A Text Generation Model that Maintains the Order of Words, Topics, and Parts of Speech via Their Embedding Representations and Neural Language Models

Noriaki Kawamae
kawamae@gmail.com
NTT Comware
Tokyo, JAPAN

ABSTRACT

Our goal is to generate coherent text accurately in terms of their semantic information and syntactic structure. Embedding methods and neural language models are indispensable in generating coherent text as they learn semantic information, and syntactic structure, respectively, and they are indispensable methods for generating coherent text. We focus here on parts of speech (POS) (e.g. noun, verb, preposition, etc.) so as to enhance these models, and allow us to generate truly coherent text more efficiently than is possible by using any of them in isolation. This leads us to derive **W**ords and **T**opics and **P**OS 2 Vec (WTP2Vec) as an embedding method, and Structure Aware Unified Language Model (SAUL) as a neural language model. Experiments show that our approach enhances previous models and generates coherent and semantically valid text with natural syntactic structure.

KEYWORDS

neural language modeling, text generation, embedding method

ACM Reference Format:
Noriaki Kawamae. 2021. A Text Generation Model that Maintains the Order of Words, Topics, and Parts of Speech via Their Embedding Representations and Neural Language Models. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI-IAT '21), December 14–17, 2021, ESSENDON, VIC, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3486622.3493968

1 INTRODUCTION

Our goal is to generate coherent text accurately in terms of their semantic information and syntactic structure. As more and more people are expressing their opinions on-line and the volume of such contents exceeds the analysis capabilities of any individual or group of individuals, topic models have become popular tools for modeling the document generation process. They represent the co-occurrence relationships of words as topics and treat each document as a mixture of topics without any supervision under the 'bag-of-words' assumption.

As with global semantic information, topic models [2] and their extensions take a global statistical view and look at the word distributions of topics across a given corpus. As with local semantic information, distributed representations [1] take a local context view and so represent words and documents as low-dimensional real value vectors in a semantic space. While recurrent Neural Networks (RNNs) [28], such as Long Short Term Memory (LSTM) [10], have been shown to be effective at capturing long patterns in data and have been employed in topic models [5, 18], their representations are generally uninterpretable and inaccessible to humans [37].

Motivated by the above research, we propose **W**ords and **T**opics and **P**OS 2 Vec (WTP2Vec) as an embedding method, and Structure Aware Unified Language Model (SAUL) as a neural language model. In combination, they allow us to generate coherent text accurately in terms of semantic information and syntactic structure. Since embedding methods and neural language models learn semantic information and syntactic structure, respectively, and they are indispensable elements for text generation model, we focus on parts of speech (POS) (e.g. noun, verb, preposition, etc.) so as to unify and enhance these state-of-the-art models. As topics and embedding representations can complement each other in exploiting semantic information when used together, WTP2Vec learns embeddings representations of words, topics, and POS while considering syntactic structures that reflect their relationships. Since we know that the order of words, topics, and POS and their long-range dependencies constitute syntactic structures, SAUL uses recurrent neural networks to learn them while preserving their structures. WTP2Vec encodes topic (global statistical information) and context (local information) as syntactic awareness embedding representations of topics, POS, and words, while SAUL learns their long-range dependencies such as topical phrases, adjacent topics, and POS through their embedding representations.

Experiments confirm the following key advantages of our models, and verify that they achieve our goal more efficiently than is possible by using any of the information in isolation.

- WTP2Vec enhances the expressive power of topic models, as it can find more distinct topics, and enhances embedding models, as it can improve the quality of topics and extract more informative embeddings.
- SAUL enhances neural language models with regard to generating coherent text by maintaining the order of POS, words and the structure of topics as the syntactic structure.

WI-IAT '21, December 14–17, 2021, ESSENDON, VIC, Australia
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-9115-3/21/12…$15.00
https://doi.org/10.1145/3486622.3493968

2 PREVIOUS WORK

Distributed representations with neural probabilistic language [1] represent words and documents as low-dimensional and dense real value vectors in single semantic space, and have achieved significant results in many tasks [33]. Herbelot and Vecchi [9] explored word embeddings and their utility for modeling language semantics. Word2vec [29] and its extensions [20, 31] are designed to capture a large number of syntactic and semantic word relationships by introducing a continuous Skip-gram model [27]; they offers excellent performance in NLP tasks. While previous topic models [13, 24] represent higher correlations between words via interpretable topics, Word2Vec and its extensions use vector geometry, and allow us to map words to real number vectors.

Because the probability distributions gained from topic models describe the statistical relationships of word occurrences in the corpus and so are not the best choice for feature representation [33], the combination of embedding models and topic models was proposed to represent words in a single semantic space. Topic2Vec [33] learns, as an alternative to probability, topic representations in the same vector space with words. The Gaussian LDA [4] uses pre-trained word embeddings and trains topics. STE [34] learns topic-specific word embeddings, term distributions of topics, and the topic distributions of the documents, to resolve the issue of polysemy. Topical Word Embedding (TWE) [25] is a composite model that forms topical word embeddings by using pre-trained topic structures and concatenating topical embedding with word embedding. Our approach focuses on POS as the syntactic information and assumes that POS can complement topics and words for exploiting semantic information; it integrates them by learning their embeddings.

Another approach to modeling sequences of words and topics is to employ RNN schemes [28], such as Long Short Term Memory (LSTM) [10]. RNN is a deep learning technique with low dimensional representations of words (i.e., word embedding). While topic models reflect the co-occurrence relationships of words and so fail to adequately model sequential dependency across sentences, LSTM models characterize long range dependencies of a sequence. Sentence Level Recurrent Topic Model [38], which uses LSTM, accepts word embedding vectors as input, and assigns topic mixture weights to each document like LDA. Word-Topic Mixture [32] trains word embedding and topic models simultaneously by combining the ideas of TW [25] and latent feature models with LDA [2]. Latent Topical Skip-Gram model [19] learns both topics and their vector representations. Yet, the main barrier to employing deep learning is that it is difficult to assign a reasonable interpretation to each dimension of the generated distribution. LSTMs require a large number of parameters, notwithstanding the simplicity of the underlying dynamics, rending them uninterpretable [41]. Topically Driven Neural Language Model (TDLM) [18] consists of a language model and a topic model, and jointly trains them. While Topic RNN [5] and Latent LSTM Allocation (LLA) [41] capture the semantic meaning in an end-to-end fashion, our approach learns topics in a pipeline fashion. Word and Topic 2 Vec (Wat2vec) and Topic Structure-Aware Neural Language Model (TSANL) [14] aim to explain the generative process of documents as based both semantic information and syntactic structures, and enhance previous embedding methods.

Table 1: Notations used in this paper

SYMBOL	DESCRIPTION
N, V, O, K	Documents, Vocabulary, POS, Topic size
M	Dimensionality of the embedding space
d_n	The n-th document
$\theta_n, t_v, g_o \in \mathbb{R}^K$	Topic representation of n-th document, the v-th word, the l-th POS
$a_k, b_v, q_o \in \mathbb{R}^M$	Embedding of k-th topic, the v-th word, the l-th POS
$c_v, c_{\dot{v}} \in \mathbb{R}^M$	Context word embedding of v-th word
$h_o, h_{\dot{o}} \in \mathbb{R}^M$	Context POS embedding of p-th POS
$D \in \mathbb{R}^{N \times V}$	Document-word matrix
$S \in \mathbb{R}^{N \times V}$	Document-POS matrix
$W \in \mathbb{R}^{V \times V}$	Word co-occurrence matrix
$E \in \mathbb{R}^{L \times O}$	POS co-occurrence matrix
$T \in \mathbb{R}^{K \times V}$	Topic-word matrix
$G \in \mathbb{R}^{K \times O}$	Topic-POS matrix
$\Theta \in \mathbb{R}^{K \times N}$	Document-topic matrix
$A \in \mathbb{R}^{M \times K}$	Topic-embedding matrix
$B \in \mathbb{R}^{M \times V}$	Word-embedding matrix
$C \in \mathbb{R}^{M \times V}$	Context word-embedding matrix
$Q \in \mathbb{R}^{M \times O}$	POS-embedding matrix
$H \in \mathbb{R}^{M \times O}$	Context POS-embedding matrix
$d_{j,i}, e_{j,i}, g_{j,i}, r_{j,i}, s_{j,i}, t_{j,i}, w_{j,i}$	The j, i-th entry in matrix D, E, G, R, S, T, W

3 MOTIVATION AND METHODOLOGY

We explain the motivation and architecture of our approach, and describe our concepts, see Figure 1 and 2. As embedding methods capture semantic information and neural language models learn syntactic structure, we are motivated by the expectation that enhancing these models will result in improving the quality of text output from generation models.

Our approach focuses on POS as the syntactic information and assumes that POS complements topics and words in exploiting semantic information, and can help to train text generation models. This leads us to learn the embeddings that reflect these pieces of information by incorporating POS into it, and to propose **W**ords and **T**opics and **P**OS 2 Vec (WTP2Vec) as an embedding method, and Structure Aware Unified Language Model (SAUL) as a neural language model.

WTP2Vec combines topic models and distributed representations to learn embedding representations of topic, words, and POS tags. Topic models assume K underlying topics, each of which is a distribution over a fixed vocabulary, V. Like other models, our model represents each document as a mixture of topics, $d_j = \{z_{j,1}, ..., z_{j,i}, ..., z_{j,N_j}\}$, where z_{ji} is the topic in the j-th document and N_j is the number of tokens in the j-th document; it generates each word $w_{j,i}$ from one topic. As shown in Figure 1, our approach hypothesizes that the causal relationships among topics, POS, and words shown in Figure 1 form the syntactic structure. It 1) selects

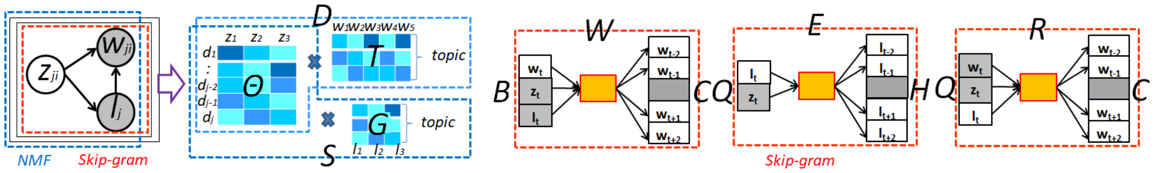

Figure 1: (left) Graphical model of WTP2Vec: The causal relationships among topics, z_j, POS, l_j, and words, w_j, is the syntactic structure of our topic model, and (right) Matrices gained from WTP2Vec: While D, and S are decomposed into Θ and T, and Θ and G, respectively, where D and S share Θ. From a given corpus, the shifted positive PMI (SPPMI) matrices, W, E, and R, are gained, where we factorize these matrices and can gain embedding matrices B, C, H, and Q.

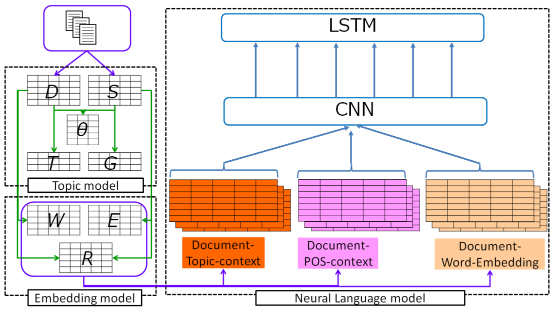

Figure 2: Learning architecture of WTP2Vec and SAUL: WTP2Vec is a combination of a MF-based topic model and an embedding model; SAUL is a neural language model. The i-th topic, z_{ji}, POS tag, l_{ji}, and word w_{ji} in j-th document are fed to LSTMs with their embeddings a_{ji}, q_{ji}, and b_{ji}, respectively

topic, z_{ji}, 2)draws the element of POS tags, l_{ji}, from a corresponding (current) topic, z_{ji}, and 3) generates a word, w_{ji}, according the pair of the topic and POS tags; previous topic models directly generate a word according to the corresponding topic.

As indicated by sentence parsing research, the syntactic structures of topics and POS have more complicated and longer range dependencies than phrases. To generate coherent text precisely in view of their semantic information and syntactic structure, SAUL trains neural language models by taking advantage of CNN and LSTM. Certain word sequences scattered throughout the document can be good indicators of the topic, but CNN can learn to find such local indicators, regardless of their position [7]. Since CNN is used as feature extractors for LSTMs on audio and textual input data, and have subsequently been shown to be effective for NLP [12, 15], our approach is to train CNN over topic/POS tags/word embeddings, **a**, **q**, and, **b**, with LSTMs on subsequent layers to support sequence prediction. While finding better ways to model long-range dependencies is an open research challenge in language modeling [5], SAUL learns them and predicts succeeding words by considering both global semantic and local semantic information through WTP2Vec.

4 WTP2VEC

4.1 Basic idea

Following the assumption that topic models and word embeddings are complementary in improving the quality of topics and embeddings, WTP2Vec (**W**ords and **T**opics and **P**OS 2 Vec) extends these models by incorporating POS. WTP2Vec discovers topics that capture the interactions among words, topics, and POS, whereas previous combination models (e.g., Topic2vec and CLM) discover word and topic embeddings. The topic models that can exploit the topics present in a given corpus fall into two groups of topic models, i.e., Bayesian generative topic models, such as latent Dirichlet allocation [2], and NMF [21]. As NMF-based models are robust to noise [35] and decompose the observed document-word and document-POS tag into topic-word and topic-POS tag factors with low calculation cost and outstanding performance [17], WTP2Vec adopts NMF to learn topics. As WTP2Vec aims to embed different types (words, topics, and POS) simultaneously, it is designed to capture the relationships between words and POS through topics, like supervised topic models in Bayesian generative topic models [42]. As WTP2Vec uses shared topics in jointly decomposing the document-word matrix and document-POS matrix with shared topics, it can be thought of as an NMF-based supervised topic model.

4.2 Representing words and POS using Matrix

Given the document-word matrix $\mathbf{D} \in \mathbb{R}^{V \times N}$ from N documents and V words, NMF decomposes the matrix into the product of documents and words denoted by $\theta_n \in \mathbb{R}^K (n = 1, ..., N)$ and $t_v \in \mathbb{R}^K (v = 1, ..., V)$, respectively. To make **D**, WTP2Vec uses TF-IDF weights instead of raw frequency. The objective function used to factorize **D** with regularization is:

$$L_D = ||D - T^T\Theta||_2^2 + \lambda_d(||T||_2^2 + ||\Theta||_2^2), \quad T \ \ and \ \ \Theta \geq 0, \tag{1}$$

where $||T||_2^2$, and $||\Theta||_2^2$ denote the l_2 norm regularization applied to document-topic matrix $\Theta \in \mathbb{R}^{K \times N}$, and topic-word matrix $T \in \mathbb{R}^{K \times V}$, respectively; λ_d is the regularization parameter that prevents overfitting.

4.3 Representing embeddings of words and POS

As previous models tag each word with a POS, sequences of POS are analogous to sequences of words. Likewise, NMF decomposes the document-POS matrix $\mathbf{S} \in \mathbb{R}^{O \times N}$ from N documents and O POS into the product of documents, and POS denoted θ_n, and $u_o \in \mathbb{R}^K (o=1, ..., O)$, respectively. As in constructing **D**, WTP2Vec weights **S** using TF-IDF. Note that g_o derived from **S** is equivalent to t_v derived from **D**.

$$L_S = ||S - G^T\Theta||_2^2 + \lambda_s(||G||_2^2 + ||\Theta||_2^2), \quad G \ \ and \ \ \Theta \geq 0, \tag{2}$$

where $||G||_2^2$, and $||\Theta||_2^2$ denote the l_2 norm regularization applied to document-topic matrix Θ, and topic-POS matrix $\mathbf{U} \in \mathbb{R}^{K\times O}$, respectively; λ_s is the regularization parameter that prevents overfitting.

As the recent success of word embedding models can be interpreted as the utility of factorizing the word co-occurrence matrix [22], our approach is based on idea that word-topic and POS-topic co-occurrence matrix factorization can be interpreted as yielding topic and POS embedding models. Levy and Goldberg [22] showed an equivalence between skip-gram Word2Vec trained with negative sampling (SGNS) values of k and implicit factorization of the pointwise mutual information (PMI) matrix shifted by log$k$. After making the connection between Word2Vec and implicit matrix factorization, Levy and Goldberg [22] performed word embedding by spectral dimensionality reduction on the shifted positive PMI (SPPMI) matrix. This matrix can be factorized by either using an eigenvalue method or solving the matrix factorization problem iteratively [40]. As SGNS does not require tuning of an optimization procedure and can capture the relationships between a word and its context within small sliding windows [27, 29], we follow this approach in Du2Vec similar to other models [23, 35].

Following this framework, we define the word co-occurrence matrix, $\mathbf{W} \in \mathbb{V}^V (v = 1, ..., V)$ as the co-occurrence SPPMI matrix instead of the raw frequency matrix, and then gain word embedding matrix $\mathbf{B}$ and context word embedding matrix $\mathbf{C} \in \mathbb{R}^{M\times V}$ by factorizing $\mathbf{W}$:

$$L_W = ||W - B^T C||_2^2 + \lambda_\omega(||B||_2^2 + ||C||_2^2), \quad B \ \ and \ \ C \geq 0, \tag{3}$$

where λ_ω is the parameter that prevents overfitting.

Likewise, we construct the POS co-occurrence matrix, $\mathbf{E} \in \mathbb{O}^O (o = 1, ..., O)$ as the co-occurrence SPPMI matrix from $\mathbf{S}$, and learn POS embedding matrix $\mathbf{Q} \in \mathbb{R}^{M\times O}$ and context POS embedding matrix $\mathbf{H} \in \mathbb{R}^{M\times O}$ by factorizing matrix $\mathbf{E}$.

$$L_E = ||E - Q^T H||_2^2 + \lambda_e(||H||_2^2 + ||Q||_2^2), \quad H \ \ and \ \ Q \geq 0, \tag{4}$$

where λ_e is the parameter that prevents overfitting.

As each word, w_{ji}, is sampled according to the topic, z_{ji}, and the corresponding POS, l_{ji}, we construct the word and POS co-occurrence matrix, $\mathbf{R} \in \mathbb{R}^{V\times O}$:

$$L_R = ||R - Q^T C||_2^2 + \lambda_r(||Q||_2^2 + ||C||_2^2), \quad Q \ \ and \ \ C \geq 0, \tag{5}$$

where λ_r is the parameter that prevents overfitting.

4.4 Joint learning embeddings of words and POS

Based on the idea that the distances between word embeddings reflect their topical similarities, CLM [39] factorizes topic-word matrix, $\mathbf{T}$, into the topic embedding matrix, $\mathbf{A}$, and context word embedding matrix, $\mathbf{C}$, similar to the approach of [23]. This factorization idea coincides with the idea of TWE-1 [25] whereby the target word with its topic predicts context words in Skip-gram; other methods predict these words using only the target word. Following this flow, WTP2Vec jointly learns embeddings by combining NMF with an embedding method via shared topics. As $\mathbf{G}$ plays the role of $\mathbf{T}$ in $\mathbf{D}$ in Eq (1), this representation is written as:

$$\begin{aligned} L_{T,G} = & \underbrace{||T - A^T C||_2^2 + \lambda_t(||A||_2^2 + ||C||_2^2)}_{topic-word} \\ & + \underbrace{||G - A^T H||_2^2 + \lambda_u(||A||_2^2 + ||H||_2^2)}_{topic-label}, \quad A \ \ , \ \ C \ \ and \ \ H \geq 0, \end{aligned} \tag{6}$$

where λ_t and λ_u are parameters that prevent overfitting. This equation is based on the fact that the probability of word w being grouped into topic z, $p(z|w)$, can be measured by the inner product of the corresponding topic embedding, and word embedding, being grouped into z can be determined in the same way [39]. That is, we use both $\mathbf{T}$ and $\mathbf{G}$ to gain embeddings, even though they are not SPPMI matrices.

Following the idea of SPPMI, our approach exploits local context information and learns POS embeddings by factorizing matrix **D,S,W,E,T,G,R**. The objective function that jointly utilizes both the global context information and local context information is given as:

$$\begin{aligned} L_{D,S,W,E,T,U,R} = & \underbrace{\omega_d||D - T^T\Theta||_2^2}_{NMF} + \underbrace{\omega_w||W - B^T C||_2^2}_{SPPMI} \\ & + \underbrace{\omega_s||S - G^T\Theta||_2^2}_{NMF} + \underbrace{\omega_e||E - Q^T H||_2^2}_{SPPMI} + \underbrace{\omega_t||T - A^T C||_2^2}_{joint(NMF,SPPMI)} \\ & + \underbrace{\omega_g||G - A^T H||_2^2}_{joint(NMF,SPPMI)} + \underbrace{\omega_r||R - Q^T C||_2^2}_{SPPMI} + \underbrace{\omega_r\lambda_r(||Q||_2^2 + ||C||_2^2)}_{regularization} \\ & + \underbrace{\omega_d\lambda_d(||T||_2^2 + ||\Theta||_2^2) + \omega_e\lambda_e(||H||_2^2 + ||Q||_2^2)}_{regularization} \\ & + \underbrace{\omega_s\lambda_s(||G||_2^2 + ||\Theta||_2^2) + \omega_g\lambda_g(||A||_2^2 + ||H||_2^2)}_{regularization} \\ & + \underbrace{\omega_t\lambda_t(||A||_2^2 + ||C||_2^2) + \omega_w\lambda_w(||B||_2^2 + ||C||_2^2)}_{regularization}, \\ & A, B, C, G, H, T, Q, \Theta > 0 \end{aligned} \tag{7}$$

where ω_d and ω_s are the parameters that control the weights of NMF of a given corpus, ω_w and ω_e are the parameters that control the weights of SPPMI of the corpus, and ω_t and ω_u are the parameters that control the weights of the joint of NMF and SPPMI. ω_s can be tuned to set the balance between topic-word and topic-POS.

Following the Alternative Least Squares (ALS) matrix factorization method [11], we obtain closed form coordinate updates by iteratively setting the gradient to zero:

$$
\begin{aligned}
\theta_n &= (\omega_d \sum_{v=1}^{V} t_v t_v^T + (\omega_d \lambda_d + \omega_s \lambda_s) I + \omega_s \sum_{o=1}^{O} g_o g_o^T)^{-1} \\
&\times (\omega_d \sum_{v=1}^{V} d_{v,n} t_v + \omega_s \sum_{o=1}^{O} s_{o,n} g_o) \\
t_v &= (\omega_d \sum_{n=1}^{N} \theta_n \theta_n^T + (\omega_t + \omega_d \lambda_d) I)^{-1} \\
&\times (\omega_d \sum_{n=1}^{N} d_{v,n} \theta_n + \omega_t A c_v) \\
a_k &= (\omega_t \sum_{v=1}^{V} c_v c_v^T + (\omega_g \lambda_g + \omega_t \lambda_t) I + \omega_g \sum_{o=1}^{O} h_o h_o^T)^{-1} \\
&\times (\omega_t \sum_{v=1}^{V} t_{k,v} c_v + \omega_g \sum_{o=1}^{O} g_{k,o} h_o) \\
b_v &= (\sum_{\dot{v}=1}^{V} c_{\dot{v}} c_{\dot{v}}^T + \lambda_w I)^{-1} \times (\sum_{\dot{v}=1}^{V} w_{v,\dot{v}} c_{\dot{v}}) \\
c_{\dot{v}} &= (\omega_w \sum_{v=1}^{V} b_v b_v^T + \omega_t \sum_{k=1}^{K} a_k a_k^T + \omega_r \sum_{o=1}^{O} q_o q_o^T \\
&+ (\omega_t \lambda_t + \omega_w \lambda_w + \omega_r \lambda_r) I)^{-1} \\
&\times (\omega_w \sum_{v=1}^{V} w_{v,\dot{v}} b_v + \omega_t \sum_{k=1}^{K} t_{k,\dot{v}} a_k + \omega_r \sum_{o=1}^{O} r_{o,\dot{v}} q_o) \\
g_o &= (\omega_s \sum_{n=1}^{N} \theta_n \theta_n^T + (\omega_g + \omega_s \lambda_s) I)^{-1} \\
&\times (\omega_s \sum_{n=1}^{N} s_{o,n} \theta_n + \omega_g A h_o) \\
q_o &= (\omega_e \sum_{\dot{o}=1}^{O} h_{\dot{o}} h_{\dot{o}}^T + \omega_r \sum_{\dot{v}=1}^{V} c_{\dot{v}} c_{\dot{v}}^T + (\omega_e \lambda_e + \omega_r \lambda_r) I)^{-1} \\
&\times (\omega_e \sum_{\dot{o}=1}^{O} e_{o,\dot{o}} h_{\dot{o}} + \omega_r \sum_{\dot{v}=1}^{V} r_{o,\dot{v}} c_{\dot{v}}) \\
h_{\dot{o}} &= (\omega_e \sum_{o=1}^{O} q_o q_o^T + \omega_g \sum_{k=1}^{K} a_k a_k^T \\
&+ (\omega_e \lambda_e + \omega_g \lambda_g) I)^{-1} \times (\omega_e \sum_{o=1}^{O} e_{o,\dot{o}} q_o + \omega_g \sum_{k=1}^{K} g_{k,\dot{o}} a_k)
\end{aligned}
\tag{8}
$$

4.5 Complexity Analysis

The computational cost of constructing the SPPMI matrices, **W**, **E** and **R**, is usually high as the problem scales quadratically with the number of words in the corpus. As shown in the previous section, all the coordinate updates can be parallelized across words, topics and POS, so these matrices only need to be constructed only once and then can be parallelized over nodes. The time complexity of WTP2Vec is $O((O_{D,W,E,R})k^2 + (O + N + V)k^3)$, where $O_{D,W,E,R}$ is the time complexity of constructing matrices D, W, E and R.

Table 2: Basic statistics of the datasets in this paper: Both datasets have 72 kinds of POS.

category	#documents (D)	#vocabulary (V)
20Newsgropus	11,296	10,785
Reuters	13,328	6,458

Because these matrices are often sparse, and can scale linearly, the computational cost of learning WTP2Vec could approximate $O(((O + N + V)k^3)$.

4.6 Training and Regularization detection of SAUL

We outline the execution of the whole learning process and inference in Figure 2. As the kernel size in the convolutional layer of CNN defines the number of words considered as the convolution is passed across the input data, by providing a grouping parameter, the multi-channel convolutional neural layer with different sized kernels can use different size n-grams (groups of topics, words, and POS) at a time when processing documents. Whilst these layers learn how to best integrate these interpretations, CNN can extract feature via max-over-time polling layers and the size of input data for following LSTM layers. This is the reason why our model adds CNN layers on the front end, follows these layers with LSTM layers, and sets a Dense layer on the output.

We represent a document of N_j words as a concatenation of word vectors, b_{ji}, and construct an $N_{ji} \times M$ Document word-context table. Similarly, we represent a POS of N_j words as a concatenation of POS vectors, q_{ji}, and construct an $N_{ji} \times M$ Document topic-context table. We apply a convolution operation to produce a feature map and then a max-over-time pooling operation to capture the most important features [3, 15]. These features are passed to a fully connected softmax layer and input to LSTMs. Here, the probability of a length-t sentence $w_{1:t}$ given the input vector consisting of a, b, q,

$$p(w_{1:1+t}|\mathbf{a},\mathbf{b},\mathbf{q}) = p(w_1|a_1, q_1) \prod_{t=2}^{T} p(w_t|a_{<t}, b_{<t}, q_{<t}). \tag{9}$$

This equation means that the first word is generated from $p(w_1|a_1, q_1)$, with this probability gained over last LSTM layer, where a_1, and q_1 denotes the embedding representation of i-th topic, and POS, respectively. Each $p(w_t|a_{<t}, q_{<t})$, where $< t = \{1, ..., t-1\}$, is recursively updated through LSTMs, or CNN from a given input data and embeddings of the previous word, topic, and POS sequence.

SAUL trains the language model, and employs categorical cross-entropy loss to train them with Eq (9), as the cross-entropy can be calculated from multiple-class classification, and measures the error between the actual probability distribution and predicted probability distribution. As with model learning, SAUL uses minibatches and SGD [16] to conduct training, where we employ embedding dropout [6, 36] to dropout on the embedding matrices at topic and word levels, and regularize SAUL. That is, SAUL uses the categorical cross-entropy loss as the loss function, where the aim is to minimize the loss value.

Table 3: Baseline and parameter settings: As TWE [25] utilizes pre-trained topics, that are the output of LDA, to learn topic and word embeddings, and does not re-learn topics, its score is the same as the score of LDA. We set the number the document-topic hyper parameter to $1/K$ and topic-word hyper parameter to 0.01. We set other hyper-parameters recorded in the original work.

Category	Models
topic model	LDA
supervised probabilistic topic model	LLDA
matrix factorization topic model	NMF
combinations of topic model and word embedding model	TWE (TWE-1)[1], Topic2Vec[2], CLM[3]

[1] https://github.com/largelymfs/topical_word_embeddings
[2] https://github.com/ukgovdatascience/topic2vec
[3] https://github.com/XunGuangxu/2in1

Table 4: Comparison of topic coherence: N is #ranked words. The dimensionality of the embedding space for Topic2Vec, CLM and WTP2Vec is set to 50, the skip length (context window size) is set to 5, and the negative sampling number is set to 5. Values in bold show best performance.

	20News			Reuters		
N	5	10	20	5	10	20
LDA(TWE)	-2.04	-2.19	-3.21	-2.75	-2.83	-3.57
NMF	-1.88	-2.23	-3.12	-2.55	-2.86	-3.73
Topic2Vec	-1.65	-2.02	-2.55	-2.19	-2.24	-2.55
LLA	-1.66	-1.91	-2.23	-2.14	-2.32	-2.71
CLM	-1.55	-1.72	-2.02	-1.36	-1.74	-2.15
WTP2Vec	**-0.94**	**-1.41**	**-1.93**	**-0.96**	**-1.57**	**-2.07**

5 EXPERIMENTS

5.1 Datasets and Experiment design

We conducted evaluations on 20 Newsgroups data sets[4] and Reuters-21578[5], as they are well-known, publicly available and used in the many related work [8, 25, 26, 39]. First, we tagged each word with POS, using NLTK[6], and converted all words to lowercase, to permit dictionary sharing with Word2Vec and its extensions in Tensorflow[7]. Because stop words and rare words are considered to be a word necessary for coherent document generation, we use all words in these experiments; the statistics of the data set are shown in Table 2 and the base model in Table 3.

5.2 Topic Coherence

The aim of this experiment is to evaluate how well each model discovers topics. This goal leads us to compare WTP2Vec with existing topic models and topic based embedding models. In order to assess their quality quantitatively, we use the topic coherence measure [30] to assess the relatedness of the top-ranked words. This measure matches the intuition that top ranked words for the same topic tend to co-occur in documents that address the same topic and all are closely semantically related. This score shows high consistency with human judgements in terms of topic quality [30], where higher scores indicate greater topic coherency.

We set the iterations of the Gibbs sampler, parameter update or epochs to 200 for all models, where the first 50 iterations were used to burn in the Gibbs sampler, where both CLM and WTP2Vec learn word embedding representation using the matrix factorization. We varied the number of top ranked words, and measured the resulting performance by using the coherence model function of gensim[8] with "u_mass" for all methods.

Table 4 shows that the top words of learned topics are semantically coherent, which coincides with the finding that using word embeddings improves the quality of latent topic models [25, 32, 39]. Compared with LDA, NMF, both CLM and WTP2Vec shows significantly better results, which implies that learning topics, POS, and embeddings simultaneously helps to discover more coherent topics. Moreover, as WTP2Vec learns topics using the order of both word and topic in a given corpus in each document and updates them iteratively, it groups semantically related words more efficiently than the others, and yields more distinct topics than the others. This result shows that WTP2Vec offers superior interpretability over regular topic models.

Table 5: Performance comparison in document classification task: P, R, and F1 denote Precision, Recall, and F1-score,respectively. The number of topics for all models over all datasets is set to 50. The skip length of all embedding models is set to 5, and the negative sampling number is set to 5. Values in bold show best performance. The hyper-parameters of TDLM follow previous work [18].)

	20News			Reuters		
	P	R	F1	P	R	F1
TDLM	0.62	0.62	0.62	0.63	0.62	0.63
NMF	0.56	0.55	0.55	0.57	0.57	0.57
TWE(z)	0.72	0.71	0.71	0.64	0.62	0.62
CLM(z)	0.71	0.70	0.70	0.63	0.64	0.64
WTP2Vec(z)	0.82	0.86	0.82	0.71	0.71	0.71
Doc2Vec	0.59	0.57	0.57	0.57	0.56	0.57
Word2Vec(g)	0.65	0.63	0.62	0.59	0.60	0.59
Word2Vec	0.66	0.67	0.64	0.63	0.63	0.62
TWE	0.71	0.70	0.69	0.66	0.64	0.64
CLM	0.72	0.72	0.71	0.67	0.67	0.66
WTP2Vec	**0.88**	**0.89**	**0.88**	**0.79**	**0.80**	**0.80**

5.3 Document Classification

We evaluated the quality of embeddings from the quality of document classification results, a popular way of evaluating the effectiveness of learned document representations yielded by topic

[4] http://qwone.com/ jason/20Newsgroups/
[5] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html
[6] NLTK 3.2.4: http://www.nltk.org
[7] https://www.tensorflow.org/tutorials/word2vec
[8] https://radimrehurek.com/gensim/models/coherencemodel.html

Table 6: The common setting of LSTM, CNN and SAUL: We use the Convolution2D in CNN.

Sequence length of topic model or input	30
Minibatch size	64
Number of filters in CNN	64
Filter size in CNN	4 (2,3,4,5)
Number of LSTM layers	1
Learning rate of optimizer	0.001
Dropout keep probability	0.5

models [34, 35]. As this experiment aims to evaluate document-level representation, we used TDLM [18] as the neural language model benchmark, Word2Vec as the word representation benchmark, and Doc2vec (PD-DBOW) as the document level representation benchmark, with TWE and CLM as the topic awareness embedding representation benchmarks; documents are classified by SVM over each representation vector. Keeping the same experimental setting as in the previous section for CLM and WTP2Vec, we varied the dimensionality of documents among 100, 200, and 300 except for Word2Vec(g)(Google[9]). Note that 1) we show only the results gained from TWE, CLM, and WTP2Vec with 100 embedding dimension, that is the worst result of each model, due to the space, 2) only $Word2Vec_g$ yields fixed-length vectors with 300 dimensions learned from other data sets; we trained Word2Vec over the same data sets using Tensorflow. The document-topic distribution of Bayesian generative topic models corresponds to the document-topic matrix of NMF based models such as CLM and WTP2Vec or document-embedding of Doc2vec and TWE, where both CLM and WTP2Vec learn embedding representation using the matrix factorization. While TWE(z), TDLM, CLM(z), and WTP2Vec(z) use document-topic representation for this task, TWE, CLM, and WTP2Vec use the average of word -embeddings. To compare overall performance, we compared them using macro-averaged precision, recall, and F1 measure coherence using gensim[10]. The classifiers of TWE, TDLM, CLM, and WTP2Vec were trained based on the topic representation of the training dataset; the SVM classifier received the document embedding as input.

Table 5 shows that our model yields word representations that are effective for document classification, so document-topic distribution beats embedding models in this task. Compared with Word2Vec(g), both CLM and WTP2Vec outperform Word2Vec(g); even the dimension of embeddings is smaller than that of Word2Vec(g). This comparison demonstrates that the word embeddings yielded by skip-gram with topics and POS tags could be improved.

5.4 Document Generation

As this task is a kind of language model evaluation, we use the well-known test-set perplexity, which has been widely used, to evaluate the quality of generated document. To measure each element of SAUL, we compare SAUL with various combination of networks and embedding methods. We adopt the experiment setting [18] to all models, show the parameter setting in Table 6, and the results in Table 7 and the samples in Table 8. Note that models fed with

[9]https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit

[10]https://radimrehurek.com/gensim/

Table 7: Comparison of test-set perplexity: N is #ranked words. In the Emb column, the embedding method in each model, where TDLM learns embedding inside and show Self. In the 1-st and 2-nd, the networks used in each layer of each hierarchical model. The setting of dimensionality, topics, the skip length and the negative sampling number follows Table 5. Values in bold show best performance.

	Emb	1-st	2-nd	20News	Reuters
Benchmark	Self	TDLM		52.2	57.3
	CLM	LSTM	LSTM	54.3	58.1
	WTP2Vec	LSTM	LSTM	50.7	55.4
	CLM	CNN	LSTM	44.8	51.5
SAUL	WTP2Vec	CNN	LSTM	**41.6**	**44.1**

Table 8: Generated samples from SAUL trained on the (top) 20News and (bottom) Reuters from the same seed words.

20News	**the miracle of theism oxford this volume** contains a comprehensive review of the principal arguments and for the existence of god includes an in which he somewhat
Reuters	**the miracle of theism oxford this volume** do not view view religious as an agnostic example atheism is closer note that here belief is not a good of religious belief

WTP2Vec contain only 10 words with max length = 30, as these models take a concatenated embeddings of topic, POS, and word, $\epsilon_{1:i} = a_1 \oplus q_1 \oplus b_1 ... a_i \oplus q_i \oplus b_i$. This table shows that SAUL offers lower perplexity than other setting, and supports our idea that utilizing WTP2Vec and SAUL yields a reduction in perplexity and could generate coherent text.

6 DISCUSSION

The problem of employing deep learning for natural language processing is that it is difficult to attach a reasonable interpretation to each dimension of the generated distributions [1, 25, 32, 33]. SAUL uses CNN and LSTM to train these structure, as it yields both the sequential dependency of word and topic coherency.

While our approach enhances the quality of topic models, and neural language models by incorporating POS, its quality depends on the quality of POS-tagging significantly. By training POS-tagging processes as subtasks in a multi-task learning setting, we would be able to improve the quality of POS-tagging, improve overall quality of contained models, and generate more coherent text. Since WTP2Vec requires SPPMI matrices and regularize these matrices, its computational cost appears higher that those of other approaches using neural networks. As POS limits each word in each token, fewer iterations are required to achieve convergence than other embedding approaches that ignore POS.

7 CONCLUSION

This paper proposed WTP2Vec, a combination of topic models and embedding models thorough POS, and SAUL, neural language models for text generation. Experiments showed that our approach learns the embeddings of topics, POS and words, and enhances previous models. while also maintaining their order for more coherent text generation. As POS does not depend strongly on the domain, we will consider a framework that can apply trained POS embeddings to other domains and decreases the computational cost for learning the embeddings of words and topics.

REFERENCES

[1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.* 3 (Mar 2003), 1137–1155.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *JMLR* 3 (2003), 993–1022.

[3] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.* 12 (nov 2011), 2493–2537.

[4] Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for Topic Models with Word Embeddings. In *ACL*. 795–804.

[5] Adji B. Dieng, Chong Wang, Jianfeng Gao, and John William Paisley. 2016. TopicRNN: A Recurrent Neural Network with Long-Range Semantic Dependency. *CoRR* abs/1611.01702 (2016).

[6] Yarin Gal and Zoubin Ghahramani. 2016. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In *NIPS*. 1027–1035.

[7] Yoav Goldberg. 2016. A Primer on Neural Network Models for Natural Language Processing. *J. Artif. Intell. Res.* 57 (2016), 345–420.

[8] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW*. 507–517.

[9] Aurélie Herbelot and Eva Maria Vecchi. 2015. Building a shared world: mapping distributional to model-theoretic semantic spaces. In *EMNLP*. 22–32.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[11] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM*. 263–272.

[12] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *ACL*. 655–665.

[13] Noriaki Kawamae. 2016. *N*-gram over Context. In *WWW*. 1045–1055.

[14] Noriaki Kawamae. 2019. Topic Structure-Aware Neural Language Model: Unified language model that maintains word and topic ordering by their embedded representations. In *WWW*. 2900–2906.

[15] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*. 1746–1751.

[16] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[17] Da Kuang, Sangwoon Yun, and Haesun Park. 2015. SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering. *J. Global Optimization* 62, 3 (2015), 545–574.

[18] Jey Han Lau, Timothy Baldwin, and Trevor Cohn. 2017. Topically Driven Neural Language Model. In *ACL*. 355–365.

[19] Jarvan Law, Hankz Hankui Zhuo, Junhua He, and Erhu Rong. 2017. LTSG: Latent Topical Skip-Gram for Mutually Learning Topic Model and Vector Representations. *CoRR* (2017).

[20] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML*. 1188–1196.

[21] Daniel D. Lee and H. Sebastian Seung. 1999. Learning the parts of objects by nonnegative matrix factorization. *Nature* 401, 6755 (1999), 788–791.

[22] Omer Levy and Yoav Goldberg. 2014. Neural Word Embedding As Implicit Matrix Factorization. In *NIPS*. 2177–2185.

[23] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei. 2016. Factorization Meets the Item Embedding: Regularizing Matrix Factorization with Item Co-occurrence. In *RecSys*. 59–66.

[24] Robert V. Lindsey, William P. Headden, III, and Michael J Stipicevic. 2012. A phrase-discovering topic model using hierarchical Pitman-Yor processes. In *EMNLP-CoNLL*. 214–222.

[25] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical Word Embeddings. In *AAAI*. 2418–2424.

[26] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR*. 43–52.

[27] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* (2013).

[28] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*. 1045–1048.

[29] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*. 3111–3119.

[30] David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2010. Optimizing Semantic Coherence in Topic Models. In *EMNLP*. 262–272.

[31] Christopher E. Moody. 2016. Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec. *CoRR* abs/1605.02019 (2016).

[32] Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving Topic Models with Latent Feature Word Representations. *TACL* 3, 4 (2015), 299–313.

[33] Liqiang Niu, Xinyu Dai, Jianbing Zhang, and Jiajun Chen. 2015. Topic2Vec: Learning distributed representations of topics. In *IALP*. 193–196.

[34] Bei Shi, Wai Lam, Shoaib Jameel, Steven Schockaert, and Kwun Ping Lai. 2017. Jointly Learning Word Embeddings and Latent Topics. In *SIGIR*. 375–384.

[35] Tian Shi, Kyeongpil Kang, Jaegul Choo, and Chandan K. Reddy. 2018. Short-Text Topic Modeling via Non-negative Matrix Factorization Enriched with Local Word-Context Correlations. In *WWW*. 1105–1114.

[36] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* 15, 1 (2014), 1929–1958.

[37] Hendrik Strobelt, Sebastian Gehrmann, Bernd Huber, Hanspeter Pfister, and Alexander M. Rush. 2016. Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks. *CoRR* abs/1606.07461 (2016).

[38] Fei Tian, Bin Gao, Di He, and Tie-Yan Liu. 2016. Sentence Level Recurrent Topic Model: Letting Topics Speak for Themselves. *CoRR* (2016).

[39] Guangxu Xun, Yaliang Li, Jing Gao, and Aidong Zhang. 2017. Collaboratively Improving Topic Discovery and Word Embeddings by Coordinating Global and Local Contexts. In *KDD*. 535–543.

[40] Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. 2018. Dynamic Word Embeddings for Evolving Semantic Discovery. In *WSDM*. 673–681.

[41] Manzil Zaheer, Amr Ahmed, and Alexander J. Smola. 2017. Latent LSTM Allocation: Joint Clustering and Non-Linear Dynamic Modeling of Sequence Datay. In *ICML*. 3967–3976.

[42] Jianwen Zhang, Yangqiu Song, Changshui Zhang, and Shixia Liu. 2010. Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora. In *KDD*. 1079–1088.