# *Better Together:* Combining Language and Social Interactions into a Shared Representation

**Yi-Yu Lai*** **Chang Li*** **Dan Goldwasser** **Jennifer Neville**
Department of Computer Science,
Purdue University, West Lafayette, Indiana
{lai49, li1873, dgoldwas, neville}@purdue.edu
* authors contributed equally

## Abstract

Despite the clear inter-dependency between analyzing the interactions in social networks, and analyzing the natural language content of these interactions, these aspects are typically studied independently. In this paper we present a first step towards finding a joint representation, by embedding the two aspects into a single vector space. We show that the new representation can help improve performance in two social relations prediction tasks.

## 1 Introduction

The interactions, social bonds and relationships between people have been studied extensively in recent years. Broadly speaking, these works fall into two, almost completely disconnected, camps. The first, focusing on *social network analysis*, looks at the network structure and information flow on it as means of inferring knowledge about the network. For example, works by (Leskovec et al., 2008; Kumar et al., 2010) model the evolution of network structure over time, and works such as (Xiang et al., 2010; Leskovec et al., 2010) use the network structure to predict properties of links (e.g., strength, sign).

The second camp, focusing on *natural language analysis*, looks into tasks such as extracting social relationships from narrative text (Elson et al., 2010; Van De Camp and van den Bosch, 2011; Agarwal et al., 2012) and analyzing the contents of the information flowing through the network. For example, works by (Danescu-Niculescu-Mizil et al., 2012; Hassan et al., 2012; Filippova, 2012; Volkova et al., 2014; West et al., 2014; Rahimi et al., 2015; Volkova et al., 2015) extract attributes of, and social relationships between, nodes by analyzing the textual communication between them. Other works (Krishnan and Eisenstein, 2014; Sap et al., 2014) use the social network to inform language analysis.

Both perspectives on social network analysis resulted in a wide range of successful applications; however, they neglect to model the interactions between the social and linguistic representations and how they complement one another. One of the few exceptions was discussed in (West et al., 2014), which inferred sentiment links between nodes in a social network by jointly modeling the local output probabilities of a sentiment analyzer looking at the textual interactions between the nodes and the global network structure. While resulting in better performance, inference is done over two *independent* representations, one capturing the linguistic information, and the other, the network structure.

Instead, in this paper we take the first step towards finding a joint representation over both linguistic and network information, rather than treating the two independently. We follow the intuition that interactions in a social network can be fully captured only by taking into account both types of information together. To achieve this goal, we embed the input social graph into a dense, continuous, low-dimensional vector space, capturing both network and linguistic similarities between nodes. Word (Mikolov et al., 2013; Pennington et al., 2014) and Network (Perozzi et al., 2014; Tang et al., 2015) embedding approaches that were recently proposed, aim to combat a similar problem in their respective domains–data sparsity. Both follow a similar approach–embed discrete objects (words or nodes in



*Proceedings of the 2016 Workshop on Graph-based Methods for Natural Language Processing, NAACL-HLT 2016*, pages 29–33,
San Diego, California, June 17, 2016. 

the graph) into a continuous vector representation, based on the context they appear in. Our approach aims to map both social and linguistic information into the same vector space, rather than embedding the two aspects into two independent spaces. The social graph, originally containing only quantitative properties of the interaction between nodes (e.g., number of messages exchanged between nodes), is extended to capture the contents of these interactions, by computing the textual similarity between the messages generated by each one of the nodes. The computed similarity is used to weight the edges between adjacent nodes. We embed the modified graph nodes into a vector space, using the embedding technique described by (Tang et al., 2015).

We evaluate the joint representation by using it in two social relationship prediction tasks and comparing it to several different word-based and network based representations. Our experiments show the advantage of the joint representation.

## 2 Problem Formulation

Our primary assumption is there is a latent space that influences the interactions we observe among people. Thus the goal of our work is to learn this latent representation from the observed data. We describe the data and problem more specifically below.

### 2.1 Data

We assume that the data comprise a graph $G = (V, E)$, where nodes $V$ correspond to entities (e.g., users in a social network), and the edges $E$ correspond to textual interactions among the entities (e.g., emails, messages). Each edge $e_{ij}^t \in E$, which refers to a message sent from node $v_i$ to node $v_j$ at time $t$, has an associated document representation $d_{ij}^t$. We refer to the set of messages (documents) between nodes $v_i$ and $v_j$ as $\mathbf{E}_{ij} := \{e_{ij}^t\}_t$ ($\mathbf{D}_{ij}$ respectively). Moreover, we refer to the set of messages (documents) sent by a node $v_i$ to any other node as $\mathbf{E}_i := \{e_{ij}^t\}_{t,j}$ ($\mathbf{D}_i$ respectively).

### 2.2 Motivation

Given this type of network data, the goal is to discover the underlying latent representation of the nodes. Our assumption is that the entities are embedded in a latent space that influences the frequency and nature of their communication. We assume that each node has a location in space (e.g., in $\mathbf{R}^2$, the location of $v_i$ is $\mathbf{v}_i := (x_i, y_i)$), and that pairwise node distances (e.g., $d(\mathbf{v}_i, \mathbf{v}_j)$) affect the likelihood of communication and the content of that communication. More specifically, we assume that nearby nodes are more likely to communicate, and talk about similar things. Thus, we assume the latent space embedding represents entities' interests and pairs of entities with similar interests are more likely to interact. These assumptions are motivated by online communities where users exhibit *homophily* (McPherson et al., 2001), i.e., users with common interests are more likely to form relationships.

### 2.3 Problem Definition

Given the framework and assumptions described above, we can now state the problem definition for the work in this paper. Assume as input, a multigraph $G = (V, E)$ with messages between nodes in the graph that can be modeled as a set of documents. The goal is to learn an embedding of the nodes $V$ in $\mathbf{R}^{\mathbf{k}}$ such that the representation reflects both the frequency and content of the messages.

To achieve this we will consider several different ways to compute the embedding based on optimizing (1) network connectivity, (2) message content, and (3) connectivity and content. Our conjecture is that jointly considering connectivity and content will produce an embedding that is more robust to noisy interaction data. Strong (but introverted) friends may talk less frequently but share more common interests, compared to gregarious users who talk more frequently but with many (weak) friends.

Since there is no ground truth for quantitative evaluation, it is difficult to directly evaluate the quality of a learned embedding. Thus, we evaluate our methods indirectly via related classification tasks. In this work, we will use the learned embeddings in two link-based prediction tasks, where we differentiate (1) strong vs. weak(er) friendships, and (2) employees working in the same vs. different groups.

## 3 Method

The input for our task is the text-enriched network graph $G$. The goal is to compute a node embedding from $G$ and then use the embedding to generate features for pairs of nodes, which can then be used for a prediction task. The process follows these steps.

• **Textual-Similarity (TS) Infused Social Graph:** Construct graph weights $\mathbf{W}_{ij}$ based on the text in $G$, according to **(1)** a *Node* or *Edge* view of the documents, and **(2)** using *Topic Model* or *Word Embedding* to represent the content.

• **Node Embedding:** Construct an embedding function $V \rightarrow R^k$, mapping the (weighted) graph nodes into a $R^k$ dimensional space. We used the LINE method (Tang et al., 2015). We omit the details due to space restrictions.

• **Feature Extraction:** Construct a feature set for each node pair, using 9 similarity measures between the nodes' $k$-dimensional vector representations from the embedding. We experiment with additional features extracted directly.

### 3.1 Creating the TS-Infused Social Graph

The TS-Infused social graph captures the interaction between node pairs by modifying the strength of the edge connecting them according to the similarity of the text generated by each one of the nodes. We identify several design decisions for the process.

**Node vs. Edge** Each edge $e_{ij} \in G$ is associated with textual content $d_{ij}$. We can characterize the textual content from the point of view of the *node* by aggregating the text over all its outgoing edges (i.e., $\mathbf{D}_i$), or alternatively, we can characterize the textual content from the edge point of view, by only looking at the text contained in the relevant outgoing *edges* (i.e., $\mathbf{D}_{ij}$).

**Representing Textual Content using Topic Models vs. Word Embedding** Before we compute the similarity between the content of two parties, we need a vector space model to represent the textual information (the set of documents $\mathbf{D}_i$, or $\mathbf{D}_{ij}$). One obvious method for this is topic modeling, in which the textual content is represented as a topic distribution. In this approach, we learn a topic model over the set of documents, and then represent each document via a set of topic weights ($\mathbf{T}_i$ or $\mathbf{T}_{ij}$). An alternative approach is using word embedding, which has been proved effective as a word representation. In this approach, we represent each document as the average of the embedding over the words in the document ($\mathbf{WE}_i$ or $\mathbf{WE}_{ij}$). Given the distributional representation of text associated with a node/edge, we assign a weight ($w_{ij}$) for each edge ($e_{ij}$) as the cosine similarity between vector representation of contents from neighboring nodes (e.g., $d(\mathbf{T}_i, \mathbf{T}_j)$ or $d(\mathbf{T}_{ij}, \mathbf{T}_{ji})$, where $d$ is cosine similarity).

### 3.2 Node Embedding

We utilize the LINE embedding technique (Tang et al., 2015), aimed at preserving network structures when generating node embedding for social and information networks. LINE uses edge weights corresponding to the number of interactions between each pair of nodes. This only makes use of the network structure, without taking advantage of the text in the network. We modify the embedding procedure by using the edges weights $\mathbf{W}_{ij}$ described above (i.e., based on the cosine similarity of the text between nodes $i, j$) and use the LINE algorithm to compute a $k$-dimensional embedding of the nodes in $G$.

### 3.3 Feature Extraction

**Distance-based Features** Given a node pair represented by their k-dimensional node embedding, we generate features for the pair according to nine similarity measures. The nine measures used by us are Bray-Curtis distance, Canberra distance, Chebyshev distance, City Block (Manhattan) distance, Correlation distance, Cosine distance, Minkowski distance, Euclidean and squared Euclidean distance.

**Additional Features** Besides the distance-based features, we can also add one or more other basic features related to nodes in the network. These include the following: (1) Network: The number of interactions between two nodes, e.g. number of emails sent and received. (2) Unigram: The unigram feature vector for text sent for each node. (3) Word embedding features: The word embedding vector for text sent for each node. Again we use the average of word embedding to represent documents.

## 4 Experiments

**Purdue Facebook Network** We analyzed the public Purdue Facebook network data from March 2007 to March 2008, which includes 3 million post activities. Members can set friends as top (close) friends to get the timely notifications without a confirmation by the other. We collected 945 mutually top friend pairs for two users who set each other as top friend and 34633 one-way top friend pairs if there is only one of them set the other as top friend. The dataset will be referred as "Facebook" in this

| Dataset | Embedding | $\emptyset$ | N | W | WE | N+W | N+WE | N+W+WE |
|---|---|---|---|---|---|---|---|---|
| Facebook (F1) | no $GE$ | 49.45 | 77.80 | 75.04 | 75.09 | 81.23 | 79.14 | 79.26 |
| | $GE$ | 53.36 | 78.54 | 75.82 | 75.68 | 82.09 | 79.11 | 78.39 |
| | $GE^N_{TM}$ | 61.58 | 80.16 | 76.33 | 76.31 | 82.69 | 78.72 | 79.68 |
| | $GE^E_{TM}$ | 78.36 | 80.51 | 77.51 | 77.51 | 80.23 | 79.82 | 80.38 |
| | $GE^N_{WE}$ | 59.81 | 79.98 | 75.44 | 76.82 | 81.19 | 79.62 | 79.15 |
| | $GE^E_{WE}$ | 80.66 | 82.49 | 81.96 | 81.07 | 83.31 | 82.06 | **83.72** |
| Avocado (F1) | no $GE$ | 49.69 | 40.91 | 55.03 | 57.89 | 53.33 | 56.64 | 55.36 |
| | $GE$ | 65.75 | 66.15 | 65.77 | 66.57 | 66.24 | 66.99 | 66.53 |
| | $GE^N_{TM}$ | 66.66 | 66.65 | 66.49 | **67.28** | 66.83 | 67.12 | 66.84 |
| | $GE^E_{TM}$ | 63.09 | 64.67 | 64.79 | 64.09 | 64.80 | 65.05 | 64.49 |
| | $GE^N_{WE}$ | 61.33 | 64.51 | 64.60 | 63.83 | 65.46 | 64.67 | 65.39 |
| | $GE^E_{WE}$ | 52.03 | 55.11 | 56.94 | 56.03 | 57.40 | 57.18 | 58.48 |

Table 1: Prediction results over the two datasets. We report the F1 score.

paper. We evaluated our method by a classification task of the two different social relationships.

**Avocado Email Collection** This collection consists of 279 e-mail accounts, from which we extracted the job titles and departments of 136 accounts. We divided these accounts into three groups, according to their positions in the company, namely executives, engineering department, and business department. We will refer to this dataset as "Avocado" in this paper. The task is defined as predicting whether two accounts belong to the same group. In order to make use of text signal. We will only consider account pairs that have correspondence between each other. There are 2232 positive and 1409 negative examples in this dataset.

### 4.1 Result

Using the features defined in the previous section, we train Logistic Regression classifier via scikit-learn in Python. We show the ten-fold cross-validation performance of our features on Facebook and Avocado datasets in Table 1. It represents the results of five different approaches to generate node embedding, and with or without adding additional features. $GE$ is the original embedding method, the superscript $N$ or $E$ represent the Node or Edge, and the subscript $TM$ or $WE$ represent Topic Model or Word Embedding used to construct the TS-Infused graph respectively. The $N$, $W$, $WE$ in the columns indicate the Network, Unigram and Word embedding as additional features. $\emptyset$ with no $GE$ shows the result of random generated embedding. In this paper, we use LINE as the node embedding method, Latent Dirichlet Allocation (Blei et al., 2003) for topic modeling with ten topics and Skip-Gram for word embedding. The regularization parameters are optimized. Since the Facebook and Avocado datasets are unbalanced, we randomly downsamples the majority class to equate the size of both classes. The results show in the Table are the average scores of ten different random downsampling.

For Facebook dataset, the results of all embeddings constructed by TS-Infused social graph outperforms the original embedding $GE$. It shows the joint representation over linguistic information and network structure is more effective than only considering one of them independently. The results on Avocado dataset also confirm the advantage of shared representation. $GE^N_{TM}$ significantly outperforms other text-based or network-based methods. The performance of aggregating text sent by a node is better than only looking at text on one outgoing edge, which is opposite to the results on Facebook dataset. This could be resulted from the difference between two prediction tasks. In the Facebook dataset, we try to distinguish strong and weak(er) friendship, in which case the messages they sent to each other are most indicative. While when we predict whether two persons belong to the same group inside a company, the interaction they had with their colleagues would tell us more about the community they are from.

## Acknowledgments

We thank the reviewers for their insightful comments. This research is supported by NSF under contract number IIS-1149789.

## References


Apoorv Agarwal, Augusto Corvalan, Jacob Jensen, and Owen Rambow. 2012. Social network analysis of alice in wonderland. In *Proceedings of the Workshop on Computational Linguistics for Literature*.

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

Cristian Danescu-Niculescu-Mizil, Lillian Lee, Bo Pang, and Jon Kleinberg. 2012. Echoes of power: Language effects and power differences in social interaction. In *Proceedings of WWW*, pages 699–708.

David K Elson, Nicholas Dames, and Kathleen R McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of ACL*, pages 138–147. Association for Computational Linguistics.

Katja Filippova. 2012. User demographics and language in an implicit social network. In *Proceedings of EMNLP-CoNLL*, pages 1478–1488. Association for Computational Linguistics.

Ahmed Hassan, Amjad Abu-Jbara, and Dragomir Radev. 2012. Detecting subgroups in online discussions by modeling positive and negative relations among participants. In *Proceedings of EMNLP-CoNLL*, pages 59–70. Association for Computational Linguistics.

Vinodh Krishnan and Jacob Eisenstein. 2014. " you're mr. lebowski, i'm the dude": Inducing address term formality in signed social networks. *arXiv preprint arXiv:1411.4351*.

Ravi Kumar, Jasmine Novak, and Andrew Tomkins. 2010. Structure and evolution of online social networks. In *Link mining: models, algorithms, and applications*, pages 337–357. Springer.

Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. 2008. Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470. ACM.

Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650. ACM.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of ACM SIGKDD*, pages 701–710. ACM.

Afshin Rahimi, Duy Vu, Trevor Cohn, and Timothy Baldwin. 2015. Exploiting text and network context for geolocation of social media users.

Maarten Sap, Gregory Park, Johannes C Eichstaedt, Margaret L Kern, David Stillwell, Michal Kosinski, Lyle H Ungar, and H Andrew Schwartz. 2014. Developing age and gender predictive lexica over social media.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of WWW*, pages 1067–1077. International World Wide Web Conferences Steering Committee.

Matje Van De Camp and Antal van den Bosch. 2011. A link to the past: constructing historical social networks. In *Proceedings of the Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 61–69. Association for Computational Linguistics.

Svitlana Volkova, Glen Coppersmith, and Benjamin Van Durme. 2014. Inferring user political preferences from streaming communications. In *Proceedings of ACL*, pages 186–196.

Svitlana Volkova, Yoram Bachrach, Michael Armstrong, and Vijay Sharma. 2015. Inferring latent user properties from texts published in social media.

Robert West, Hristo S Paskov, Jure Leskovec, and Christopher Potts. 2014. Exploiting social network structure for person-to-person sentiment analysis. *TACL*.

Rongjing Xiang, Jennifer Neville, and Monica Rogati. 2010. Modeling relationship strength in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 981–990. ACM.