Improving Textual Network Embedding with Global Attention via Optimal Transport

Liqun Chen, Guoyin Wang, Chenyang Tao, Dinghan Shen, Pengyu Cheng, Xinyuan Zhang, Wenlin Wang, Yizhe Zhang, Lawrence Carin
liqun.chen@duke.edu

Abstract

Constituting highly informative network embeddings is an important tool for network analysis. It encodes network topology, along with other useful side information, into low-dimensional node-based feature representations that can be exploited by statistical modeling. This work focuses on learning context-aware network embeddings augmented with text data. We reformulate the network-embedding problem, and present two novel strategies to improve over traditional attention mechanisms: (i) a content-aware sparse attention module based on optimal transport, and (ii) a high-level attention parsing module. Our approach yields naturally sparse and self-normalized relational inference. It can capture long-term interactions between sequences, thus addressing the challenges faced by existing textual network embedding schemes. Extensive experiments are conducted to demonstrate our model can consistently outperform alternative state-of-the-art methods.

1 Introduction

When performing network embedding, one maps network nodes into vector representations that reside in a low-dimensional latent space. Such techniques seek to encode topological information of the network into the embedding, such as affinity (Tang and Liu, 2009), local interactions (*e.g,* local neighborhoods) (Perozzi et al., 2014), and high-level properties such as community structure (Wang et al., 2017). Relative to classical network-representation learning schemes (Zhang et al., 2018a), network embeddings provide a more fine-grained representation that can be easily repurposed for other downstream applications (*e.g.,* node classification, link prediction, content recommendation and anomaly detection).

For real-world networks, one naturally may have access to rich side information about each node. Of particular interest are *textual networks*, where the side information comes in the form of natural language sequences (Le and Lauw, 2014). For example, user profiles or their online posts on social networks (*e.g.,* Facebook, Twitter), and documents in citation networks (*e.g.,* Cora, arXiv). The integration of text information promises to significantly improve embeddings derived solely from the noisy, sparse edge representations (Yang et al., 2015).

Recent work has started to explore the joint embedding of network nodes and the associated text for abstracting more informative representations. Yang et al. (2015) reformulated DeepWalk embedding as a matrix factorization problem, and fused text-embedding into the solution, while Sun et al. (2016) augmented the network with documents as auxiliary nodes. Apart from direct embedding of the text content, one can first model the topics of the associated text (Blei et al., 2003) and then supply the predicted labels to facilitate embedding (Tu et al., 2016).

Many important downstream applications of network embeddings are *context-dependent*, since a static vector representation of the nodes adapts to the changing context less effectively (Tu et al., 2017). For example, the interactions between social network users are context-dependent (*e.g.,* family, work, interests), and contextualized user profiling can promote the specificity of recommendation systems. This motivates context-aware embedding techniques, such as CANE (Tu et al., 2017), where the vector embedding dynamically depends on the context. For textual networks, the associated texts are natural candidates for context. CANE introduced a simple mutual attention weighting mechanism to derive context-aware dynamic embeddings for link prediction. Following

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5193–5202
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

the CANE setup, WANE (Shen et al., 2018) further improved the contextualized embedding, by considering fine-grained text alignment.

Despite the promising results reported thus far, we identify three major limitations of existing context-aware network embedding solutions. First, mutual (or cross) attentions are computed from pairwise similarities between local text embeddings (word/phrase matching), whereas global sequence-level modeling is known to be more favorable across a wide range of NLP tasks (MacCartney and Manning, 2009; Liu et al., 2018; Malakasiotis and Androutsopoulos, 2007; Guo et al., 2018). Second, related to the above point, low-level affinity scores are directly used as mutual attention without considering any high-level parsing. Such an over-simplified operation denies desirable features, such as noise suppression and relational inference (Santoro et al., 2017), thereby compromising model performance. Third, mutual attention based on common similarity measures (*e.g.*, cosine similarity) typically yields dense attention matrices, while psychological and computational evidence suggests a sparse attention mechanism functions more effectively (Martins and Astudillo, 2016; Niculae and Blondel, 2017). Thus such naive similarity-based approaches can be suboptimal, since they are more likely to incorporate irrelevant word/phrase matching.

This work represents an attempt to improve context-aware textual network embedding, by addressing the above issues. Our contributions include: (i) We present a principled and more-general formulation of the network embedding problem, under reproducing kernel Hilbert spaces (RKHS) learning; this formulation clarifies aspects of the existing literature and provides a flexible framework for future extensions. (ii) A novel global sequence-level matching scheme is proposed, based on optimal transport, which matches key concepts between text sequences in a sparse attentive manner. (iii) We develop a high-level attention-parsing mechanism that operates on top of low-level attention, which is capable of capturing long-term interactions and allows relational inference for better contextualization. We term our model ***G**lobal **A**ttention **N**etwork **E**mbedding* (**GANE**). To validate the effectiveness of GANE, we benchmarked our models against state-of-the-art counterparts on multiple datasets. Our models consistently outperform competing methods.

2 Problem setup

We introduce basic notation and definitions used in this work.

Textual networks. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ be our textual network, where $\mathcal{V}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are the edges between the nodes, and $\mathcal{T} = \{S_v\}_{v \in \mathcal{V}}$ are the text data associated with each node. We use $S_v = [\omega_1, \cdots, \omega_{n_v}]$ to denote the token sequence associated with node $v \in \mathcal{V}$, of length $n_v = |S_v|$ where $|\cdot|$ denotes the counting measure. To simplify subsequent discussion, we assume all tokens have been pre-embedded in a p-dimensional feature space. As such, S_v can be directly regarded as a $\mathbb{R}^{p \times n_v}$ matrix tensor. We use $\{u, v\}$ to index the nodes throughout the paper. We consider directed unsigned graphs, meaning that for each edge pair $(u, v) \in \mathcal{E}$ there is a non-negative weight w_{uv} associated with it, and w_{uv} does not necessarily equal w_{vu}.

Textual network embedding. The goal of textual network embedding is to identify a d-dimensional embedding vector $\boldsymbol{z}_v \in \mathbb{R}^d$ for each node $v \in \mathcal{V}$, which encodes network topology ($\mathcal{E}$) via leveraging information from the associated text ($\mathcal{T}$). In mathematical terms, we want to learn an encoding (embedding) scheme $\mathbf{Z}_{\mathcal{G}} \triangleq \{\boldsymbol{z}_v = \text{Enc}(v; \mathcal{G})\}_{v \in \mathcal{V}}$ and a probabilistic decoding model with likelihood $p_\theta(E; \mathbf{Z})$, where $E \subseteq \mathcal{V} \times \mathcal{V}$ is a random network topology for node set $\mathcal{V}$, such that the likelihood for the observed topology $p_\theta(\mathcal{E}|\mathbf{Z}_{\mathcal{G}})$ is high. Note that for efficient coding schemes, the embedding dimension is much smaller than the network size (*i.e.*, $d \ll |\mathcal{V}|$). In a more general setup, the decoding objective can be replaced with $p_\theta(\mathcal{A}|\mathbf{Z})$, where $\mathcal{A}$ denotes observed attributes of interest (*e.g.*, node label, community structure, *etc.*).

Context-aware embedding. One way to promote coding efficiency is to contextualize the embeddings. More specifically, the embeddings additionally depend on an exogenous context c. To distinguish it from the context-free embedding $\boldsymbol{z}_u$, we denote the context-aware embedding as $\boldsymbol{z}_{u|c}$, where c is the context. For textual networks, when the embedding objective is network topology reconstruction, a natural choice is to treat the text as context (Tu et al., 2017). In particular, when modeling the edge w_{uv}, S_v and S_u are respectively treated as the context for context-aware embeddings $\boldsymbol{z}_{u|c}$ and $\boldsymbol{z}_{v|c}$, which are then used in the prediction of edge likelihood.

Attention & text alignment. Much content can be contained in a single text sequence, and retrieving them with a fixed length feature vector can be challenging. A more flexible solution is to employ an attention mechanism, which only attends to content that is relevant to a specific query (Vaswani et al., 2017). Specifically, attention models leverage a gating mechanism to de-emphasize irrelevant parts in the input; this method pools information only from the useful text, which is also a fixed length vector but that only encodes information with respect to one specific content (Santos et al., 2016). Popular choices of attention include normalized similarities in the feature space (*e.g.*, Softmax normalized cosine distances). For two text sequences, one can build a mutual attention by cross-relating the content from the respective text (Santoro et al., 2017). In text alignment, one further represents the content from one text sequence using the mutual attention based attentive-pooling on the other sequence (Shen et al., 2018).

Optimal transport (OT). Consider $\boldsymbol{\mu} = \{(\boldsymbol{x}_i, \mu_i)\}_{i=1}^n$ and $\boldsymbol{\nu} = \{(\boldsymbol{y}_j, \nu_j)\}_{j=1}^m$, a set of locations and their associated nonnegative mass (we assume $\sum_i \mu_i = \sum_j \nu_j = 1$). We call $\pi \in \mathbb{R}_+^{n\times m}$ a valid transport plan if it properly redistributes mass from $\boldsymbol{\mu}$ to $\boldsymbol{\nu}$, *i.e.*, $\sum_i \pi_{ij} = \nu_j$ and $\sum_j \pi_{ij} = \mu_i$. In other words, π breaks mass at $\{\boldsymbol{x}_i\}$ into smaller parts and transports π_{ij} units of $\boldsymbol{x}_i$ to $\boldsymbol{y}_j$. Given a cost function $c(\boldsymbol{x}, \boldsymbol{y})$ for transporting unit mass from $\boldsymbol{x}$ to $\boldsymbol{y}$, discretized OT solves the following constrained optimization for an optimal transport plan π^* (Peyré et al., 2017):

$$\mathcal{D}_c(\boldsymbol{\mu}, \boldsymbol{\nu}) = \inf_{\pi \in \Pi(\mu,\nu)} \left\{ \sum_{ij} \pi_{ij} c(\boldsymbol{x}_i, \boldsymbol{y}_j) \right\}, \quad (1)$$

where $\Pi(\boldsymbol{\mu}, \boldsymbol{\nu})$ denotes the set of all viable transport plans. Note that $c(\boldsymbol{x}, \boldsymbol{y})$ is a distance metric on $\mathcal{X}$, and $D_c(\boldsymbol{\mu}, \boldsymbol{\nu})$ induces a distance metric on the space of probability distributions supported on $\mathcal{X}$, commonly known as the Wasserstein distance (Villani, 2008). Popular choices of cost include Euclidean cost $\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$ for general probabilistic learning (Gulrajani et al., 2017) and cosine similarity cost $\cos(\boldsymbol{x}, \boldsymbol{y})$ for natural language models (Chen et al., 2018). Computationally, OT plans are often approximated with Sinkhorn-type iterative schemes (Cuturi, 2013). Algorithm 1 summarizes a particular variant used in our study (Xie et al., 2018).

Algorithm 1 Optimal transport solver (SolveOT)

Input: Sentence matrices $\mathbf{S} = \{\boldsymbol{w}_i\}_1^n$, $\mathbf{S}' = \{\boldsymbol{w}'_j\}_1^m$ and generalized stepsize $1/\beta$,
$\boldsymbol{\sigma} = \frac{1}{m}\mathbf{1_m}$, $\mathbf{T}^{(1)} = \mathbf{1_n}\mathbf{1_m}^\top$
$\mathbf{C}_{ij} = c(\boldsymbol{z}_i, \boldsymbol{z}'_j)$, $\mathbf{A}_{ij} = \mathrm{e}^{-\frac{\mathbf{C}_{ij}}{\beta}}$
for $t = 1, 2, 3 \ldots$ **do**
  $\mathbf{Q} = \mathbf{A} \odot \mathbf{T}^{(t)}$ // $\odot$ is Hadamard product
  for $k = 1, \ldots K$ **do** // $K = 1$ in practice
    $\boldsymbol{\delta} = \frac{1}{n\mathbf{Q}\boldsymbol{\sigma}}$, $\boldsymbol{\sigma} = \frac{1}{m\mathbf{Q}^\top\boldsymbol{\delta}}$
  end for
  $\mathbf{T}^{(t+1)} = \mathrm{diag}(\boldsymbol{\delta})\mathbf{Q}\mathrm{diag}(\boldsymbol{\sigma})$
end for
Return T

3 Proposed Method

3.1 Model framework overview

To capture both the topological information (network structure $\mathcal{E}$) and the semantic information (text content $\mathcal{T}$) in the textual network embedding, we explicitly model two types of embeddings for each node $v \in \mathcal{V}$: (i) the topological embedding $\boldsymbol{z}_u^t$, and (ii) the semantic embedding $\boldsymbol{z}_u^s$. The final embedding is constructed by concatenating the topological and semantic embeddings, *i.e.*, $\boldsymbol{z}_u = [\boldsymbol{z}_u^t; \boldsymbol{z}_u^s]$. We consider the topological embedding $\boldsymbol{z}^t$ as a static property of the node, fixed regardless of the context. On the other hand, the semantic embedding $\boldsymbol{z}^s$ dynamically depends on the context, which is the focus of this study.

Motivated by the work of (Tu et al., 2017), we consider the following probabilistic objective to train the network embeddings:

$$\ell(\Theta) = \mathbb{E}_{e\sim\mathcal{E}}\left\{\ell(e;\Theta)\right\}, \quad (2)$$

where $e = (u, v)$ represents sampled edges from the network and $\Theta = \{\mathbf{Z}, \theta\}$ is the collection of model parameters. The edge loss $\ell(e;\Theta)$ is given by the cross entropy

$$\ell(e_{uv};\Theta) = -w_{uv}\log p_\Theta(u|v), \quad (3)$$

where $p_\Theta(u|v)$ denotes the conditional likelihood of observing a (weighted) link between nodes u and v, with the latter serving as the context. More specifically,

$$p_\Theta(u|v) = \langle \boldsymbol{z}_u, \boldsymbol{z}_v\rangle - \log(Z), \quad (4)$$

where $Z = \sum_{u'\in\mathcal{V}} \exp(\langle \boldsymbol{z}_{u'}, \boldsymbol{z}_v\rangle)$ is the normalizing constant and $\langle\cdot,\cdot\rangle$ is an inner product operation, to be defined momentarily. Note here we have suppressed the dependency on Θ to simplify notation.

To capture both the topological and semantic information, along with their interactions, we propose to use the following decomposition for our inner product term:

$$\langle \boldsymbol{z}_u, \boldsymbol{z}_v \rangle = \underbrace{\langle \boldsymbol{z}_u^t, \boldsymbol{z}_v^t \rangle_{tt}}_{\text{topology}} + \underbrace{\langle \boldsymbol{z}_u^s, \boldsymbol{z}_v^s \rangle_{ss}}_{\text{semantic}} + \underbrace{\langle \boldsymbol{z}_u^t, \boldsymbol{z}_v^s \rangle_{ts} + \langle \boldsymbol{z}_u^s, \boldsymbol{z}_v^t \rangle_{st}}_{\text{interaction}} \tag{5}$$

Here we use $\langle \boldsymbol{z}_u^a, \boldsymbol{z}_v^b \rangle_{ab}, a, b \in \{s, t\}$ to denote the inner product evaluation between the two feature embeddings $\boldsymbol{z}_u^a$ and $\boldsymbol{z}_v^b$, which can be defined by a semi-positive-definite kernel function $\kappa_{ab}(\boldsymbol{z}_u^a, \boldsymbol{z}_v^b)$ (Alvarez et al., 2012), *e.g.*, Euclidean kernel, Gaussian RBF, IMQ kernel, *etc.* Note that for $a \neq b$, $\boldsymbol{z}_u^a$ and $\boldsymbol{z}_v^b$ do not reside on the same feature space. As such, embeddings are first mapped to the same feature space for inner product evaluation. In this study, we use the Euclidean kernel

$$\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle_{\mathcal{X}} = \boldsymbol{x}_1^T \boldsymbol{x}_2$$

for inner product evaluation with $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X} \subseteq \mathbb{R}^d$, and linear mapping

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{X}\mathcal{Y}} = \langle \boldsymbol{x}, \mathbf{A}\boldsymbol{y} \rangle_{\mathcal{X}}, \text{ where } \mathbf{A} \in \mathbb{R}^{d \times d'}$$

for feature space realignment with $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d, \boldsymbol{y} \in \mathcal{Y} \subseteq \mathbb{R}^{d'}$. Here $\mathbf{A}$ is a trainable parameter, and throughout this paper we omit the bias terms in linear maps to avoid notational clutter.

Note that our solution differs from existing network-embedding models in that: (i) our objective is a principled likelihood loss, while prior works heuristically combine the losses of four different models (Tu et al., 2017), which may fail to capture the non-trivial interactions between the fixed and dynamic embeddings; and (ii) we present a formal derivation of network embedding in a reproducing kernel Hilbert space.

Negative sampling. Direct optimization of (3) requires summing over all nodes in the network, which can be computationally infeasible for large-scale networks. To alleviate this issue, we consider other more computationally efficient surrogate objectives. In particular, we adopt the negative sampling approach (Mikolov et al., 2013), which replaces the bottleneck Softmax with a more tractable approximation given by

$$\log p(v|u) \approx \log \sigma(\langle \boldsymbol{z}_u, \boldsymbol{z}_v \rangle) + \sum_{j=1}^{K} \mathbb{E}_{v_k \sim p_n}[\log \sigma(-\langle \boldsymbol{z}_u, \boldsymbol{z}_{v_k} \rangle)], \tag{6}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function, and $p_n(v)$ is a noise distribution over the nodes. Negative sampling can be considered as a special variant of noise contrastive estimation (Gutmann and Hyvärinen, 2010), which seeks to recover the ground-truth likelihood by contrasting data samples with noise samples, thereby bypassing the need to compute the normalizing constant. As the number of noise samples K goes to infinity, this approximation becomes exact[1] (Goldberg and Levy, 2014). Following the practice of Mikolov et al. (2013), we set our noise distribution to $p_n(v) \propto d_v^{\frac{3}{4}}$, where d_v denotes the out-degree of node v.

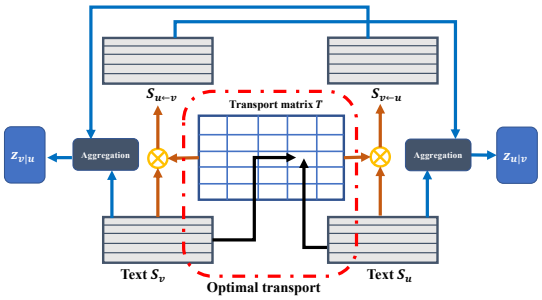

Figure 1: Schematic of the proposed mutual attention mechanism. In this setup, bag-of-words feature matchings are explicitly abstracted to infer the relationship between vertices.

Context matching. We argue that a key to the context-aware network embedding is the design of an effective attention mechanism, which cross-matches the relevant content between the node's associated text and the context. Over-simplified dot-product attention limits the potential of existing textual network embedding schemes. In the following sections, we present two novel, efficient attention designs that fulfill the desiderata listed in our Introduction. Our discussion follows the setup used in CANE (Tu et al., 2017) and WANE (Shen et al., 2018), where the text from the interacting node is used as the context. Generalization to other forms of context is straightforward.

3.2 Optimal-transport-based matching

We first consider reformulating content matching as an optimal transport problem, and then repurpose the transport plan as our attention score to aggregate context-dependent information. More specifically, we see a node's text and context as two (discrete) distributions over the content space. Related content will be matched in the sense that they yield a higher weight in the optimal transport plan π^*. The following two properties make the optimal transport plan more appealing for use as attention score. (i) Sparsity: when solved exactly, π^* is a sparse matrix with at most $(2m-1)$ non-zero elements, where m is the number of

[1] This is a non-trivial result, for completeness we provide an intuitive justification in Supplementary Material.

contents (Brualdi et al. (1991), §8.1.3); (ii) Self-normalized: row-sum and column-sum equal the respective marginal distributions.

Implementation-wise, we first feed embedded text sequence S_u and context sequence S_v into our OT solver to compute the OT plan,

$$\mathbf{T}_{uv} = \text{SolveOT}(S_u, S_v) \in \mathbb{R}^{n_u \times n_v}. \quad (7)$$

Note that here we treat pre-embedded sequence S_u as n_u point masses in the feature space, each with weight $1/n_u$, and similarly for S_v. Next we "transport" the semantic content from context S_v according to the estimated OT plan with matrix multiplication

$$S_{u \leftarrow v} = \mathbf{T}_{uv} S_v \in \mathbb{R}^{n_u \times p}, \quad (8)$$

where we have treated S_v as a $\mathbb{R}^{n_v \times p}$ matrix. Intuitively, this operation *aligns* the context with the target text sequence via averaging the context semantic embeddings with respect to the OT plan for each content element in S_u. To finalize the contextualized embedding, we *aggregate* the information from both S_u and the aligned $S_{u \leftarrow v}$ with an operator F_{agg},

$$\boldsymbol{z}_{u|v} = F_{\text{agg}}(S_u, S_{u \leftarrow v}) \in \mathbb{R}^{d \times 1}. \quad (9)$$

In this case, we practice the following simple aggregation strategy: first concatenate S_u and the aligned $S_{u \leftarrow v}$ along the feature dimension, and then take max-pooling along the temporal dimension to reduce the feature vector into a $2p$ vector, followed by a linear mapping to project the embedding vector to the desired dimensionality.

3.3 Attention parsing

Direct application of attention scores based on a low-level similarity-based matching criteria (*e.g.*, dot-product attention) can be problematic in a number of ways: (i) low-level attention scores can be noisy (*i.e.*, spurious matchings), and (ii) similarity-matching does not allow relational inference. To better understand these points, consider the following cases. For (i), if the sequence embeddings used do not explicitly address the syntactic structure of the text, a relatively dense attention score matrix can be expected. For (ii), consider the case when the context is a query, and the matching appears as a cue in the node's text data; then the information needed is actually in the vicinity rather than the exact matching location (*e.g.*, shifted a few steps ahead). Inspired by the work of Wang et al. (2018), we propose a new mechanism called *attention parsing* to address the aforementioned issues.

As the name suggests, attention parsing recalibrates the raw low-level attention scores to better integrate the information. To this end, we conceptually treat the raw attention matrix $\mathbf{T}_{\text{raw}}$ as a two-dimensional image and apply convolutional filters to it:

$$\mathbf{H} = \text{ReLU}(\text{Conv2d}(\mathbf{T}_{\text{raw}}, \mathbf{W}_F)) \in \mathbb{R}^{n_u \times n_v \times c}, \quad (10)$$

where $\mathbf{W}_F \in \mathbb{R}^{h \times w \times c}$ denotes the filter banks with h, w and c respectively as window sizes and channel number. We can stack more convolutional layers, break sequence embedding dimensions to allow multi-group (channel) low-level attention as input, or introduce more-sophisticated model architectures (*e.g.*, ResNet (He et al., 2016), Transformer (Vaswani et al., 2017), *etc.*) to enhance our model. For now, we focus on the simplest model described above, for the sake of demonstration.

With $\mathbf{H} \in \mathbb{R}^{n_u \times n_v \times c}$ as the high-level representation of attention, our next step is to reduce it to a weight vector to align information from the context S_v. We apply a max-pooling operation with respect to the context dimension, followed by a linear map to get the logits $\boldsymbol{h} \in \mathbb{R}^{n_u \times 1}$ of the weights

$$\boldsymbol{h} = \text{MaxPool}(\mathbf{H}, \text{column}) \cdot \mathbf{B}, \quad (11)$$

where $\mathbf{B} \in \mathbb{R}^{c \times 1}$ is the projection matrix. Then the parsed attention weight $\boldsymbol{w}$ is obtained by

$$\boldsymbol{w} = \text{Softmax}(\boldsymbol{h}) \in \mathbb{R}^{n_u \times 1}, \quad (12)$$

which is used to compute the aligned context embedding

$$\boldsymbol{s}_{u \leftarrow v} = \boldsymbol{w}^T S_v \in \mathbb{R}^{1 \times p}. \quad (13)$$

Note that here we compute a globally aligned context embedding vector $\boldsymbol{s}_{u \leftarrow v}$, rather than one for each location in S_u as described in the last section ($S_{u \leftarrow v}$). In the subsequent aggregation operation, $\boldsymbol{s}_{u \leftarrow v}$ is broadcasted to all the locations in S_u. We call this *global alignment*, to distinguish it from the *local alignment* strategy described in the last section. Both alignment strategies have their respective merits, and in practice they can be directly combined to produce the final context-aware embedding.

4 Related Work

Network embedding models. Prior network embedding solutions can be broadly classified into two categories: (i) *topology embedding*, which

only uses the link information; and (ii) *fused embedding*, which also exploits side information associated with the nodes. Methods from the first category focus on encoding high-order network interactions in a scalable fashion, such as LINE (Tang et al., 2015), DeepWalk (Perozzi et al., 2014). However, models based on topological embeddings alone often ignore rich heterogeneous information associated with the vertices. Therefore, the second type of model tries to incorporate text information to improve network embeddings. For instance, TADW (Yang et al., 2015), CENE (Sun et al., 2016), CANE (Tu et al., 2017), WANE (Shen et al., 2018), and DMTE (Zhang et al., 2018b).

Optimal Transport in NLP. OT has found increasing application recently in NLP research. It has been successfully applied in many tasks, such as topic modeling (Kusner et al., 2015), text generation (Chen et al., 2018), sequence-to-sequence learning (Chen et al., 2019), and word-embedding alignment (Alvarez-Melis and Jaakkola, 2018). Our model is fundamentally different from these existing OT-based NLP models in terms of how OT is used: these models all seek to minimize OT distance to match sequence distributions, while our model used the OT plan as an attention mechanism to integrate context-dependent information.

Attention models. Attention was originally proposed in QA systems (Weston et al., 2015) to overcome the limitations of the sequential computation associated with recurrent models (Hochreiter et al., 2001). Recent developments, such as the Transformer model (Vaswani et al., 2017), have popularized attention as an integral part of compelling sequence models. While simple attention mechanisms can already improve model performance (Bahdanau et al., 2015; Luong et al., 2015), significant gains can be expected from more delicate designs (Yang et al., 2016; Li et al., 2015). Our treatment of attention is inspired by the LEAM model (Wang et al., 2018), which significantly improves mutual attention in a computationally efficient way.

5 Experiments

5.1 Experimental setup

Datasets and tasks. We consider three benchmark datasets: (i) *Cora*[2], a paper citation network with text information, built by McCallum et al. (2000). We prune the dataset so that it only has papers on the topic of machine learning. (ii) *Hepth*[3], a paper citation network from Arxiv on high energy physics theory, with paper abstracts as text information. (iii) *Zhihu*, a Q&A network dataset constructed by (Tu et al., 2017), which has 10,000 active users with text descriptions and their collaboration links. Summary statistics of these three datasets are summarized in Table 1. Preprocessing protocols from prior studies are used for data preparation (Shen et al., 2018; Zhang et al., 2018b; Tu et al., 2017).

	Cora	*Hepth*	*Zhihu*
#vertices	2,227	1,038	10,000
#edges	5,214	1,990	43,894
#avg text len	90	54	190
#labels	7	NA	NA

Table 1: Dataset statistics.

For quantitative evaluation, we tested our model on the following tasks: (a) *Link prediction*, where we deliberately mask out a portion of the edges to see if the embedding learned from the remaining edges can be used to accurately predict the missing edges. (b) *Multi-label node classification*, where we use the learned embedding to predict the labels associated with each node. Note that the label information is not used in our embedding. We also carried out ablation study to identify the gains. In addition to the quantitative results, we also visualized the embedding and the attention matrices to qualitatively verify our hypotheses.

Evaluation metrics. For the link prediction task, we adopt the area under the curve (AUC) score to evaluate the performance, AUC is employed to measure the probability that vertices in existing edges are more similar than those in the nonexistent edge. For each training ratio, the experiment is executed 10 times and the mean AUC scores are reported, where higher AUC indicates better performance. For multi-label classification, we evaluate the performance with Macro-F1 scores. The experiment for each training ratio is also executed 10 times and the average Macro-F1 scores are reported, where a higher value indicates better performance.

Baselines. To demonstrate the effectiveness of the proposed solutions, we evaluated our model along with the following strong baselines. (i) *Topology only embeddings*: MMB (Airoldi et al.,

[2] https://people.cs.umass.edu/~mccallum/data.html

[3] https://snap.stanford.edu/data/cit-HepTh.html

	Cora					Hepth				
%Training Edges	**15%**	**35%**	**55%**	**75%**	**95%**	**15%**	**35%**	**55%**	**75%**	**95%**
MMB	54.7	59.5	64.9	71.1	75.9	54.6	57.3	66.2	73.6	80.3
node2vec	55.9	66.1	78.7	85.9	88.2	57.1	69.9	84.3	88.4	89.2
LINE	55.0	66.4	77.6	85.6	89.3	53.7	66.5	78.5	87.5	87.6
DeepWalk	56.0	70.2	80.1	85.3	90.3	55.2	70.0	81.3	87.6	88.0
Naive combination	72.7	84.9	88.7	92.4	94.0	78.7	84.7	88.7	92.1	92.7
TADW	86.6	90.2	90.0	91.0	92.7	87.0	91.8	91.1	93.5	91.7
CENE	72.1	84.6	89.4	93.9	95.5	86.2	89.8	92.3	93.2	93.2
CANE	86.8	92.2	94.6	95.6	97.7	90.0	92.0	94.2	95.4	96.3
DMTE	91.3	93.7	96.0	97.4	98.8	NA	NA	NA	NA	NA
WANE	91.7	94.1	96.2	97.5	99.1	92.3	95.7	97.5	97.7	98.7
GANE-OT	92.0	95.7	97.3	98.6	99.2	93.4	97.0	97.9	98.2	98.8
GANE-AP	**94.0**	**97.2**	**98.0**	**98.8**	**99.3**	**93.8**	**97.3**	**98.1**	**98.4**	**98.9**

Table 2: AUC scores for link prediction on the *Cora* and *Hepth* dataset.

%Training Edges	**15%**	**25%**	**35%**	**45%**	**55%**	**65%**	**75%**	**85%**	**95%**
DeepWalk	56.6	58.1	60.1	60.0	61.8	61.9	63.3	63.7	67.8
node2vec	54.2	57.1	57.3	58.3	58.7	62.5	66.2	67.6	68.5
LINE	52.3	55.9	59.9	60.9	64.3	66.0	67.7	69.3	71.1
MMB	51.0	51.5	53.7	58.6	61.6	66.1	68.8	68.9	72.4
Naive combination	55.1	56.7	58.9	62.6	64.4	68.7	68.9	69.0	71.5
TADW	52.3	54.2	55.6	57.3	60.8	62.4	65.2	63.8	69.0
CENE	56.2	57.4	60.3	63.0	66.3	66.0	70.2	69.8	73.8
CANE	56.8	59.3	62.9	64.5	68.9	70.4	71.4	73.6	75.4
DMTE	58.4	63.2	67.5	71.6	74.0	76.7	78.5	79.8	81.5
WANE	58.7	63.5	68.3	71.9	74.9	77.0	79.7	80.0	82.6
GANE-OT	61.6	66.4	70.8	73.0	77.3	80.6	80.4	81.8	83.2
GANE-AP	**64.6**	**69.4**	**72.8**	**74.2**	**79.1**	**82.6**	**81.8**	**83.0**	**84.3**

Table 3: AUC scores for link prediction on the *Zhihu* dataset.

2008), DeepWalk (Perozzi et al., 2014), LINE (Tang et al., 2015), Node2vec (Grover and Leskovec, 2016). (ii) *Joint embedding of topology & text*: Naive combination, TADW (Yang et al., 2015), CENE (Sun et al., 2016), CANE (Tu et al., 2017), WANE (Shen et al., 2018), DMTE (Zhang et al., 2018b). A brief summary of these competing models is provided in the Supplementary Material (SM).

5.2 Results

We consider two variants of our model, denoted as GANE-OT and GANE-AP. GANE-OT employs the most basic OT-based attention model, specifically, global word-by-word alignment model; while GANE-AP additionally uses a one-layer convolutional neural network for the attention parsing. Detailed experimental setups are described in the SM.

Link prediction. Tables 2 and 3 summarize the results from the link-prediction experiments on all three datasets, where a different ratio of edges are used for training. Results from models other than GANE are collected from Tu et al. (2017), Shen et al. (2018) and Zhang et al. (2018b). We have also repeated these experiments on our own, and the results are consistent with the ones reported. Note that Zhang et al. (2018b) did not report results on DMTE. Both GANE variants consistently outperform competing solutions. In the low-training-sample regime our solutions lead by a large margin, and the performance gap closes as the number of training samples increases. This indicates that our OT-based mutual attention framework can yield more informative textual representations than other methods. Note that GANE-AP delivers better results compared with GANE-OT, suggesting the attention parsing mechanism can further improve the low-level mutual attention matrix. More results on *Cora* and *Hepth* are provided in the SM.

Multi-label Node Classification. To further evaluate the effectiveness of our model, we consider multi-label vertex classification. Following the setup described in (Tu et al., 2017), we first computed all context-aware embeddings. Then we averaged over each node's context-aware embeddings with all other connected nodes, to obtain a global embedding for each node, *i.e.*, $\boldsymbol{z}_u = \frac{1}{d_u}\sum_v \boldsymbol{z}_{u|v}$, where d_u denotes the degree of node u. A linear SVM is employed, instead of a

%training labels	10%	30%	50%	70%
LINE	53.9	56.7	58.8	60.1
TADW	71.0	71.4	75.9	77.2
CANE	81.6	82.8	85.2	86.3
DMTE	81.8	83.9	86.3	87.9
WANE	81.9	83.9	86.4	88.1
GANE-OT	82.0	84.1	86.6	88.3
GANE-AP	**82.3**	**84.2**	**86.7**	**88.5**

Table 4: Test Macro-F1 scores for multi-label node classification on *Cora*.

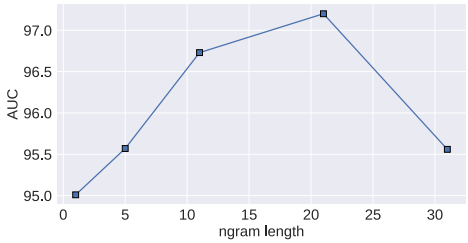

Figure 2: n-gram length *VS* AUC on Cora.

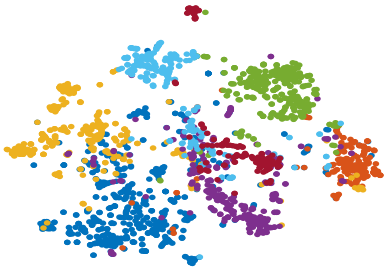

Figure 3: t-SNE visualization on Cora dataset.

sophisticated deep classifier, to predict the label attribute of a node. We randomly sample a portion of labeled vertices with embeddings $(10\%, 30\%, 50\%, 70\%)$ to train the classifier, using the rest of the nodes to evaluate prediction accuracy. We compare our results with those from other state-of-the-art models in Table 4. The GANE models delivered better results compared with their counterparts, lending strong evidence that the OT attention and attention parsing mechanism promise to capture more meaningful representations.

Ablation study. We further explore the effect of n-gram length in our model (*i.e.*, the filter size for the covolutional layers used by the attention parsing module). In Figure 2 we plot the AUC scores for link prediction on the Cora dataset against varying n-gram length. The performance peaked around length 20, then starts to drop, indicating a moderate attention span is more preferable. Similar results are observed on other datasets (results not shown). Experimental details on the ablation study can be found in the SM.

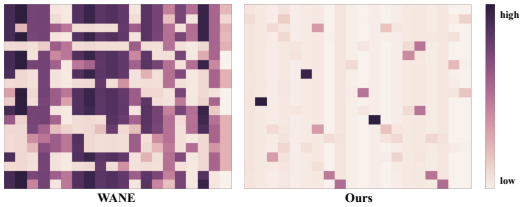

Figure 4: Mutual attention between two nodes in Cora. Left: WANE attention. Right: OT attention (ours).

5.3 Qualitative Analysis

Embedding visualization. We employed t-SNE (Maaten and Hinton, 2008) to project the network embeddings for the *Cora* dataset in a two-dimensional space using GANE-OT, with each node color coded according to its label. As shown in Figure 3, papers clustered together belong to the same category, with the clusters well-separated from each other in the network embedding space. Note that our network embeddings are trained without any label information. Together with the label classification results, this implies our model is capable of extracting meaningful information from both context and network topological.

Attention matrix comparison. To verify that our OT-based attention mechanism indeed produces sparse attention scores, we visualized the OT attention matrices and compared them with those simarlity-based attention matrices (*e.g.*, WANE). Figure 4 plots one typical example. Our OT solver returns a sparse attention matrix, while dot-product-based WANE attention is effectively dense. This underscores the effectiveness of OT-based attention in terms of noise suppression.

6 Conclusion

We have proposed a novel and principled mutual-attention framework based on optimal transport (OT). Compared with existing solutions, the attention mechanisms employed by our GANE model enjoys the following benefits: (*i*) it is naturally sparse and self-normalized, (*ii*) it is a global sequence matching scheme, and (*iii*) it can capture long-term interactions between two sentences. These claims are supported by experimental evidence from link prediction and multi-label vertex classification. Looking forward, our attention mechanism can also be applied to tasks such as relational networks (Santoro et al., 2017), natural language inference (MacCartney and Manning, 2009), and QA systems (Zhou et al., 2015).

Acknowledgments

This research was supported in part by DARPA, DOE, NIH, ONR and NSF.

References

Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. 2008. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014.

Mauricio A Alvarez, Lorenzo Rosasco, Neil D Lawrence, et al. 2012. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266.

David Alvarez-Melis and Tommi S Jaakkola. 2018. Gromov-wasserstein alignment of word embedding spaces. *arXiv preprint arXiv:1809.00013*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Richard A Brualdi, Herbert J Ryser, et al. 1991. *Combinatorial matrix theory*, volume 39. Springer.

Liqun Chen, Shuyang Dai, Chenyang Tao, Dinghan Shen, Zhe Gan, Haichao Zhang, Yizhe Zhang, and Lawrence Carin. 2018. Adversarial text generation via feature-mover's distance. In *NIPS*.

Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Improving sequence-to-sequence learning via optimal transport. In *ICLR*.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of Wasserstein GANs. In *NIPS*.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. In *ACL*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *ICML*.

Tuan MV Le and Hady W Lauw. 2014. Probabilistic latent document network embedding. In *2014 IEEE International Conference on Data Mining*, pages 270–279. IEEE.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *ACL*.

Xiaodong Liu, Kevin Duh, and Jianfeng Gao. 2018. Stochastic answer networks for natural language inference. *arXiv preprint arXiv:1804.07888*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv:1508.04025*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Bill MacCartney and Christopher D Manning. 2009. *Natural language inference*.

Prodromos Malakasiotis and Ion Androutsopoulos. 2007. Learning textual entailment using svms and string similarity measures. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 42–47.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *ICML*.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Vlad Niculae and Mathieu Blondel. 2017. A regularized framework for sparse and structured neural attention. In *NIPS*.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*.

Gabriel Peyré, Marco Cuturi, et al. 2017. Computational optimal transport. Technical report.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. In *NIPS*.

Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.

Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Improved semantic-aware network embedding with fine-grained word alignment. In *EMNLP*.

Xiaofei Sun, Jiang Guo, Xiao Ding, and Ting Liu. 2016. A general framework for content-enhanced network representation learning. In *arXiv preprint arXiv:1610.02906*.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale information network embedding. In *WWW*.

Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *KDD*.

Cunchao Tu, Han Liu, Zhiyuan Liu, and Maosong Sun. 2017. CANE: Context-aware network embedding for relation modeling. In *ACL*.

Cunchao Tu, Weicheng Zhang, Zhiyuan Liu, Maosong Sun, et al. 2016. Max-margin deepwalk: Discriminative learning of network representation. In *IJCAI*.

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

Cédric Villani. 2008. *Optimal Transport: Old and New*. Springer.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint embedding of words and labels for text classification. In *ACL*.

Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. 2017. Community preserving network embedding. In *AAAI*.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *ICLR*.

Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. 2018. A fast proximal point method for Wasserstein distance. In *arXiv:1802.04307*.

Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Chang. 2015. Network representation learning with rich text information. In *IJCAI*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL*.

Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. 2018a. Network representation learning: A survey. *IEEE transactions on Big Data*.

Xinyuan Zhang, Yitong Li, Dinghan Shen, and Lawrence Carin. 2018b. Diffusion maps for textual network embedding. In *NIPS*.

Guangyou Zhou, Tingting He, Jun Zhao, and Po Hu. 2015. Learning continuous word embedding with metadata for question retrieval in community question answering. In *ACL*.