in addition to utilizing word co-occurrence analysis, knowledge validation is also considered to prevent incorrect and general words from being added to the knowledge set.

### 2.2. Lexicon-based methods

Here, lexicon knowledge is used in order to improve the quality of the aspects. As an example, in Ma, Zhang, Yan, and Kim (2013), various aspects of opinions in the Chinese language are extracted using a combination of the LDA model and a lexicon (including synonyms of the words). Similarly, Yao et al. (2016) use information in Wikipedia to detect aspects in news articles. Another approach employs the combination of the PageRank algorithm and lexicons for topic extraction (Yan, Xing, Zhang, & Ma, 2015).

However, the main challenge in this category is dependency on the language in terms of lexicons and other linguistic resources. Conversely, the proposed method has no particular dependency on the written language and can be applied to all languages with reasonable accuracy. To prove this, our proposed ELDA method is evaluated with a dataset in the Persian language. The experimental results will be further discussed in Section 4.

### 2.3. Graph-based methods

In this category, words are formed as a graph, then graph clustering algorithms are used to generate aspects. Furthermore, the co-occurrences of words are used to assign weight to the edges (Sayyadi & Raschid, 2013; Wang, Xu, Hu, & Ohsawa, 2013; Yang, Liu, Lin, & Lin, 2016). Recently, efforts have been made to utilize probabilistic models in addition to co-occurrences (Colace, Casaburi, De Santo, & Greco, 2014; Zhang et al., 2016). For example, in Zhang et al. (2016), words are formed in graph nodes and the weight of each edge is calculated based on the integration of LDA as a topic model and co-occurrence relations. Lastly, a graph clustering method is used to exploit the graph to detect topics. This method has been named LDA-IG.

Although, the proposed ELDA method is a mixed model consisting of a topic model and the co-occurrence of the words, and it is similar to the model proposed in Zhang et al. (2016), there are some differences between the proposed ELDA and LDA-IG.

- Since LDA-IG produces a different number of aspects for each dataset, some of the unrelated aspects should be deleted manually. Conversely, in the proposed method, it is possible to determine the desired number of aspects and words that are clustered based on the number of aspects.
- In LDA-IG, the number of words in each aspect is not balanced. Therefore, it may be possible for an aspect to be shown with only two words, with numerous words for the rest of the aspects. Hence, many of the detected aspects will be incoherent. In contrast, ELDA does not face this limitation.
- In our proposed method, a mixture of the topic model and co-occurrence is exploited iteratively and each iteration improves the quality of aspects, while LDA-IG is a one-step method without any improvement taken into account.
- In LDA-IG, several separate thresholds are needed to specify for graph clustering. So, the outcome results are mostly dependent on the parameter values.

## 3. ELDA method

In this section, our approach is outlined, which is composed of five main phases: (I) preprocessing and applying LDA; (II) hub and bridge extraction; (III) finding similar aspects; (IV) knowledge extraction; and (V) knowledge injection. The overall algorithm is shown in Algorithm 1.

**Algorithm 1**
ELDA method.

Input:
$D = \{D_1, D_2, \ldots, D_n\}$, dataset from n domains;
R, number of iterations
Output:
$T = \{T_1, T_2, \ldots, T_n\}$, set of all aspects for each domain
1- Preprocessing(D)
2- For each domain corpus $D_i \in D$
3- $T_i \leftarrow$ LDA($D_i$)
4- End for
5- $T \leftarrow \cup_i T_i$
6- For iter = 1 to R
7- For each domain corpus $D_i \in D$
8- For each aspect $A_j \in T_i$
9- HubSet($A_j$) $\leftarrow$ Hub&BridgeExtraction($T_i$)
10- SimilarSet($A_j$) $\leftarrow$ FindingSimilarAspects(T)
11- K($A_j$) $\leftarrow$ KnowledgeExtraction(HubSet($A_j$), SimilarSet($A_j$))
12- End for
13- K($T_i$) $\leftarrow \cup_j$ K($A_j$)
14- $T_i \leftarrow$ KnowledgeInjection($D_i$, K($T_i$))
15- End for
16- $T \leftarrow \cup_i T_i$
17- End for

The input of the ELDA method consists of several domain corpora, and the output is the set of all aspects for each domain. A domain ($D_i$) in our case is a collection of documents about a type of product and an aspect refers to the set of informative words belonging to each subtopic/property of a domain. Also, topic $T_i$ denotes the set of all aspects for domain $D_i$.

Lines 2–4 apply LDA on each domain $D_i$ of dataset $D$. The outcomes of the phase are preliminary aspects attached by the probability of each word in the aspect. In line 9, the hub and bridge words of each aspect are extracted as an outcome of the hub and bridge extraction phase. Detection of these words is beneficial in order to detect relative aspects and ensure extraction of useful knowledge.

Afterwards, taking the hub and bridge words into account, similar aspects are selected (line 10). Then, in line 11, the domain knowledge is generated based on the hubs and the similar extracted aspects. The extracted knowledge is integrated into the model in a weighted manner to exclude incorrect knowledge.

Finally, the generated prior knowledge is injected into the LDA model with a Bernoulli distribution, which determines whether the observed word is sampled based on the knowledge set or independent words (line 14). Lines 7–16 iteratively improve the quality of the detected aspects, which will be discussed in the evaluation section. The five phases of Algorithm 1 are explained in more detail in Sections 3.1–3.5, respectively.

### 3.1. Preprocessing and applying LDA

In the first step of the proposed method, basic preprocessing tasks, including stemming, stop words elimination and removing words with very low document frequency, are performed on the dataset. Since preprocessing is commonly used in natural language processing research, any further explanation is omitted.

Afterwards, all documents are decomposed into the constitutive sentences. Assuming each sentence is one independent document, the basic LDA algorithm (Blei et al., 2003) is applied to each domain of the dataset. The outcome of this step is the primary detection of different aspects in each topic, along with the probability of each word.

### 3.2. Hub and bridge extraction

The aspect words can be categorized according to their relation with other words as hubs and bridges. As mentioned before, a bridge is a word that has strong relations with different aspects of a topic. Bridges frequently co-occur with other aspects, which is why they cannot be used as good indicators of specific aspects. Hubs, on the other hand, are words with three main features. 1) They frequently co-occur with other words of an aspect, in which case they can be a suitable indicator of the aspect. 2) Bridge words should not belong to the hub words of an aspect. Bridge words are removed from the hub set because they are not good indicators of a specific aspect and can be seen in any aspect of a topic. 3) The hub words of each aspect should not belong to the hub set of the other aspects. To put it another way, hub words of different aspects of a topic may not include duplicate words. This feature prevents duplicate aspects and, hence, leads to the improvement of aspect extraction.

The hub and bridge extraction phase is inspired by the idea proposed in Zhang et al. (2016). The details of the method and how it differs from the proposed method are discussed in Section 2.3. Since hubs and bridges are detected based on the correlations between words, Eq. 1 shows how the relation of two words *w*

**Algorithm 2**
Hub & bridge extraction.

Input:
$T_i = \{A_1, A_2, ..., A_m\}$, set of all aspects in topic $T_i$
HubNum, number of hub words
BridgeNum, number of bridge words
Output:
$HubSet(A_j) = \{w_1, ..., w_{HunNum}\}$, hub words of aspect $A_j$
1- For each word $w \in A_j$
2- Calculate the bridge weight of word w based on Eq. 2
3- End for
4- $BridgeSet(A_j) \leftarrow$ Select the top BridgeNum words based on bridge weights
5- For each word $w \in A_j$
6- Calculate the hub weight of word w based on Eq. 3
7- End for
8- $HSet(A_j) \leftarrow$ Sort all words based on hub weights
9- $HSet(A_j) \leftarrow HSet(A_j)$ - $BridgeSet(A_j)$
10- For k=1 to k=j-1 do
11- $HSet(A_j) \leftarrow HSet(A_j)$ - $HubSet(A_k)$
12- End for
13- $HubSet(A_j) \leftarrow$ Select the top HubNum words from $HSet(A_j)$.

and $w'$ in topic $T$ can be measured.

$$Rel_T(w, w') = \frac{\sum_{d \in T | (w, w') \in d} \frac{1}{|d|}}{\sum_{d \in T | w' \in d} \frac{1}{|d|}} \tag{1}$$

where $d$ is a document in topic $T$ and $|d|$ is document length. According to Eq. 1, in a case when two words have more co-occurrences in different documents and $w'$ is a more specific word in the topic, the two have a strong relationship with each other. Document length is used to measure relation because, in a large document, a word may frequently co-occur with numerous words with potentially low relation. Therefore, in this equation, co-occurrences between words as well as the specificity of a word, are normalized through document length.

With the help of word relation, bridge and hub words are extracted for each aspect (Algorithm 2). To determine bridge words, the bridge weight of each word is calculated (lines 1–3). After being sorted in descending order, a fixed number of the words is chosen as the bridge set (line 4). Eq. 2 shows how the bridge weight of each word is calculated. This is based on the observation that, if a word is more related to the words of other aspects, it is a good candidate to gain more bridge weight.

$$bridge(w, A_i \in T) = \sum_{A_j \in T, A_j \neq A_i} \sum_{w' \in A_j, w' \neq w} Rel_T(w, w') \tag{2}$$

where $A_i$ is $i$th aspect in topic $T$ and $Rel_T$ is calculated by Eq. 1. In order to find the hub words in each aspect, the hub weight of each word is calculated according to Eq. 3 (lines 5–7). The hub weight is based on the observation that, when a word bears more relation to other words in the aspect, it is more central and, therefore, more important to the aspect. Words are sorted based on the calculated metric and the words with the highest hub weight that do not belong to the bridge set and hub words of other aspects are determined as the hub set (lines 8–13).

$$hub(w, A_i \in T) = \sum_{w' \in A_i, w' \neq w} Rel_T(w, w') \tag{3}$$

For instance, the aspect concerning *CPU-RAM* in a *laptop* includes certain words such as *fast*, *box*, and *notebook* as bridge words. Because the words frequently co-occur with other aspects, they are not good representatives of the aspect. In contrast, other words such as *processor*, *memory*, and *performance* construct the aspect's hub set.

**Algorithm 3**
Finding similar aspects.

Input:
$T = \{T_1, T_2, ..., T_n\}$, set of all aspects for each domain
SimNum, number of similar aspects
Output:
$SimilarSet(A_i \in T_t) = \{A_1, ..., A_{SimNum}\}$, set of similar aspects to $A_i$
1- For each topic $T_k \in T$
2- If($T_k$ != $T_t$)
3- For each aspect $A_j \in T_k$
4- Calculate similarity $A_i$, $A_j$ based on Eq. 4
5- End for
6- End if
7- End for
8- $SimilarSet(A_i) \leftarrow$ Select the top SimNum aspects based on similarity weight

**Algorithm 4**
Knowledge extraction.

Input:
$HubSet(A_i)$, hub words of aspect $A_i$
$SimilarSet(A_i)$, set of similar aspects to $A_i$
Output:
$K(A_i) = \{(h_1, k_1, Rel_T(h_1, k_1)), (h_2, k_2, Rel_T(h_2, k_2)), ...\}$, knowledge set for aspect $A_i$
1- For each word $h \in HubSet(A_i)$
2- For each aspect $A_j \in SimilarSet(A_i)$
3- For each word $k \in A_j$
4- Calculate $Rel_T(h,k)$ based on Eq. 1.
5- $K(A_i) \leftarrow K(A_i) \cup (h, k, Rel_T(h,k))$
6- End for
7- End for
8- End for

### 3.3. Finding similar aspects

Following the detection of hub words, similar aspects are extracted for knowledge construction. Algorithm 3 is run on other topics to find similarities to aspect $A_i$. To do this, in lines 1–7 by using Eq. 4, the similarity between $A_i$ and other aspects is computed.

$$similarity(A_i \in T', A_j \in T) = \sum_{w \in hubset(A_i)} \sum_{w' \in A_j, w' \neq w} Rel_T(w, w') \tag{4}$$

where $A_i$ and $A_j$ are different aspects and $hubset(A_i)$ indicates the set of words that is selected as the hub set for aspect $A_i$. Aspect similarity is based on the assumption that similar and related aspects tend to have more related hub words than expected. The output of this phase is the set of similar aspects to aspect $A_i$, which refers to the set of top-ranked aspects based on similarity weight (line 8). For example, the aspects related to *CPU* or *RAM* in *cellphone* or *tablet*, are chosen as the similar set in the aspect concerning *CPU-RAM* of *laptop* domain.

### 3.4. Knowledge extraction

In the knowledge extraction phase, the domain knowledge for each aspect is generated using hub words and the extracted similar aspects. The knowledge is produced as pairs ($h$,$k$) consisting of a hub word together with each word of any similar aspects. The knowledge extraction phase is introduced in Algorithm 4.

However, the existing challenge in this regard involves the possibility of incorrect and general knowledge. For example, considering an aspect in the field of a *wireless network*, it is possible that the existence of *wireless* as a hub word can cause similar aspects to be found such as *wireless mouse* and, hence, a knowledge pair such as (*wireless, mouse*) or (*wireless, device*) is incorrectly added to the *wireless network* aspect. In order to prevent incorrect knowledge

**Table 1**
Model parameters.

| | |
|---|---|
| $N_D$ | Number of documents in domain D |
| $N_d$ | Number of words in document d |
| $N_T$ | Number of aspects in a topic |
| V | Number of distinct words |
| A, W, X | Aspect, word and word dependency, respectively |
| K | Knowledge pair |
| $\theta$, $\varphi$ | Multinomial distribution over aspects and words, respectively |
| $\psi$ | Bernoulli process over X |
| $\alpha$, $\beta$, $\delta$ | hyper-parameters of the model |

injection, each knowledge set is assigned a weight proportional to the relation with its domain.

Two features are considered for weighting. The first deals with incorrect knowledge; for example, in the aspect *wireless network*, a knowledge pair such as (*wireless, mouse*) should not enter this aspect having a high weight. Because, under the circumstances, words related to *mouse* are added to the *wireless network* aspect eventually making the aspect incoherent.

The second feature in the weighting phase prevents the inclusion of general knowledge, which refers to the type of knowledge without much information about the aspect. For instance, the pair (*wireless, device*) is an example of general knowledge. *Device* is a general concept and its inclusion in a particular aspect not only adds no new valuable information to the aspect, but also may cause incoherence.

In order to address the aforementioned features, the weight value is calculated based on the similarity and specialty of the knowledge pair. The weight of each knowledge pair (*h, k*) is calculated by Eq. 1 (line 4). According to Eq. 1, whichever two words tend to appear together in more documents than expected, and word *k* is more specific in the topic, the pair is more informative. The weighted knowledge pairs ($h$, $k$, $Rel_T(h, k)$) are passed to the knowledge injection phase.

### 3.5. Knowledge injection

In the final phase, the generated knowledge is injected into the LDA model. Several studies address this issue mostly by adding the generated knowledge to the model in a manual or semi-automatic fashion. In these methods, since the knowledge is added to system by an expert, the probability of failure is low, which is why validation of the knowledge is not needed. Thus, in the methods, the extracted aspects strongly overfit to the prior knowledge.

LTM (Chen & Liu, 2014b) tries to automatically handle wrong knowledge by setting prior weight using the PMI of words. However, since the words are only sampled from prior knowledge, there is a possibility of overfitting, which deteriorates the quality of aspects.

Although dealing automatically with knowledge mismatches is particularly difficult, the proposed ELDA adopts LTM and CTM (Yang, Wen, Kinshuk, Chen, & Sutinen, 2015), which are useful for the knowledge injection task. In ELDA, to involve prior knowledge automatically as well as to avoid overfitting, sampling can be implemented from either the knowledge pair or an independent word. To this end, word dependency is sampled from hidden variable $X_i$ as a result of a Bernoulli process. Under this circumstance, $W_i$ can influence $W_j$, and the knowledge pair ($W_i$, $W_j$) is injected into the model if and only if $X_i$ is observed. The diagram of the knowledge injection phase is presented in Fig. 1. Also, Table 1 describes the model parameters.

As shown in Fig. 1, ELDA uses Gibbs sampling for word selection. The sampling process has the following main steps: (1) sampling from either knowledge pair or independent word is determined by the Bernoulli process $\psi$ with parameter $\delta$; (2) the probability of aspect $A_i$ in document *d* is modeled by $\theta$ with the Dirichlet hyperparameter $\alpha$; and (3) the distribution $\varphi$ models the probability of words associated with aspect $A_i$ and knowledge pair *K* with $\beta$ as the Dirichlet hyperparameter. In this way, the definition of distribution for aspect extraction can be expressed as Eq. 5.

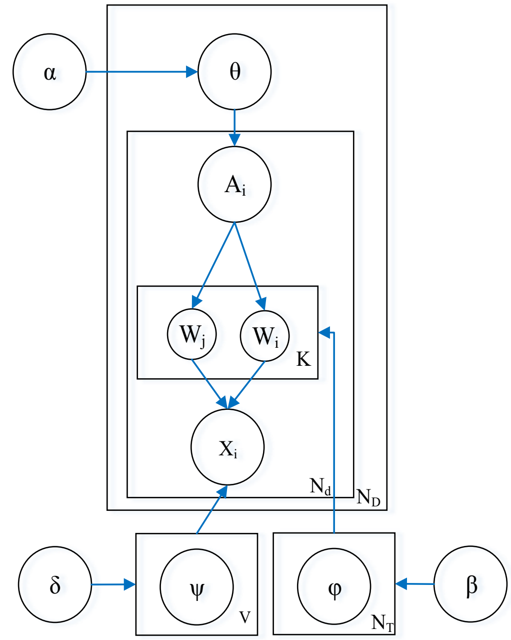


**Fig. 1.** Knowledge injection phase.

$$P(A_{d,i}|A_{-(d,i)}, X, W, K, \alpha, \beta, \delta) \propto P(A_{d,i}\ |A_{-(d,i)}, \alpha) \times \begin{cases} P(W_{d,i}\ |A, W_{-(d,i)}, \beta)\ if\left(X_{d,i}=0\right) \\ P(W_{d,i}\ |A, K, W_{-(d,i)}, \beta)\ if\left(X_{d,i}=1\right) \end{cases} \tag{5}$$

In Eq. 5, $A_{d,i}$ represents the aspects associated with word *i* in document *d*. $X_{d,i}$ is the dependency variable for the *i*th word in document *d*. If $X_{d,i}=1$ then *i*th word sampled from a knowledge set and $X_{d,i}=0$ means that the word is independent. $A_{-(d,i)}$ is an indicator of the aspect assignments except aspect $A_{d,i}$, and $X_{-(d,i)}$ is also defined in this way. The approximate conditional distribution can be calculated using Eq. 6.

$$P(A_{d,i}|A_{-(d,i)}, X, W, K, \alpha, \beta, \delta) \propto \left(\frac{n_{d,a,-i}+\ \alpha}{\sum_{a'=1}^{N_T}\left(n_{d,a',-i}+\ \alpha\right)}\right) \times \begin{cases} \frac{n_{w,a,-i}+\ \beta}{\sum_{w'=1}^{V}\left(n_{w',a,-i}+\ \beta\right)} & if\left(X_{d,i}=0\right) \\ \frac{\sum_{w'=1}^{V} Rel_T(w,w')\times n_{w',a,-i}+\ \beta}{\sum_{v=1}^{V}\left(\sum_{w'=1}^{V} Rel_T(v,w')\times n_{w',a,-i}+\ \beta\right)} & if\left(X_{d,i}=1\right) \end{cases} \tag{6}$$

In Eq. 6, $n_{d,a,-i}$ is the number of words with aspect *a* in document *d*, except the current word; $n_{w,a,-i}$ is the number of occurrences of the word *w* in aspect *a*, except the current word; and $Rel_T(w, w')$ is the produced weight of the knowledge pair from Eq. 1, if $(w, w')$ is a knowledge pair in the current aspect; otherwise, it is set to zero. Eq. 7 is used to find the probability of sampling a word from the knowledge pair or independently.

$$p(\ X_{d,i}=1\ |X_{-(d,i)}, \delta) = \ \psi\ _{d,i}^{X_{d,i}}\left(1-\psi_{d,i}\right)^{1-X_{d,i}} \tag{7}$$

where $\psi_{d,\ i}$ is the posterior of a Beta prior $\delta|\ \delta_0$, $\delta_1$, and it is estimated by Eq. 8, as in Yang and Wen et al. (2015).

$$\psi_{d,i} = \frac{\delta_c + q_{i,c}}{\sum_{c=0}^{1}\left(\delta_c + q_{i,c}\right)} \tag{8}$$

**Algorithm 5**
Knowledge injection.

```
Input:
 D_i, a domain corpus
 K(T_i), knowledge sets for topic T_i
 N, number of iterations for Gibbs sampling
Output:
 T_i={A_1, A_2, ..., A_n}, set of all aspects in topic T_i
1- For iter=0 to N/5 do
2- T_i ← GibbsSampling(D_i) //Run Gibbs sampling with no knowledge injection
3- End for
4- For iter=N/5 to N do
5- T_i ← GibbsSampling(D_i,K(T_i)) //Run Gibbs sampling with knowledge injection based on Eq. 6
6- End for
```

where $\delta_c$ is the hyper-parameter for Beta prior and $q_{i,\,c}$ is the number of variable $X_{d,i} = c$ (0 or 1) given the $i$th word. There is a slight difference between how Eq. 8 is solved in our ELDA method and the procedure prescribed by Yang and Wen et al. (2015). In Yang and Wen et al. (2015), $X_{d,i} = 1$ means that the $i$th word is sampled from bigram. On the other hand, in ELDA, the word is sampled from a knowledge pair that can occur separately at different locations on a sentence. Please note that, contrary to the other phases, $\psi_{d,\,i}$ is calculated at the document level and the obtained value is used for all the sentences in a document.

According to Eq. 8, for whichever word has been used more as a knowledge pair in sentences of a document, the probability of occurrence $X_{d,i} = 1$ will be more and vice versa, so that the more independent occurrence of a word causes higher probability of $X_{d,i} = 0$.

Algorithm 5 shows the pseudocode for the knowledge injection phase. Lines 1–3 apply the basic Gibbs sampler for $N/5$ iterations to make a set of primary aspects. Then, other iterations inject extracted knowledge into the model (lines 4–6).

Finally, after finishing the knowledge injection phase, the aspects are combined with the prior knowledge in a weighted manner and enriched aspects are formed. As previously indicated in Algorithm 1, the four final phases (Sections 3.2–3.5) of the proposed ELDA method are executed iteratively and, in each iteration, the extracted aspects are improved.

## 4. Evaluation

In this section, the proposed ELDA method is compared to four state-of-the-art baselines:

- LDA (Blei et al., 2003): The LDA topic model is an unsupervised method that does not use prior knowledge.
- DF-LDA (Andrzejewski et al., 2009): A semi-automatic method that incorporates the generated knowledge with the LDA model manually. This method does not provide any validation analysis of the input knowledge. It should be mentioned that the input knowledge of the DF-LDA algorithm must be entered manually by the user. So, in order to compare the algorithm with the proposed method, the generated knowledge by the ELDA method is given as input to DF-LDA.
- LTM (Chen & Liu, 2014b): In this model, the prior knowledge is integrated into the topic model in the form of frequent item sets and the MI metrics are used in order to exclude irrelevant knowledge.
- LDA-IG (Zhang et al., 2016): A graph-based model for topic detection that combines LDA and co-occurrence of the words.

### 4.1. Datasets

To evaluate our proposed ELDA, two different datasets are employed. First, a Persian dataset is crawled from different online shopping websites in the Persian language, consisting of 44 different product domains. Altogether, approximately 100,000 different documents are collected from 44 product domains.[1] The list of domains and the number of documents in each category, along with the average number of words per document, are listed in Table 2. Second, the English dataset used in LTM (Dataset1K) contains 50 domains from Amazon.com where each domain has 1000 reviews. More information about the dataset can be found in Chen and Liu (2014b).

### 4.2. Parameter initialization and primary settings

The ELDA model runs for 1000 iterations for the Gibbs sampler. The hyper-parameters $\alpha$ and $\beta$ are set to 1 and 0.1 respectively, the same values are in Chen and Liu, (2014a,b), Chen et al. (2013), and Mukherjee and Liu (2012). The parameter $\delta$ is set to 0.4, based on our preliminary experiments. Different executions of ELDA indicate that small variations in the mentioned parameters do not affect the result. Please note that since the English dataset consists of discrete sentences wherein their source document is not specified, the results of ELDA on the English dataset are obtained by sampling only from knowledge pairs ($X_{d,i} = 1$).

The numbers of hub and bridge words are selected as five and three, respectively; in each cycle, two similar aspects are used in order to extract the knowledge. Also, 15 aspects per topic are extracted and the 20 top words are characterized for each aspect. Sensitivity analysis of knowledge extraction parameters will be discussed in Section 4.6. It should be noted that, for the baselines, the values of required parameters and the iterations of steps are set according to the reported values in their original articles.

### 4.3. Topic coherence

In order to evaluate and compare the ELDA method with the introduced baselines (i.e. LDA, DF-LDA, LTM, and LDA-IG), the topic coherence metric is measured (Mimno, Wallach, Talley, Leenders, & McCallum, 2011). This metric can be calculated by Eq. 9.

$$C\left(A, V^{(a)}\right) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log_e \frac{CODF\left(v_m^{(a)}, v_l^{(a)}\right) + 1}{DF\left(v_l^{(a)}\right)} \tag{9}$$

In Eq. 9, $V^{(a)} = (v_1^{(a)}, \ldots, v_M^{(a)})$ is the list of $M$ words with high probability in the aspect, $DF$ is the document frequency, and $CODF$ is the co-occurrence of two words in different documents. The higher value of the topic coherence metric indicates the better quality of the extracted aspects. In Fig. 2, the experimental results of the proposed ELDA and baselines on the Persian and English datasets can be seen.

[1] This dataset will be published freely via the laboratory websites to be used later by other researchers.

**Table 2**
List of 44 domain names.

| # | Category | #docs | AVG. words | # | Category | #docs | AVG. words |
|---|---|---|---|---|---|---|---|
| 1 | Laptop | 3109 | 1081 | 2 | Phone Case | 1786 | 155 |
| 3 | Cell Phone | 1693 | 1393 | 4 | Computer | 2140 | 392 |
| 5 | Networking | 1428 | 417 | 6 | Laptop Accessories | 2889 | 261 |
| 7 | Tablets | 1749 | 811 | 8 | Computer Accessories | 1115 | 242 |
| 9 | Camera | 1402 | 1011 | 10 | Office Supplies | 1731 | 342 |
| 11 | Cell Phone Accessories | 2016 | 282 | 12 | Camera Accessories | 1764 | 218 |
| 13 | Drives & Storage | 1570 | 384 | 14 | Bedding & Bath | 1732 | 152 |
| 15 | Home Décor | 2615 | 138 | 16 | Kitchen & Dining | 3792 | 147 |
| 17 | Home Audio & Theater | 2438 | 578 | 18 | Electrical Appliances | 2334 | 254 |
| 19 | Food Service | 2767 | 204 | 20 | Power & Hand Tools | 1265 | 336 |
| 21 | Washing & Cleaning | 1352 | 209 | 22 | Personal Care | 2074 | 295 |
| 23 | Luxury Beauty | 3623 | 257 | 24 | Health Care Tools | 1058 | 362 |
| 25 | Fragrance | 2315 | 284 | 26 | Electrical Personal Care | 756 | 379 |
| 27 | Magazines | 2163 | 195 | 28 | Books | 11,946 | 188 |
| 29 | Musical Instruments | 1824 | 340 | 30 | Arts & Crafts | 2362 | 259 |
| 31 | Software & Games | 1564 | 313 | 32 | Gifts | 3631 | 181 |
| 33 | Writing Instruments | 1278 | 220 | 34 | Jewelry | 3465 | 214 |
| 35 | Baby Clothing | 1302 | 168 | 36 | Entertainment | 3315 | 192 |
| 37 | Baby & Child Care | 1474 | 226 | 38 | Toys & Games | 2344 | 240 |
| 39 | Athletic Clothing | 1872 | 319 | 40 | Bags & Backpacks | 1165 | 283 |
| 41 | Camping & Hiking | 1360 | 334 | 42 | Shoes | 1436 | 559 |
| 43 | Sports & Fitness | 1460 | 375 | 44 | Watches | 4968 | 223 |

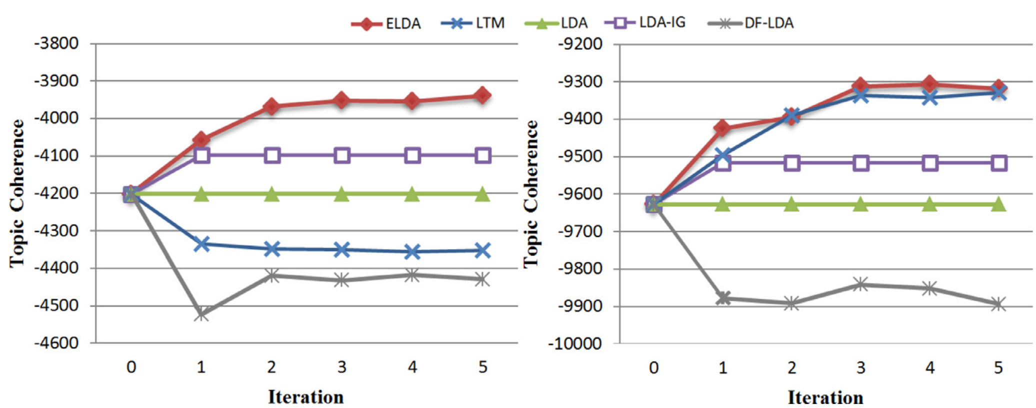


**Fig. 2.** Topic coherence of each model on the Persian (left) and English (right) corpora.

In all of the methods, the evaluation is started from LDA, which is indeed the common initial step of the methods. Since the LDA-IG method is based on the graph and does not have iterations, all iterations yield the same results. It should also be noted that, in order to keep the method consistent with other methods, a zero step is considered. During the execution of LDA-IG, the aspects for each topic are found separately, but, for extracting the words relations, the whole dataset is used.

As can be seen in Fig. 2, the weakest method is DF-LDA, which is due to a lack of validation of prior knowledge. Since the knowledge is generated automatically and may contain errors, so the arrival of knowledge without validation can cause a weakening of the results.

Although LTM is efficient (close to ELDA) as far as English are concerned, it does not produce acceptable results for the Persian dataset because of general knowledge. In LTM, first, the aspects are clustered and then frequent item sets are extracted as prior knowledge. This may decrease the quality of aspects, particularly in the case of datasets that contain the same words with high frequency in all topics (general knowledge).

LDA-IG, which combines the co-occurrence of words and Dirichlet distribution, achieves better results than both LDA and DF-LDA. This proves that the idea of combining co-occurrence and the topic model can be useful. As can be seen the results of ELDA is better than LDA-IG.

The final point concerns fluctuations in topic coherence (e.g. between iterations 3 and 4) of the proposed method. This is due to changes in the hub words of each aspect, which causes changes in input knowledge. Despite slight modifications, the proposed method is still superior to the other methods and generates promising results even in the worst-case cycle.

### 4.4. Human evaluation

In order to have a more precise evaluation of the ELDA method, the existing aspects in each topic are identified by three experts who are aware of the specification of products. Then, the top 20 words extracted using the ELDA and two baseline methods, i.e. LDA and LTM, are labeled manually as being either related or unrelated. The ultimate label for each case is obtained via the majority of opinions.

Since the extracted aspect words are ranked and the total number of correct aspect words remains unknown, *Precision@n* (where $n$ is a rank position) is used for evaluation. It should be mentioned that the efficiency of LDA-IG and DF-LDA are not evaluated in this subsection. LDA-IG does not rank aspect words; therefore, it is clearly impossible to calculate *Precision@n*. Besides, the topic coherency of DF-LDA is lower than that of the other two baselines; hence, the method is ignored.

In Fig. 3 (top), *Precision@*10 for different topics is calculated for the Persian dataset. The topics are enumerated according to

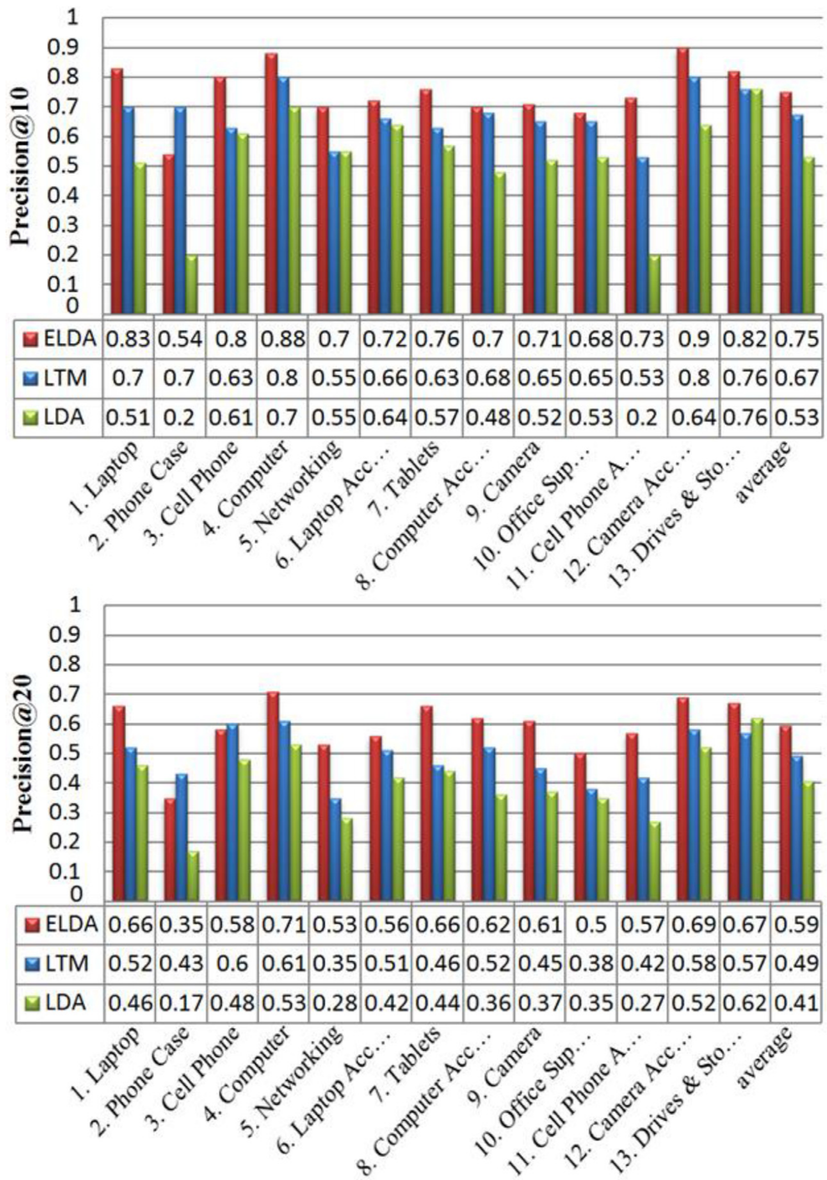


**Fig. 3.** *Precision*@10 (top) and *Precision*@20 (bottom) of the three models on the Persian dataset (the topics are enumerated according to Table 2).

Table 2. As can be seen, most of the time, the proposed method performs better than the two baseline methods, with, on average, 8% and 22% better accuracy compared to LTM and LDA, respectively. This improvement is due to the fact that the co-occurrence relations could enrich the extracted aspects and incorrect knowledge is also well excluded.

As can be seen in Fig. 3 (bottom), with increasing the rank position ($n$), the precision decreases in all of the methods, but the proposed method is still much better than the two baseline methods. The proposed method is approximately 10% and 18% better than the LTM and LDA methods, respectively, in aspect extraction.

In the following, the proposed method is also evaluated for the English dataset. To do so, the aspect selection routine and labeling of the words are repeated for eight out of the 50 available domains of English dataset. *Precision*@10 and *Precision*@20 are then calculated for the eight domains, as shown in Fig. 4.

Fig. 4 shows the superiority of ELDA in word selection for different aspects in such a way that, on average, for *Precision*@10 and *Precision*@20, the proposed method performs 18% and 15% better than LTM and 28% and 23% better than LDA.

Paired $t$-tests confirmed that the improvements of ELDA compared to all baselines are significant (all $p < 0.01$). The evaluations also show that the proposed method is independent from linguistic resources and applicable to all languages.

As another evaluation, in Table 3, a list of the top 10 words in some extracted aspects concerning *laptops* is provided by ELDA and LTM. The red marks are the words that are detected incorrectly by each method.

As clearly evident from Table 3, in most aspects, ELDA generates better words; in particular, the first word that has the highest probability in each aspect is the best word in ELDA. The problem of general knowledge, such *netbook* and *program* in the *CPU-RAM* aspect or *movie* and *music* in the *speaker* aspect, can be seen in the LTM extracted words.

### *4.5. Evaluation on small corpora*

In ELDA, the aspects of each topic are enriched using the knowledge extracted from similar aspects of other topics. This knowledge may be used to enhance aspect extraction in domains with lim-

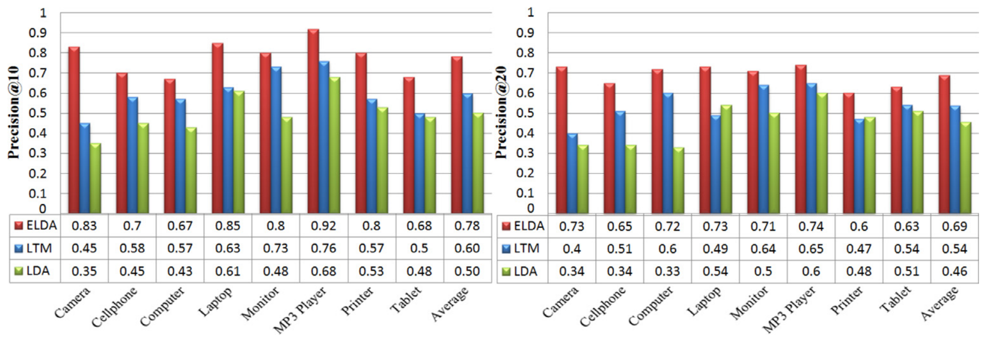


**Fig. 4.** *Precision*@10 (left) and *Precision*@20 (right) of the three models on the English dataset.

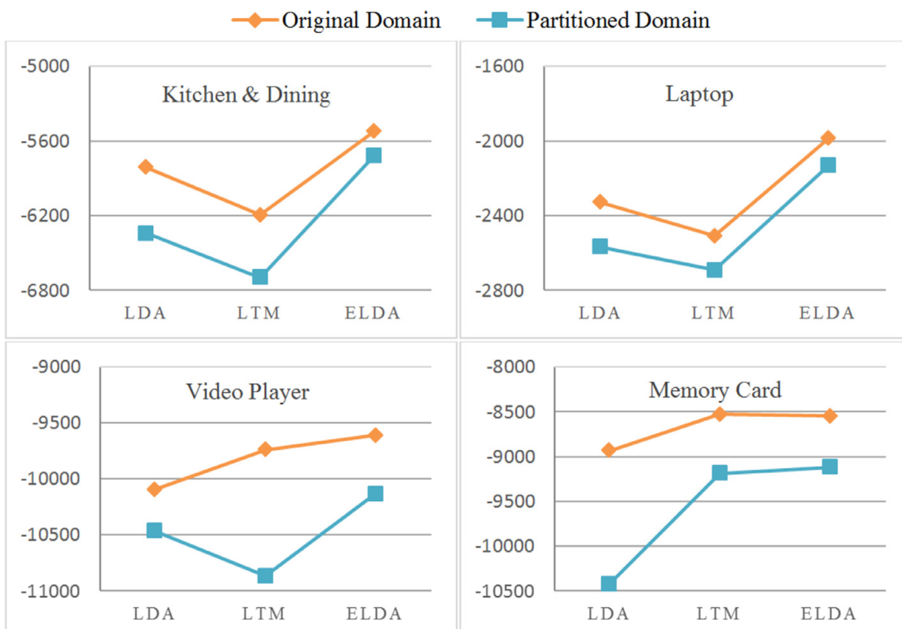


**Fig. 5.** Average topic coherence of three models tested on original/partitioned domains taken from the Persian (top) and English (bottom) datasets.

**Table 3**
Example aspects from LTM and ELDA in the *Laptop* domain. Errors are italicized in red.

| CPU-RAM | | Screen | | Software | | Speaker | | Fan | | Battery | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ELDA | LTM | ELDA | LTM | ELDA | LTM | ELDA | LTM | ELDA | LTM | ELDA | LTM |
| ram | ram | screen | screen | windows | vista | speaker | *movie* | fan | fan | battery | battery |
| gig | machine | bright | *macbook* | mac | website | sound | *player* | cooler | cooler | life | life |
| memory | *program* | inch | *keyboard* | software | windows | bass | music | quiet | hot | hour | hour |
| gb | *window* | black | *air* | operating | xp | high | *dvd* | hot | *bottom* | capacity | charge |
| ghz | *netbook* | size | *mouse* | xp | driver | music | *file* | cool | cool | *dell* | long |
| intel | *desktop* | clear | *pro* | *pc* | software | low | *tv* | speed | *pad* | charge | power |
| processor | *easy* | display | *apple* | window | mac | *crisp* | *video* | cooling | *top* | charger | *single* |
| machine | performance | *feature* | inch | program | *pc* | volume | speaker | *bottom* | *lap* | power | hr |
| performance | fast | large | *full* | system | *cd* | clear | volume | metal | *con* | hr | capacity |
| cpu | box | *case* | *key* | user | system | *end* | bass | noise | quiet | minute | saver |

ited datasets. To assess the efficiency of the proposed method with respect to domains with a small number of documents, two domains with minimum and maximum coherency are selected from the Persian (i.e. *Kitchen&Dining* and *Laptop*), and English (i.e. *Video Player* and *Memory Card*) datasets. Each domain is then randomly divided into 10 equal partitions (e.g. 3100 documents in *Laptop* are divided into ten parts having 310 documents). Thereafter, the partitions iteratively replace the original domain and ELDA is applied to the new dataset. After 10 iterations, average coherency is calculated based on the original (undivided) dataset, as shown in Fig. 5. The evaluation is inspired by Chen and Liu (2014b).

Next, for the four topics, the top 20 words extracted using ELDA and LTM are labeled according to the procedure in Section 4.4. Based on the generated labels, *Precision*@20 is examined for existing words in the aspects and the results are shown in Fig. 6.

As evident from Figs. 5 and 6, by decreasing the size of each domain, an overall reduction of topic coherences is observed. However, ELDA is able to obtain knowledge from similar aspects of other topics to compensate for the small number of documents. Ultimately, the results demonstrate the effectiveness of ELDA in small corpora.

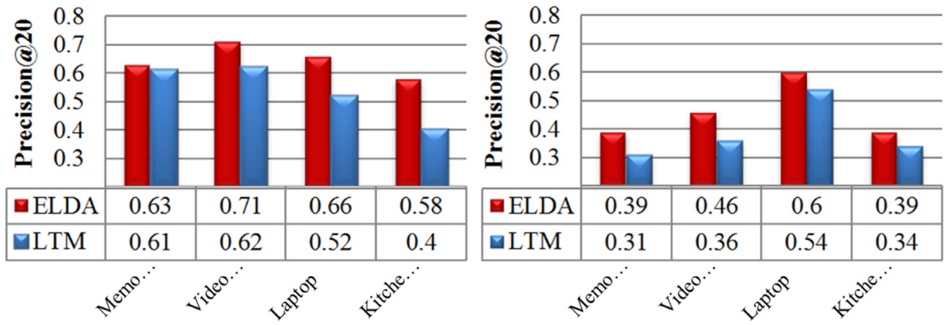


**Fig. 6.** *Precision*@20 of ELDA and LTM tested on the original (left) and partitioned (right) domains.

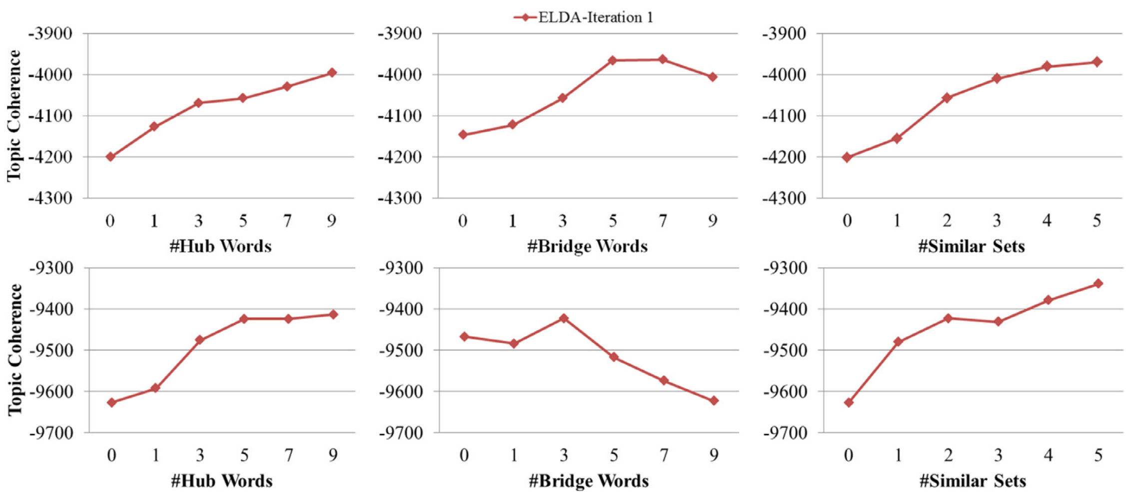


**Fig. 7.** Sensitivity analysis of ELDA tested on Persian (top) and English (bottom) datasets.

### 4.6. Sensitivity analysis of knowledge extraction parameters

The final evaluation concerns parameter sensitivity of ELDA in regard to knowledge extraction parameters. In each experiment, all but one of the parameters are initialized with the original values in Section 4.2. Topic coherence is then measured for different target parameter values.

As shown in Fig. 7, small values of the hub, bridge, and similar sets lead to a dramatic drop in topic coherence; this is due to the selection of inappropriate and insufficient words to enrich each aspect. Accordingly, if the number of hub words or similar sets is reduced to zero, the topic coherence of ELDA coincides with that of the basic LDA. In contrast, ELDA is sufficiently robust against relativity large values for the hub words and similar set because, despite having many knowledge pairs, weighting and validation steps exclude invalid and global knowledge.

## 5. Conclusion

In this paper, a method based on incorporating prior domain knowledge into the LDA topic model for aspect extraction was proposed. In the proposed method, the LDA topic model is combined with co-occurrence of the words. In this way, in an iterative algorithm in each cycle based on co-occurrence, prior knowledge from similar aspects is extracted and this knowledge is added to the LDA model in knowledge pair sets. The experimental results of the algorithm indicate that addition of the knowledge in different cycles improves the quality of aspects and causes the aspect to become closer to human judgment. The proposed method does not depend on the writing language of the text and is applicable to various datasets.

Although our proposed ELDA has shown potential in the aspect extraction task, several limitations have to be considered for further improvement. First, the runtime of ELDA increases linearly with the number of documents, like LTM and LDA. However, ELDA tends to have a larger runtime because it requires a relatively costly knowledge generation step, unlike the mentioned baselines. Nevertheless, the generated knowledge clearly improves the quality of the extracted aspects. Second, a list of relevant topics is needed for proper application of ELDA. Since the required knowledge for enriching each aspect is extracted from the similar aspects of other topics, a dataset of related topics is necessary in order to achieve sufficient efficiency. Suppose a dataset includes only three topics, namely *Laptop*, *Food Service*, and *Baby Clothing*. The results of the proposed method are not very different from those of the basic LDA because none of the topics include aspects that enhance other topics. (Indeed, the three topics do have some common aspects such as *price* or *warranty*, though they share no specific aspects.) Please note that, given the vast amounts of data on the Internet and the large number of products hosted on online shopping websites (as the primary application of the proposed method), the limitation is expected to be easily resolved.

For future works, it would be beneficial to add other types of contextual information such as n-grams, dependency relations, and similarity between sentences in order to strengthen the knowledge extraction phase. In addition, ELDA can be leveraged to rank the sentiment words for each individual aspect. Finally the proposed method can be extended to different tasks such as aspect-based sentiment analysis, text summarization, and entity ranking.

## References


Abbasi Moghaddam, S. (2013). *Aspect-Based Opinion Mining in Online Reviews*. Canada: Simon Fraser University.

Andrzejewski, D., Zhu, X., & Craven, M. (2009). Incorporating domain knowledge into topic modeling via Dirichlet Forest priors. In *Proceedings of the 26th annual international conference on machine learning - ICML '09: Vol. 382* (pp. 1–8). http://doi.org/10.1145/1553374.1553378.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Chang, J., Gerrish, S., Wang, C., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems* (pp. 288–296).

Chen, Z., & Liu, B. (2014a). Mining topics in documents: Standing on the shoulders of big data. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining - KDD '14* (pp. 1116–1125). http://doi.org/10.1145/2623330.2623622.

Chen, Z., & Liu, B. (2014b). Topic modeling using topics from many domains, lifelong learning and big data. In *Proceedings of the 31st international conference on machine learning - ICML '14: Vol. 32* (pp. 703–711).

Chen, Z., Mukherjee, A., Hsu, M., & Castellanos, M. (2013). Exploiting domain knowledge in aspect extraction. In *Proceedings of the conference on empirical methods in natural language processing - EMNLP '13* (pp. 1655–1667).

Colace, F., Casaburi, L., De Santo, M., & Greco, L. (2014). Sentiment detection in social networks and in collaborative learning environments. *Computers in Human Behavior, 51*, 1061–1067. http://doi.org/10.1016/j.chb.2014.11.090.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 50–57). http://doi.org/10.1145/312624.312649.

Lin, C., & He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceeding of the 18th ACM conference on information and knowledge management - CIKM '09* (pp. 375–384). http://doi.org/10.1145/1645953.1646003.

Liu, B. (2012). *Sentiment analysis and opinion mining*. Morgan & Claypool Publishers http://doi.org/10.2200/S00416ED1V01Y201204HLT016.

Ma, B., Zhang, D., Yan, Z., & Kim, T. (2013). An LDA and synonym lexicon based approach to product feature extraction from online consumer product reviews. *Journal of Electronic Commerce Research, 14*(4), 304–314. http://doi.org/10.1016/j.im.2015.02.002.

Marrese-Taylor, E., Velásquez, J. D., & Bravo-Marquez, F. (2014). A novel deterministic approach for aspect-based opinion mining in tourism products reviews. *Expert Systems with Applications, 41*(17), 7764–7775. http://doi.org/10.1016/j.eswa.2014.05.045.

Mei, Q., Ling, X., Wondra, M., Su, H., & Zhai, C. (2007). Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web* (pp. 171–180). http://doi.org/10.1145/1242572.1242596.

Mimno, D., Wallach, H. M., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the conference on empirical methods in natural language processing - EMNLP '11* (pp. 262–272).

Mukherjee, A., & Liu, B. (2012). Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th annual meeting of the association for computational linguistics* (pp. 339–348).

Rana, T. A., & Cheah, Y.-N. (2016). Aspect extraction in sentiment analysis: Comparative analysis and survey. *Artificial Intelligence Review, 46*(4), 459–483. http://doi.org/10.1007/s10462-016-9472-z.

Sayyadi, H., & Raschid, L. (2013). A graph analytical approach for topic detection. *ACM Transactions on Internet Technology, 13*(2), 1–23. http://doi.org/10.1145/2542214.2542215.

Wang, H., Xu, F., Hu, X., & Ohsawa, Y. (2013). IdeaGraph: A graph-based algorithm of mining latent information for human cognition. In *International conference on systems, man, and cybernetics* (pp. 952–957). http://doi.org/10.1109/SMC.2013.167.

Wang, S., Chen, Z., & Liu, B. (2016). Mining aspect-specific opinion using a holistic lifelong topic model. In *Proceedings of the 25th international conference on World Wide Web - WWW '16* (pp. 167–176). http://doi.org/10.1145/2872427.2883086 .

Wang, W., Wang, H., & Song, Y. (2017). Ranking product aspects through sentiment analysis of online reviews. *Journal of Experimental & Theoretical Artificial Intelligence, 29*(2), 227–246. http://doi.org/10.1080/0952813X.2015.1132270.

Yan, Z., Xing, M., Zhang, D., & Ma, B. (2015). EXPRS: An extended pagerank method for product feature extraction from online consumer reviews. *Information & Management, 52*(7), 850–858. http://doi.org/10.1016/j.im.2015.02.002.

Yang, G., Wen, D., Kinshuk, Chen, N.-S., & Sutinen, E. (2015). A novel contextual topic model for multi-document summarization. *Expert Systems with Applications, 42*(3), 1340–1352. http://doi.org/10.1016/j.eswa.2014.09.015.

Yang, L., Liu, B., Lin, H., & Lin, Y. (2016). Combining local and global information for product feature extraction in opinion documents. *Information Processing Letters, 116*(10), 623–627. http://doi.org/10.1016/j.ipl.2016.04.009.

Yang, Z., Kotov, A., Mohan, A., & Lu, S. (2015). Parametric and non-parametric user-aware sentiment topic models. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval - SIGIR '15* (pp. 413–422). http://doi.org/10.1145/2766462.2767758 .

Yao, L., Zhang, Y., Wei, B., Li, L., Wu, F., & Zhang, P. (2016). Concept over time: The combination of probabilistic topic model with wikipedia knowledge. *Expert Systems with Applications, 60*, 27–38. http://doi.org/10.1016/j.eswa.2016.04.014.

Yuan, Q., Cong, G., Ma, Z., Sun, A., & Thalmann, N. M. (2013). Who, where, when and what: Discover spatio-temporal topics for twitter users. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13* (pp. 605–613). http://doi.org/10.1145/2487575.2487576.

Zhang, C., Wang, H., Cao, L., Wang, W., & Xu, F. (2016). A hybrid term–term relations analysis approach for topic detection. *Knowledge-Based Systems, 93*, 109–120. http://doi.org/10.1016/j.knosys.2015.11.006.