World Wide Web (2020) 23:2429–2447
https://doi.org/10.1007/s11280-020-00806-x

Understanding a bag of words by conceptual labeling with prior weights

Haiyun Jiang[1] · Deqing Yang[2] · Yanghua Xiao[1] · Wei Wang[1]

Received: 4 April 2019 / Revised: 4 October 2019 / Accepted: 21 February 2020 /
Published online: 14 April 2020

Abstract

In many natural language processing tasks, e.g., text classification or information extraction, the weighted bag-of-words model is widely used to represent the semantics of text, where the importance of each word is quantified by its weight. However, it is still difficult for machines to understand a weighted bag of words (WBoW) without explicit explanations, which seriously limits its application in downstream tasks. To make a machine better understand a WBoW, we introduce the task of conceptual labeling, which aims at generating the minimum number of concepts as labels to explicitly represent and explain the semantics of a WBoW. Specifically, we first propose three principles for label generation and then model each principle as an objective function. To satisfy the three principles simultaneously, a multi-objective optimization problem is solved. In our framework, a taxonomy (i.e., Microsoft Concept Graph) is used to provide high-quality candidate concepts, and a corresponding search algorithm is proposed to derive the optimal solution (i.e., a small set of proper concepts as labels). Furthermore, two pruning strategies are also proposed to reduce the search space and improve the performance. Our experiments and results prove that the proposed method is capable of generating proper labels for WBoWs. Besides, we also apply the generated labels to the task of text classification and observe an increase in performance, which further justifies the effectiveness of our conceptual labeling framework.

✉ Yanghua Xiao
shawyh@fudan.edu.cn

Haiyun Jiang
jianghy16@fudan.edu.cn

Deqing Yang
yangdeqing@fudan.edu.cn

Wei Wang
weiwang1@fudan.edu.cn

[1] Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai, China

[2] School of Data Science, Fudan University, Shanghai, China

Keywords Conceptual labeling · Microsoft concept graph · Weighted bag of words · Multi-objective optimization · Concept pruning

1 Introduction

The weighted bag-of-word (WBoW) model[1] [8, 21] is an extension of the bag-of-words model [45], where the importance of each word in a WBoW is quantified by a weight. In general, the bag-of-words model can be considered as a special case of the weighted bag-of-word model where all the words are associated with a uniform weight. Intuitively, a WBoW is more informative than a bag of words (BoW) because the weight associated with each word can precisely quantify the importance of each word in characterizing the semantics of the original text. There are lots of mature technologies to construct WBoWs from texts, including (1) the keywords extraction-based methods: TextRank [29], RAKE [33] and TAKE [28], as well as some other extractors [1, 3, 30, 36]; (2) topic model-based methods: LDA [4], hierarchical topic models [20] and structural topic models [32].

Although a WBoW contains the most important and representative words of text, it is difficult for machines to understand the semantics of a WBoW without explicit explanation. As a result, the text cannot be well understood by machines. For example, in topic modeling, a topic found by LDA [4] is usually represented by a distribution over some words (i.e., a WBoW). However, it is still unclear what the words of each topic mean and further what the topic is about. Therefore, explicitly explaining WBoWs for machines becomes one of the critical issues to use WBoWs.

Concepts are strong evidence for the explanation of a WBoW, because humans usually understand the world by classifying objects into concepts [41]. Psychologist Gregory Murphy acclaimed that "*Concepts are the glue that holds our mental world together. Without concepts, there would be no mental world in the first place*" [5, 27]. Based on this point, the task of conceptual labeling, i.e., generating concepts as labels, is proposed to explicitly explain a bag of words (BoW) [34, 35, 37]. However, to the best of our knowledge, most of the existing work does not consider the prior weight of a word in a BoW. In this paper, we focus on conceptual labeling for a WBoW, that is, *generating the minimum number of concepts about the words in the WBoW to represent and explain the semantics of this WBoW*.

We illustrate this task with two toy examples:

- `basketball(4.78), soccer(3.18), tennis (2.83), swimming(1.76)` → *sport*
- `rosewood(1.44), poplar(2.48), cherry (6.7), tulip(2.66), carnation (1.83), marigold(1.53)` → *tree*, *flower*

For human beings, the labels on the right are the concepts that come to our mind given the words and phrases on the left. That is, we can unconsciously generate proper concepts in our mind as an explanation to understand a WBoW.

But how to generate a suitable set of conceptual labels for a WBoW? In other words, how do the prior weights affect the conceptualization? Intuitively, the label generation is supported to bias toward conceptualizing the words with a large weight. In contrast, a word with a smaller weight should be secondarily considered. Thus, the naive solution is to filter out

[1] In this paper, we only consider the case that all the words in a bag are entity mentions, e.g., *Obama, notebook, rose, etc.* Because entities are core components in most text analysis tasks.

the words with smaller weights and then conceptualize the rest of the words. However, this solution will *lose some useful information from words with smaller weights.* We illustrate how weights affect the conceptualization results:

- `watermelon(1.21), apple(1.8), banana (0.7), pear(0.99), China(0.01)` → *fruit*
- `watermelon(1.21), apple(1.8), banana (0.7), pear(0.99), China(0.01), Japan (0.18)` → *fruit, Asian country*

In the first example, the weight of `China` is very small and it is difficult to merge `China` with other words. So `China` is likely to be a noise word and discarding it does not affect the semantic understanding of the original text. For the rest words, *fruit* is a good label. Thus, the proper label for this WBoW can be *fruit.* In contrast, in the second example, `Japan` is a core keyword and `China` is semantically related to `Japan`, so `China` is unlikely to be a noise word and the proper labels for this WBoW could be *fruit, Asian country*. However, the naive solution will delete `China` from the WBoW because of its small weight, which loses useful information from `China`. Therefore, it is necessary to *model the complex dependencies between the word set and the weight distribution.*

1.1 Our solution

The solution to our task mainly contains two steps: conceptual labeling modeling and optimal label generation (searching). In particular, we also provide two pruning strategies to accelerate the label searching process. In this paper, we take a large-scale knowledge base: Microsoft Concept Graph (MCG)[2] to provide candidate concepts.

To model the conceptual labeling, we propose three principles to guide the optimal label generation, i.e., (1) *the least number of conceptual labels*, (2) *the strongest conceptualization ability* and (3) *the maximum coverage of words*. In particular, principle 2 incorporates the words weights into the conceptual labeling. As a result, the label generation problem is formalized as a *multi-objective discrete optimization problem*, where each objective function corresponds to one principle.

To obtain the optimal conceptual labels, the multi-objective optimization problem is required to be solved. That is, we need to search a small set of concepts from MCG by optimizing the multiple objective problem. In this paper, we propose a simple but effective *hybrid approach* for the optimization.

We have to point out that there are more than 5.4 million concepts in MCG, which makes the search complexity unacceptable, especially for large-scale WBoWs. To overcome this challenge, we propose two strategies to prune the candidate concepts during the concept search process, which is motivated by the observation that *a large number of candidate concepts are too vague to be labels or they can be replaced by other concepts in semantic characterization.* The pruning operations significantly improve both the effectiveness of conceptualization and efficiency.

1.2 Applications

Conceptual labeling for a WBoW is very useful for many real applications. For example:

[2] https://concept.research.microsoft.com/

- Text classification. Given a text, we can construct the corresponding WBoW and then generate a set of conceptual labels. These labels are used as additional features to enhance the existing text classification models [15]. For example, a text containing the keywords "Donald Trump", "Xi Jinping" and "Hillary" is very likely to describe *politics*, which is is strongly implied by the generated concept "politician", although "politician" is unlikely to appear in the text. Compared with the traditional text classification [15], conceptual labels are very useful background knowledge for improving this task.
- Explaining the results of topic modeling. In topic modeling, a latent topic is represented by a distribution over words, i.e., a WBoW. The previous work [20, 25] explains the latent topics by conceptualizing their topic words with conceptual labels. But in these methods, all the topic words are viewed as equally important, which discards the important weight information and it is hard to achieve the desired performance. In contrast, conceptual labeling with prior weights makes the explanation of topics more precise.
- Understanding user intent. In the item recommendation systems [7], a popular method is to mine the historical queries to understand the user intent. Conceptual labeling is very effective to assist it by explaining the bag of queries, where the weights can be defined as the click frequencies. For example, if a user queried "IPhone X" and "HUAWEI P30", then we infer that the user is interested in high-end smartphones by generating a conceptual label "High-end Phone" for the two queries. As a result, the system could recommend some other high-end smartphones for the user.

1.3 Contributions

The main contributions of this paper are summarized as follows. (1) To the best of our knowledge, this is the first work to generate conceptual labels for a WBoW. (2) We propose three principles for this task and also propose a simple hybrid approach for optimization. (3) We present two strategies to prune the candidate concepts, which significantly accelerates the label generation process.

The rest of the paper is organized as follows. Section 2 introduces the proposed principles and formalizes conceptual labeling as a multi-objective discrete optimization problem. Section 3 discusses the solution to the problem as well as the pruning strategies. Section 4 conducts the experiments and presents the analysis. Besides, the related work is presented in Section 5. Finally, the simple conclusion and the several issues for future work are given in Section 6.

2 Principles and modeling of conceptual labeling

In this section, we first briefly introduce the MCG knowledge used in our framework and then present the principles as well as the corresponding objective functions for conceptual labeling.

2.1 Using MCG knowledge

The candidate concepts are required in our framework, which can be obtained from existing knowledge bases. In this paper, we use MCG [43] as the concept source that contains more than 5.4 million concepts. MCG contains more than 87 million *concept-instance* pairs with isA relations that are extracted from text corpora.

For example, MCG contains the pair <*flower*, `rose`> that is extracted from the sentence "the rose is a kind of flower", where *flower* is a concept and `rose` is one of its instance.

The *typicality* score is defined to measure how likely we think of a concept given an instance.[3] Typicality is defined as:

$$p\left(c\,|x\right) = \frac{n\left(c, x\right)}{\sum_{c_i \in MCG} n\left(c_i, x\right)} \tag{1}$$

Where x is an instance, c and c_i denote concepts, and $n\left(c, x\right)$ is the frequency provided by MCG that quantifies the confidence of the isA pair <*c,x*>. Intuitively, a large $p(c|x)$ indicates that people are more likely to think of the concept c compared with other concepts c_i given the instance x. For example, $p\left(\textit{flower}|\texttt{rose}\right) > p\left(\textit{plant}|\texttt{rose}\right)$. The typicality score allows us to choose *flower* instead of *plant* to better conceptualize `rose`.

2.2 Problem modeling

Notations We denote a WBoW as $X = \{\langle x_1, w_1\rangle, \langle x_2, w_2\rangle, \cdots, \langle x_M, w_M\rangle\}$, where x_i is the i-th instance and w_i is the normalized weight of x_i, i.e., $\sum_{i=1}^{M} w_i = 1$. The concept set of the instance x_i can be queried from MCG and we denote it as C_i. Then $C = \cup_{i=1}^{M} C_i$ (where $N = |C|$) is queried as the candidate concept set, from which the label set will be selected. We also define a *typicality matrix* $\mathbf{T} \in \mathbb{R}^{M\times N}$, where $T_{ij} = p(c_j|x_i)$ is the typicality score about concept c_j and instance x_i. Notice that $p(c_j|x_i)$ is usually very small, we normalize $\mathbf{T}$ by updating $p(c_j|x_i) = p(c_j|x_i)/\max(\mathbf{T})$, where $\max(\mathbf{T})$ is the maximum element in $\mathbf{T}$.

In our framework, the conceptual label set for X will be selected from C under the guidance of the proposed principles. Since each column in $\mathbf{T}$ is related to a distinct concept, we transform the label set selection into a *column subset selection problem* [6, 10]. That is, selecting a "best" column subset $\mathbf{T}_0$ from $\mathbf{T}$ by optimizing a multi-objective function, where each objective function corresponds to a distinct principle.

Multi-objective function. Based on the relationship between concept and column, our task is formalized as the following multi-objective optimization problem:

$$\mathbf{T}_0 = \arg\max_{\mathbf{T}'\subset\mathbf{T}} \{f_1(\mathbf{T}'), f_2(\mathbf{T}'), f_3(\mathbf{T}')\} \tag{2}$$

where $\mathbf{T}'$ denotes a column subset of $\mathbf{T}$ and its corresponding concept set is C'. That is, we need to select an optimal column subset from $\mathbf{T}$ that maximizes all the three objective functions $f_i(\cdot)$ as much as possible, where $f_i(\cdot)$ is derived from the i-th principle ($i \in \{1, 2, 3\}$). In this way, the three principles will be satisfied as much as possible by optimizing (2). We denote the optimal solution as $\mathbf{T}_0$ and the corresponding concept subset as C_0, which is the optimal conceptual label set for X.

2.3 Principles

In this section, we propose three principles to guide the conceptual label set generation for a WBoW. The principles can be expressed as: (1) the least number of conceptual labels, (2) the strongest conceptualization ability and (3) the maximum coverage of words. For each principle, we elaborate how to formalize it as an objective function.

[3] For simplicity, the words in WBoWs are also known as instances.

Principle 1: the least number of conceptual labels In general, we hope to conceptualize a WBoW using as few labels as possible, so that machines can understand the WBoW as well as the corresponding text more easily. Since the size of a concept set C' is also equal to the number of columns of $\mathbf{T}'$ (noted as $|\mathbf{T}'|$), principle 1 is expressed as max $f_1(\mathbf{T}')$, where $f_1(\mathbf{T}') = -|\mathbf{T}'|$.

Principle 2: the strongest conceptualization ability Principle 2 means the selected labels should strongly conceptualize a WBoW, that is, the labels can strongly represent the semantics of the WBoW. We formalize the conceptualization ability of a concept set to a WBoW as follows.

Given an isA pair (c_j, x_i), the conceptualization ability of c_j with respect to x_i is quantified by the typicality score $p(c_j|x_i)$ [41]. In particular, $p(c_j|x_i) = 0$ means (c_j, x_i) has no isA relationship, so x_i cannot be conceptualized by c_j. In general, a concept can only conceptualize a subset of X. For example, let X ={`Microsoft(0.30)`, `Google(0.449)`, `banana(0.201)`}, the concept *company* can only conceptualize the first two instances well. We denote the subset that can be conceptualized by c_j as $X_{c_j} \subseteq X$, i.e., $p(c_j|x_i) > 0$ $(\forall x_i \in X_{c_j})$.

The possibility of selecting c_j as a label is influenced by three aspects: *(1) the size of* X_{c_j}*, (2) the typicality scores* $p(c_j|x_i)$ $(x_i \in X_{c_j})$*, (3) the weight of the instances in* X_{c_j}. The conceptualization ability of c_j to X_{c_j} is measured by considering the three aspects simultaneously. Specifically, the ability γ is defined as

$$\gamma = \sum_{x_i \in X_{c_j}} [p(c_j|x_i)w_i]^2 \tag{3}$$

where w_i is the normalized weight of x_i. Intuitively, a concept c_j with a large γ tends to be selected as one of the conceptual labels. Furthermore, the conceptualization ability of the candidate concept set C' with respect to X is measured by

$$\gamma' = \sum_{c_j \in C'} \sum_{x_i \in X_{c_j}} [p(c_j|x_i)w_i]^2 \tag{4}$$

We define the *weight matrix* of X as $\mathbf{W} = diag(w_1, ..., w_M)$. Since $p(c_j|x_i) = 0$ for $x_i \notin X_{c_j}$, (4) can be rewritten as $\gamma' = ||\mathbf{W}\mathbf{T}'||_F^2$. Thus, principle 2 can be expressed as max $f_2(\mathbf{T}')$, where $f_2(\mathbf{T}') = \gamma'$.

Principle 3: the maximum coverage of words. In general, we hope the selected label set could conceptualize all the words in a WBoW. However, (1) it is not easy to be satisfied as the size of the label set is required to be small (i.e., Principle 1). (2) it is unnecessary in cases where some words are noise. We define *coverage* to measure the number of words conceptualized by the label set.

Definition 1 (Coverage) Given a word x_i in X and a concept set C', if there exists a concept $c_j \in C'$ that makes $p(c_j|x_i) > 0$, then x_i can be semantically covered by the concept set C'. Thus, the coverage is defined as the ratio of the number of words covered by C' to the size of X.

Since we have introduced the typicality matrix $\mathbf{T}$, the number of words covered by C' in X is equal to the number of non-zero rows in $\mathbf{T}'$, i.e., $||\mathbf{T}'\mathbf{1}||_0$, where $\mathbf{1} = [1, ..., 1]^T \in \mathbb{R}^n$

and $\mathbf{T}'\mathbf{1}$ denotes the vector where $\mathbf{T}'$ is summed by column. $||\cdot||_0$ is the zero norm. The coverage is computed by

$$f_3(\mathbf{T}') = ||sum\mathbf{T}'\mathbf{1}||_0/M \tag{5}$$

So principle 3 is expressed as max $f_3(\mathbf{T}')$.

3 Solution and pruning

In this section, we discuss how to generate the conceptual label set, i.e., the solution to (2). Besides, we also describe how to prune the candidate concepts to accelerate the label generation process.

3.1 Generating conceptual label set

We denote the solution to (2) as $\mathbf{T}_0$ whose corresponding concept set C_0 is the generated label set. There are two challenges in optimizing (2): (1) the maximization of $f_1(\mathbf{T}')$ conflicts with the maximization of $f_2(\mathbf{T}')$ and $f_3(\mathbf{T}')$, (2) the existing algorithms usually provide mature solutions for the continuous optimization problems, while our problem is discrete. Therefore, we propose a simple but effective *hybrid approach* to select the best $\mathbf{T}_0$ from $\mathbf{T}$. The basic idea is to enumerate the size of the label set iterating from 1, and derive the best concept set under each size. Then we explore the proper size, thus obtaining the proper label set. Specifically, our approach contains the following two steps.

(1) *Selecting the best $\mathbf{T}_{0,k}$ given a label size k.* When $f_1(\mathbf{T}') = k$ is fixed, the optimization problem in (2) is simplified as

$$\mathbf{T}_{0,k} = \arg \max_{\substack{\mathbf{T}' \subset \mathbf{T} \\ f_1(\mathbf{T}')=k}} [f_2(\mathbf{T}'), f_3(\mathbf{T}')]^T \tag{6}$$

The optimization in (6) can be achieved using the linear weighting method [9], that is,

$$\mathbf{T}_{0,k} = \arg \max_{\substack{\mathbf{T}' \subset \mathbf{T} \\ f_1(\mathbf{T}')=k}} \lambda f_2(\mathbf{T}') + (1-\lambda) f_3(\mathbf{T}') \tag{7}$$

Where λ denotes the importance weight between principles 2 and 3. The direct solution to (7) can be obtained by the exhaustive search.

(2) *Selecting the proper k_0.* Let

$$f(k) = \arg \max_{\substack{\mathbf{T}' \subset \mathbf{T} \\ f_1(\mathbf{T}')=k}} \lambda f_2(\mathbf{T}') + (1-\lambda) f_3(\mathbf{T}') \tag{8}$$

in (7). Intuitively, $f(k)$ is *monotonically* increasing with k, implying that at least one of the principles 2 and 3 will be more satisfied when k increases. However, a large k will violate the principle 1. To balance principles 1 and 2,3, we require that $f(k)$ increases largely enough when k increases by 1. This is measured by an *incremental threshold* δ.

That is, we enumerate k starting at 1 and stop the enumeration when $f(k_0+1) - f(k_0) < \delta$. Then k_0 is the proper size. Further, $\mathbf{T}_0 = \mathbf{T}_{0,k_0}$ and C_0 are the selected column set and the generated conceptual label set, respectively.

The proposed approach is very easy to be parallelized. For example, the search space in (7) can be divided into g groups. We first get the local optimal solution in each group and then obtain $\mathbf{T}_{0,k}$ among these local optimal solutions.

3.2 Pruning

In our approach, the label set generation is conducted by exhaustive search, where the complexity is directly determined by the number of candidate concepts. In this section, we significantly reduce the search space by two kinds of pruning.

3.2.1 Pruning based on substitutability

In fact, most candidate concepts in C can never be selected as conceptual labels for X. We present the following theorem with a simple proof.

Theorem 1 *If two concepts c_j and $c_{j'}$ in C satisfy $p(c_j|x_i) \geq p(c_{j'}|x_i)$ for all the words x_i in X, then $c_{j'}$ can never be selected as a label.*

Proof $p(c_j|x_i) \geq p(c_{j'}|x_i)$ $(\forall x_i \in X)$ indicates that (1) the conceptualization ability of $c_{j'}$ is less than c_j for all the words, (2) the contribution of $c_{j'}$ to the coverage is also less than c_j. Therefore, $c_{j'}$ is less important than c_j to conceptualize X according to the proposed principles, and the selection of $c_{j'}$ can be replaced by c_j without performance loss. We call this property as *substitutability*. To reduce the search space, we pre-delete the concept $c_{j'}$ from C as well as the j'-th column from $\mathbf{T}$. □

Statistically, our experiments show that this pruning operation can delete up to 90% of the candidate concepts from $\mathbf{T}$ in average, which greatly reduces the search complexity without losing the optimal concepts.

3.2.2 Pruning based on vagueness

MCG was created by data-driven approaches, thus containing many vague concepts, such as *simple element*, *proper name*, *everyone* and so on. These concepts should not be selected as conceptual labels for their poor conceptualization ability for most instances. We pre-delete these vague concepts from C, thus improving both the efficiency and the conceptualization performance.

Our statistical analysis in MCG shows that a vague concept c_j has the following two characteristics simultaneously. (1) *c_j covers a large number of instances in MCG*, i.e., $|I_{c_j}| = L$ is very large, where I_{c_j} is the instance set in MCG covered by c_j. (2) *the corresponding frequencies are very small*, i.e., $n(c_j, x)$ is very small for all $x \in I_{c_j}$.

We present a simple method to delete the vague concepts from C. Specifically, we set two thresholds $\bar{L}$ and $\bar{N}$, then any concept c_j satisfying $L > \bar{L}$ and $\max_{x \in I_{c_j}} n(c_j, x) < \bar{N}$ will be deleted from C. In our experiments, we set $\bar{L} = 300$ and $\bar{N} = 15$.

In general, the two pruning strategies significantly reduce the search space, which makes the conceptualization for very large WBoWs possible.

4 Experiments

We evaluate the effectiveness of our conceptual labeling scheme from *three* aspects: (1) generating labels for BoWs, (2) for WBoWs and (3) the application of labels in text classification. The first two aspects aim to directly evaluate the generated labels, where we

construct three datasets for evaluation, i.e., one synthetic dataset and two real datasets (including WikipediaData data and FlickrData). Besides, we also consider a downstream task, i.e., text classification, to indirectly prove the effectiveness of the conceptual labeling scheme. The hyperparameters $\lambda = 0.5$ and $\delta = 0.2$ are used in all experiments and they will be further discussed in Section 4.4.

4.1 Experiments on BoWs

In this section, we conduct experiments on BoWs, a special kind of WBoWs.

Dataset We take two real datasets [37], i.e., FlickrData and WikipediaData, to evaluate the performance on BoWs.

- *FlickrData* was collected from manually labeled tags in Flickr [22], where each BoW consists of the instance tags in an image. Image tags in Flickr are generally redundant and contain some noise. Conceptual labels can refine the tags and help machines to understand the images more deeply.
- Each BoW in *WikipediaData*[4] contains the topic words of an English Wikipedia page, which is obtained by LDA [2, 4]. Conceptual labeling provides rich background knowledge for machines to understand the latent topics as well as the documents.

Baselines We compare our model with two strong baselines.

- *Clustering-then-conceptualization (CC)*. CC is an extension of the model proposed by [34]. In CC, we first cluster the words in a BoW by K-means [12] according to the semantic similarity. Then we generate the best single concept for each individual cluster using a naive Bayes model [34].
- *MDL-based model* [37]. This model proposes two criteria for conceptual labeling of a BoW, i.e., semantic coverage and minimality. To balance the semantic *coverage* of a BoW and the *minimality* of the output labels, the minimum description length (MDL) principle [31] is used to select the best label set.

Evaluation criteria It is difficult to provide the ground-truth label set for a BoW. Some BoWs can be well conceptualized by several label sets. For example, either *European country* or *developed country* are acceptable for the BoW {`French`, `UK`, `Germany`}. Therefore, we consider a *manual scoring*-based evaluation. Specifically, we divide the quality of the generated labels into the following four levels:

- **Perfect (4)**. *A label set is scored with 4 if it appropriately represents the semantics of the input BoW.* For example, given a BoW {`volleyball`, `basketball`, `football`}, we can easily think of *ball game*, which is an appropriate label with a score of 4.
- **Minor loss in conceptualization ability (3)**. *If a candidate label set has minor loss in conceptualization ability, then it is scored with 3*. For another example, given {`French`, `UK`, `Germany`, `Italy`}, the label *country* loses minor conceptualization ability compared with *European country*, thus getting a score of 3.
- **Minor loss in coverage (2)**. *If a candidate label set has minor coverage, then it is scored with 2.* For example, the label *meal* gets a score of 2 for

[4] https://dumps.wikipedia.org/

{meal, dinner, food, breakfast, ceremony, wedding} because it only conceptualizes the first four words, which loses minor coverage.

- **Much loss in conceptualization ability or coverage (1)**. *If the label set loses much conceptualization ability or coverage, its score is 1.* For example, given {poplar, pine, cherry, rose}, the label *tree* loses much coverage, since 50% of the words, i.e., cherry and rose, can not be semantically represented by *tree*. In fact, a suitable label set can be {*tree, flower*}. For another example, given {puppy, kitten, piggy}, the label *creature* loses more conceptualization ability compared with *pet* or *animal* because these two labels are more specific in explaining the BoW. So the label *creature* is only scored with 1.
- **Unrelated (0)**. *A conceptual label set that is not related to the BoW will be scored with 1.* For example, given {walkway, swimming pool, vehicle}, the label *activity* has a score of 0.

In addition to the scoring criteria above, we also provide a large number of samples to volunteers so that they have a deeper understanding of the scoring criteria.

Metric We randomly select 300 BoWs from each dataset (i.e., FlickrData or WikipediaData) and take all the models to generate conceptual label sets for them. We recruit seven volunteers to evaluate the labeling results according to the evaluation criteria above. All the seven volunteers are in the field of natural language processing or data mining. Supposing the i-th volunteer's score for the generated label set of the j-th BoW is $s_{i,j}$, then the average score on each dataset is computed as:

$$S = \frac{1}{IJ} \sum_{i=1}^{I} \sum_{j=1}^{J} s_{i,j} \tag{9}$$

Where $J = 300$ is the number of BoWs and $I = 5$ is the number of volunteers. Obviously, the full score $S = 4$ when all the label sets are scored with 4.

Results Generating conceptual labels for a BoW is just a special case of our conceptual labeling scheme. To deal with BoWs, we simply replace the weighted matrix $\mathbf{W}$ in principle 2 with the identity matrix. The results are presented in Table 1. We conclude that, (1) our proposed model can effectively conceptualize BoWs. (2) Our model is superior to CC and MDL in performance. In CC, the number of clusters is difficult to be determined in advance, and the error caused by clustering will mislead the label generation. In MDL, only the criteria of coverage and label size are considered. In our model, the principle of conceptualization ability is also considered, which further improves the quality of the generated labels. (3) The score on FlickrData is lower than that on WikipediaData for all the models. An important reason is that the former is redundant and contains more noise, which inevitably affects the results of all the models.

Table 1 Evaluation scores on FlickrData and WikipediaData for different models

Model	FlickrData	WikipediaData
CC	2.34	2.40
MDL	2.52	2.61
Proposed Model	2.64	2.73

Table 2 The effect of n_i on performance

(n_c=5, n_n=5)	n_i=2	n_i=4	n_i=6	n_i=8	n_i=10
Precision	0.81	0.86	0.90	0.92	0.94
Recall	0.74	0.80	0.88	0.91	0.93
F-score	0.77	0.83	0.89	0.91	0.93

4.2 Experiments on WBoWs

In this section, we evaluate the performance on WBoWs from two perspectives: We first evaluate on a synthetic dataset based on MCG, where the ground-truth label sets of WBoWs are available. Then we conduct experiments on the real dataset, where the human evaluation is adopted.

4.2.1 Evaluation on synthetic dataset

Dataset Similar to [37], we use MCG to generate synthetic WBoWs that have ground-truth label sets for automatic evaluation. The first two steps for dataset construction are the same as those in [37].

Step 1 We first randomly select n_c concepts from MCG. Then n_i instances of each selected concept are randomly sampled.

Step 2 We also randomly select n_n instances of other concepts as noise. Then the selected instance set constitutes a BoW of size $n_c n_i + n_n$.

Step 3 We assign a random weight distribution to the BoW, thus getting a synthetic WBoW.[5] The ground-truth labels for this WBoW are the selected n_c concepts in Step 1.

We construct the synthetic WBoWs with different parameter settings. Specifically, (1) $n_c = 5$, $n_n = 5$, and $n_i \in \{2, 4, 6, 8, 10\}$, (2) $n_i = 4$, $n_n = 5$, and $n_c \in \{2, 3, 4, 5, 6\}$. The former (latter) setting makes us to analyze the effect of n_i (n_c) on performance. For each parameter setting, we construct 500 WBoWs.

Metrics We introduce three metrics: *precision* (P), *recall* (R) and F*-score* (F) as follows:

$$P = \frac{\sum_{j=1}^{J} q_j}{\sum_{j=1}^{J} p_j} \quad R = \frac{\sum_{j=1}^{J} q_j}{J n_c} \quad F = \frac{2PR}{P+R} \tag{10}$$

Where $J = 500$ is the number of WBoWs. For the j-th WBoW, our model generates p_j conceptual labels and q_j of them are in the ground-truth label set.

Results We present the results in Tables 2 and 3 and conduct the analysis as follows.

– In Table 2, the performance of the results becomes better as n_i increases. Particularly, the three metrics exceed 93% when $n_i = 10$, which means more instances of a concept will help to generate this concept. As a result, the task of conceptual labeling is very suitable to help understand the topics of long documents, where many semantically related words expressing the same topic can be extracted into a WBoW.

[5] Note that the noise instances are required to have smaller weights than the non-noise.

Table 3 The effect of n_c on performance

(n_i=4, n_n=5)	n_c=2	n_c=3	n_c=4	n_c=5	n_c=6
Precision	0.95	0.91	0.88	0.86	0.82
Recall	0.96	0.89	0.85	0.80	0.76
F-score	0.95	0.90	0.86	0.83	0.79

– In Table 3, the conceptualization performance is very high for small n_c (e.g., $n_c = 2$), and begins to decline as n_c increases. Intuitively, the average semantic distance between two instances of different concepts becomes smaller when n_c increases. Thus a WBoW derived from more concepts is harder to be conceptualized with the ground-true labels. This can also be explained from the perspective of topic discovery of a document. That is, if we view the true labels of a WBoW from a document as the latent topics, then a document containing more topics will make machines difficult to find all these topics.

4.2.2 Evaluation on real dataset

Besides the synthetic dataset, we also conduct the experiments on the real dataset.

Dataset We also consider the datasets: *FlickrData* and *WikipediaData* that have been used in the experiments on BoWs (see Section 4.1). For FlickrData, the weight of an instance is defined as the occurrence frequency of this instance in the image. For WikipediaData, the unnormalized weight for each topic word is defined as $p(w|t)\,p(t)$, where $p(x|t)$ denote the conditional probability of word w given its topic t. $p(t)$ denotes the probability of topic t. For both the datasets, the weights will be normalized over the words in a WBoW.

Baselines To the best of our knowledge, there is no previous work dealing with the labeling for WBoWs, so we construct three strong baselines for comparison.

– *Improved MDL-based model (IMDL).* In the original MDL model [37], the prior weight $p(x_i)$ is equal for all the instances in a BoW. To deal with WBoWs, we simply modify $p(x_i) = w_i$, thus incorporating the weight information into the MDL-based model and generating labels for WBoWs.
– *Maximal clique segmentation-based model (MCS).* In this model, we first construct a semantic graph for a WBoW, where the nodes correspond to instances and the weight of an edge reflects the similarity between two instances as well as their weights. Then we take the operation of maximal clique segmentation [35, 38] to divide the graph into several subgraphs and the instances in each subgraph is conceptualized by one concept.
– *Clustering-based model (Cluster).* In this model, we take a regularized K-means-based method [17] to cluster a WBoW X into several clusters according to the semantic similarity, where the feature vector of an instance comes from the result of Word2Vec [26]. The clusters containing only one instance will be deleted if the corresponding instance weight is smaller than $1/2|X|$, where $|X|$ is the size of X. This is reasonable because the small-weight words that cannot be clustered with other words are very likely to be noise. Finally, we generate one concept for each cluster, thus obtaining the label set.

Evaluation criteria Similar to the evaluation for BoWs, we still take a manual scoring approach to evaluate the generated labels of WBoWs. Generally, a good conceptual label set should strongly conceptualize all the words with a large weight in a WBoW. To simplify

Table 4 Average scores on FlickrData and WikiData for different models

Dataset	IMDL	MCS	Cluster	Ours
FlickrData	2.13	1.99	1.72	2.21
WikipediaData	2.16	2.02	1.83	2.37

the evaluation, the "large" weight is defined as $\geq 1/2|X|$ for X, i.e., the half of the average weight. The evaluation criteria are roughly the same as those for BoWs (see evaluation criteria in Section 4.1), except that the conceptualization ability and coverage are judged on the words in X with a "large" weight.

Results and analysis According to the criteria, we invite the previous seven volunteers to evaluate the candidate label set by scoring. We define the average score as

$$S = \frac{1}{IJ} \sum_{i=1}^{I} \sum_{j=1}^{J} s_{i,j} \tag{11}$$

where $I = 5$ is the number of volunteers and $J = 300$ is the number of sampled WBoWs. $s_{i,j}$ is the i-th volunteer's score for the j-th WBoW. The full score $S = 4$ if and only if all the label sets are scored with 4.

The results are presented in Table 4. We conclude that the proposed model outperforms the other three baselines on conceptual labeling for real WBoWs. In our case study, we find that some WBoWs get lower scores for all the models, which are mainly caused by two reasons. (1) There are many polysemous words in English language. For example, the instance "apple" denotes both the "Apple company" and "apple (fruit)" without distinction, so there are both facts "apple is a company" and "apple is a fruit" in MCG. As a result, "apple (fruit)" may be wrongly conceptualized as "company" with some other company instances. (2) The missing or wrong facts in MCG. For example, there is a wrong fact "software is a world", which may mislead the conceptualization for WBoWs containing "software".

4.3 Experiments on text classification with conceptualization

In Sections 4.1 and 4.2, our conceptualization framework has been *directly* evaluated by scoring the generated labels. In this section, we further evaluate the framework by considering a downstream task: text classification [16, 19, 44]. We generate conceptual labels for texts and evaluate whether the text classification performance can be improved with the help of the conceptual labels.

Dataset Two standard text classification datasets are chosen in our experiments: AG's News [44] and 20NG.[6] AG's corpus is obtained from news article on the web.[7] It contains 496,835 news articles from more than 2000 news sources. There are 4 classes in AG's News and each of them contains 30,000 training samples and 1900 testing ones, respectively. The 20NG dataset is construed based on 20 Newsgroups and it contains 11,314 training documents and 7,532 test documents.

[6] http://qwone.com/~jason/20Newsgroups/

[7] http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

WBoW construction for texts. Given a text s to be classified, we obtain the corresponding WBoW X_s as follows. We take Microsoft Entity Linking[8] to identify all the entities in s and denote them as $\{x_1, ..., x_M\}$. The weight w for entity x is computed with TF-IDF, i.e.,

$$w = \log(1 + f_{es}) \log \frac{|\mathcal{S}|}{n_e} \tag{12}$$

where $\mathcal{S}$ is the set of training texts in the dataset. f_{es} denotes the raw frequency (i.e., the number of occurrences) of entity x in text s. n_e is the number of texts containing entity x. All the weights of the entities are normalized. For each WBoW X_s, we generate the conceptual labels using our conceptualization framework. We have to point out that the conceptualization for text classification is slightly different from that in Section 4.2. Specifically, we change the conceptualization criterion 1 as selecting k_0 concepts. As a result, we selected k_0 concepts for X_s based on (8). In this paper, we set $k_0 = 30$. The motivation is that our experiments find more informative concepts will help text understanding. We denote the selected concepts as C_s.

Text classification models Text classification has been extensively studied in recent years and many outstanding models were proposed [16, 19, 44]. As an example, we consider two representative solutions: **Char-CNN** [44] and **BERT**. For both the two models, the inputs are texts and the outputs are the class distribution scores. Specifically, Char-CNN [44] is under the setting of supervised deep learning and it takes character-level convolutional networks to learn text representations for text classification. BERT first takes the state-of-the-art pre-training model BERT [11] to encode a text into its distributional representation. Then a text classifier is trained on the specific text classification dataset, where the input is the distributional representation of a text.

Improved text classification models with conceptualization As we described in Applications (see Section 1.1), entities are very important elements for text understanding. The context-aware labels of entities provide abundant background knowledge for understanding entities as well as the text. Thus, the text classification performance can be further improved by incorporating conceptual labels as additional inputs.

In this paper, we propose a feature fusion-based solution to incorporate conceptual labels into the text classification models. The overview of the structure is shown in Figure 1 and the key modules are formalized as follows.

- **Inputs**. Given a label set C_s, we map each label $c_j \in C_s$ to a low-dimension vector $\mathbf{c}_j \in \mathbb{R}^{d_2}$, i.e., initialized embedding. We resort to word embeddings in Word2Vec.[9] Since a conceptual label usually contains several words, we merge these embeddings by element-wise addition and take it as the initialized embedding of the conceptual label. We denote the inputs as $\mathbf{C} = [\mathbf{c}_1, ..., \mathbf{c}_{N_1}] \in \mathbb{R}^{d_2 \times N_1}$, where N_1 is the label size.
- **Self-attention**. The self-attention mechanism [39] is used to model the interaction between different labels in C_s. The output of the self-attention module is denoted as $\mathbf{c} \in \mathbb{R}^{d_2}$, which is computed by

$$\mathbf{c} = \sigma\left(\text{sum}\left(\mathbf{W}_1^{\mathbf{T}}\mathbf{C}'\right) + \mathbf{b}_1\right), \tag{13}$$

[8] https://docs.microsoft.com/en-us/azure/cognitive-services/entitylinking/home
[9] https://code.google.com/archive/p/word2vec/

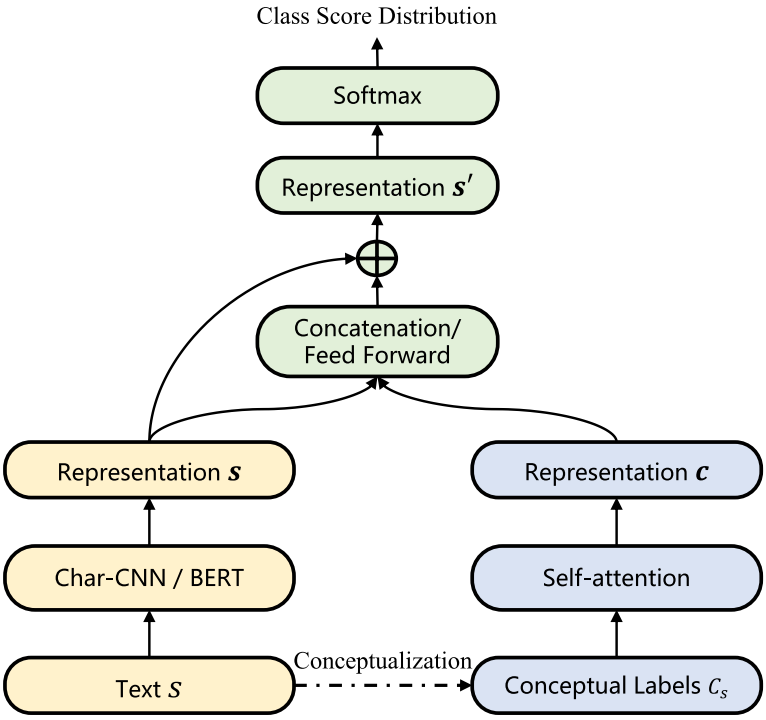

Figure 1 The overview of the feature fusion-based solution. On the left, Char-CNN or BERT encode a text s into a low-dimensional representation $\mathbf{s} \in \mathbb{R}^{d_1}$. On the left, we take the self-attention mechanism [39] to encode the conceptual label set C_s into the representation $\mathbf{c} \in \mathbb{R}^{d_2}$. Then a feed forward network with a residual connection [13] is used to output the unified representation $\mathbf{s}' \in \mathbb{R}^{d_1}$ based on a text and its conceptual label set. Finally, the representation $\mathbf{s}'$ will be input to the Softmax layer for classification

where $\mathbf{W}_1 \in \mathbb{R}^{d_2 \times d_2}$ and $\mathbf{b}_1 \in \mathbb{R}^{d_2}$ are parameters to be learned. $\sigma = \tanh(\cdot)$ is the activation function. sum$(\cdot)$ denotes the operation that sums over all the elements in each row in $\mathbf{W}_1^{\mathbf{T}}\mathbf{C}'$, thus transforming the matrix $\mathbf{W}_1^{\mathbf{T}}\mathbf{C}'$ to the vector "sum $\left(\mathbf{W}_1^{\mathbf{T}}\mathbf{C}'\right)$" with dimension d_2. The matrix $\mathbf{C}' \in \mathbb{R}^{d_2 \times N_1}$ is the result of the self-attention operation with the embeddings $\mathbf{C}$ as inputs. The i-th column in $\mathbf{C}'$ (denoted as $\mathbf{c}'_i$) is computed by

$$\mathbf{c}'_i = \sum_{j=1}^{N_1} \alpha_{ij}\mathbf{c}_j, \tag{14}$$

where α_{ij} is the normalized weight of α'_{ij}, i.e.,

$$[\alpha_{i1}, \alpha_{i2}, ..., \alpha_{iN_1}] = \text{Softmax}[\alpha'_{i1}, \alpha'_{i2}, ..., \alpha'_{iN_1}]. \tag{15}$$

α'_{ij} denotes the attention weight of $\mathbf{c}_i$ to $\mathbf{c}_j$, which is computed by

$$\alpha'_{ij} = \mathbf{c}_\mathbf{i}^{\mathbf{T}}\mathbf{c}_j \tag{16}$$

– **Concatenation/Feed Forward**. We take a feed-forward network with a residual connection [13] to integrate the representations of the text s and the label set C_s. In this way, we obtain a new mixed representation $\mathbf{s}' \in \mathbb{R}^{d_1}$, i.e.,

$$\mathbf{s}' = \sigma(\mathbf{W}_2\text{Concat}(\mathbf{s}, \mathbf{c}) + \mathbf{b}_2) + \mathbf{s} \tag{17}$$

where Concat($\mathbf{s}$, $\mathbf{c}$) $\in \mathbb{R}^{d_1+d_2}$ is the concatenation of $\mathbf{s}$ and $\mathbf{c}$. $\mathbf{s} \in \mathbb{R}^{d_1}$ is the representation of the text s and $\mathbf{c} \in \mathbb{R}^{d_2}$ is the representation of the label set C_s (the output of the self-attention module). $\mathbf{W}_2 \in \mathbb{R}^{d_2 \times (d_1+d_2)}$, $\mathbf{b}_2 \in \mathbb{R}^{d_2}$ are the parameters.

Training and testing details For the convenience of implementation, the label size N_1 is truncate or padded to $N_1 = 5$. The truncation is conducted by randomly deleting the labels in C_s and the padding is realized by adding zero vectors as the initialized embedding. The dimension of the text representation d_1 is set as 1024 for Char-CNN, respectively. The dimension of the conceptual label representation d_2 is set as 512 for both the two models. We ran our model on a computer with GPU of GTX 1080, 8G memory and operating system of ubuntu 16.04.5. We implement our framework using TensorFlow with mini-batch gradient descent. The batch size is 64 and the learning rate is 0.001.

Results and analysis We report the accuracy [44] in Table 5. We conclude that, with the conceptual labels as the additional inputs, the performance of both the models is improved. The results prove that the task of conceptualization with prior weights can effectively guide the task of text classification. In turn, the performance improvement indicates the generated conceptual labels are capable of capturing the semantics of the weighted bag of words. Besides, we also observe that the performance improvement on AG's News is more significant compared with 20NG for Char-CNN and BERT. This is because the text in AG's News contains more entities than that in 20NG. As a result, the generated labels for AG's News contains more background knowledge to understand the texts.

4.4 Hyperparameter settings

In our experiments, we set $\lambda = 0.5$. This parameter can also be heuristically re-selected according to the real applications. To select δ, we take additional 50 WBoWs from WikipediaData to calculate the average scores for different δ under $\lambda = 0.5$ and gets the highest score 2.53 when $\delta = 0.2$. Our hyperparameter settings may not guarantee the optimality on all WBoWs, but produce a good performance in general.

5 Related work

We mainly investigate the previous works on conceptualization as well as the applications in the topic modeling.

Table 5 The accuracy of different models on AG's News and 20NG datasets

Model	AG's News	20NG
Char-CNN	0.893	0.721
Char-CNN(Conceptualization)	0.912	0.743
BERT	0.930	0.839
BERT(Conceptualization)	0.943	0.851

[x]For Char-CNN, we use the results from the source code released by the authors. For BERT, we train the text classifier based on the outputs of BERT

5.1 Conceptualization

Conceptualization is an important task for natural language understanding (NLU), and it maps a text to several concepts that are pre-defined in a certain taxonomy or knowledge base [14, 18, 34, 37, 42]. Wang [41] proposed a Bayesian model using typicality and PMI to label *one* instance with a basic-level concept. Hua [14] leveraged co-occurrence network for concept inference. Song [34] used a Bayesian model as well as clustering to generate multiple labels for a short text. Sun used the minimum description length (MDL) principle to generate a set of conceptual labels [37] for a bag of words. These solutions aimed at generating conceptual labels for short texts, instance or unweighted bag of words.

However, none of them focus on conceptual labeling of a WBoW, a widely used text representation framework. Moreover, extending the existing solutions to conceptualize WBoWs is nontrivial in general, because (1) the influence of weights on the conceptual labels is complicated, (2) the existing solutions have their own specific solution framework, and are not general enough to be adjusted for our problem settings.

5.2 Conceptualization in topic modeling

Conceptualization is also widely combined with topic modeling, which aims at generating conceptual labels to explain the topics represented by a distribution over words. The early effort relies on humans to find meaningful labels [23, 24]. However, manual labeling requires a great human effort and is prone to subjectivity [40]. To alleviate it, probabilistic approaches were proposed to interpret the multinomial topic models automatically and objectively [25]. This approach achieved the automatic interpretation of topics, but the candidate labels available were limited to phrases inside documents. To overcome this limitation, Lau et al [20] proposed an automatic topic label generation method which obtains candidate labels from Wikipedia articles containing the top-ranking topic terms, top-ranked document titles, and sub-phrases.

The conceptualization above was conducted without supervision. To improve the labeling accuracy, supervised labeling was proposed, such as supervised latent Dirichlet allocation (sLDA) [3], labeled LDA (LLDA) [30], etc.

6 Conclusion and future work

In this paper, we introduce the task of conceptual labeling of a WBoW. We propose three conceptualization principles for this task and model it as a multi-objective optimization problem. The solution is given by a simple hybrid approach. Our extensive experiments show the high performance in generating conceptual labels for BoWs and WBoWs. Besides, we also apply the generated labels for the task of text classification and observe performance improvement.

Conceptual labeling has extensive applications. In addition to text classification and interpretation for the topic model, other tasks involving text understanding, e.g., text summarization and reading comprehension, will also benefit from the conceptual labels. Therefore, our future work focuses on how to properly incorporate conceptual labels into some NLP tasks.

Funding Information This paper was supported by National Key R&D Program of China No. 2017YFC1201203, National NSF of China No.U1636207 and Shanghai Science and technology innovation action plan (No. 19511120400).

References

1. Beliga, S., Meštrović, A., Martinčić-Ipšić, S.: An overview of graph-based keyword extraction methods and approaches. J. Inf. Organ. Sci. **39**(1), 1–20 (2015)
2. Blei, D.M.: Probabilistic topic models. Commun. ACM **55**(4), 77–84 (2012)
3. Blei, D.M., Mcauliffe, J.D.: Supervised topic models. Adv. Neural Inf. Process. Syst. **3**, 327–332 (2010)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
5. Bloom, P.: Glue for the mental world. Nature **421**(6920), 212–213 (2003)
6. Boutsidis, C., Mahoney, M.W., Drineas, P.: An improved approximation algorithm for the column subset selection problem. In: Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms, pp. 968–977. Society for Industrial and Applied Mathematics (2009)
7. Chaney, A.J., Blei, D.M., Eliassi-rad, T.: A probabilistic model for using social networks in personalized item recommendation. In: Proceedings of the 9th ACM Conference on Recommender Systems, pp. 43–50. ACM (2015)
8. Chasanis, V., Kalogeratos, A., Likas, A.: Movie segmentation into scenes and chapters using locally weighted bag of visual words. In: Proceedings of the ACM International Conference on Image and Video Retrieval, pp. 35. ACM (2009)
9. Deb, K.: Multi-objective optimization. Springer US, 403–449 (2014)
10. Deshpande, A., Rademacher, L.: Efficient volume sampling for row/column subset selection. In: 2010 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 329–338. IEEE (2010)
11. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2018)
12. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. J. R. Stat. Soc. Ser. C (Appl. Stat.) **28**(1), 100–108 (1979)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
14. Hua, W., Wang, Z., Wang, H., Zheng, K.: Short text understanding through lexical-semantic analysis. In: IEEE International Conference on Data Engineering, pp. 495–506 (2015)
15. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv:1607.01759 (2016)
16. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: EACL (2016)
17. Kang, S.H., Sandberg, B., Yip, A.M.: A regularized k-means and multiphase scale segmentation. Inverse Probl. Imaging **5**(2), 407–429 (2017)
18. Kim, D., Wang, H., Oh, A.: Context-dependent conceptualization. In: International Joint Conference on Artificial Intelligence, pp. 2654–2661 (2013)
19. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: AAAI (2015)
20. Lau, J.H., Grieser, K., Newman, D., Baldwin, T.: Automatic labelling of topic models. In: The Meeting of the Association for Computational Linguistics Human Language Technologies, Proceedings of the Conference, Portland, Oregon, pp. 1536–1545 (2012)
21. Lebanon, G., Mao, Y., Dillon, J.: The locally weighted bag of words framework for document representation. J. Mach. Learn. Res. **8**(Oct), 2405–2441 (2007)
22. Liu, C., Sharan, L., Adelson, E.H., Rosenholtz, R.: Exploring features in a bayesian framework for material recognition. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 239–246. IEEE (2010)
23. Mei, Q., Zhai, C.X.: Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In: Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 198–207 (2005)
24. Mei, Q., Liu, C., Su, H., Zhai, C.X.: A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In: International Conference on World Wide Web, pp. 533–542 (2006)
25. Mei, Q., Shen, X., Zhai, C.X.: Automatic labeling of multinomial topic models. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 490–499 (2007)

26. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. Adv. Neural Inf. Process. Syst. **26**, 3111–3119 (2013)
27. Murphy, G.L.: The big book of concepts. MIT Press, Cambridge (2004)
28. Pay, T.: Totally automated keyword extraction. 2016 IEEE International Conference on Big Data (Big Data) pp. 3859–3863 (2016)
29. Prabhumoye, S., Botros, F., Chandu, K., Choudhary, S., Keni, E., Malaviya, C., Manzini, T., Pasumarthi, R., Poddar, S., Ravichander, A., et al.: Building cmu magnus from user feedback. Alexa Prize Proceedings (2017)
30. Ramage, D., Hall, D., Nallapati, R., Manning, C.D.: Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora. In: Conference on Empirical Methods in Natural Language Processing: Volume, pp. 248–256 (2009)
31. Rissanen, J.: Minimum description length principle. Encyclopedia of Statistical Sciences (1985)
32. Roberts, M.E., Stewart, B.M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S.K., Albertson, B., Rand, D.G.: Structural topic models for open-ended survey responses. Am. J. Polit. Sci. **58**(4), 1064–1082 (2014)
33. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic keyword extraction from individual Documents. Wiley, New York (2010)
34. Song, Y., Wang, H., Wang, Z., Li, H., Chen, W.: Short text conceptualization using a probabilistic knowledgebase. The Journal of machine Learning research, pp. 2330–2336 (2011)
35. Song, Y., Wang, H., Wang, H.: Open domain short text conceptualization: a generative + descriptive modeling approach. In: International Conference on Artificial Intelligence, pp. 3820–3826 (2015)
36. Su, Y., Liu, H., Yavuz, S., Gur, I., Sun, H., Yan, X.: Global relation embedding for relation extraction. arXiv:1704.05958 (2017)
37. Sun, X., Xiao, Y., Wang, H.: On conceptual labeling of a bag of words. IJCAI **22**, 1326–1332 (2015)
38. Tomita, E.: Efficient algorithms for finding maximum and maximal cliques and their applications. In: International Workshop on Algorithms and Computation, pp. 3–15 (2017)
39. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NIPS (2017)
40. Wang, X., Mccallum, A.: Topics over time: a non-markov continuous-time model of topical trends. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 424–433 (2006)
41. Wang, Z., Wang, H., Wen, J.R., Xiao, Y.: An inference approach to basic level of categorization. In: The ACM International, pp. 653–662 (2015)
42. Wang, Z., Zhao, K., Wang, H., Meng, X., Wen, J.R.: Query understanding through knowledge-based conceptualization. In: International Conference on Artificial Intelligence, pp. 3264–3270 (2015)
43. Wu, W., Li, H., Wang, H., Zhu, K.Q.: Probase: A probabilistic taxonomy for text understanding. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 481–492. ACM (2012)
44. Zhang, X., Zhao, J.J., LeCun, Y.: Character-level convolutional networks for text classification. In: NIPS (2015)
45. Zhang, Y., Jin, R., Zhou, Z.-H.: Understanding bag-of-words model: a statistical framework. Int J Mach Learn Cybern **1**(1-4), 43–52 (2010)