Hierarchical Concept-Driven Language Model

YASHEN WANG and HUANHUAN ZHANG, China Academy of Electronics and Information Technology of CETC
ZHIRUN LIU and QIANG ZHOU, Beijing Institute of Technology

For guiding natural language generation, many semantic-driven methods have been proposed. While clearly improving the performance of the end-to-end training task, these existing semantic-driven methods still have clear limitations: for example, (i) they only utilize shallow semantic signals (e.g., from topic models) with only a single stochastic hidden layer in their data generation process, which suffer easily from noise (especially adapted for short-text etc.) and lack of interpretation; (ii) they ignore the sentence order and document context, as they treat each document as a bag of sentences, and fail to capture the long-distance dependencies and global semantic meaning of a document. To overcome these problems, we propose a novel semantic-driven language modeling framework, which is a method to learn a Hierarchical Language Model and a Recurrent Conceptualization-enhanced Gamma Belief Network, simultaneously. For scalable inference, we develop the auto-encoding Variational Recurrent Inference, allowing efficient end-to-end training and simultaneously capturing global semantics from a text corpus. Especially, this article introduces concept information derived from high-quality lexical knowledge graph Probase, which leverages strong interpretability and anti-nose capability for the proposed model. Moreover, the proposed model captures not only intra-sentence word dependencies, but also temporal transitions between sentences and inter-sentence concept dependence. Experiments conducted on several NLP tasks validate the superiority of the proposed approach, which could effectively infer meaningful hierarchical concept structure of document and hierarchical multi-scale structures of sequences, even compared with latest state-of-the-art Transformer-based models.

CCS Concepts: • **Computing methodologies → Lexical semantics**; **Semantic networks**;

Additional Key Words and Phrases: Language modeling, text generation, concept semantic information, interpretation, recurrent conceptualization-enhanced gamma belief network, hierarchical language modeling, representation learning

This work is funded by: (i) the National Natural Science Foundation of China (No. U19B2026); (ii) the New Generation of Artificial Intelligence Special Action Project (No. AI20191125008); (iii) the National Key Research and Development Project (No. 2017YFC0820503); (iv) the National Integrated Big Data Center Pilot Project (No. 20500908, 17111001, and 17111002); and (v) the Joint Advanced Research Foundation of China Electronics Technology Group Corporation (CETC) (No. 6141B08010102).

Authors' addresses: Y. Wang (corresponding author) and H. Zhang, National Engineering Laboratory for Risk Perception and Prevention (RPP), China Academy of Electronics and Information Technology of CETC, Shuangyuan Street 11, Shijingshan District, Beijing, China, 100041; emails: yswang@bit.edu.cn, huanhuanz_bit@126.com; Z. Liu and Q. Zhou, Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications, School of Computer, Beijing Institute of Technology, Zhongguancun South Street 5, Haidian District, Beijing, China, 100081; emails: {zrliu, qzhou}@bit.edu.cn.

1556-4681/2021/05-ART104 $15.00
https://doi.org/10.1145/3451167

ACM Reference format:
Yashen Wang, Huanhuan Zhang, Zhirun Liu, and Qiang Zhou. 2021. Hierarchical Concept-Driven Language Model. *ACM Trans. Knowl. Discov. Data* 15, 6, Article 104 (May 2021), 22 pages.
https://doi.org/10.1145/3451167

1 INTRODUCTION

Traditional **language models (LMs)** only utilize *un-structured* information embedded in plain text, without extra semantic information beyond the given text. Semantic-driven LM, enhances traditional LMs, by combining text's un-structured information and extra *structured* semantic information (e.g., concept information from lexical **knowledge graph (KG)** such as Probase) or the *distilled* high-level semantic information (e.g., topics derived from topic models). With help of these kinds of extra semantic information, prior knowledge (including common sense and language-using knowledge etc.) has been successfully introduced for modeling language. Recent years have witnessed great advance of semantic-driven LM, which generates semantically meaningful latent representation for multi-granularity text. LMs have become key components of various **Natural Language Processing (NLP)** tasks, such as text summarization [16], information retrieval [73], speech recognition [19], machine translation [64], multi-granularity text embeddings [74], and caption generation [53]. The primary purpose of an LM is to capture the distribution of a word sequence, commonly with a **Recurrent Neural Network (RNN)** [45, 47] or a Transformer-based neural network [12, 66]. In this article, we focus on improving **RNN-based language models (RNN-LMs)** that often have much fewer parameters and are easier to perform end-to-end training [45, 46], with help of extra semantic information.

Multiple extra semantic information is widely used for enhancing LM [38, 43, 63, 74]. E.g., topic models, such as **Latent Dirichlet Allocation (LDA)** [3, 25] and its non-parametric Bayesian generalizations [50, 65], are well suited to extract document-level word concurrence patterns into latent topics from a text corpus. Their modeling power has been further enhanced by introducing multi-layer deep representation [86, 91]. While having semantically meaningful latent representation, they typically treat each document as a **Bag-of-Words (BOW)**, *ignoring* word order [20]. Moreover, topic models suffer easily from noise (especially adapted for short-text etc.) and lack of interpretation. Especially, traditional topic model quickly boosts the general and vague topics co-occurred with all the observed words, and dismisses the topics partially matching the words, which would be more specific and descriptive for representing the context. Hence, this article introduces *concept* information derived from high-quality lexical KG Probase [80], which has strong interpretability and anti-nose capability [28, 73], for boosting LM (especially adopted in short-text context). Because concept information has been proved to be effective and robust in helping understanding semantic in many NLP tasks [73, 75, 80, 84] and could be combined with the distributional knowledge.

Figure 1 sketches the architecture of the proposed hierarchical concept-driven LM. Overall, In this article, we propose to use **Recurrent Conceptualization-enhanced Gamma Belief Network (RCGB**, in blue part) to guide a stacked RNN for a **Hierarchical Language Modeling (HLM**, in green part). We refer to the model as RCGB+HLM, which integrates RCGB (i.e., a deep recurrent conceptualization model) and a HLM (i.e., stacked RNN [8, 19, 34]), into a novel larger-context RNN-LM, as depicted in Figure 3. It simultaneously learns: (i) a deep recurrent conceptualization model, extracting *document-level* multi-layer word concurrence patterns and sequential concept vectors for sentences; and (ii) an expressive LM, capturing both *short-distance* and *distance-range* word sequential dependencies. For inference, we equip RCGB+HLM (decoder) with a novel **Variational Recurrent Inference (VRI)** network (encoder), and train it end-to-end by maximizing the **evidence lower bound (ELBO)**. Different from the stacked RNN-LM

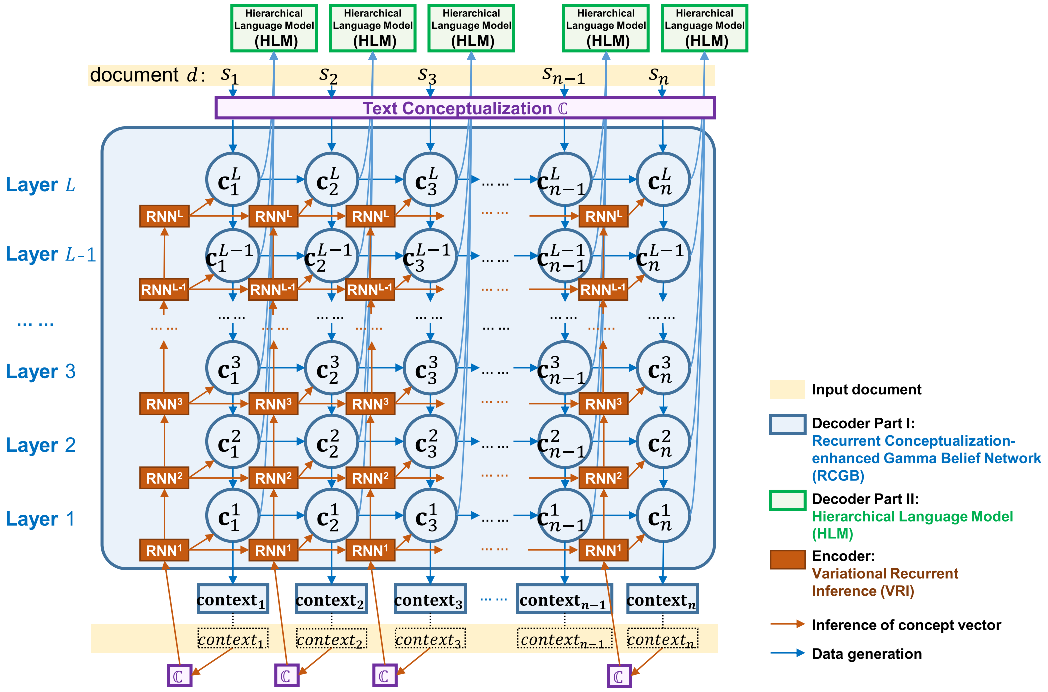
Hierarchical Language Model (HLM)
document d: s_1 s_2 s_3 s_{n-1} s_n
Text Conceptualization $\mathbb{C}$
Layer L
Layer L-1
... ...
Layer 3
Layer 2
Layer 1
$\mathbf{c}_1^L$ $\mathbf{c}_2^L$ $\mathbf{c}_3^L$ $\mathbf{c}_{n-1}^L$ $\mathbf{c}_n^L$
$\mathbf{c}_1^{L-1}$ $\mathbf{c}_2^{L-1}$ $\mathbf{c}_3^{L-1}$ $\mathbf{c}_{n-1}^{L-1}$ $\mathbf{c}_n^{L-1}$
$\mathbf{c}_1^3$ $\mathbf{c}_2^3$ $\mathbf{c}_3^3$ $\mathbf{c}_{n-1}^3$ $\mathbf{c}_n^3$
$\mathbf{c}_1^2$ $\mathbf{c}_2^2$ $\mathbf{c}_3^2$ $\mathbf{c}_{n-1}^2$ $\mathbf{c}_n^2$
$\mathbf{c}_1^1$ $\mathbf{c}_2^1$ $\mathbf{c}_3^1$ $\mathbf{c}_{n-1}^1$ $\mathbf{c}_n^1$
RNN^L
RNN^{L-1}
RNN^3
RNN^2
RNN^1
$\mathbf{context}_1$ $\mathbf{context}_2$ $\mathbf{context}_3$ $\mathbf{context}_{n-1}$ $\mathbf{context}_n$
$context_1$ $context_2$ $context_3$ $context_{n-1}$ $context_n$
$\mathbb{C}$
Input document
Decoder Part I: Recurrent Conceptualization-enhanced Gamma Belief Network (RCGB)
Decoder Part II: Hierarchical Language Model (HLM)
Encoder: Variational Recurrent Inference (VRI)
Inference of concept vector
Data generation

Fig. 1. The overall architecture of the proposed model, including: (i) the decoder, consisting of RCGB (blue part, details in Section 4.3) and HLM (green part, details in Section 4.4); and (ii) the encoder, VRI (brown part, details in Section 4.5). Wherein, the brown arrows denote the inference of latent concept vectors, blue ones indicate the data generation.

proposed in [8], which relies heavily on three types of customized training operations (i.e., UPDATE, COPY, and FLUSH) to extract multi-scale structures, the LM in the proposed RCGB+HLM here learns such structures purely under the guidance of the temporally and hierarchically connected stochastic layers of RCGB. The effectiveness of RCGB+HLM as a new larger-context LM is demonstrated quantitatively in many NLP tasks.

The contributions of the proposed model, could be concluded as follows:

(i) We leverage concept information derived from high-quality lexical KG Probase, which has strong interpretability and anti-nose capability, for proposing a novel semantic-driven LM.

(ii) We propose a novel RCGB for capturing both *global* semantics across documents and long-distance *inter-sentence* dependencies within a document.

(iii) We propose a novel HLM, which could learn the *local* syntactic and lexical relationships between the words within a sentence, and characterize the "word-sentence-document" hierarchy to incorporate both *intra-* and *inter-*sentence semantics, which enhances the interactions among the semantic signals from: word-level, sentence-level, and document-level.

2 RELATED WORK

RNN-based Language Model: Neural LMs have recently achieved remarkable advances [44]. The RNN-LM is superior for its ability to model longer-distance temporal dependencies without imposing a strong conditional independence assumption [71]. Beside, it has recently been shown to outperform carefully-tuned traditional *n*-gram based LMs [29]. Unfortunately, the traditional RNN-LMs often assume that the sentences of a document are *independent* to each other. This simplifies the modeling task to independently assigning probabilities to individual sentences, *ignoring* their orders and document context [70]. Therefore, such LMs may consequently *fail to* capture the long-distance dependencies and global semantic meaning of a document [13].

Semantic-Driven Language Model: An RNN-LM or neural LM could be further improved by utilizing the extra semantic signals derived from broad document context [38, 43, 63, 74]. Such models typically extract latent semantic categories (e.g., topics) via a topic analysis model, and then send the topic vector to a LM for sentence generation. Important work in this direction includes [13, 21, 65, 86, 91]. The key differences of these methods is in either the topic model itself or the method of integrating the topic signals into the LM. Concerning the method of incorporating the topic vector into the LM, [47] extended the RNN cell with additional topic features. [13] used a hybrid model combining the predicted word distribution given by both a topic model and a standard RNN-LM. [38] leveraged topical information to enhance word embeddings with help of large-context. To relax the sentence independence assumption in language modeling, [70] proposed larger-context LMs that model the context of a sentence by representing its preceding sentences as either a single or a sequence of BOW vectors, which were then fed directly into the sentence modeling RNN. An alternative approach attracting significant recent interest was leveraging topic models to improve RNN-LMs. [47] used pre-trained topic model features as an additional input to the RNN hidden states and/or output.

While clearly improving the performance of the end task, these existing semantic-driven methods still have clear limitations. For example, they *only* utilize *shallow* semantic signals (e.g., from topic models) with only a single stochastic hidden layer in their data generation process. Note that, several neural topic models use deep neural networks to construct their variational encoders, but still use *shallow* generative models (decoders) [62, 90]. Another key limitation lies in *ignoring* the sentence order, as they treat each document as a bag of sentences. Thus once the semantic vector learned from the document context is given, the task is often reduced to independently assigning probabilities to individual sentences [72]. Moreover, conventional latent topic modeling suffers easily from *noise* (especially adapted for short-text context etc.) and lack of *interpretation*. Especially, traditional topic model quickly boosts the *general* and *vague* topics co-occurred with all the observed words, and dismisses the topics partially matching the words, which would be more specific and descriptive for representing the context. Hence, this article explore to provides solution according these problems mentioned above.

Concept-Enhanced Methodology for NLP: Psychologist Gregory Murphy began his highly acclaimed book [48] with the statement "*Concepts* are the glue that holds our mental world together." Still, Nature magazine book review calls it an understatement, because "Without *concepts*, there would be no mental world in the first place" [4]. Doubtless to say, the ability to conceptualize is a defining characteristic of humanity. [28, 67] The idea of using learned concepts [82] to improve NLP tasks has been explored previously, including text embedding [74], text conceptualization [58], information retrieval [73], entity disambiguation [5], query understanding [60], semantic conceptualization [52], text segmentation [27], KG completion [75], and so on. Previous work has shown concept's strong interpretability and anti-nose capability, which is effective and robust in helping understanding semantic and could be combined with the distributional knowledge. Distinct from them, we propose the use of concepts to guide the prior of a Gamma Belief Network and a HLM. This provides more flexibility in text modeling and also the ability to infer the posterior on latent codes, which could be useful for visualization and downstream NLP tasks.

3 MOTIVATION

3.1 Leveraging "Word-Sentence-Document" Hierarchy Interaction

The traditional RNN-LMs often assume that the sentences of a document are *independent* to each other, wherein a document is still treated as a bag of sentences. This simplifies the modeling task to independently assigning probabilities to individual sentences, *ignoring* their orders and document context. Therefore, such LMs may consequently *fail to* capture the long-distance dependencies and global semantic meaning of a document. *To overcome this problem*, we propose a novel RCGB, which extracts document-level multi-layer word concurrence patterns and sequential concept vectors for sentences, for capturing both *global inter-document* semantics across documents and long-distance *inter-sentence* dependencies within a document. Moreover, we utilize the HLM, which captures both short-distance and distance-range word sequential dependencies, to learn the *local* syntactic and lexical relationships between the words within a sentence. Overall, the goal of characterizing the "word-sentence-document" hierarchy to incorporate both *intra-* and *inter-*sentence/document is achieved.

3.2 Leveraging "Concept" Semantics

Previous semantic-driven LM models typically extract *shallow* latent semantic categories (e.g., topics) via a topic analysis model, and then send the topic vector to a LM for sentence generation, *without* exploiting the deep semantic information from extra (lexical or encyclopedic) knowledge source and hence failing to understand the language. However, conventional latent topic modeling suffers easily from *noise* (especially adapted for short-text context etc.) and lack of *interpretation.* Moreover, traditional topic model quickly boosts the general and vague topics co-occurred with all the observed words, and dismisses the topics partially matching the words, which would be more specific and descriptive for representing the context. *To overcome this problem*, we leverage *concept* semantic information, derived from high-quality lexical KG Probase, for proposing a novel semantic-driven LM. Previous research has demonstrated concept's strong interpretability and anti-nose capability for boosting language modeling tasks (especially adopted in short-text context) [28, 58, 73].

4 METHODOLOGY

4.1 Overview

Denote a document of n sentences as $d = (s_1, s_2, \ldots, s_n)$, wherein $s_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,|s_i|})$ consists of $|s_i|$ words from a vocabulary V (of size $|V|$). Notation $w_{i,j}$ indicates the j-th word in sentence s_i. Conventional statistical LMs often only focus on the word sequence within a sentence. Assuming that the sentences of a document are *independent* to each other, they often define $\mathcal{P}(d) \approx \prod_{i=1}^{n} \mathcal{P}(s_i) = \prod_{i=1}^{n} \prod_{j=2}^{|s_j|} \mathcal{P}(w_{i,j}|w_{j,<j})\mathcal{P}(w_{i,1})$. RNN-based neural LMs define the conditional probability of each word $w_{i,j}$ given all the previous words $\{w_{i,<j}\}$ within the sentence s_i, through the softmax function of a hidden state $\mathbf{h}_{i,j}$, as follows:

$$\mathbf{h}_{i,j} = \sigma(\mathbf{h}_{i,<j}, w_{i,j-1}) \tag{1}$$

$$\mathcal{P}(w_{i,j}|w_{j,<j}) = \mathcal{P}(w_{i,j}|\mathbf{h}_{i,j}). \tag{2}$$

Wherein $\sigma(\cdot)$ indicates a non-linear function typically defined as an RNN cell, such as **Long Short-Term Memory (LSTM)** [24] and **Gated Recurrent Unit (GRU)** [7]. These RNN-based statistical LMs are typically applied *only* at the word level, *without* exploiting the document context, and hence often fail to capture *long-distance* dependencies. What is more, these RNN-based statistical LMs are typically applied *only* at the word-surface level, without exploiting the semantic information from extra (lexical or encyclopedic) knowledge source, and hence often *fail to* understand the language. While [13, 33, 71] remedied the issue by guiding the LM with a topic model, they still treated a document as a bag of sentences, *ignoring* the order of sentences, and lacked the ability to extract hierarchical and recurrent semantic structures.

We introduce another widely-used semantic information here—*concept*, which has been demonstrated to enable to help better understand text semantic, and extend the texts with high-level semantic categorical information [14, 26, 27, 58, 77], beneficial for text many NLP tasks (e.g., context embeddings, short-text clustering, and microblog retrieval)[28, 30, 60, 67, 73–75]. Prior work on concept enhanced contextualized embedding (for word [28], entity [75] and sentence [74] etc.) has shown that steering distributional models towards capturing robust semantic representation has a positive impact on natural language understanding. Moreover, we propose RCGB+HLM, as depicted in Figure 3, as a novel larger-context neural LM. It consists of two key components: (i) a hierarchical recurrent conceptualization model (RCGB, shown in blue part in Figure 3); and (ii) a stacked RNN-based HLM (shown in green part in Figure 3). We leverage RCGB for capturing both *global* semantics across documents and long-distance *inter-sentence* dependencies within a document d, and utilize the HLM to learn the *local* syntactic and lexical relationships between the words within a sentence. We represent a document d as a sequence of sentence-context pairs as $\{\{s_1, context_1\}, \ldots, \{s_n, context_n\}\}$, where $context_i$ summarizes the document excluding sentence s_i, specifically $context_i = \{s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n\}$. Denote embedded vector $context_i \in \mathbb{R}^{|V|}$ (shown at the bottom of Figures 1 and 3) as the "context vector" respect to $context_i$, with $|V|$ as the size of the vocabulary excluding stop-words. Generation details of $context_i$ will be described in latter Section 4.3. Note a naive way is to treat each sentence as a document, use a co-ranking based text conceptualization model [28] to capture the temporal dependencies of the latent concept vectors, which is fed to the RNN to model the word sequence of the corresponding sentence. In our setting, as $context_i$ summarizes the document-level context of sentence s_i, it is in general sufficiently long for conceptualization modeling. Note that, during testing, we redefine $context_i$ as the context vector summarizing only the preceding sentences, i.e., $s_{1:i-1}$, which will be further clarified when presenting experimental results.

4.2 Preliminary

To enhance the representation ability of the proposed model, this article introduces extra lexical knowledge (i.e., *concept* knowledge from Probase [58, 81] emphasized here), which has been proved to be effective in helping understanding semantic in many NLP tasks [73, 81, 84].

4.2.1 Definition.

(Def.1) **Concept.** Following [28, 67], we define a "concept" as a set or class/category of "entities" or "things" within a domain, such that words belonging to similar classes get similar representations. E.g., "microsoft" and "amazon" could be represented by concept COMPANY. Probase [82] is used in our study as KG.

(Def.2) **Text Conceptualization.** Given a text $s_i = \{w_1, w_2, \ldots, w_{|s_i|}\}$, wherein w_i denotes a word, text conceptualization algorithm enables to select the open-domain concepts $C_{s_i} = \{< c_i, p_i > | i = 1, \ldots, \}$ from the KG Probase which own the optimal ability for discriminatively representing the given text s_i. E.g., given a text as input (e.g., "microsoft unveils office for apple's ipad"), we generate the concepts C_{s_i} = {<COMPANY,0.8567>, <BRAND,0.7457>, <PRODUCT,0.5471> , ...} from Probase for this text context. Besides, the concept vector θ_i is generated based on c_i and its corresponding probability p_i: each dimensionality of θ_i represents the probability p_i of the concept c_i in the given text. Note that, concept plays an important role in the proposed model, and concept vector θ_i mentioned above is utilized in the following Section 4.3 and Section 4.5. In this

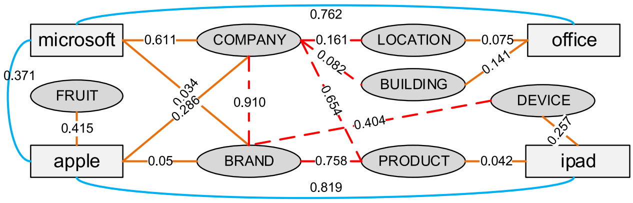

Fig. 2. The example sketch of the lexical KG Probase [82]. Rectangles indicate instances, and ellipses indicate concept defined in Probase. Orange solid links indicate isA relationship between instances and concepts. Red dashed lines indicate correlation relationship among between two concepts, and blue solid lines indicate correlation relationship among between two instances, respectively. Numerical values on the line is corresponding probabilities (i.e., scores).

article, we utilize the state-of-the-art text conceptualization algorithm proposed in [28][1], which co-ranks the concepts and words simultaneously in an iterative procedure, and this algorithm respects to the notation $\mathbb{C}$ in Figure 1 and Figure 3 (illustrated as purple letters).

4.2.2 Probase. Probase[2] is widely used in research about text understanding [59, 61, 80], text representation [28, 74], information retrieval [73], and KG completion [75]. Probase uses an automatic and iterative procedure to extract concept knowledge from 1.68 billion Web pages. It contains 2.36 millions of open-domain terms, and each term is a concept, an instance (respect to a word occurring in given text in this study), or both. Meanwhile, it provides around 14 millions relationships with two kinds of important knowledge related to concepts: concept-attribute co-occurrence (isAttrbuteOf) and concept-instance co-occurrence (isA). For clarity, Figure 2 sketches the organization of instances and their corresponding concepts defined in Probase [74]. Moreover, Probase provides huge number of high-quality and robust concepts without builds. Therefore, lexical KG Probase is utilized in this article for leveraging lexical semantics for boosting efficiency of language modeling, with help of its strong interpretability and anti-nose capability.

4.3 Decoder Part I: RCGB

Shown in Figure 3, to model the time-varying sentence-context context vectors $context_i$ in document d, the generative process of the RCGB component, from the top to bottom hidden layers, is expressed as follows:

$$\begin{aligned} \mathbf{c}_i^L &\sim \text{Gam}(\theta_i + \mathbf{M}^L \mathbf{c}_{i-1}^L, \mu), \ldots, \\ \mathbf{c}_i^l &\sim \text{Gam}(\mathbf{U}^{l+1}\mathbf{c}_i^{l+1} + \mathbf{M}^l \mathbf{c}_{i-1}^l, \mu), \ldots, \\ \mathbf{c}_i^1 &\sim \text{Gam}(\mathbf{U}^2\mathbf{c}_i^2 + \mathbf{M}^1 \mathbf{c}_{i-1}^1, \mu), \mathbf{context}_i \sim \text{Pois}(\mathbf{U}^1\mathbf{c}_i^1) \end{aligned} \tag{3}$$

Wherein, for layer l: (i) $\mathbf{c}_i^l \in \mathbb{R}^{|C|_l}$ denotes the gamma distributed concept vectors of sentence s_i at layer l; (ii) $\mathbf{M}^l \in \mathbb{R}^{|C|_l \times |C|_l}$ indicates the transition matrix of layer l that captures cross-concept temporal dependencies; and (iii) $\mathbf{U}^l \in \mathbb{R}^{|C|_{l-1} \times |C|_l}$ indicates the loading matrix at layer l. Where, $|C|_l$ represents the number of concepts of layer l, which affects the model's performance and will be discussed in experimental section latter. μ is a scaling hyper-parameter. Especially, when $i = 1$

[1] Note that, although many text conceptualization algorithms could be adopted here, we choose the state-of-the-art one [28], because it is not central to this study.
[2] https://concept.research.microsoft.com/.

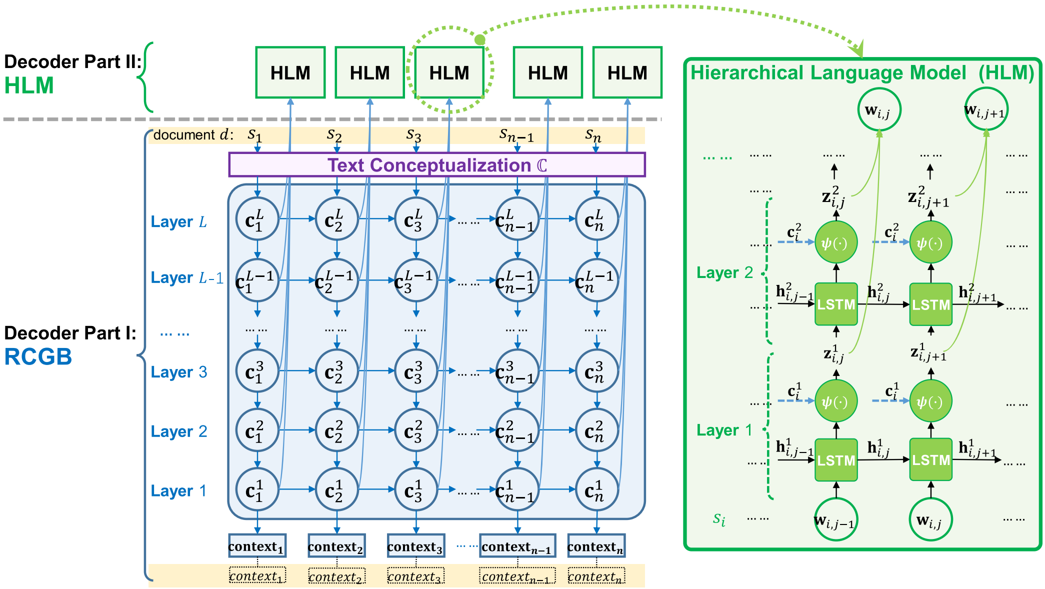
Decoder Part II:
HLM
HLM
HLM
HLM
HLM
HLM
Hierarchical Language Model (HLM)
document d: s_1 s_2 s_3 s_{n-1} s_n
Text Conceptualization $\mathbb{C}$
Layer L
Layer L-1
… …
Layer 3
Layer 2
Layer 1
Decoder Part I:
RCGB
$\mathbf{c}_1^L$ $\mathbf{c}_2^L$ $\mathbf{c}_3^L$ $\mathbf{c}_{n-1}^L$ $\mathbf{c}_n^L$
$\mathbf{c}_1^{L-1}$ $\mathbf{c}_2^{L-1}$ $\mathbf{c}_3^{L-1}$ $\mathbf{c}_{n-1}^{L-1}$ $\mathbf{c}_n^{L-1}$
$\mathbf{c}_1^3$ $\mathbf{c}_2^3$ $\mathbf{c}_3^3$ $\mathbf{c}_{n-1}^3$ $\mathbf{c}_n^3$
$\mathbf{c}_1^2$ $\mathbf{c}_2^2$ $\mathbf{c}_3^2$ $\mathbf{c}_{n-1}^2$ $\mathbf{c}_n^2$
$\mathbf{c}_1^1$ $\mathbf{c}_2^1$ $\mathbf{c}_3^1$ $\mathbf{c}_{n-1}^1$ $\mathbf{c}_n^1$
$\mathbf{context}_1$ $\mathbf{context}_2$ $\mathbf{context}_3$ $\mathbf{context}_{n-1}$ $\mathbf{context}_n$
$context_1$ $context_2$ $context_3$ $context_{n-1}$ $context_n$
$\mathbf{w}_{i,j}$ $\mathbf{w}_{i,j+1}$
$\mathbf{z}_{i,j}^2$ $\mathbf{z}_{i,j+1}^2$
$\mathbf{c}_i^2$ $\psi(\cdot)$ $\mathbf{c}_i^2$ $\psi(\cdot)$
Layer 2
$\mathbf{h}_{i,j-1}^2$ LSTM $\mathbf{h}_{i,j}^2$ LSTM $\mathbf{h}_{i,j+1}^2$
$\mathbf{z}_{i,j}^1$ $\mathbf{z}_{i,j+1}^1$
$\mathbf{c}_i^1$ $\psi(\cdot)$ $\mathbf{c}_i^1$ $\psi(\cdot)$
Layer 1
$\mathbf{h}_{i,j-1}^1$ LSTM $\mathbf{h}_{i,j}^1$ LSTM $\mathbf{h}_{i,j+1}^1$
s_i $\mathbf{w}_{i,j-1}$ $\mathbf{w}_{i,j}$

Fig. 3. The decoder architecture of the proposed RCGB+HLM: (i) RCGB, shown in blue part; and (ii) HLM, shown in green part. Wherein, (i) left side: the generative model of a L-hidden-layer RCGB+HLM, where the bottom (blue) part is the RCGB (details in Section 4.3), document contexts of consecutive sentences (i.e., $\{context_1, \ldots, context_n\}$) are used as observed data, and upper (green) part is the HLM (details in Section 4.4). (ii) Right side: overview of the HLM component, wherein input $\mathbf{w}_{i,j}$ denotes the embedded vector of j-th word in i-th sentence of a document, $\mathbf{h}_{i,j}^l$ indicates the hidden state of the stacked RNN at time step j and layer l, and $\mathbf{c}_i^l$ represents the concept vector of i-th sentence s_i at layer l. Besides, notation $\mathbb{C}$ (in purple letters) represents the text conceptualization algorithm [28], which could help to generate the concept vector $\mathbf{c}_i^L$ for each sentence s_i.

(i.e., 1st sentence occurred in current document), $\mathbf{c}_1^l \sim \text{Gam}(\mathbf{U}^{l+1}\mathbf{c}_1^{l+1}, \mu)$ for $l \in [1, L-1]$ and the concept vector in last layer $\mathbf{c}_1^L \sim \text{Gam}(\theta_1, \mu)$. Wherein, $\theta_i \in \mathbb{R}^{|C|_L} (i \in [1, n])$ is computed by introducing a co-ranking text conceptualization algorithm [28], as discussed in former Section 4.2.1.

Finally, Dirichlet priors [50, 83] could be placed on the columns of $\mathbf{M}^l$ and $\mathbf{U}^l$, which not only makes the latent representation more identifiable and interpretable, but also facilitates inference. In this condition: (i) the context vector $\mathbf{context}_i$ (shown at the bottom of Figure 3) could be factorized into the $\mathbf{M}^1$ and $\mathbf{U}^1$ under the Poisson likelihood (shown as Equation (3)); (ii) the parameters of $\mathbf{c}_i^l \in \mathbb{R}^{|C|_l}$ could be factorized into the sum of $\mathbf{U}^{l+1}\mathbf{c}_i^{l+1}$ (capturing *inter*-layer hierarchical dependence), and $\mathbf{M}^l\mathbf{c}_{i-1}^l$ of previous sentence s_{i-1} (capturing *intra*-layer temporal dependence). With efforts above, RCGB not only captures the *document-level* word occurrence patterns inside the training text corpus, but also the sequential dependencies of the sentences *inside* a document.

Note that: (i) the proposed RCGB will reduce to the Gamma Belief Network [55, 87], if the recurrent structure is ignored; and (ii) the proposed RCGB will reduce to Poisson-Gamma dynamical systems [55], when hierarchical structure is ignored (i.e., $L = 1$) [21]. We refer to the the proposed RCGB+HLM without its **r**ecurrent structure as CGB+HLM, which no longer models sequential sentence dependencies, and the comparison among them will be provided in the ablation experiment in the experimental Section 5 later.

4.4 Decoder Part II: HLM

Generally, a conventional RNN-LM [45, 46] predicts the next word *only* using the preceding words within the current sentence. In this article, we integrate the hierarchical recurrent concept vectors $\mathbf{c}_i^l$ into the LM to predict the word sequence in the i-th sentence s_i (as shown in Figure 3). Especially, the proposed LM is constructed upon the stacked RNN [8, 19, 40]; however, with the help of the proposed RCGB discussed above in Section 4.3, it no longer requires specialized training heuristics to extract multi-scale structures. As shown in the right side of Figure 3, to generate $w_{i,j}$, the j-th word of i-th sentence s_i in a document, we construct the hidden states $\mathbf{h}_{i,j}^l$ of the LM, from the bottom to top layers, as follows:

$$\mathbf{h}_{i,j}^l = \begin{cases} \text{LSTM}^l(\mathbf{h}_{i,j-1}^l, \mathbf{w}_{i,j-1}), & \text{if} \quad l = 1 \\ \text{LSTM}^l(\mathbf{h}_{i,j-1}^l, \mathbf{z}_{i,j}^{l-1}), & \text{if} \quad 1 < l \leq L \end{cases} \tag{4}$$

Wherein $\text{LSTM}^l(\cdot,\cdot)$ denotes the word-level LSTM at layer l, and $\mathbf{w}_{i,j} \in \mathbb{R}^k$ indicates the word embeddings respect to word $\mathbf{w}_{i,j}$. Following [21], we introduce a gating unit $\psi(\cdot)$ similar to GRU [7] to compute $\mathbf{z}_{i,j}^l$, as follows: $\mathbf{z}_{i,j}^l = \psi(\mathbf{h}_{i,j}^l, \mathbf{c}_i^l)$. Therefore, $\mathbf{z}_{i,j}^l$ could combine $\mathbf{c}_i^l$ of i-th sentence s_i with its hidden state $\mathbf{h}_{i,j}^l$ of word-level LSTM at layer l and time j. Denote $\mathbf{W}^\star$ as a weight matrix with $|V|$ rows and $\mathbf{z}_{i,j}^{1:L}$ as the concatenation of $\mathbf{z}_{i,j}^l$ across all layers. Different from Equation (2), the conditional probability of $w_{i,j}$ could be reformed as follows:

$$\mathcal{P}(w_{i,j}|w_{j,<j}, \mathbf{c}_i^l) = \text{softmax}\left(\mathbf{W}^\star \cdot \mathbf{z}_{i,j}^{1:L}\right). \tag{5}$$

We argue that, there are two main reasons for combining all the latent representations $\mathbf{z}_{i,j}^{1:L}$ for the proposed Hierarchical Language Modeling: (i) first of all, the latent representations exhibit different statistical properties at different stochastic layers of the proposed RCGB+HLM, and hence are combined together to enhance their representation power; and (ii) secondly, having skip-connections from all hidden layers to the output one, makes it easier to train the proposed entire network, reducing the number of processing steps between the bottom of the network and the top and hence mitigating the "vanishing gradient" problem [19, 21].

Form the overall architecture mentioned above, we could conclude that, concept information from lexical KG (embedded as as θ_i and concept vector $\mathbf{c}_i^l$ here) plays an extremely vital role here, as depicted in Figure 3: (i) the concept vector $\mathbf{c}_i^l$ of i-th sentence quantifies the concept usage of its document context $context_i$ at layer l; moreover (ii) it is further used as an additional feature of the the proposed HLM to guide the word generation inside i-th sentence, as shown in Figure 3. It is clear that the proposed RCGB+HLM has two temporal structures: (i) a deep recurrent conceptualization model to extract the temporal concept vectors $\mathbf{c}_i^l$ from the sequential document contexts (details in Section 4.3); and (ii) a novel HLM to estimate the probability of each sentence given its corresponding hierarchical concept vector $\mathbf{c}_i^l$ (details in this section). Characterizing the "word-sentence-document" hierarchy to incorporate both *intra*- and *inter*-sentence semantics, the proposed RCGB+HLM could learn more *coherent* and *interpretable* concepts, and increases the generative power of the proposed HLM. Distinct from existing semantic-driven LMs, the temporally related hierarchical concepts of RCGB+HLM exhibit different statistical properties across layers, which better guides LM to improve its language generation ability.

4.5 Encoder: VRI

For the proposed RCGB+HLM (Section 4.3), given $\{\mathbf{U}^l, \mathbf{M}^l\}_{l=1}^L$, the likelihood of the sequence of sentence-context pairs $\{\{s_1, \mathbf{context}_1\}, \ldots, \{s_n, \mathbf{context}_n\}\}$ of document d is defined as follows:

$$\begin{aligned}\mathcal{P}(d|\{\mathbf{U}^l, \mathbf{M}^l\}_{l=1}^L) = \int \prod_{i=1}^{n} \Bigg\{ \mathcal{P}(\mathbf{context}_i|\mathbf{U}^1\mathbf{c}_i^1) \left[\prod_{j=1}^{|s_i|} \mathcal{P}(w_{i,j}|w_{i,<j}, \mathbf{c}_i^{1:L}) \right] \\ \left[\prod_{l=1}^{L} \mathcal{P}(\mathbf{c}_i^l|\mathbf{U}^{l+1}\mathbf{c}_i^{l+1} + \mathbf{M}^l\mathbf{c}_{i-1}^l, \mu) \right] \Bigg\} d\mathbf{c}_{1:n}^{1:L}.\end{aligned} \tag{6}$$

The inference task is to learn the parameters of both the RCGB (in Section 4.3) and HLM (in Section 4.4) components. One intuitive solution is to alternate the training between these two components in each iteration: (i) firstly, the RCGB model is trained using a co-ranking based iterative conceptualization algorithm provided in [28] and a sampling-based strategy provided in [22]; (ii) secondly, the HLM is trained with maximum likelihood estimation under a standard cross-entropy loss. However, we argue that this solution faces with the following problems: (i) While this naive solution could utilize readily available inference algorithms for both RCGB and the HLM, it may suffer from *stability* and *convergence* issues; (ii) moreover, the need to perform a sampling based iterative algorithm for RCGB inside each iteration limits the *scalability* of the model for both training and inferencing.

To this end, we introduce a VRI network (encoder) to learn the latent temporal concept vectors $\mathbf{c}_{1:n}^{1:L}$. Given $Q = \prod_{i=1}^n \prod_{l=1}^L \phi(\mathbf{c}_i^l|\mathbf{context}_{\leq i})$, the ELBO of the log margin likelihood shown in Equation (6) can be constructed as follows:

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{n} \mathbb{E}_Q \left[\ln \mathcal{P}(\mathbf{context}_i|\mathbf{U}^1\mathbf{c}_i^1) + \sum_{j=1}^{|s_i|} \ln \mathcal{P}(w_{i,j}|w_{i,<j}, \mathbf{c}_i^{1:L} \right] \\ - \sum_{i=1}^{n} \sum_{l=1}^{L} \mathbb{E}_Q \left[\ln \frac{\phi(\mathbf{c}_i^l|\mathbf{context}_{\leq i})}{\mathcal{P}(\mathbf{c}_i^l|\mathbf{U}^{l+1}\mathbf{c}_i^{l+1} + \mathbf{M}^l\mathbf{c}_{i-1}^l, \mu)} \right],\end{aligned} \tag{7}$$

which unites both the terms which are primarily responsible for training the RCGB component (Section 4.3), and terms for training the HLM component (Section 4.4). Similar to [85], we define a random sample $\phi(\mathbf{c}_i^l|\mathbf{context}_{\leq i}) = \text{Weibull}(\kappa_i^l, \lambda_i^l)$, which can be obtained by transforming standard uniform variables ϵ_i^l, as follows:

$$\mathbf{c}_i^l = \lambda_i^l[1 - \ln(1 - \epsilon_i^l)]^{1/\kappa_i^l}. \tag{8}$$

In the sentence-level RNN (denoted as $\text{RNN}^l(\cdot)$ here) of RCGB (described in Section 4.3), its hidden state $\mathbf{h}_i^l$ could be defined as follows:

$$\mathbf{h}_i^l = \text{RNN}^l\left(\mathbf{h}_{i-1}^l, \mathbf{h}_i^{l-1}\right). \tag{9}$$

Especially, $\mathbf{h}_i^0$ indicates the conceptualization results for context of sentence s_i (i.e., $context_i$ corresponding to other sentences except for s_i occurred in the current document), by using text conceptualization algorithm [28] described above in Section 4.3. We could use an intuitive way to achieve this goal, as follows: for each $s^\star \in \{s_1, \ldots, s_{i-1}, s_{i+1} \ldots, s_n\}$, we implement co-ranking based text conceptualization algorithm [28] to generate its corresponding $\theta_{s^\star}$, and then $\mathbf{h}_i^0 = \sum_{s^\star \neq s_i} \theta_{s^\star}/(n-1)$. Besides, $\mathbf{h}_0^l = \mathbf{0}$. $\text{RNN}^l(\cdot)$ indicates the sentence-level recurrent encoder at l-th layer implemented with a basic RNN cell, capturing the sequential relationship between sentences within a document. To capture the temporal dependencies between the concept vectors, both κ_i^l and λ_i^l from the bottom to top layers can be expressed as follows:

$$\kappa_i^l = \sigma_\kappa^l(\mathbf{h}_i^l), \tag{10}$$

$$\lambda_i^l = \sigma_\lambda^l(\mathbf{h}_i^l). \tag{11}$$

Both function $\sigma_\lambda^l(\cdot)$ in Equation (11) and function $\sigma_\kappa^l(\cdot)$ in Equation (10), are nonlinear functions mapping hidden state $\mathbf{h}_i^l$ with $\sigma(\mathbf{x}) = \ln(1 + \exp(\mathbf{W}_\sigma \cdot \mathbf{x} + \mathbf{b}_\sigma))$, wherein $\{\mathbf{W}_\sigma, \mathbf{b}_\sigma\}$ is the set of parameters to be trained.

Rather than finding a point estimate of the global parameters $\{\mathbf{U}^l, \mathbf{M}^l\}_{l=1}^L$ of the proposed RCGB+HLM, we adopt a hybrid inference algorithm by combining TLASGR-MCMC [9, 85] and our proposed VRI network. That is to say: (i) the global parameters $\{\mathbf{U}^l, \mathbf{M}^l\}_{l=1}^L$ can be sampled with TLASGR-MCMC, while (ii) the parameters of the proposed HLM and VRI network, denoted by Θ, can be updated via Stochastic Gradient Descent [49] by maximizing the ELBO in Equation (7). We describe a hybrid variational/sampling inference for the proposed RCGB+HLM in Algorithm 1.

ALGORITHM 1: Hybrid TLASGR-MCMC [9] and Variational Recurrent Inference (VRI) for the Proposed RCGB+HLM

```
Require: mini-batch size $m$ and the number of layer $L$
Initialization:
Encoder and hierarchical language model parameter parameter $\Theta$, and RCGB's parameter $\{\mathbf{U}^l, \mathbf{M}^l\}_{l=1}^L$.
Procedure:
for iter=1,2,$\cdots$ do
    Randomly select a mini-batch of $m$ documents consisting of $n$ sentences to form a subset $\mathbf{D} = \{s_{o,1:n}, \mathbf{context}_{o,1:n}\}_{o=1}^m$;
    Draw random variables $\{\epsilon_{o,i}^l\}_{o=1,i=1,l=1}^{m,n,L}$ from uniform distribution;
    Calculate $\nabla_\Theta \mathcal{L}(\Theta, \mathbf{U}^l, \mathbf{W}^l; \mathbf{D}, \epsilon_{o,i}^l)$ according to Equation (7), and update $\Theta$;
    Sample $\mathbf{c}_{o,i}^l$ from Equation (8), Equation (10) and Equation (11) via $\Theta$ to update $\{\mathbf{M}^l\}_{l=1}^L$ and $\{\mathbf{U}^l\}_{l=1}^L$.
end for
```

To sum up, as shown in the brown part (from bottom to top) in Figure 1, the proposed RCGB+HLM works with a VRI framework, which takes the document context of the i-th sentence s_i within a document as input and learns hierarchical concept vectors $\mathbf{c}_i^{1:L}$ that evolve sequentially with i. The learned concept vectors $\mathbf{c}_i^{1:L}$ in different layer are then used to reconstruct the document context input $context_i$, and as an additional feature for the proposed HLM to generate the i-th sentence.

5 EXPERIMENTS

We evaluate our RCGB+HLM on text generation task, text summarization task and table-to-text generation task, and interpret its improvements quantitatively.

5.1 Experiments on Text Generation

5.1.1 Datasets. We present experimental results on three publicly available corpora: APNEWS, IMDB, and BNC. APNEWS[3] is a collection of Associated Press news articles from 2009 to 2016. IMDB

[3] https://www.ap.org/en-gb/.

Table 1. Summary Statistics for the Datasets Used in the Text Generation Experiments

	Training			Development			Testing		
	#Docs	#Sents	#Tokens	#Docs	#Sents	#Tokens	#Docs	#Sents	#Tokens
APNEWS	50K	0.7M	15M	2K	27.4K	0.6M	2K	26.3K	0.6M
IMDB	75K	0.9M	20M	12.5K	0.2K	0.3M	12.5K	0.2K	0.3M
BNC	15K	0.8M	18M	1K	44K	1M	1K	52K	0.6M

is a set of movie reviews collected by [41], and BNC is the written portion of the British National Corpus, which contains excerpts from journals, books, letters, essays, memoranda, news and other types of text. These three datasets can be downloaded from GitHub.

We follow the preprocessing steps in [33]. Specifically, words and sentences are tokenized using Stanford *CoreNLP* [42]. We lowercase all word tokens, and filter out word tokens that occur less than 10 times. We additionally remove stopwords in the documents and exclude the top 0.1% most frequent words and also words that appear in less than 100 documents. All these datasets are divided into training, development and testing sets. A summary statistic of these datasets is provided in Table 1.

5.1.2 Baselines. In order to demonstrate the advantage of the proposed model, we compare RCGB+HLM with the following baselines:

(i) **LSTM:** A standard LSTM LM [24].
(ii) **LC+LM:** A larger-context LM that incorporates context from preceding sentences, which are treated as a bag of words [70].
(iii) **LDA+LSTM:** A standard LSTM LM incorporating the topic information of a separately trained LDA.
(iv) **LDA+RNN** [13]: A hybrid model rescoring the prediction of the next word by incorporating the topic information through a linear transformation.
(v) **LDA+jLM:** A joint learning framework which learns a convolutional based topic model and a LM simultaneously [33].
(vi) **LDA+cLM:** A model which extracts the global semantic coherence of a document via a neural topic model, with the probability of each learned latent topic further adopted to build a mixture-of-experts LM [71].
(vii) **LDA+VAE:** A model which combines a variational auto-encoder based neural sequence model with a neural topic model [72].
(viii) **BERT+VAE:** [36] proposes a Transformer-based architecture and augment the decoder with an LSTM LM layer to fully exploit information of latent variables, compared to the previous variational auto-encoder for natural text.

Besides, for the proposed RCGB+HLM, we also carry out the ablation experiment: we denote CGB+HLM as a variant of the proposed RCGB+HLM, which is a simplified RCGB+HLM that removes the recurrent structure of its RCGB component (Section 4.3).

5.1.3 Settings. We consider a VRI network for RCGB+HLM to infer $\mathbf{c}_i^l$, as shown in Figure 1, whose number of hidden units in Equation (9) are set the same as the number of concepts in the corresponding layer. Following [33], word embeddings are pre-trained 300-dimension word2vec vectors. Dropout with a rate of 0.4 is used to the input of the stacked-RNN at each layer in order to alleviate over-fitting, i.e., $\mathbf{z}_{i,j}^l$ or $\mathbf{w}_{i,j-1}$ in Equation (4). The gradients are clipped if the norm of the parameter vector exceeds 5. We use the Adam optimizer [31] with learning rate 10^{-3}.

The length of an input sentence is fixed to 30. We set the mini-batch size m as 8, number of training epochs as 5, and scaling hyper-parameter μ as 1. In terms of the LSTM [24] part, we consider two settings: (i) a *small* 1-layer LSTM model with 600 hidden units, and (ii) a *large* 2-layer LSTM model with 900 hidden units in each layer. In addition, adaptive softmax is used to speed up the training process. All the hyper-parameters are tuned based on the performance on the development set. We empirically find that the optimal settings are fairly robust across the three datasets. For fair comparison, we use standard LM perplexity as the evaluation metric.

Besides, to improve computational efficiency, [35] adopt a K-Medoids clustering algorithm to group all the concepts defined in Probase into 5,000 disjoint concept clusters following, e.g., concept ANIMAL, concept WILD ANIMAL, concept JUNGLE ANIMAL are all mapped into a concept cluster ANIMAL. One concept cluster could represent one sense or a general topic, recognized with its center concept [74], and this strategy is widely recognized [67, 77]. Hence, similar to previous research, this study also does *not* carry out comparative algorithms on huge amount of individual concepts defined in Probase, but on the above-mentioned concept clusters. Because using these concept clusters as features, it not only covers the basic categories of all the concepts of Probase, but also avoids the high computational cost caused by using all the concepts of Probase, taking into account the semantic integrity and computational efficiency. Therefore, for our CGB+HLM and RCGB+HLM, we utilize these 5,000 concepts as the features in the following evaluation task. From the other perspective, as discussed in Section 4.3, $|C|_l$ indicates the number of concepts of layer l, and we test different choices of $|C|_l$: (i) 5,000 as the number of concept clusters mentioned above, and (ii) the number of all the concepts of Probase. The evaluation results show that, the difference of efficiency between these two choices is not very large, while we prefer to the former one because of its low complexity calculation.

Besides, the statistical t-test is employed here: To decide whether the improvement by model A over model B is significant, the t-test calculates a value p^* based on the performance of model A and model B. The smaller p^* is, the more significant the improvement is. If the p^* is small enough ($p^* < 0.05$), we conclude that the improvement is statistically significant.

5.1.4 Performance Summary. Experimental results are shown in Table 2, wherein notation "small" indicates LSTM model with 600 hidden units and "large" indicates LSTM model with 900 hidden units (as discussed in Section 5.1.3). As shown in Table 2, the proposed RCGB+HLM outperforms all baselines (including latest Transformer-based baselines such as BERT and BERT+VAE), and the trend of improvement continues as its number of layers increases, indicating the effectiveness of assimilating recurrent hierarchical concept information. RCGB+HLM consistently outperforms CGB+HLM, suggesting the benefits of exploiting the sequential dependencies of the sentence-contexts for language modeling. Moreover, RCGB+HLM, with its hierarchical and temporal concept guidance, achieves better performance with fewer parameters than comparable RNN-based baselines. Note that for language modeling, there has been significant recent interest in replacing RNNs with Transformer [66], which consists of stacked multi-head self-attention modules, and its variants [10, 12]. While Transformer-based LMs have been shown to be powerful in various NLP tasks, they often have significantly more parameters (although their lite variants have been proposed such as [32]), require much more training data, and take much longer to train than RNN-LMs. For example, Transformer-XL with 12L and that with 24L [10], which improve Transformer to capture longer-range dependencies, have 41M and 277M parameters, respectively, while the proposed RCGB+HLM with three stochastic hidden layers has as few as 12.2M parameters, when used for language modeling. From the perspective of performance, experimental results show that the generated texts by our approach are more meaningful and the semantics are more coherent, compared with state-of-the-art Transformer-based model. Even the ablative variant of the

Table 2. Test Perplexities of Different Models on Dataset APNEWS, Dataset IMDB, and Dataset BNC in Text Generation Task

Model	LSTM Size	APNEWS	IMDB	BNC
LSTM [24]	small	64.13	72.14	102.89
	large	58.89	66.47	94.23
LC+LM [70]	small	54.18	67.78	96.50
	large	50.63	67.86	87.77
LDA+LSTM [33]	small	55.52	69.64	96.50
	large	50.75	63.04	87.77
LDA+RNN [13]	small	54.54	67.83	93.57
	large	50.24	61.59	84.62
LDA+jLM [33]	small	52.75	63.45	85.99
	large	48.97	59.04	81.83
LDA+cLM [71]	small	52.63	62.64	86.44
	large	47.81	56.38	80.14
LDA+VAE [72]	large	48.73	57.11	102.89
BERT [12]		46.72	52.21	82.60
BERT+VAE [36]	small	46.28	55.68	81.82
	large	42.65	$51.29^{\dagger}$	79.02
CGB+HLM (Ours)	small	45.10	54.22	82.16
	large	$42.45^{\ddagger}$	52.70	$78.89^{\ddagger}$
RCGB+HLM (Ours)	small	44.08	53.03	77.92
	large	$\mathbf{40.62}^{\dagger\ddagger}$	$\mathbf{48.84}^{\dagger\ddagger}$	$\mathbf{75.25}^{\dagger\ddagger}$

The superscript † and ‡, respectively, denote statistically significant improvements over state-of-the-art BERT+VAE [36] and CGB+HLM ($p^* < 0.05$).

proposed model, i.e., CGB+HLM runs neck-and-neck with BERT+VAE. Besides, from a structural point-of-view, we consider the proposed RCGB+HLM as complementary to rather than competing with Transformer-based LMs, and consider replacing RNN with Transformer to construct RCGB guided Transformer as a promising future extension.

5.2 Experiments on Text Summarization

5.2.1 Datasets. We further test the proposed model for text summarization task on two popular datasets, following [72]. Firstly, we follow the same setup as in [54] and consider the GIGAWORDS corpus,[4] which consists of 3.8M training pair samples, 190K validation samples, and 1.9K test samples for evaluation. An input-summary pair consists of the first sentence and the headline of the source articles. Secondly, we also evaluate various models on the DUC-2004 dataset,[5] which has 500 news articles. Different from dataset GIGAWORDS, each article in dataset DUC-2004 is paired with four expert-generated reference summaries. The length of each summary is limited to 75 bytes.

5.2.2 Baselines. In order to demonstrate the advantage of the proposed model, we compare RCGB+HLM with the following baselines:

(i) **Seq2Seq:** The conventional Seq2Seq attention model [1].
(ii) **LDA+VAE:** A model which combines a variational auto-encoder based neural sequence model with a neural topic model [72].

[4] https://catalog.ldc.upenn.edu/ldc2012t21.
[5] http://duc.nist.gov/duc2004.

Table 3. Test Experimental Results of Different Models on Dataset GIGAWORDS and Dataset DUC-2004 in Text Summarization Task

Models	GIGAWORDS		DUC-2004	
	RF-1	RF-2	RR-1	RR-2
Seq2Seq [1]	34.03	15.93	28.39	9.26
LDA+VAE [72]	35.63	17.27	29.65	9.55
VAE [72]	34.22	16.10	28.78	9.11
BERT [12]	36.83	18.05	67.69	9.98
BERT+cMLM [6]	37.57	18.59†	69.05	10.28†
CGB+HLM (Ours)	38.77‡	18.48	71.25‡	10.22
RCGB+HLM (Ours)	**39.69**†‡	**19.54**†‡	**77.54**†‡	**10.81**†‡

The superscript † and ‡, respectively, denote statistically significant improvements over state-of-the-art BERT+cMLM [6] and CGB+HLM ($p^* < 0.05$).

(iii) **VAE:** A model similar to **LDA+VAE**, but without the usage of the topic dependent prior.

(iv) **BERT+cMLM:** [6] proposes a novel **conditional Masked Language Model (cMLM)** to enable the finetuning of BERT on target generation task.

5.2.3 *Settings.* We have a similar data tokenization as we have in the former text generation. Additionally, for the vocabulary, we count the frequency of words in both the source article the target summary, and maintain the top 30,000 tokens as the source article and target summary vocabulary.

5.2.4 *Performance Summary.* We evaluate the performance of the comparative models with the ROUGE score, which counts the number of overlapping content between the generated summaries and the reference summaries, e.g., overlapped n-grams. Following previous research, this article utilizes F-measures of ROUGE-1 (RF-1) and ROUGE-2 (RF-2) for dataset GIGAWORDS and Recall measures of ROUGE-1 (RR-1) and ROUGE-2 (RR-2) for dataset DUC-2004.

The results in Section 5.2.4 show that our RCGB+HLM achieves better performance than a variety of strong baseline methods on both dataset GIGAWORDS and dataset DUC-2004, demonstrating the practical value of our model. As discussed in [72], it is worthwhile to note that recently several much more complex CNN/RNN architectures have been proposed for abstract text summarization, such as [68][88]. In this work, we focus on a relatively simple RNN architecture for fair comparison. In such a way, we are able to conclude that the improvements on the results are mainly from our concept-guided text generation strategy. Our method shows improvement over Transformer-based models (BERT and BERT+cMLM), which obviously outstrip conventional methods, on all the metrics as shown in Table 3. Our best model outperforms these state-of-the-art models that use much more complex architectures specifically designed for summarization. As can be seen, though the VAE model achieves comparable performance with the standard Seq2Seq model, the usage of extra semantic information (i.e., topic and concept) for more flexible posterior inference boosts the performance, such as LDA+VAE and our RCGB+HLM. Additionally, compared with LDA+VAE and BERT+cMLM, we yield further performance improvements, with help of its deep architecture making the inter-sentence dependency signals fully interplayed, as well as concept's strong interpretability and anti-nose capability, demonstrating the importance of semantic guidance for text summarization.

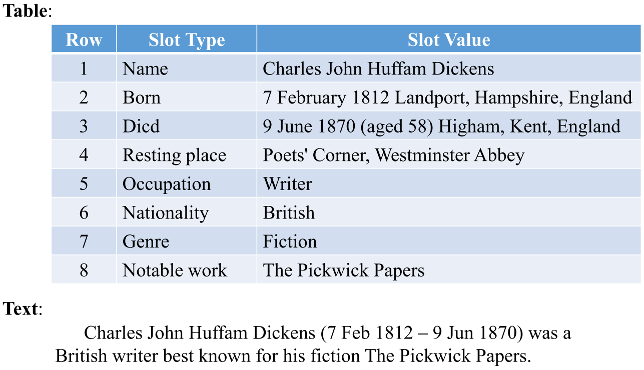

Table:

Row	Slot Type	Slot Value
1	Name	Charles John Huffam Dickens
2	Born	7 February 1812 Landport, Hampshire, England
3	Dicd	9 June 1870 (aged 58) Higham, Kent, England
4	Resting place	Poets' Corner, Westminster Abbey
5	Occupation	Writer
6	Nationality	British
7	Genre	Fiction
8	Notable work	The Pickwick Papers

Text:

Charles John Huffam Dickens (7 Feb 1812 – 9 Jun 1870) was a British writer best known for his fiction The Pickwick Papers.

Fig. 4. Wikipedia infobox about Charles Dickens and its corresponding generated description.

5.3 Experiments on Table-to-Text Description

Table-to-text description task aims to generate a text description for a structured table (e.g., infobox in Wikipedia) which can be viewed as a series of slot-value records [37, 57]. As shown in Figure 4, for example, a biographic infobox is a fixed-format table that describes a person with many “slot-value” records like “(Name, Charles John Huffam Dickens)”. This task aims at filling in this knowledge gap by developing a model that can take a table (consisted of a set of slot types and their values) about an entity as input, and automatically generate a natural language description. We introduce dataset `WIKMul` to compare our model with several baselines, and after that, we assess the performance of our model on table-to-text description.

5.3.1 Dataset. Using “person” and “animal” entities as case studies, [69] creates a new dataset based on Wikipedia dump (2018/04/01) and Wikidata (2018/04/12) as follows: (i) extract Wikipedia pages and Wikidata tables about person and animal entities, and align them according to their unique KG IDs. (ii) For each Wikidata table, filter out the slot types of which frequency is less than three. For each Wikipedia article, use its anchor links (clickable texts in hyperlinks) to locate all the entities and determine their KG IDs. (iii) For each Wikidata table, search each value (including number, date) and entity contained in the table in the corresponding Wikipedia article according to its KG ID, and remove the values and entities which cannot be found in the corresponding Wikipedia article. (iv) For each Wikipedia article, remove the sentences which contain no values, and remove sentences which only contain entities that do not exist in the Wikidata table. The remaining sentences will be taken as ground-truth reference descriptions. (v) Index the row numbers for each slot type according to their orders in the Wikidata table. The ground-truth structured table is then created. (vi) Build a fixed vocabulary for the whole corpus of ground-truth descriptions and label the words with frequency less than five as OOV. [69] further randomly shuffles and splits the dataset into training set (80%), validation set (10%) and testing set (10%) for person and animal entities, respectively.

Table 4 shows the detailed statistics. This dataset contains multiple sentences to cover as many facts as possible in the input structured KG. Therefore, this dataset constructed is denoted as `WIKMul`.

5.3.2 Baselines. In early years, researchers apply LM and neural networks to generate texts from structured data, where a neural encoder captures table-formed information and, a RNN decodes these information to a natural language sentence [37, 57, 76]. We compare the proposed

Table 4. Summary Statistics for the Dataset WIKMul Used in the Table-to-text Description Experiment

Entity type	# entity	# types before filtering	# types after filtering	# slots per sent.	# tokens per sent.	# slots per table	# tokens per entity	# sent. per entity
Person	100,000	109	76	1.9	16.8	8.0	70.9	4.2
Animal	6,216	30	12	1.3	17.1	3.2	42.2	2.5

model with several statistical LMs, other competitive sequence-to-sequence models and state-of-the-art Transformer-based models. The baselines are listed as follows:

(i) **KN:** The **Kneser-Ney (KN)** model is a widely used LM proposed by [23]. We use the *KenLM* toolkit to train 5-gram models without pruning.
(ii) **NLM:** A naive statistical neural LM proposed by [34], which uses only the slot value as input without slot type information, position information, and link information.
(iii) **Pointer:** Pointer-generator [56] which introduces a soft switch to choose between generating a word from the fixed vocabulary and copying a word from the input sequence. Here, we concatenate all slot values as the input sequence.
(iv) **Seq2Seq:** The Seq2Seq attention model [1]. We concatenate slot types and values as a sequence, and apply the sequence to sequence with attention model to generate a description.
(v) **Vanilla Seq2Seq:** The vanilla seq2seq neural architecture is also provided as a state-of-the-art baseline which uses the concatenation of word embedding, slot embedding and position embedding as the model input [37]. The model can operate local addressing over the infobox by the natural advantages of LSTM units and word level attention mechanism.
(vi) **BERT+DA:** [18] presents a Transformer-based table-to-text generation model, which learns content selection and summary generation in an end-to-end fashion, with help of data augmentation techniques.
(v) **BERT+SELF:** [89] proposes a new self-attention mechanism to incorporate richer structural information, on the basis of the Transformer [12].

5.3.3 Performance Summary. We apply the standard BLEU [51], METEOR [11], and ROUGE [15] metrics to evaluate the table-to-text generation performance, because they can measure the content overlap between model output and ground-truth, and also check whether the system output is written in sufficiently good English.

Table 5 shows the performance of various models with standard metrics. We can see that our model achieve consistent improvement. We conduct paired *t*-test between our proposed model and all the other baselines on 10 randomly sampled subsets. The differences are statistically significant with $p^* \leq 0.016$ for all settings. From the experimental results, we could conclude that, neural network models perform much better than statistical LMs, and even Vanilla Seq2Seq architecture with word level attention outperform the most competitive statistical model by a great margin. Besides, modern neural networks are considerably better than traditional KN models with or without templates. The proposed RCGB+HLM architecture can further improve the text generation compared with the competitive Vanilla Seq2Seq and Pointer. From the experimental results, we also find that, the our model outperforms current state-of-the-art Transformer-based models (BERT+DA and BERT+SELF) on all metrics. Even the ablated variant CGB+HLM is comparable with the latest Transformer-based work for table-to-text generation task.

Generally, we could also consider natural language as the most expressive way for knowledge transmission via a noisy channel. If we are able to reconstruct the input table from the generated

Table 5. Test Experimental Results of Different Models on Dataset WIKMul in Table-to-text Description Task (%)

Models	Person			Animal		
	BLEU	METEOR	ROUGE	BLEU	METEOR	ROUGE
KN [23]	2.2	6.4	7.5	3.1	5.2	6.3
NLM [34]	6.3	10.2	17.3	4.3	8.2	14.9
Seq2seq [1]	11.1	16.3	27.9	5.5	10.9	20.2
Vanilla Seq2seq [37]	17.2	19.9	38.1	5.2	12.9	35.2
Pointer [56]	16.6	19.7	36.8	5.9	13.1	37.7
BERT+SELF [89]	22.4	22.3	40.2	15.2	17.0	43.3
BERT+DA [18]	22.9	22.7	41.0	15.5	17.4	44.2
CGB+HLM (Ours)	23.1	20.8	41.3	**16.3**	**18.3**	43.7
RCGB+HLM (Ours)	**24.1**	**23.9**	**43.2**	15.9	18.2	**46.5**

Table 6. Overall Slot Filling Precision (P), Recall (R), and F-score (F-1) (%)

Models	Person			Animal		
	P	R	F-1	P	R	F-1
Seq2seq [1]	73.5	29.2	41.9	82.3	26.1	39.4
Pointer [56]	71.9	56.0	62.2	57.8	37.2	44.9
CGB+HLM (Ours)	76.1	59.3	66.4	66.3	63.5	65.3
RCGB+HLM (Ours)	**76.2**	**64.4**	**69.2**	**72.6**	**71.9**	**72.3**

description, our generator achieves a 100% success rate at knowledge propagation. Hence, [69] propose a Table Reconstruction based metric as follows: for each entity, construct a table from the generated paragraph, and compute precision, recall and F-score by comparing it with the input KG from two aspects: (i) *Overall Slot Filling*: if a pair of slot type and its slot value exists in both of the reconstructed table and the input table, it is considered as a correct slot. (ii) *Inter-Dependent Slot Filling*: if a row that consists one or multiple slot types and their slot values exist in both of the reconstructed table and the input table, it is considered as a correct row. If the same slot/row is correctly described multiple times in the system generation output, it is only counted as correct once, i.e., redundant descriptions will be penalized. It is similar to the relation extraction based generation evaluation metric proposed by [78] and entity (or event) extraction based metric proposed by [39]. They compared automatic Information Extraction results from the reference description and the system generation output. However, the performance of state-of-the-art open-domain slot filling [2, 79] is still far from satisfactory to serve as an automatic extraction tool for evaluating generation results. Therefore for the pilot study in this article we manually reconstruct tables from the generation output for evaluation. Notably none of the above automatic metrics is sufficient to capture adequacy, grammaticality and fluency of the generated descriptions. However extrinsic metrics such as system purpose and user task are expensive, while cheaper metrics such as human rating do not correlate with extrinsic metrics [17]. Moreover the task we address in this article requires essential domain knowledge for a human user to assess the generated descriptions. As shown in Table 6 and Table 7, the tables reconstructed from models with the proposed RCGB and HLM, achieve much higher quality.

Table 7. Inter-Dependent Slot Filling Precision (P), Recall (R), and F-score (F1) (%)

Models	Person			Animal		
	P	R	F-1	P	R	F-1
Seq2seq [1]	73.2	30.3	42.9	81.8	27.2	41.3
Pointer [56]	72.8	56.0	62.8	56.5	36.7	44.1
CGB+HLM (Ours)	76.1	59.3	67.3	66.2	65.3	64.4
RCGB+HLM (Ours)	**78.5**	**62.2**	**73.1**	**73.5**	**71.9**	**72.5**

6 CONCLUSIONS

We propose a novel semantic-driven language modeling framework driven by RCGB and HLM, which is a method to learn a LM and a deep recurrent conceptualization model simultaneously. For scalable inference, we develop a auto-encoding VRI methodology, allowing efficient end-to-end training. Experiments conducted on three NLP tasks validate the superiority of the proposed approach, which could effectively generate the sentences from the designated multi-level concepts or noise, while inferring more meaningful hierarchical concept structure of document and hierarchical multi-scale structures of sequences, even compared with the latest state-of-the-art Transformer-based models.

For future work, we plan to extend the proposed models to specific NLP tasks, such as machine translation and image paragraph captioning. Moreover, note that for language modeling, there has been significant recent interest in replacing RNNs with Transformer, which consists of stacked multi-head self-attention modules, and its variants. While Transformer-based LMs have been shown to be powerful in vious NLP tasks, they often have significantly more parameters (though their lite variants have been proposed), require much more training data, and take much longer to train. From a structural point-of-view, we consider the proposed RCGB+HLM model as *complementary* to rather than competing with Transformer-based LMs. Therefore, another promising extension is to replace the stacked-RNN in RCGB+HLM with Transformer, i.e., constructing an RCGB-*guided Transformer* as a new larger-context neural LM.

ACKNOWLEDGMENTS

We thank anonymous reviewers for valuable comments.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *Eprint Arxiv* (2014).

[2] Nikita Bhutani, H. V. Jagadish, and Dragomir R. Radev. 2016. Nested propositions in open information extraction. In *EMNLP*.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.

[4] Paul Bloom. 2003. Glue for the mental world. *Nature* 421 (2003), 212–213.

[5] Lihan Chen, Jiaqing Liang, Chenhao Xie, and Yanghua Xiao. 2018. Short text entity linking with fine-grained topics. In *CIKM*.

[6] Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jing jing Liu. 2020. Distilling knowledge learned in BERT for text generation. In *ACL*. 7893–7905.

[7] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*.

[8] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. 2016. Hierarchical multiscale recurrent neural networks. *ArXiv* abs/1609.01704 (2016).

[9] Yulai Cong, Bo Chen, Hongwei Liu, and Mingyuan Zhou. 2017. Deep latent Dirichlet allocation with topic-layer-adaptive stochastic gradient Riemannian MCMC. In *ICML*. 864–873.

[10] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL*.

[11] Michael J. Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *WMT@ACL*.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

[13] Adji B. Dieng, Chong Wang, Jianfeng Gao, and John W. Paisley. 2016. TopicRNN: A recurrent neural network with long-range semantic dependency. *ArXiv* abs/1611.01702 (2016).

[14] Ofer Egozi, Shaul Markovitch, and Evgeniy Gabrilovich. 2011. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems* 29, 2 (2011), 8.

[15] Carlos Flick. 2004. ROUGE: A package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out*.

[16] Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-up abstractive summarization. In *EMNLP*.

[17] Dimitra Gkatzia and Saad Mahamood. 2015. A snapshot of NLG evaluation practices 2005–2014. In *ENLG*.

[18] Li Gong, Josep Maria Crego, and Jean Senellart. 2019. Enhanced transformer model for data-to-text generation. In *NGT@EMNLP-IJCNLP*.

[19] Alex Graves. 2013. Generating sequences with recurrent neural networks. *ArXiv* abs/1308.0850 (2013).

[20] Thomas R. L. Griffiths and Mark Steyvers. 2004. Finding scientific topics.*Proceedings of the National Academy of Sciences of the United States of America* 101, Suppl 1 (2004), 5228–35.

[21] Dandan Guo, Bo Chen, Ruiying Lu, and Mingyuan Zhou. 2019. Recurrent hierarchical topic-guided neural language models. *ArXiv* abs/1912.10337 (2019).

[22] Dandan Guo, Bo Chen, Hao Zhang, and Mingyuan Zhou. 2018. Deep poisson gamma dynamical systems. In *NeurIPS*. 8451–8461.

[23] Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *ACL*. 690–696.

[24] Sepp Hochreiter and Jrgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[25] Matthew D. Hoffman, David M. Blei, and Francis R. Bach. 2010. Online learning for latent Dirichlet allocation. In *NIPS*.

[26] Wen Hua, Yangqiu Song, Haixun Wang, and Xiaofang Zhou. 2013. Identifying users' topical tasks in web search. In *WSDM*. 93–102.

[27] Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. 2015. Short text understanding through lexical-semantic analysis. In *IEEE ICDE*. 495–506.

[28] Heyan Huang, Yashen Wang, Chong Feng, Zhirun Liu, and Qiang Zhou. 2018. Leveraging conceptualization for short-text embedding. *IEEE Transactions on Knowledge and Data Engineering* 30, 7 (2018), 1282–1295.

[29] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *ArXiv* abs/1602.02410 (2016).

[30] Dongwoo Kim, Haixun Wang, and Alice Oh. 2013. Context-dependent conceptualization. In *IJCAI*. 2654–2661.

[31] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980 (2014).

[32] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. *ArXiv* abs/1909.11942 (2020).

[33] Jey Han Lau, Timothy Baldwin, and Trevor Cohn. 2017. Topically driven neural language model. In *ACL*.

[34] Remi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *EMNLP*.

[35] Peipei Li, Haixun Wang, Kenny Q. Zhu, Zhongyuan Wang, and Xindong Wu. 2013. Computing term similarity by large probabilistic isa knowledge. In *CIKM*. ACM, 1401–1410.

[36] Danyang Liu and Gongshen Liu. 2019. A transformer-based variational autoencoder for sentence generation. In *IJCNN*. 1–7.

[37] Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. *CoRR* abs/1711.09724 (2018).

[38] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings. In *AAAI*.

[39] Di Lu, Spencer Whitehead, Lifu Huang, Heng Ji, and Shih-Fu Chang. 2018. Entity-aware image caption generation. In *EMNLP*.

[40] Weixin Luo, Wen Liu, and Shenghua Gao. 2017. A revisit of sparse coding based anomaly detection in stacked RNN framework. In *ICCV*. 341–349.

[41] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL*.

[42] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL*.

[43] Donald Metzler and W. Bruce Croft. 2007. Latent concept expansion using Markov random fields. In *SIGIR*.

[44] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*. 1045–1048.

[45] Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Vcernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*.

[46] Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan vCernocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *ICASSP*. 5528–5531.

[47] Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *SLT*. 234–239.

[48] Gregory L. Murphy. 2002. *The Big Book of Concepts*. MIT Press

[49] Deanna Needell, Nathan Srebro, and Rachel Ward. 2016. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Mathematical Programming* 155, 1–2 (2016), 549–573.

[50] John W. Paisley, Chong Wang, David M. Blei, and Michael I. Jordan. 2015. Nested hierarchical Dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 256–270.

[51] Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *ACL*.

[52] Jin-woo Park, Seung-won Hwang, and Haixun Wang. 2016. Fine-grained semantic conceptualization of FrameNet. In *AAAI*. 2638–2644.

[53] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2016. Self-critical sequence training for image captioning. In *CVPR*. 1179–1195.

[54] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*.

[55] Aaron Schein, Hanna M. Wallach, and Mingyuan Zhou. 2016. Poisson-gamma dynamical systems. In *NIPS*.

[56] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.

[57] Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. 2018. Order-planning neural text generation from structured data. In *AAAI*. 5414–5421.

[58] Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. 2011. Short text conceptualization using a probabilistic knowledgebase. In *IJCAI*. 2330–2336.

[59] Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. 2011. Short text conceptualization using a probabilistic knowledgebase. In *IJCAI*. 2330–2336.

[60] Yangqiu Song, Shusen Wang, and Haixun Wang. 2015. Open domain short text conceptualization: A generative + descriptive modeling approach. In *IJCAI*. 3820–3826.

[61] Yangqiu Song, Shusen Wang, and Haixun Wang. 2015. Open domain short text conceptualization: A generative + descriptive modeling approach. In *ICAI*.

[62] Akash Srivastava and Charles A. Sutton. 2017. Autoencoding variational inference for topic models. In *ICLR*.

[63] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced representation through knowledge integration. *ArXiv* abs/1904.09223 (2019).

[64] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*. 3104–3112.

[65] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association* 101, 476 (2006), 1566–1581.

[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

[67] Fang Wang, Zhongyuan Wang, Zhoujun Li, and Ji Rong Wen. 2014. Concept-based short text classification and ranking. In *CIKM*. 1069–1078.

[68] Li Wang, Junlin Yao, Yunzhe Tao, Li Zhong, Wei Liu, and Qiang Du. 2018. A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. In *IJCAI*. 4453–4460.

[69] Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018. Describing a knowledge base. In *INLG*.

[70] Tian Wang and Kyunghyun Cho. 2016. Larger-context language modelling with recurrent neural network. In *ACL*.

[71] Wenlin Wang, Zhe Gan, Wenqi Wang, Dinghan Shen, Jiaji Huang, Wei Ping, Sanjeev Satheesh, and Lawrence Carin. 2017. Topic compositional neural language model. In *AISTATS*.

[72] Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-guided variational autoencoders for text generation. In *NAACL-HLT*.

[73] Yashen Wang, Heyan Huang, and Chong Feng. 2017. Query expansion based on a feedback concept model for microblog retrieval. In *WWW*. 559–568.

[74] Yashen Wang, Heyan Huang, Chong Feng, Qiang Zhou, Jiahui Gu, and Xiong Gao. 2016. CSE: Conceptual sentence embeddings based on attention model. In *ACL*. 505–515.

[75] Yashen Wang, Yifeng Liu, Huanhuan Zhang, and Haiyong Xie. 2019. Leveraging lexical semantic information for learning concept-based multiple embedding representations for knowledge graph completion. In *APWeb/WAIM*.

[76] Yashen Wang, Huanhuan Zhang, Yifeng Liu, and Haiyong Xie. 2019. KG-to-text generation with slot-attention and link-attention. In *NLPCC*.

[77] Zhongyuan Wang, Kejun Zhao, Haixun Wang, Xiaofeng Meng, and Ji Rong Wen. 2015. Query understanding through knowledge-based conceptualization. In *IJCAI*. 3264–3270.

[78] Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *EMNLP*.

[79] Fei Wu and Daniel S. Weld. 2010. Open information extraction using Wikipedia. In *ACL*.

[80] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*.

[81] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*. 481–492.

[82] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*. 481–492.

[83] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. 2007. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research* 8, 2 (2007), 35–63.

[84] Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2016. Understanding short texts through semantic enrichment and hashing. *IEEE Transactions on Knowledge and Data Engineering* 28, 2 (2016), 566–579.

[85] Hao Zhang, Bo Chen, Dandan Guo, and Mingyuan Zhou. 2018. WHAI: Weibull hybrid autoencoding inference for deep topic modeling. In *ICLR*.

[86] He Zhao, Lan Du, Wray L. Buntine, and Mingyuan Zhou. 2018. Dirichlet belief networks for topic structure learning. In *NeurIPS*.

[87] Mingyuan Zhou, Yulai Cong, and Bo Chen. 2016. Augmentable gamma belief networks. *Journal of Machine Learning Research* 17, 163 (2016), 163:1–163:44.

[88] Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *ACL*.

[89] Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. Modeling graph structure in transformer for better AMR-to-text generation. In *EMNLP/IJCNLP*.

[90] Qile Zhu, Zheng Feng, and Xiaolin Li. 2018. GraphBTM: Graph enhanced autoencoded variational inference for biterm topic model. In *EMNLP*.

[91] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *SIGIR*.

Received June 2020; revised December 2020; accepted February 2021