Hybrid Context-Aware Word Sense Disambiguation in Topic Modeling based Document Representation

1st Wenbo Li
Information Science and Electrical Engineering
Kyushu University
Fukuoka, Japan
liwenbo_923@hotmail.com

2nd Einoshin Suzuki
Information Science and Electrical Engineering
Kyushu University
Fukuoka, Japan
suzuki@inf.kyushu-u.ac.jp

***Abstract*—We propose a hybrid context based topic model for word sense disambiguation in document representation. Document representation is an essential part of various document based tasks, and word sense disambiguation is to capture the distinctions of word senses in the representation. Traditional methods mainly rely on knowledge libraries for data enrichment; however, semantics division for a word may vary from different domain-specific datasets. We aim to discover more particular word semantic differences for each input dataset and handle the disambiguation problem without data enrichment. The challenge for this disambiguation is to (1) divide various senses for each polysemous word while (2) preserve the differences between synonyms. Most of the existing models are either based on separate context clusters or integrating an auxiliary module to specify word senses. They can hardly achieve both (1) and (2) since different senses of a word are assumed to be independent and their intrinsic relationships are ignored. To solve this problem, we estimate a word sense by both the context in which it occurs and the contexts of its other occurrences. Besides, we introduce the "Bag-of-Senses" (BoS) assumption: a document is a multiset of word senses, and the senses are generated instead of the words. Our experiments on three standard datasets show that our proposal outperforms other state-of-the-art methods in terms of accuracy of word sense estimation, topic modeling, and document classification.**

***Index Terms*—document representation, topic model, word sense disambiguation**

I. Introduction

Document representation is the task of mapping the sparse high-dimensional features of documents to low-dimensional space while reflecting their latent semantics [1], [2]. Word Sense Disambiguation (WSD) is the process of identifying a sense of polysemic words [3]. As the basic unit of documents, words are often ambiguous [4]. Simply ignoring the distinctions of word senses could fairly obscure the differences between documents in the semantic space. Therefore, the WSD problem has always been an essential topic in document representation studies [5].

Traditional solutions typically introduce an external standard knowledge library (e.g., Wikipedia, WordNet [6]) as machine-readable sense inventories for data enrichment[1] [8]–[12]. However, in most lexicographic practice, word senses are abstractions from clusters of corpus citations, i.e., the category and the explanation for each sense strongly depend on the semantic coverage[2] of the related dataset. A more extensive semantic coverage corresponds to a coarser granularity of the word semantic division. Therefore, in these knowledge libraries, word senses which are rare, emerging, or confined to a specific domain are typically ignored [13]. For instance, for the word "*religion*" in a politics-related dataset, we may be more concerned about a finer-grained semantic division of different religious groups (e.g., "*the Islam*" and "*the Christian*"), rather than just handling them abstractly as "*a belief in one or more gods*". Besides, the semantics understanding of words in a dataset may also exhibit a unique perspective, which is often domain-specific and implicit in the co-occurrence pattern between each word and its contexts within a dataset [13], [14]. Therefore, handling the WSD problem without data enrichment is a critical issue in the document representation task. The challenge for this disambiguation problem is to divide various senses of each polysemous word while preserving the differences between different words, especially synonyms.

Several researchers model multiple word senses without data enrichment by separate context clusters [15]–[17]. Specifically, they group the contexts of all occurrences for each word into discriminated sense clusters, use these clusters to re-label the words based on the contexts of each occurrence, and then learn word or document representations based on these re-labeled words. These context clustering-based methods can capture different usages of word senses in a dataset without external knowledge libraries. However, relying solely on each clustered contexts is likely to decrease the differences between synonyms, since they often occur in highly similar contexts when representing similar or identical senses, such as the sense "*belief*" for words "*faith*" and "*religion*". Besides, the context in which a word occurs is not necessarily sufficient to specify its sense. For example, "*kick*" contributes more to clarifying the sense of the word "*ball*" than "*play*" because "*play*" has

A part of this work is supported by Grant-in-Aid for Scientific Research JP18H03290 from the Japan Society for the Promotion of Science (JSPS) and the State Scholarship Fund of China Scholarship Council (grant 201706680067).

[1]Data enrichment is defined as merging third-party data from an external authoritative source with an existing database of first-party customer data [7].

[2]Semantic coverage is the coverage of themes relative to a dataset.

a broader sense than "*kick*".

Another kind of solution is to introduce an auxiliary module which is linked through an intermediate variable $\boldsymbol{t}$, e.g., the topic assignment for each word [8], [18], [19]. Identical words combined with different values of $\boldsymbol{t}$ correspond to different senses. This approach can take advantage of the complementary of different models and improve document representation performance. However, there are two risks for the applicability of the word sense division: (1) the differences in senses of identical words with the same value of $\boldsymbol{t}$ could be ignored, and (2) identical words with different $\boldsymbol{t}$ values could be misinterpreted as representing different senses. For example, the word "*key*" in the topic of "*electronic*", might has at least two senses of "*buttons on a keyboard*" and "*string of bits for scrambling and unscrambling*", and the sense "*buttons on a keyboard*" may correspond to at least two topics of "*electronic*" and "*music*". Therefore, it is not always appropriate to impose such a semantic division for each word.

Either of these two kinds of solutions seems unable to construct a common word sense disambiguation standard in document representation. The fundamental reason is that the different senses of a word are mainly assumed to be independent and their intrinsic relationships are ignored. These relationships are an essential basis to clarify the usage differences in other words. For example, the difference between the senses of "*belief*" for "*religion*" and "*faith*" lies in that "*faith*" in something does not necessarily pre-suppose that the belief could not be proven wrong, while "*religion*" is not [20]. Such internal differences of synonyms are challenging to be captured only according to the sense related to their contexts in which they occur, and should also depend on their other senses, e.g., another sense "*ceremonies and duties related to a belief*" of "*religion*" may help clarify its difference to "*faith*" [21].

In this paper, we propose a hybrid context based word sense aware topic model (named HCT), where each sense of a word is estimated by integrating their topic distributions of both the context words in which it occurs and those of its other occurrences. Besides, we introduce the "Bag-of-Senses" (BoS) assumption that a document is a multiset of word senses, based on which HCT generates a word sense instead of the words themselves. The proposed model enjoys two substantial merits over the state-of-the-art methods: (1) no data enrichment or auxiliary module is needed, (2) it is an end-to-end model in which the topic vectors for hybrid contexts as well as their weights for each word are all considered as variables and learned jointly.

II. Related Works

Document representation has long been studied in various areas [22]. Topic modeling and word embedding are two important paradigms for this task. The former takes a global view of the word distributions across the corpus to assign a topic to each word occurrence. The latter is based on a view of the local word collocation patterns observed in a text corpus. For the traditional versions of the two paradigms, such as LDA [23] and Word2vec [24], despite their significant progress in various tasks and applications, there is a common issue that one word corresponds to one topic distribution or embedded vector, while in many cases, the semantics of a word may vary from different senses.

In recent years, lots of studies have been proposed for word sense disambiguation in the document representation task [8], [10]–[12], [25]. Conventionally, they mainly rely on data enrichment, e.g., using knowledge libraries or pre-training datasets, for word sense induction, such as the WordNet [6] based Seeded-LDA [8] and SemLDA [10], the Wikipedia based Token-SDM [26] and LTTM [27], as well as the BooksCorpus [28] and Wikipedia based model ELMO [29], GPT [30], and BERT [12]. All of them have achieved significant progress in word disambiguation performance in document representation, and especially for the recent neural network based language models such as ELMO [29], GPT [30] and BERT [12], have rapidly improved the state-of-the-art on many NLP tasks. Despite their empirical success, the requirements for scales of pre-training datasets and computational efficiency are widely recognized issues due to their large number of parameters (94M for ELMO [29], 340M for BERT [12], and 1542M for GPT [30])) [31]. More importantly, for most of the data enrichment based methods, they assume that word senses are within the scope of the auxiliary text data, while senses in the auxiliary data may not constantly match the ones in a specific dataset. In contrast, we aim to discover more particular word semantic differences for a dataset related to a specific domain, in which we cannot always obtain sufficient scales of domain-specific data for the enrichment.

One solution to solve the WSD problem without data enrichment is by contexts clustering [15]–[17]. DPMM [15] and EHModel [16] both obtain multi-prototype word embeddings by conducting clustering on all context word features for each word. Though useful, they generate multi-prototype word vectors in isolation, ignoring complicated correlations between word senses and their contexts [18]. MSSG [17] improves them by providing flexibility for the number of context clusters, allowing the cluster number varies according to the different distances of contexts in which each word occurs to their nearest sense cluster. Clustering contexts for each word can effectively divide their senses; however, it is challenging to clarify the differences between synonyms due to their similar contexts. Moreover, because of the independence between different clusters, the degree of relationship between a sense and a specific context is ignored.

Another solution is to introduce an additional module to support the word sense disambiguation in document representation. SA-SLDA [8] integrated a Word Sense Induction (WSI) model and are topic models, and the two modules are linked by the topics corresponding to each word. CGTM [32] and w2v-LDA [33] is a topic model using word embedding as an additional component. Topic2Vec [34] and TWE [18] introduce topic vectors in the embedding process, in which the word vector is embedded by concatenating the corresponding word and topic. STE [19] holds the same basic idea as TWE

that combines both the latent topics and word embeddings, but the difference is to learn topical word embeddings in a unified manner. Essentially, they all attempt to link each word occurrence and a specific sense through topics. However, this explicit and compulsory division for word semantics inevitably overlooks the influence of other senses on further clarifying the differences to other words, which leads to either splitting of a sense corresponding to more than one topic or neglecting of multiple senses of a word sharing one topic. Besides, many studies have shown that the two paradigms of word embedding and topic modeling are complementary in how they represent the semantics of documents; thus, the improvement for either of them can contribute to optimizing the performance of their integrated model in document representation [19].

III. Sense Aware Topic Modeling

This section describes in detail the "Bag-of-Senses" (BoS) model, how a word sense is estimated and generated in our BoS based topic model using the hybrid context, and the Gibbs-Sampling [35] based parameter estimation.

A. Bag-of-Senses

As a topic model, the basic task is, for a set of n documents $\mathbb{D} = \{D_0, D_1, ..., D_n\}$, to obtain the topic distribution $\boldsymbol{\theta}_{D_i}$ for each document D_i and word distribution $\boldsymbol{\phi}_k$ for each topic k. Following most traditional topic models [23], $\boldsymbol{\theta}_{D_i}$ and $\boldsymbol{\phi}_k$ are both assumed to be Dirichlet distributions. The number K of topics for each document is assumed fixed and known. Each word corresponds to a K-dimensional topic vector. For the "Bag-of-Words" (BoW)[3] based models, all the word occurrences are mapped to one topic vector, whereas the vector cannot reflect the difference between various senses. Therefore, we propose the "Bag-of-Senses" (BoS) hypothesis: a document d in a dataset, is represented as a multiset of word senses s_w, $d = \{s_w\colon n_{s_w} | w \in \boldsymbol{w}_d\}$, where n_{s_w} is the counts of s_w in d, $\boldsymbol{w}_d$ refers to the multiset of words in d. Each word in d corresponds to a word sense and each sense corresponds to a topic vector. For example, suppose that a document D_i consists of three words "*religion*" where two of them refer to the sense of "*the Islam*" (s_1) and the other one refers to "*the Christian*" (s_2). In the BoW model, D_i is represented as $\{religion : 3\}$, whereas in BoS, it is $\{religion_{s_1} : 2, religion_{s_2} : 1\}$.

B. Hybrid-Context based Sense Estimation

The primary problem is how to define the topic vector of the sense for a word occurrence. Based on the compilation rules of a dictionary that each group of similar usage corresponds to a sense cluster [29], we can reasonably assume that the senses of each word in a specific dataset hold the similar clustered properties. Moreover, the usage differences of senses for each word are reflected by their corresponding different contexts. Therefore, we give the definitions of the Dataset-Specific Word Sense and the Context Words as follows.

[3]The "Bag-of-Words" (BoW) is the most widely used simplifying model in document representation, which assumes a document is a multiset of words, disregarding grammar and even word order but keeping multiplicity [36].

Definition 1. Dataset-Specific Word Sense In BoS, the Dataset-Specific Word Sense for a word is defined as a cluster of similar usage of all its senses in a specific dataset.

Definition 2. Context Words For a word w in a document, given a window size L, the context words $\boldsymbol{w'}$ of word w refers to a set of words within the window.

To ensure both differences between various senses for each polysemes as well as those between synonyms, based on Definitions 1 and 2, a sense vector $\boldsymbol{v}_{\boldsymbol{w}',w}$ for a word w in a specific context $\boldsymbol{w}'$ is estimated by a hybrid of all its sense clusters, where $\boldsymbol{w}'$ is the context words of w. Specifically, let each sense cluster s' of word w correspond to a specific vector $\boldsymbol{v}_w^{s'}$; thus, $\boldsymbol{v}_{\boldsymbol{w}',w}$ is obtained by a mixture as:

$$\boldsymbol{v}_{\boldsymbol{w}',w} = \sum_{s' \in \boldsymbol{S}} \mu_{\boldsymbol{w}',w}^{s'} \boldsymbol{v}_w^{s'}, \tag{1}$$

where $\boldsymbol{S}$ is a set of all sense clusters of word w and $\mu_{\boldsymbol{w}',w}^{s'}$ is its corresponding weight ($\sum_{s' \in \boldsymbol{S}} \mu_{\boldsymbol{w}',w}^{s'} = 1$). Now the problem is how to define $\boldsymbol{v}_w^{s'}$ in a topic space. Explicitly estimating all the sense clusters of each word is difficult, since the cluster number for each word is quite different; thus, to estimate the word sense vector directly by Eq. (1) is intractable. To solve this problem, inspired by the "Distributional Hypothesis" [37] which states that words in similar contexts have similar meanings, we can assume that sense clusters can be reflected in different sets of contexts. Therefore, given a set of contexts of each sense cluster, the vector $\boldsymbol{v}_w^s$ for cluster s can be represented by the average of the vectors for the words in the set of its contexts:

$$\boldsymbol{v}_w^s = \sum_{w' \in \boldsymbol{w}_s'} \boldsymbol{v}_{w'}^g, \tag{2}$$

where $\boldsymbol{w}_s'$ refers to the set of words occurring in the context of all senses in cluster s and $\boldsymbol{v}_{w'}^g$ is the global topic vector of word w'.

Nevertheless, obtaining $\boldsymbol{w}_s'$ is also difficult since we cannot know all the possible contexts of each sense cluster in a dataset beforehand. Therefore, we rewrite Eq. (1) as:

$$\boldsymbol{v}_{\boldsymbol{w}',w} = \mu_{\boldsymbol{w}',w}^{s} \boldsymbol{v}_w^s + \sum_{s' \in \boldsymbol{S}_{-s}} \mu_{\boldsymbol{w}',w}^{s'} \boldsymbol{v}_w^{s'},$$

where $\boldsymbol{S}_{-s}$ is the set of all sense clusters except for s. We see that $\boldsymbol{v}_{\boldsymbol{w}',w}$ can be represented as a combination of one sense cluster vector and a weighted sum of other cluster vectors, while the latter can be approximately regarded as a general vector of w since it contains most of its senses. Hence, we can always find a combination of weights to let $\boldsymbol{v}_{\boldsymbol{w}',w}$ be represented as a weighted sum of a local sense vector $\boldsymbol{v}_w^l$ and a global topic vector $\boldsymbol{v}_w^g$, where $\boldsymbol{v}_w^l$ only concerned about the current context $\boldsymbol{w}'$ ($\boldsymbol{v}_w^l = \sum_{w' \in \boldsymbol{w}'} \boldsymbol{v}_{w'}^g$). Hence, the sense $v_{\boldsymbol{w}',w}$ in a specific context can be calculated as:

$$\boldsymbol{v}_{\boldsymbol{w}',w} = \mu_{\boldsymbol{w}',w} \boldsymbol{v}_w^l + (1 - \mu_{\boldsymbol{w}',w}) \boldsymbol{v}_w^g, \tag{3}$$

where $\mu_{\boldsymbol{w}',w}$ is the corresponding weight. Eq. (3) avoids obtaining all the sense clusters of each word beforehand since the calculation of $\boldsymbol{v}_w$ is independent of sense clusters.

We name the topic vector of a word Global Sense Vector (denoted by $\boldsymbol{v}^g$), the mean vector of its context words Local Sense Vector (denoted by $\boldsymbol{v}^l$), the topic vector of word sense Word Sense Vector (denoted by $\boldsymbol{v}_w$), and the weight of $\boldsymbol{v}^l$ Specific Sense Weight (denoted by $\mu_{\boldsymbol{w}',w}$). Therefore, $\boldsymbol{v}_{\boldsymbol{w}',w}$ for a word w within context words $\boldsymbol{w}'$ can be estimated by its $\boldsymbol{v}^g_w$ and $\boldsymbol{v}^l_{\boldsymbol{w}',w}$. Their definitions are as follows.

Definition 3. Global Sense Vector For a K dimensional topic space, the Global Sense Vector $\boldsymbol{v}^g_w$ is the probability distribution of w for the K topics.

Definition 4. Local Sense Vector For a word w in a context of $\boldsymbol{w}'$, the Local Sense Vector $\boldsymbol{v}^l_{\boldsymbol{w}',w}$ of w is a mean of $\boldsymbol{v}^g$s of its context words:

$$\boldsymbol{v}^l_{\boldsymbol{w}',w} = \sum_{w' \in \boldsymbol{w}'} \boldsymbol{v}^g_{w'}.$$

Definition 5. Word Sense Vector For a word w with context words $\boldsymbol{w}'$, its sense $\boldsymbol{v}_{\boldsymbol{w}',w}$ is a weighted average of its $\boldsymbol{v}^g$ and $\boldsymbol{v}^l$:

$$\boldsymbol{v}_{\boldsymbol{w}',w} = \mu_{\boldsymbol{w}',w}\boldsymbol{v}^l_{\boldsymbol{w}',w} + (1-\mu_{\boldsymbol{w}',w})\boldsymbol{v}^g_w,$$

where $\mu_{\boldsymbol{w}',w}$ is named Local Sense Weight.

C. Word Sense Generation

Based on the above definitions, for a BoS based topic model, a document is generated by word senses, while a specific sense consists of a word and its context words. Therefore, given a topic to the ith word of document d, not only a word is generated but also its context words.

According to Definition 4, using joint probabilities to estimate the generating possibilities of context words is inappropriate because the Local Sense Vector of a word is defined as the mean topic vector of the context words. Therefore, we assume the set of context words $\boldsymbol{w}'$ of word w to be a pseudo word $c_{\boldsymbol{w}'}$ as an observed variable, and takes the average of topic vectors for all the involved words as its own vector. Following LDA [23], the topic-word distribution $\boldsymbol{\phi}_k$ for w and the topic-pseudo word distribution $\boldsymbol{\pi}_k$ for $c_{\boldsymbol{w}'}$ follow two Dirichlet distributions as:

$$\boldsymbol{\phi}_k \sim Dir(\boldsymbol{\beta}), \boldsymbol{\pi}_k \sim Dir(\boldsymbol{\gamma}).$$

Therefore, given a topic k, the word w and its corresponding pseudo word $c_{\boldsymbol{w}'}$ follow two Categorical distributions:

$$w \sim Cat(\boldsymbol{\phi}_k), c_{\boldsymbol{w}'} \sim Cat(\boldsymbol{\pi}_k).$$

According to the conjugate of Dirichlet distribution and Categorical (or Multinomial) distribution, their expectations are calculated as follows:

$$E_{\boldsymbol{\beta}}(\phi_{k,w}) = \frac{n^w_{k,-(d,i)} + \beta_w}{\sum_{f=1}^{V}(n^f_{k,-(d,i)} + \beta_f)} \tag{4}$$

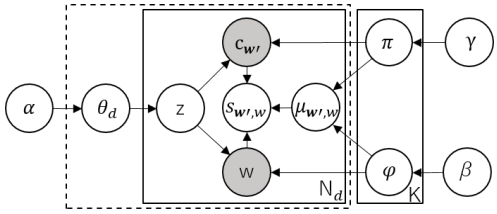


Fig. 1. Plate notation of HCT.

$$E_{\boldsymbol{\gamma}}(\pi_{k,\boldsymbol{w}'}) = \frac{\sum_{t \in \boldsymbol{w}'}(n^t_{k,-(d,i)} + \gamma_t)}{L\sum_{f=1}^{V}(n^f_{k,-(d,i)} + \gamma_f)}, \tag{5}$$

where $\phi_{k,w}$ and $\pi_{k,\boldsymbol{w}'}$ respectively refer to the probabilities of generating word w and $c_{\boldsymbol{w}'}$ given topic k. L is the size of context window. $n^t_{k,-(d,i)}$ is the number of word t belonging to topic k without the ith word in document d. Based on Definition 5, we introduce a hidden variable $s_{\boldsymbol{w}',w}$ to present the sense of word w in context $\boldsymbol{w}'$, where $s_{\boldsymbol{w}',w}$ is generated from:

$$s_{\boldsymbol{w}',w} \sim (1-\mu_{\boldsymbol{w}',w})\phi_{k,w} + \mu_{\boldsymbol{w}',w}\pi_{k,\boldsymbol{w}'}.$$

For weight $\mu_{\boldsymbol{w}',w}$, based on Definition 5, it can be regarded as the probability for $v^l_{\boldsymbol{w}',w}$ in a mixture of topic distributions of $\boldsymbol{v}^g_w$ and $\boldsymbol{v}^l_{\boldsymbol{w}',w}$. Therefore, given topic k, $\mu_{\boldsymbol{w}',w}$ can be estimated by the Bayes Rule [38]. Specifically, for the ith word w in document d, we obtain $\mu_{\boldsymbol{w}',w}$ by the probabilities of topic k in $\boldsymbol{v}^g_w$ and $\boldsymbol{v}^l_{\boldsymbol{w}',w}$, as Eq. (6):

$$\mu_{\boldsymbol{w}',w} \triangleq P(v^l_{\boldsymbol{w}',w}|k) = \frac{P(k, v^l_{\boldsymbol{w}',w})}{P(k, \boldsymbol{v}^g_w) + P(k, v^l_{\boldsymbol{w}',w})}, \tag{6}$$

where $P(k, \boldsymbol{v}^g_w)$ refers to the probability of topic k in $\boldsymbol{v}^g$ and $P(k, \boldsymbol{v}^l_{\boldsymbol{w}',w})$ refers to that in $\boldsymbol{v}^l_{\boldsymbol{w}',w}$. Their calculations are:

$$P(k, \boldsymbol{v}^g_w) = \frac{\phi_{k,w}}{\sum_{s=1}^{K}(\phi_{s,w})} \propto \frac{n^w_{k,-(d,i)} + \beta_w}{\sum_{s=1}^{K}(n^w_{s,-(d,i)} + \beta_w)},$$

$$P(k, v^l_{\boldsymbol{w}',w}) = \frac{\pi_{k,\boldsymbol{w}'}}{\sum_{s=1}^{K}(\pi_{s,\boldsymbol{w}'})} \propto \frac{1/L\sum_{t \in \boldsymbol{w}'} n^t_{k,-(d,i)} + \gamma_t}{\sum_{s=1}^{K}\left[1/L(\sum_{t \in \boldsymbol{w}'} n^t_{k,-(d,i)} + \gamma_t)\right]},$$

where $n^w_{k,-(d,i)}$ is the number of word w in the dataset which belongs to topic k without the ith word in document d.

D. Model Description

The plate notation is as shown in Figure 2. We introduce five new variables $\boldsymbol{\pi}$, $\boldsymbol{\gamma}$, $c_{\boldsymbol{w}'}$, $s_{\boldsymbol{w}',w}$ and $\boldsymbol{\mu}_{\boldsymbol{w}',w}$ to traditional LDA, where $\boldsymbol{\pi}$ represents the topic-pseudo word distribution with a parameter of $\boldsymbol{\gamma}$. $c_{\boldsymbol{w}'}$ refers to a pseudo word for the average of context words. $s_{\boldsymbol{w}',w}$ represents the sense of word w in context $\boldsymbol{w}'$. $\boldsymbol{\mu}_{\boldsymbol{w}',w}$ refers to the Local Sense Weight. For the other variables, $\boldsymbol{\theta}_d$ represents the topic distribution of document d with parameter $\boldsymbol{\alpha}$. $\boldsymbol{\phi}_k$ is the topic-word distribution of topic k with parameter $\boldsymbol{\beta}$. w is a word in a document and z is its corresponding topic. For a dataset of D documents with a

Algorithm 1: Parameter Estimation Algorithm

Input: A set of D documents of length N_d; number N_{iter} of iterations; number K of topics; Dirichlet parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$; context window size L

Output: For each document d, topic distribution θ_d; for each topic k, word distribution ϕ_k $(1 \leq k \leq K)$;

Initialize topic assignments randomly and set $\mu_{\boldsymbol{w}',w}$ by 0.5 for all words in documents D with context words of $\boldsymbol{w}'$

for *iteration = 1 to* N_{iter} **do**
  for d *= 1 to* D **do**
    for i *= 1 to* N_d **do**
      Update $\mu_{\boldsymbol{w}'_{(d,i)},w_{(d,i)}}$ by Eq. (6).
      Assign a topic $z_{(d,i)}$ from $\boldsymbol{P}_{(d,i)}$ by Eq. (7).
    Update θ_d and topic-word matrix $\boldsymbol{M}$.

return topic-word matrix $\boldsymbol{M}$, θ_d for each document d as well as $\mu_{\boldsymbol{w}'_{(d,i)},w_{(d,i)}}$ for each word.

vocabulary of size V and latent topics indexed in $\{1, ..., K\}$, the generative process of HCT is described as follows:

1) Generate $\boldsymbol{\phi}_k$ for each topic k: $\boldsymbol{\phi}_k \sim Dir(\boldsymbol{\beta})$.
2) For each document d:
   a) Generate $\boldsymbol{\theta}_d$ for document d: $\boldsymbol{\theta}_d \sim Dir(\boldsymbol{\alpha})$.
   b) For each word w in d (index by i):
      i) Assign topic $z_{(d,i)}$ by $\boldsymbol{\theta}_d$: $z_{(d,i)} \sim Cat(\boldsymbol{\theta}_d)$.
      ii) Obtain context words $\boldsymbol{w'}$ and generate topic-pseudo word distribution $\boldsymbol{\pi}_k$: $\boldsymbol{\pi}_k \sim Dir(\boldsymbol{\gamma})$.
      iii) Generate w by $\boldsymbol{\phi}_k$: $w \sim Cat(\boldsymbol{\phi}_k)$.
      iv) Generate $c_{\boldsymbol{w}'}$ by $\boldsymbol{\pi}_k$: $c_{\boldsymbol{w}'} \sim Cat(\boldsymbol{\pi}_k)$.
      v) Calculate $\mu_{\boldsymbol{w}',w}$ by Eq. (6).
      vi) Generate $s_{\boldsymbol{w}',w}$ by $z_{(d,i)}$, $\phi_{k,w}$ and $\pi_{k,\boldsymbol{w}'}$: $s_{\boldsymbol{w}',w} \sim (1-\mu_{\boldsymbol{w}',w})\phi_{z_{(d,i)},w} + \mu_{\boldsymbol{w}',w}\pi_{z_{(d,i)},\boldsymbol{w}'}$.

$\boldsymbol{\phi}_k$ and $\boldsymbol{\pi}_k$ share the same topic-word matrix $\boldsymbol{M}$ which records the number of occurrence for each word in different topics. Based on $\boldsymbol{M}$, $\boldsymbol{\phi}_k$ and $\boldsymbol{\pi}_k$ are calculated with reference to their respective Dirichlet parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. Each row and column of $\boldsymbol{M}$ respectively corresponds to a topic-word distribution and a topic vector of a word.

E. Parameter Estimation

For complex probability models, obtaining the optimal parameters directly by point estimation is difficult. Therefore, except for $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, the parameters of our model are approximately estimated by Gibbs sampling [35], which is one of the widely used sampling methods based on Markov chain Monte Carlo (MCMC) [39]. In the estimation procedure, we need to calculate conditional distribution $P_{(d,i),k}$= $P(z_{(d,i)}{=}k|w_{d,i},\boldsymbol{z}_{d,-(d,i)},\boldsymbol{w}'_{(d,i)},\boldsymbol{\mu}_{(d,i)},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})$, for each document d, where $w_{(d,i)}$ represents the ith word in d and $\boldsymbol{z}_{d,-(d,i)}$ refers to the topic assignments for all words in d except word $w_{(d,i)}$. $\boldsymbol{w}'_{(d,i)}$ is the context words of $w_{(d,i)}$ and $\mu_{(d,i)}$ refers to the Local Sense Weight of $w_{(d,i)}$. $P_{(d,i),k}$ is computed as follows (See Appendix B for detailed derivation):

$$P_{(d,i),k} \propto E_{\boldsymbol{\alpha}}(\theta_{d,k}) \left[(1-\mu_{(d,i)})E_{\boldsymbol{\beta}}(\phi_{k,t}) + \mu_{(d,i)}E_{\boldsymbol{\gamma}}(\pi_{k,\boldsymbol{w}'_{(d,i)}})\right], \quad (7)$$

where $E_{\boldsymbol{\alpha}}(\theta_{d,k})$ refers to the expectation of the probability for topic k in document d, which can be estimated by:

$$E_{\boldsymbol{\alpha}}(\theta_{d,k}) \propto (n_{d,k,-(d,i)} + \alpha), \quad (8)$$

where $n_{d,k,-(d,i)}$ is the number of words in d belonging to topic k. $E_{\boldsymbol{\beta}}(\phi_{k,t})$ and $E_{\boldsymbol{\gamma}}(\pi_{k,\boldsymbol{w}'_{(d,i)}})$ are the expectations of the probabilities for word w_t and pseudo word of context words $\boldsymbol{w}'_{(d,i)}$. They are calculated by Eqs. (4) and (5). Based on Eq. (7), we obtain topic assignment probability $P_{(d,i),k}$ for each word in d, so as to compute their corresponding topic distribution $\boldsymbol{P}_{(d,i)}$. Detailed steps are shown in Algorithm 1.

IV. Experiments

We conducted both quantitative and qualitative analyses. Firstly, we use three benchmark datasets 20Newsgroups[4] (20NG), Toxic Comments[5] (T-COM) and Sanders Tweet[6] (Tweet) in the quantitative analysis for evaluating the word sense estimation qualities, document classification effects, and topic modeling accuracy. In qualitative analysis, we use 20NG and T-COM to verify the effects of our approach in capturing various domain-specific word senses.

20NG is a collection of approximately 20,000 newsgroup documents, organized into 20 different newsgroups, each corresponding to a different topic. T-COM is a dataset of Wikipedia comments which human raters have labeled for toxic behavior, i.e., comments which are rude, disrespectful, or controversial. Tweet is a twitter sentiment corpus created by Sanders Analytics, which consists of 5513 hand-classified tweets. Each tweet was classified for one of four different topics. For all the datasets, stop words were removed in advance.

To validate the proposed model HCT, we test the following baseline methods: a traditional topic model LDA [23], two word embedding methods combined with topic modeling, TWE-1 [18] and STE [19], two topic models CGTM [32] and w2v-LDA [33] which are combined with a Skip-gram based word embedding model, as well as two sense cluster based embedding methods EHModel [16] and MSSG [17]. Moreover, in the quantitative analysis, we combine our model HCT with a skip-gram based word embedding framework [24] as another testing method (denoted by HCT-S). Its integration principle is similar to TWE-1 [18], where the difference is that we take the sense vector for each word occurrence rather than its topic assignment as additional input features. The hyper-parameters ($\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$) were all fixed to 0.05, and the size

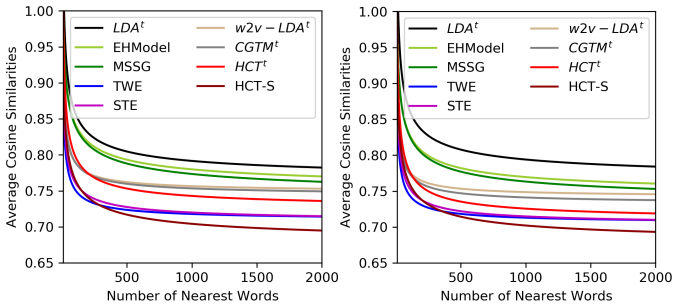
[4]http://qwone.com/~jason/20Newsgroups/

[5]http://kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data

[6]https://github.com/zfz/twitter_corpus

TABLE I
COMPARISON OF THE AVERAGE SIMILARITIES BETWEEN SENSE CLUSTERS OF EACH WORD ($\overline{C_{sc}}$).

	K	EHModel	MSSG	TWE-1	STE	w2v-LDA^t	CGTM^t	HCT^t	HCT-S
$\overline{C_{sc}}$	100	**0.819**	0.822	0.849	0.832	0.887	0.876	0.848*	0.833
	200	0.815	**0.814**	0.837	0.821	0.875	0.864	0.836*	0.825



(a) K=100 (b) K=200

Fig. 2. Comparison of average similarities between vectors of each word and its Top-n ($n \in [1, 2000]$) nearest words based on the cosine similarity on 20NG dataset.

L of context window L=10 (See Appendix A for parameter setting experiments).

A. Quantitative Analysis

The quantitative experiments are conducted on 20NG, T-COM, and Tweet, in terms of three aspects: classification effect, sense estimation quality, and topic modeling accuracy.

1) Sense Estimation Quality: To investigate sense estimation qualities, we evaluate the differences among various senses for each word and those among its synonyms on 20NG. In this analysis, the differences are measured by the cosine similarity. A lower value reflects higher discrimination between different senses or words, corresponding to a better sense estimation quality. We used KMeans [40] to cluster the sense vectors, where the cluster number are determined by Silhouette Coefficient[7]. We chose the cluster number (from 2 to 10) with the highest Silhouette value as the parameter for KMeans, while other parameters remained default.

We calculated the average cosine similarities between sense clusters of each word (denoted by $\overline{C_{sc}}$, as shown in Table I) , as well as the average cosine similarities between each word vector and its Top-n ($n \in [1, 2000]$) nearest word vectors (denoted by $\overline{C_w}$, as shown in Figure 2), where the word vector is represented by the mean of its sense vectors. * indicates the best scores yielded by topic models (labeled by ^{t}), and bold fonts indicate the best ones of all the baselines. We see that the lowest $\overline{C_{sc}}$ are achieved by the clustered based methods (EHModel and MSSG). In all cases, HCT-S is superior to most of the others, and HCT performs the best in the topic models. The word semantic divisions by contexts clustering can directly clarify the distinctions between different senses of a word. However, it may obscure the differences to other words, especially those with similar usages. There is a trade-off between improving the sense differences of a word and preserving the word distinctions with other words. Ignoring the differentiation to other words while dividing the word senses is likely to confuse their similar sense clusters, and thus reducing the differences between words. Furthermore, for the word embedding based methods, their $\overline{C_w}$ are lower than the topic models as the number of nearby words involved increases. One possible reason is that the optimization targets of these two paradigms are different. The embedding models focus on optimizing word vectors, while topic models aim at optimizing the topic distributions of documents. This difference directs the embedding vectors more sufficiently to reflect the semantic similarities and differences between words. In the following experiments, we will discuss their complementary nature in the document representation task.

[7]The Silhouette Coefficient ranges in $[-1, 1]$, where a higher value indicates that the object is better matched to its own cluster and poorly matched to other clusters [26].

TABLE II
COMPARISON OF CLASSIFICATION PERFORMANCE ON T-COM.

Method	K	Precision	Recall	F-Score
LDA^t	100	0.774±0.009	0.767±0.009	0.771±0.008
EHModel		0.771±0.009	0.763±0.009	0.767±0.007
MSSG		0.783±0.009	0.779±0.007	0.781±0.007
TWE-1		0.819±0.007	0.824±0.007	0.821±0.006
STE		0.825±0.008	0.828±0.008	0.826±0.008
w2v-LDA^t		0.826±0.009	0.822±0.009	0.824±0.008
CGTM^t		0.828±0.008	0.825±0.008	0.827±0.007
HCT^t		0.835±0.009*	0.831±0.009*	0.832±0.008*
HCT-S		**0.851±0.011**	**0.854±0.009**	**0.852±0.011**
LDA^t	200	0.806±0.005	0.788±0.005	0.797±0.005
EHModel		0.808±0.005	0.801±0.005	0.804±0.005
MSSG		0.814±0.005	0.811±0.005	0.812±0.005
TWE-1		0.825±0.004	0.823±0.004	0.824±0.004
STE		0.831±0.004	0.834±0.004	0.832±0.003
w2v-LDA^t		0.831±0.007	0.830±0.007	0.830±0.007
CGTM^t		0.835±0.005	0.832±0.005	0.832±0.005
HCT^t		0.849±0.005*	0.852±0.005*	0.849±0.005*
HCT-S		**0.862±0.007**	**0.868±0.007**	**0.865±0.007**

TABLE III
COMPARISON OF CLASSIFICATION PERFORMANCE ON 20NG.

Method	K	Precision	Recall	F-Score
LDA^t	100	0.689±0.014	0.663±0.015	0.676±0.012
EHModel		0.799±0.013	0.797±0.013	0.794±0.013
MSSG		0.814±0.014	0.812±0.015	0.813±0.012
TWE-1		0.848±0.012	0.847±0.012	0.847±0.011
STE		0.851±0.011	0.857±0.011	0.854±0.010
w2v-LDA^t		0.839±0.0011	0.831±0.0011	0.835±0.0011
CGTM^t		0.841±0.009	0.835±0.009	0.838±0.008
HCT^t		0.857±0.012*	0.851±0.013*	0.855±0.012*
HCT-S		**0.877±0.014**	**0.871±0.014**	**0.875±0.014**
LDA^t	200	0.708±0.005	0.707±0.005	0.707±0.005
EHModel		0.811±0.013	0.807±0.013	0.808±0.013
MSSG		0.824±0.014	0.821±0.015	0.823±0.012
TWE-1		0.857±0.012	0.855±0.012	0.856±0.011
STE		0.863±0.011	0.859±0.011	0.861±0.010
w2v-LDA^t		0.845±0.007	0.838±0.007	0.841±0.007
CGTM^t		0.848±0.005	0.841±0.005	0.845±0.005
HCT^t		0.868±0.005*	0.872±0.005*	0.871±0.005*
HCT-S		**0.878±0.007**	**0.882±0.007**	**0.881±0.007**

2) Document Classification: To evaluate the quality of document representation vectors, we conducted classification

TABLE IV
COMPARISON OF CLASSIFICATION PERFORMANCE ON TWEET.

Method	K	Precision	Recall	F-Score
LDA^t	100	0.650±0.021	0.651±0.021	0.651±0.020
EHModel		0.668±0.011	0.664±0.011	0.665±0.011
MSSG		0.677±0.013	0.675±0.011	0.675±0.011
TWE-1		0.682±0.008	0.681±0.008	0.682±0.008
STE		0.691±0.009	0.688±0.009	0.689±0.008
w2v-LDA^t		0.683±0.008	0.682±0.007	0.682±0.007
CGTM^t		0.690±0.008*	0.686±0.007	0.688±0.007*
HCT^t		0.688±0.008	0.687±0.008*	0.687±0.008
HCT-S		**0.717±0.009**	**0.716±0.009**	**0.716±0.009**
LDA^t	200	0.651±0.008	0.653±0.008	0.653±0.007
EHModel		0.671±0.006	0.675±0.006	0.672±0.005
MSSG		0.682±0.006	0.679±0.006	0.680±0.005
TWE-1		0.687±0.005	0.685±0.005	0.685±0.005
STE		0.702±0.006	0.697±0.006	0.697±0.005
w2v-LDA^t		0.687±0.005	0.686±0.004	0.686±0.004
CGTM^t		0.697±0.005	0.695±0.004	0.695±0.004
HCT^t		0.699±0.005*	0.702±0.005*	0.700±0.005*
HCT-S		**0.725±0.007**	**0.728±0.007**	**0.727±0.007**

experiments on three benchmark datasets. We randomly sampled 12000 documents from 20NG, 10000 documents from T-COM, and all the four classes in Tweet. We use Support Vector Machines (SVM) [41] to predict ground truth labels from the topic vectors of documents and used WEKA [42] for learning a classifier with ten-fold cross-validation and default parameters. The precision and recall as well as the macro averaged F1-Score [43] (with the representation vector dimension K=100, 200) are presented as the evaluation metrics for this task. The results (the mean and standard deviation) are reported in Tables II, III, and IV, where * indicates the best scores achieved by the topic models, and bold fonts indicate the best scores achieved by all the models. Topic models are labeled by ^{t}. We see that HCT shows the best results in the topic modeling based methods in most cases, and the integrated method HCT combined with skip-gram is superior to all the other baseline models on the three datasets.

Classification performance reflects the ability to distinguish different classes of documents in their representation spaces. HCT considers the relationships between the different senses of each word in topic modeling, and thus achieve a better trade-off between the differentiation of various senses of each word and the semantic differences of synonyms. For short text datasets, the context words for each word may occupy the vast majority of the document; thus, their influence on the word sense may counteract the role of the document itself, such as the topic distribution. Therefore, our model has limited improvement in classification accuracy on short text datasets compared to other baselines. On the other hand, the integrated models, which combine both the topic modeling and word embedding, are more effective than the other baselines. However, they both assume word senses under each topic dimension are different. Nevertheless, it is common that a sense could belong to multiple topics, and the number of senses for each word is different; thus, this explicit and compulsory division for word senses is likely to decrease the accuracy of their embedded vectors. Another significant issue is that all the baseline models neglect the degree of dependency for a word sense on its context words. However, these degrees of the dependencies of a sense varies from its usage frequency. For example, a non-standard use of a word is more dependent on its context than its standard use [13], [30]. These problems might be the leading causes of their performance bottlenecks.

TABLE V
COMPARISON OF NPMIS (THE MEAN AND STANDARD DEVIATION) ON DATASETS T-COM, 20NG, AND TWEET ($K = 100, 200$).

Method	K	NPMI		
		T-COM	20NG	Tweet
LDA	100	-12.2±0.3	-8.2±0.3	-15.3±0.4
w2v-LDA		-9.6±0.3	-7.7±0.5	-12.8±0.6
CGTM		-8.4±0.5	-6.7±0.4	**-11.3±0.5**
HCT		**-7.8±0.3**	**-6.5±0.3**	-11.4±0.3
LDA	200	-13.6±0.4	-9.4±0.2	-16.6±0.2
w2v-LDA		-11.5±0.4	-8.3±0.3	-13.7±0.4
CGTM		-10.3±0.4	-7.8±0.3	-12.2±0.2
HCT		**-9.7±0.3**	**-7.2±0.3**	**-11.8±0.2**

Besides, the complementarity of topic modeling and word embedding improves the performance of the integrated methods. For most topic modeling based methods, embeddings are mainly used to improve the accuracy of the topic assignment for each word (CGTM and w2v-LDA). This indirect influence on topic modeling cannot sufficiently reflect the context information captured by the embedding models. For the embedding based models (TWE-1 , STE, and HCT-S), the topic modeling results are inputted as additional features and directly utilized in the word vector estimation, which might be the main reason for embedding-based integrated methods being generally better than other integrated ones in this analysis.

3) Topic Modeling Accuracy: As a topic modeling method, we evaluate the accuracy of the discovered topics by calculating the average normalized pointwise mutual information (NPMI) for each method. NPMI is a popular metric of topic modeling quality by measuring the coherence of a topic based on point-wise mutual information [44]. It assumes that a topic is more coherent if the most probable words in the topic co-occur more frequently [45]. Besides, topic coherence can also reflect the matching between the topic assignment and semantics for each word, since semantic expressions in a document are usually coherent and segmented (such as paragraphs and sections) [46]. A higher NPMI score indicates that the topic distributions are semantically more coherent. Given the T most probable words of topic k, the NPMI is:

$$\mathrm{NPMI}(k) = \sum_{1 \le i < j \le T} \frac{1}{-\log P(w_i, w_j)} \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)},$$

where $P(w_i, w_j)$ and $P(w_i)$ are the probabilities of word pair (w_i, w_j) and word w_i, respectively, and are both estimated from an external dataset[8].

The results of topic models (HCT, LDA, w2v-LDA, and CGTM) are shown in Table V, where bold fonts highlight the

[8]We use English Wikipedia as the external dataset, and collected words that co-occur in a window of ±5 (https://dumps.wikimedia.org/enwiki/).

best results. We see HCT shows the best results in most cases, which confirms that our model can generate more accurate document vectors. HCT generates both words and context words as well as uses the context and adaptive weights to clarify word semantics, reducing the uncertainty of word topic assignment. The others use embedding vectors to clarify the word topic assignment [33]. However, the embedding vectors are learned by all their contexts, which is difficult to help specify a rare sense for a word in a specific context.

B. Qualitative Analysis

We conducted qualitative experiments on 20NG and T-COM, where we set the representation topic number K=200. Firstly, we verify whether our model can capture useful domain-specific senses by the estimated sense vectors. We randomly sampled 10000 documents from 20NG with 20 classes[9] and 7000 comments from T-COM covering three sensitive themes of "religion", "race", and "homosexuality". We respectively select three high frequent words which are likely to cover the most related themes of each datasets according to the Longman Dictionary[10] ("*card*", "*power*", "*key*" for 20NG, and "*religion*", "*race*", "*homosexuality*" for T-COM) as examples and compute the Word Sense Vector $\boldsymbol{v}_{\boldsymbol{w}',w}$ of each word w within each context $\boldsymbol{w}'$ by Definition 5. We used the same settings as those in quantitative analysis for sense vectors clustering, and visualized the results by t-SNE [40].

As shown in Figure 3 (a-f), each point represents a sense vector, and each color refers to a sense cluster. We see that the sense vectors exhibit varying degrees of clustered properties. This observation verifies our hypothesis in Definition 1. For further study of the semantics for each cluster, we then counted the high frequent context words for each cluster and presented the interpretations which are likely to be relevant to these clusters based on the Longman Dictionary, as shown in Tables VI and VII. From Table VI, we see that although not all groups of context words can be abstracted to an exact meaning, the differences between them are clear. For instance, the two sense clusters "*power*" possibly correspond to "*a kind of energy*" ($\boldsymbol{c}_1$) and "*a supernatural ability*" ($\boldsymbol{c}_2$), respectively. The two clusters ($\boldsymbol{c}_1$ and $\boldsymbol{c}_2$) of "*card*" respectively refers to "*a computer-related equipment*" and "*a person identification certificate*". For the word "*key*", the differences between the clusters are obvious, where the senses of $\boldsymbol{c}_1$ are possibly relevant to the sense "*encryption*", the ones of $\boldsymbol{c}_2$ may refer to "*the keyboard buttons*", $\boldsymbol{c}_3$ possibly refer to "*a tool to lock or unlock a door*", and $\boldsymbol{c}_4$ may represents "*a kind of password or serial number*". Combining Tables VI and VII, we see that the interpretations for the above senses might be found in the knowledge library, while the following three examples show more particular and fine-grained senses for a specific dataset. For example, our model captured three entities that the word "*religion*" refers to: "*the Christianity*" ($\boldsymbol{c}_1$), "*the Islam*" ($\boldsymbol{c}_2$) and "*the communist*" ($\boldsymbol{c}_3$). Moreover, we can also recognize the positions or tendencies represented by different clusters according to obviously uncomfortable or discriminatory context words, such as $\boldsymbol{c}_2$ and $\boldsymbol{c}_3$ of the word "*race*", as well as $\boldsymbol{c}_2$ of "*homosexual*". These results confirm our assumptions about the word sense vector, e.g., Definition 5, and the effectiveness of obtaining domain-specific senses.

[9]These 20 classes mainly cover the themes of "electronics", "sports", "religion", "politics", "industry" (http://qwone.com/~jason/20Newsgroups/).

[10]https://www.ldoceonline.com/dictionary/

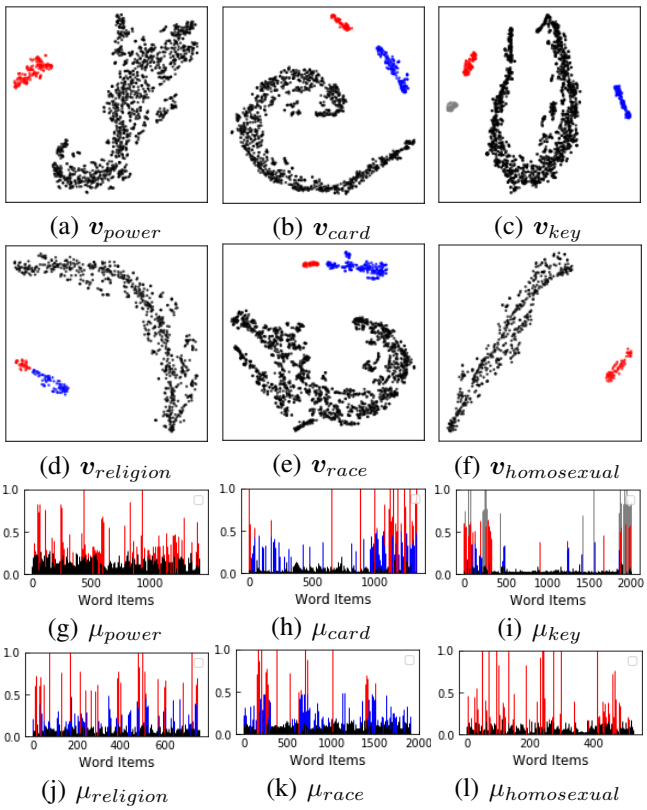


Fig. 3. Visualization for each example word w by their Word Sense Vectors ($\boldsymbol{v}_w$) and the corresponding Local Sense Weights (μ_w) in the 20NG. Each point in (a-f) or bar in (g-l) refers to a word item in the dataset, where each color corresponds to a sense cluster.

Besides, we further analyze the relationships between the Local Sense Weight $\mu_{\boldsymbol{w}',w}$ and $\boldsymbol{v}_{\boldsymbol{w}',w}$. Figure 3 (g-l) shows the weights for each example word, where the colors of bars correspond to those of the clusters in Figure 3 (a-f). We observe that clusters with fewer sense vectors have higher weights than others, and vectors that belong to the same cluster correspond to similar weights. These observations signify that $\mu_{\boldsymbol{w}',w}$ reflects a difference between a sense cluster and its corresponding general one. The higher the weights, the more different from its general sense and the more dependent on its context words. These phenomena also confirm a viewpoint of a lexicography sect about the formation of word senses, that corpus citations of a word fall into one or more distinct but related clusters. Each of these clusters, if large enough and distinct enough from others, forms a distinct word sense [13].

V. Conclusion

We propose a hybrid context based topic model for handling the WSD problem in document representation without data

TABLE VI

CONTEXT WORDS FOR EACH SENSE CLUSTER OF EXAMPLE WORDS. BOLD FONTS INDICATE THE HIGH-FREQUENCY CONTEXT WORDS WHICH HELP CLARIFY THE SEMANTIC DIFFERENCE. THE COLOR FOR EACH CLUSTER SYMBOL $\boldsymbol{c}$ CORRESPONDS TO THAT OF EACH CLUSTER IN FIGURE 3.

Word	Cluster	Top-10 High Frequent Words in Context
power	$\boldsymbol{c}_1$	"people", "problem", "**drive**", "**supply**", "**connector**", "hard", "disk", "**nuclear**", "**battery**", "**device**"
	$\boldsymbol{c}_2$	"**god**", "**lord**", "play", "person", "**christ**", "**jesus**", "son", "human", "town", "believe"
card	$\boldsymbol{c}_1$	"**video**", "**drive**", "system", "**graphic**", "problem", "**controller**", "support", "**monitor**", "vga", "bit"
	$\boldsymbol{c}_2$	"people", "citizen", "carry", "**identify**", "letter", "recognize", "signed", "**nationality**", "**number**", "**authority**"
	$\boldsymbol{c}_3$	"key", "lose", "**tool**", "guess", "remember", "law", "insurance", "left", "hold", "**game**"
key	$\boldsymbol{c}_1$	"chip", "escrow", "**system**", "public", "bit", "**encryption**", "**number**", "message", "**security**", "**algorithm**"
	$\boldsymbol{c}_2$	"keyboard', "**character**", "**application**", "file", "**program**", "**system**", "change", "monitor", "sequence", "**code**"
	$\boldsymbol{c}_3$	"**home**", "**car**", "**door**", "**lock**", "know", "line", "people", "work", "launch", "available"
	$\boldsymbol{c}_4$	"window"', "**drive**", "write", "**machine**", "**number**", "theory", "release", "cable", "printer", "**series**"
religion	$\boldsymbol{c}_1$	"god", "state", "**jewish**", "**christian**", "**judaism**", "history", "faith", "people", "source", "life"
	$\boldsymbol{c}_2$	"**islam**", "god", "faith", "people", "politics", "truth", "culture", "**muslim**", "history", "believe"
	$\boldsymbol{c}_3$	"god", "eastern", "**socialist**" , "**communist**", "science", "people", "politics", "believe", "ethnicity", "sex"
race	$\boldsymbol{c}_1$	"people", "concept", "**religion**", "ethnicity", "difference", "article", "**language**", "**human**", "world", "author"
	$\boldsymbol{c}_2$	"**nazi**", "**holocaust**", "victims", "sex", "people", "dark", "age", "prison", "crime", "family"
	$\boldsymbol{c}_3$	"**white**", "ethnic", "people", "**black**", "group", "world", "**african**", "**asian**", "**nationalism**", "war"
homosexual	$\boldsymbol{c}_1$	"article", "**gay**", "people", "life", "**sex**", "children", "enjoy", "female", "male", "parent"
	$\boldsymbol{c}_2$	"man", "**ass**", "**fuck**", "**shit**", "**hole**", "**beat**", "dog", "think", "pick", "piece"

TABLE VII

INTERPRETATIONS IN THE LONGMAN DICTIONARY FOR THE GENERATED SENSE CLUSTERS. $\boldsymbol{c}_i$(S) IN EACH ROW REPRESENTS THE POSSIBLY RELATED CLUSTER(S).

Word	Cluster(s)	Interpretation
power	$\boldsymbol{c}_1$	"*energy that make a machine work*"
card	$\boldsymbol{c}_1$	"*a piece of equipment in a computer*"
	$\boldsymbol{c}_2$	"*a small piece of plastic or paper that contains information about a person*"
key	$\boldsymbol{c}_2$	"*the buttons on a computer keyboard*"
	$\boldsymbol{c}_3$	"*a specially shaped piece of metal to lock or unlock a door, start a car etc*"
religion	$\boldsymbol{c}_1$, $\boldsymbol{c}_2$	"*a belief in one or more gods*"
race	$\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\boldsymbol{c}_3$	"*one of the main groups that humans can be divided into by their colour of skin or other physical features*"
homo-sexual	$\boldsymbol{c}_1$, $\boldsymbol{c}_2$	"*someone, especially a man, is sexually attracted to people of the same sex*"

enrichment. By integrating topic distributions of both the context in which a word occurs and those of its other occurrence in sense estimation, the proposed model effectively captures domain-specific word senses and preserves the differences between synonyms. Besides, we proposed the "Bag-of-Senses" hypothesis, based on which our model generates senses instead of words. Our experiments confirm the effectiveness of our model to obtain the domain-specific word sense vectors and showed that our proposal outperforms the baseline models in terms of sense estimation quality, classification performance, and topic modeling accuracy. In future work, we will further optimize the parameter estimation steps and use more efficient algorithms (e.g., the Variational Inference [47]) to improve the adaptiveness of our model for more substantial scale datasets.

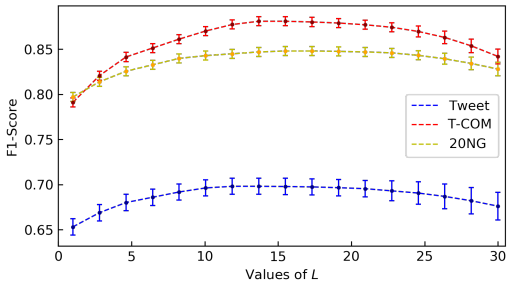


Fig. 4. F1-Scores (the mean and standard deviation) with different values of L on "Tweet", "T-COM" and "20NG".

APPENDIX A
DERIVATION OF OUR GIBBS SAMPLING

For each document d, the posterior probability, $P_{(d,i),k}$ is computed as follows:

$$
\begin{aligned}
P_{(d,i),k} &\propto P(z_{(d,i)} = k, s_{\boldsymbol{w}'_{(d,i)},(d,i)} = s_t | \boldsymbol{w}'_{(d,i)}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \\
&= P(z_{(d,i)} = k, w_{(d,i)} = t, c_{\boldsymbol{w}'_{(d,i)}} = c_{\boldsymbol{t}'} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \\
&= \int P(z_{(d,i)} = k | \boldsymbol{\theta}_d) P(\boldsymbol{\theta}_d | \boldsymbol{\alpha}) d\boldsymbol{\theta}_d \\
&\quad \Bigg[(1 - \mu_{(d,i)}) \int P(w_{(d,i)} = t | \boldsymbol{\phi}_k) P(\boldsymbol{\phi}_k | \boldsymbol{\beta}) d\boldsymbol{\phi}_k \\
&\quad + \mu_{(d,i)} \int P(c_{\boldsymbol{w}'_{(d,i)}} = c_{\boldsymbol{t}'} | \boldsymbol{\pi}_{\boldsymbol{w}'_{(d,i)},k}) P(\boldsymbol{\pi}_{\boldsymbol{w}'_{(d,i)},k} | \boldsymbol{\gamma}) d\boldsymbol{\pi}_{\boldsymbol{w}'_{(d,i)},k} \Bigg]
\end{aligned}
$$

Based on the definition of Dirichlet distribution, conditional distribution $P_{(d,i),k}$ can be simplified as:

$$
P_{(d,i),k} \propto E_{\alpha}(\theta_{d,k}) \Big[(1 - \mu_{(d,i)}) E_{\beta}(\phi_{k,t}) + \mu_{(d,i)} E_{\gamma}(\pi_{\boldsymbol{w}'_{(d,i)},k}) \Big]
$$

According to the definition of the expectation of Dirichlet Distribution, we obtain conditional probability $P_{(d,i),k}$ as below:

$$P_{(d,i),k} \propto (n_{d,k,-(d,i)} + \alpha_k)\Bigg[\mu_{(d,i)}\frac{1}{L}\frac{\sum_{t\in \boldsymbol{w}'}(n^t_{k,-(d,i)} + \gamma_t)}{\sum_{f=1}^{V}(n^f_{k,-(d,i)} + \gamma_f)} + (1-\mu_{(d,i)})\frac{n^t_{k,-(d,i)} + \beta_t}{\sum_{f=1}^{V}(n^f_{k,-(d,i)} + \beta_f)}\Bigg].$$

APPENDIX B
PARAMETER SENSITIVITY

For investigating the sensitivity on L, we tested the classification performances in 20NG, T-COM, and Tweet with different values of L with K=200, fixing other parameters as previous settings. The results are shown in Figure 5. We see the F1-Score increases sharply as L increases and tends to saturate when L reaches around 10. As L continues to increase, the F-Score starts to gradually decline. Although the best value of L varies with the dataset, setting L=10, in general, can well reflect the semantic context of a word.

REFERENCES

[1] L. M. Manevitz and M. Yousef, "One-Class SVMs for Document Classification," *JMLR*, vol. 2, no. Dec, pp. 139–154, 2001.

[2] Y. Liu, "On Document Representation and Term Weights in Text Classification," in *Handbook of Research on Text and Web Mining Technologies*. IGI Global, 2009, pp. 1–22.

[3] E. F. Kelly and P. J. Stone, "Computer recognition of english word senses," vol. 13, 1975.

[4] B. K. Britton, "Lexical Ambiguity of Words Used in English Text," *Behavior research methods & Instrumentation*, vol. 10, no. 1, pp. 1–7, 1978.

[5] R. Navigli, "Word Sense Disambiguation: A Survey," *ACM computing surveys*, vol. 41, no. 2, pp. 1–69, 2009.

[6] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[7] N. Meghanathan, S. Boumerdassi, N. Chaki, and D. Nagamalai, "Recent Trends in Networks and Communications," in *International Conferences, NeCoM 2010, WiMoN 2010, WeST 2010*, vol. 90. Springer, 2010.

[8] J. Boyd-Graber, D. Blei, and X. Zhu, "A Topic Model for Word Sense Disambiguation," in *Proc. EMNLP-CoNLL*, 2007, pp. 1024–1033.

[9] D. S. Chaplot and R. Salakhutdinov, "Knowledge-based Word Sense Disambiguation Using Topic Models," in *Proc. AAAI*, 2018.

[10] A. Ferrugento, H. G. Oliveira, A. Alves, and F. Rodrigues, "Can Topic Modelling benefit from Word Sense Information?" in *Proc. LREC*, 2016, pp. 3387–3393.

[11] A. Huang, D. Milne, E. Frank, and I. H. Witten, "Clustering Documents Using A Wikipedia-based Concept Representation," in *Proc. PAKDD*. Springer, 2009, pp. 628–636.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[13] A. Kilgarriff, "I Don't Believe in Word Senses," *Computers and the Humanities*, vol. 31, no. 2, pp. 91–113, 1997.

[14] ——, "Dictionary Word Sense Distinctions: An Enquiry into Their Nature," *Computers and the Humanities*, vol. 26, no. 5-6, pp. 365–387, 1992.

[15] J. Reisinger and R. Mooney, "A Mixture Model with Sharing for Lexical Semantics," in *Proc. EMNLP*. ACL, 2010, pp. 1173–1182.

[16] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving Word Representations via Global Context and Multiple Word Prototypes," in *Proc. ACL*. ACL, 2012, pp. 873–882.

[17] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient Non-Parametric Estimation of Multiple Embeddings Per Word in Vector Space," *arXiv preprint arXiv:1504.06654*, 2015.

[18] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical Word Embeddings," in *Proc. AAAI*, 2015.

[19] B. Shi, W. Lam, S. Jameel, S. Schockaert, and K. P. Lai, "Jointly Learning Word Embeddings and Latent Topics," in *Proc. SIGIR*, 2017, pp. 375–384.

[20] L. L. Newman, "Faith, spirituality, and religion: A model for understanding the differences." *College Student Affairs Journal*, vol. 23, no. 2, pp. 102–110, 2004.

[21] K. Tudor, "Religion, faith, spirituality, and the beyond in transactional analysis," *Transactional Analysis Journal*, vol. 49, no. 2, pp. 71–87, 2019.

[22] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," in *Proc. ICML*, 2014, pp. 1188–1196.

[23] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *JMLR*, vol. 3, no. Jan, pp. 993–1022, 2003.

[24] Y. Goldberg and O. Levy, "Word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Rmbedding Method," *arXiv preprint arXiv:1402.3722*, 2014.

[25] J. Boyd-Graber, D. Blei, and X. Zhu, "A Topic Model for Word Sense Disambiguation," in *Proc. EMNLP-CoNLL*, 2007, pp. 1024–1033.

[26] L. Li, B. Roth, and C. Sporleder, "Topic Nodels for Word Sense Disambiguation and Token-Based Idiom Detection," in *Proc. ACL*. ACL, 2010, pp. 1138–1147.

[27] B. Skaggs and L. Getoor, "Topic modeling for wikipedia link disambiguation," *ACM TOIS*, vol. 32, no. 3, pp. 1–24, 2014.

[28] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books," in *Proc. ICCV*, 2015, pp. 19–27.

[29] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep Contextualized Word Representations," *arXiv preprint arXiv:1802.05365*, 2018.

[30] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-training," 2018.

[31] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient Knowledge Distillation for Bert Model Compression," *arXiv preprint arXiv:1908.09355*, 2019.

[32] G. Xun, Y. Li, W. X. Zhao, J. Gao, and A. Zhang, "A Correlated Topic Model Using Word Embeddings," in *IJCAI*, 2017, pp. 4207–4213.

[33] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson, "Improving topic models with latent feature word representations," *arXiv preprint arXiv:1810.06306*, 2018.

[34] L. Niu, X. Dai, J. Zhang, and J. Chen, "Topic2Vec: Learning Distributed Representations of Topics," in *Proc. IALP*. IEEE, 2015, pp. 193–196.

[35] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans.PAMI*, vol. 6, no. 6, pp. 721–741, 2009.

[36] J. Sivic and A. Zisserman, "Efficient Visual Search of Videos Cast as Text Retrieval," *IEEE Trans.PAMI*, vol. 31, no. 4, pp. 591–606, 2008.

[37] Z. S. Harris, "Distributional Structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[38] H. Jeffreys, *Scientific Inference*. Read Books Ltd, 2013.

[39] C. M. Carlo, "Markov Chain Monte Carlo and Gibbs Sampling," *Lecture notes for EEB*, vol. 581, 2004.

[40] L. v. d. Maaten and G. Hinton, "Visualizing Data using t-SNE," *JMLR*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[41] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[42] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[43] C. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval," *Natural Language Engineering*, vol. 16, no. 1, pp. 100–103, 2010.

[44] M. Röder, A. Both, and A. Hinneburg, "Exploring the Space of Topic Coherence Measures," in *Proc. WSDM*. ACM, 2015, pp. 399–408.

[45] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum, "Optimizing Semantic Coherence in Topic Models," in *Proc. EMNLP*. ACL, 2011, pp. 262–272.

[46] M. Purver, "Topic Segmentation," *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pp. 291–317, 2011.

[47] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An Introduction to Variational Methods for Graphical Models," *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.