Text Mining
(Part II: Embedding, Information Extraction & Text Cube)

JIAWEI HAN
COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

MARCH 30, 2017
Outline

- Learning Embeddings in Networks and Text
  - LINE: Large-scale Information Network Embedding
- Biological Relationship Discovery with Network Embedding
- LAKI: Representing Documents via Latent Keyphrase Inference
- MetaPAD: Meta Pattern Discovery from Massive Text Corpora
- TextCube, EventCube and CaseOLAP
- Summary
Information Network Analysis

- A lot of interesting problems
  - Node classification
  - Link prediction
  - Visualization

- Input of most algorithms
  - Low-dimensional feature vector of every vertex

- How to get the feature vectors?
  - From the context information of vertices 😞
  - From the network structure 😊

- How?
LINE and Its Hypothesis

- **LINE paper:** J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “LINE: Large-scale information network embedding”, WWW'15
- Nodes with strong ties turn to be similar
  - 1\(^{st}\) order similarity
- Nodes share many neighbors turn to be similar
  - 2\(^{nd}\) order similarity
- **Well-learnt embedding should preserve both 1\(^{st}\) order and 2\(^{nd}\) order similarity**

Nodes 6 & 7: high 1\(^{st}\) order similarity
Nodes 5 & 6: high 2\(^{nd}\) order similarity
LINE: 1st Order Similarity

- Try to capture local pairwise proximity between the nodes (vertices) in the network
- Vertices that are connected with larger weight are more similar
- Embedding vector for each vertex: $\vec{v}$
- To model the 1st order proximity, for each undirected edge $(i, j)$, we define the joint probability between vertex $v_i$ and $v_j$ as
  $$p_1(v_i, v_j) = \frac{1}{1 + \exp(-\vec{u}_i^T \cdot \vec{u}_j)}$$
  where $\vec{u}_i \in \mathbb{R}^d$ is the low-dimensional vector representation of vertex $v_i$
- **Joint probability** between vertex $v_i$ and $v_j$: $P(v_i, v_j) = \frac{\exp(\vec{v}_i \cdot \vec{v}_j)}{\sum_{i,j} \exp(\vec{v}_i \cdot \vec{v}_j)}$
- **Empirical joint probability** between vertex $v_i$ and $v_j$: $\hat{P}(v_i, v_j) = \frac{w_{ij}}{\sum_{i,j} w_{ij}}$
- Objective: Minimize the KL-divergence of two probability distributions
  $$O_1 = KL(\hat{P} || P) = -\sum_{i,j} w_{ij} \log P(v_i, v_j)$$
KL Divergence: Comparing Two Probability Distributions

- The Kullback-Leibler (KL) divergence: Measure the difference between two probability distributions over the same variable \( x \)
- From information theory, closely related to relative entropy, information divergence, and information for discrimination
- \( D_{KL}(p(x) \parallel q(x)) \): divergence of \( q(x) \) from \( p(x) \), measuring the information lost when \( q(x) \) is used to approximate \( p(x) \)
- Discrete form:
  \[
  D_{KL}(p(x) \parallel q(x)) = \sum_{x \in X} p(x) \ln \frac{p(x)}{q(x)}
  \]
- The KL divergence measures the expected number of extra bits required to code samples from \( p(x) \) (“true” distribution) when using a code based on \( q(x) \), which represents a theory, model, description, or approximation of \( p(x) \)
- Its continuous form:
  \[
  D_{KL}(p(x) \parallel q(x)) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} \, dx
  \]
- Not a distance measure, not a metric: asymmetric, not satisfy triangular inequality
LINE: 2nd Order Similarity

- 2nd order similarity: proximity between two vertices \((u, v)\) is the similarity between their neighborhood network structures.
- Each vertex is treated as a specific context, and vertices with similar distributions over the contexts are assumed to be similar.
- Introduce two vectors for each vertex:
  - Embedding vector: \(\vec{v}\), and context vector: \(\vec{v}'\)
- The probability of “context” \(v_j\) generated by vertex \(v_i\) and its empirical probability as,
  \[
P_i(v_j | v_i) = \frac{\exp(\vec{v}_i \cdot \vec{v}_j')}{\sum_{i,j} \exp(\vec{v}_i \cdot \vec{v}_j')}
  \]
  \[
  \hat{P}_i(v_j | v_i) = \frac{w_{ij}}{\sum_j w_{ij}}
  \]
- Minimizing the objective function:
  \[
  O_2 = \sum_i d_i KL(\hat{P}_i || P_i) = - \sum_{i,j} w_{ij} \log P_i(v_j | v_i)
  \]
Combining LINE 1st and LINE 2nd Order Similarity

- Concatenate the embedding learnt by LINE 1st and that learnt by LINE 2nd
- 1st order similarity and 2nd order similarity are complementary
- Jointly optimize $O_1$ and $O_2$
- Difficult to determine the weights of the two objective functions

**Optimization**

- Directly optimizing the objective function is difficult
- Softmax units
- Negative sampling:
  - Approximation algorithm
  - Can retain the quality of the embedding

- $O_1 = \sum_{i,j} \hat{P}_{ij} \log \sigma(\bar{v}_i \cdot \bar{v}_j) + K \sum_{i,j} \hat{P}_i \hat{P}_j \log \sigma(-\bar{v}_i \cdot \bar{v}_j)$
- $O_2 = \sum_{i,j} \hat{P}_{ij} \log \sigma(\bar{v}_i \cdot \bar{v}_j') + K \sum_{i,j} \hat{P}_i \hat{P}_j \log \sigma(-\bar{v}_i \cdot \bar{v}_j')$
Algorithm LINE and Some Practical Issues

- **LINE:**
  - Train the LINE model which preserves the *first-order* proximity and *second-order* proximity separately
  - Then concatenate the embeddings trained by the two methods for each vertex
  - A more principled way (future work)
  - Jointly train the two objective functions
- **Implementation consideration:**
  - Information networks can be very sparse: Reconstruct them to make them denser
  - Strategy: Add links between every vertex and its high order neighbors
  - Add second order neighbors
  - \[ w_{ij} = \sum_{k \in N(i)} w_{ik} \frac{w_{kj}}{d_k} \]
**Previous Work**

- **Graph Embedding**
  - Construct an affinity matrix & compute eigenvectors of the affinity matrix
  - Weakness: High time complexity and space complexity, and
    - Only preserve 1st order similarity

- **Matrix Factorization**
  - Objective: $\sum_{i,j}(x_i^T x_j - W_{ij})^2$  Optimization: SGD (Stochastic Gradient Descent Alg.)
  - Weakness: Can’t converge when the range of weights is very large
    - Only preserve 1st order similarity

- **Deep Walk (Perozzi, et al., KDD’14)**
  - Sample a chain of vertex by truncated random walk
  - Treat the chain as a sentence and run word2vec on this sentence
  - Weakness: Only preserve 2nd order similarity
Experiment Setup

- **Dataset**

<table>
<thead>
<tr>
<th>Name</th>
<th>Language Network</th>
<th>Social Network</th>
<th>Citation Network</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WIKIPEDIA</strong></td>
<td>undirected,weighted</td>
<td>undirected,binary</td>
<td>directed,weighted</td>
</tr>
<tr>
<td><strong>Flickr</strong></td>
<td>1,985,098</td>
<td>1,715,256</td>
<td>524,061</td>
</tr>
<tr>
<td><strong>YouTube</strong></td>
<td>1,000,924,086</td>
<td>22,613,981</td>
<td>20,580,238</td>
</tr>
<tr>
<td><strong>Avg. degree</strong></td>
<td>504.22</td>
<td>26.37</td>
<td>78.54</td>
</tr>
<tr>
<td><strong>#Labels</strong></td>
<td>7</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td><strong>#train</strong></td>
<td>70,000</td>
<td>75,958</td>
<td>20,684</td>
</tr>
</tbody>
</table>

- **Task**
  - Word analogy: Evaluated on Accuracy
  - Document classification: Evaluated on Macro-F1 Micro-F1
  - Vertex classification: Evaluated on Macro-F1 Micro-F1

- **Result visualization**

  (a) GF  
  (b) DeepWalk  
  (c) LINE(2nd)
Results: Language Networks

- Word Analogy
  - GF (Graph Factorization) 
    Ahmed et al., WWW2013)
- Document Classification

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Semantic (%)</th>
<th>Syntactic (%)</th>
<th>Overall (%)</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>GF</td>
<td>61.38</td>
<td>44.08</td>
<td>51.93</td>
<td>2.96h</td>
</tr>
<tr>
<td>DeepWalk</td>
<td>50.79</td>
<td>37.70</td>
<td>43.65</td>
<td>16.64h</td>
</tr>
<tr>
<td>SkipGram</td>
<td>69.14</td>
<td>57.94</td>
<td>63.02</td>
<td>2.82h</td>
</tr>
<tr>
<td>LINE-SGD(1st)</td>
<td>9.72</td>
<td>7.48</td>
<td>8.50</td>
<td>3.83h</td>
</tr>
<tr>
<td>LINE-SGD(2nd)</td>
<td>20.42</td>
<td>9.56</td>
<td>14.49</td>
<td>3.94h</td>
</tr>
<tr>
<td>LINE(1st)</td>
<td>58.08</td>
<td>49.42</td>
<td>53.35</td>
<td>2.44h</td>
</tr>
<tr>
<td>LINE(2nd)</td>
<td><strong>73.79</strong></td>
<td><strong>59.72</strong></td>
<td><strong>66.10</strong></td>
<td><strong>2.55h</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric</th>
<th>Algorithm</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro-F1</td>
<td>GF</td>
<td>79.63</td>
<td>80.51</td>
<td>80.94</td>
<td>81.18</td>
<td>81.38</td>
<td>81.54</td>
<td>81.63</td>
<td>81.71</td>
<td>81.78</td>
</tr>
<tr>
<td></td>
<td>DeepWalk</td>
<td>78.89</td>
<td>79.92</td>
<td>80.41</td>
<td>80.69</td>
<td>80.92</td>
<td>81.08</td>
<td>81.21</td>
<td>81.35</td>
<td>81.42</td>
</tr>
<tr>
<td></td>
<td>SkipGram</td>
<td>79.84</td>
<td>80.82</td>
<td>81.28</td>
<td>81.57</td>
<td>81.71</td>
<td>81.87</td>
<td>81.98</td>
<td>82.05</td>
<td>82.09</td>
</tr>
<tr>
<td></td>
<td>LINE-SGD(1st)</td>
<td>76.03</td>
<td>77.05</td>
<td>77.57</td>
<td>77.85</td>
<td>78.08</td>
<td>78.25</td>
<td>78.39</td>
<td>78.44</td>
<td>78.49</td>
</tr>
<tr>
<td></td>
<td>LINE-SGD(2nd)</td>
<td>74.68</td>
<td>76.53</td>
<td>77.54</td>
<td>78.18</td>
<td>78.63</td>
<td>78.96</td>
<td>79.19</td>
<td>79.40</td>
<td>79.57</td>
</tr>
<tr>
<td></td>
<td>LINE(1st)</td>
<td>79.67</td>
<td>80.55</td>
<td>80.94</td>
<td>81.24</td>
<td>81.40</td>
<td>81.52</td>
<td>81.61</td>
<td>81.69</td>
<td>81.67</td>
</tr>
<tr>
<td></td>
<td>LINE(2nd)</td>
<td>79.93</td>
<td>80.90</td>
<td>81.31</td>
<td>81.63</td>
<td>81.80</td>
<td>81.91</td>
<td>82.00</td>
<td>82.11</td>
<td>82.17</td>
</tr>
<tr>
<td></td>
<td>LINE(1st+2nd)</td>
<td><strong>81.04</strong></td>
<td><strong>82.08</strong></td>
<td><strong>82.58</strong></td>
<td><strong>82.93</strong></td>
<td><strong>83.16</strong></td>
<td><strong>83.37</strong></td>
<td><strong>83.52</strong></td>
<td><strong>83.63</strong></td>
<td><strong>83.74</strong></td>
</tr>
</tbody>
</table>

| Macro-F1 | GF        | 79.49 | 80.39 | 80.82 | 81.08 | 81.26 | 81.40 | 81.52 | 81.61 | 81.68 |
|          | DeepWalk  | 78.78 | 79.78 | 80.30 | 80.56 | 80.82 | 80.97 | 81.11 | 81.24 | 81.32 |
|          | SkipGram  | 79.74 | 80.71 | 81.15 | 81.46 | 81.63 | 81.78 | 81.88 | 81.98 | 82.01 |
|          | LINE-SGD(1st) | 75.85 | 76.90 | 77.40 | 77.71 | 77.94 | 78.12 | 78.24 | 78.29 | 78.36 |
|          | LINE-SGD(2nd) | 74.70 | 76.45 | 77.43 | 78.09 | 78.53 | 78.83 | 79.08 | 79.29 | 79.46 |
|          | LINE(1st) | 79.54 | 80.44 | 80.82 | 81.13 | 81.29 | 81.43 | 81.51 | 81.60 | 81.59 |
|          | LINE(2nd) | 79.82 | 80.81 | 81.22 | 81.52 | 81.71 | 81.82 | 81.92 | 82.00 | 82.07 |
|          | LINE(1st+2nd) | **80.94** | **81.99** | **82.49** | **82.83** | **83.07** | **83.29** | **83.42** | **83.55** | **83.66** |

Significantly outperforms GF at the: ** 0.01 and * 0.05 level, paired t-test.
## Results: Social Networks

- **Flickr dataset**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Algorithm</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro-F1</td>
<td>GF</td>
<td>53.23</td>
<td>53.08</td>
<td>53.98</td>
<td>54.14</td>
<td>54.32</td>
<td>54.38</td>
<td>54.43</td>
<td>54.50</td>
<td>54.48</td>
</tr>
<tr>
<td></td>
<td>DeepWalk</td>
<td>60.38</td>
<td>60.77</td>
<td>60.90</td>
<td>61.05</td>
<td>61.13</td>
<td>61.18</td>
<td>61.19</td>
<td>61.29</td>
<td>61.22</td>
</tr>
<tr>
<td></td>
<td>DeepWalk (256dim)</td>
<td>60.41</td>
<td>61.09</td>
<td>61.35</td>
<td>61.52</td>
<td>61.69</td>
<td>61.76</td>
<td>61.80</td>
<td>61.91</td>
<td>61.83</td>
</tr>
<tr>
<td></td>
<td>LINE (1st)</td>
<td>63.27</td>
<td>63.69</td>
<td>63.82</td>
<td>63.92</td>
<td>63.96</td>
<td>64.03</td>
<td>64.06</td>
<td>64.17</td>
<td>64.10</td>
</tr>
<tr>
<td></td>
<td>LINE (2nd)</td>
<td>62.83</td>
<td>63.41</td>
<td>63.34</td>
<td>64.14</td>
<td>63.55</td>
<td>63.55</td>
<td>63.59</td>
<td>63.66</td>
<td>63.69</td>
</tr>
<tr>
<td></td>
<td>LINE (1st+2nd)</td>
<td><strong>63.20</strong></td>
<td><strong>63.97</strong></td>
<td><strong>64.25</strong></td>
<td><strong>64.39</strong></td>
<td><strong>64.65</strong></td>
<td><strong>64.55</strong></td>
<td><strong>64.61</strong></td>
<td><strong>64.75</strong></td>
<td><strong>64.74</strong></td>
</tr>
</tbody>
</table>

- **Macro-F1**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Algorithm</th>
<th>1%</th>
<th>2%</th>
<th>3%</th>
<th>4%</th>
<th>5%</th>
<th>6%</th>
<th>7%</th>
<th>8%</th>
<th>9%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro-F1</td>
<td>GF</td>
<td>25.43</td>
<td>26.16</td>
<td>26.60</td>
<td>26.91</td>
<td>27.32</td>
<td>27.61</td>
<td>27.88</td>
<td>28.13</td>
<td>28.30</td>
<td>28.51</td>
</tr>
<tr>
<td></td>
<td>DeepWalk</td>
<td>39.88</td>
<td>41.78</td>
<td>42.78</td>
<td>43.55</td>
<td>43.96</td>
<td>44.31</td>
<td>44.61</td>
<td>44.89</td>
<td>45.06</td>
<td>45.23</td>
</tr>
<tr>
<td></td>
<td>DeepWalk (256dim)</td>
<td>39.94</td>
<td>42.17</td>
<td>43.19</td>
<td>44.05</td>
<td>44.47</td>
<td>44.84</td>
<td>45.17</td>
<td>45.43</td>
<td>45.65</td>
<td>45.81</td>
</tr>
<tr>
<td></td>
<td>LINE (1st)</td>
<td>35.43</td>
<td>38.08</td>
<td>39.33</td>
<td>40.21</td>
<td>40.77</td>
<td>41.24</td>
<td>41.53</td>
<td>41.89</td>
<td>42.07</td>
<td>42.21</td>
</tr>
<tr>
<td></td>
<td>LINE (2nd)</td>
<td>32.98</td>
<td>36.70</td>
<td>38.93</td>
<td>40.26</td>
<td>41.08</td>
<td>41.79</td>
<td>42.28</td>
<td>42.70</td>
<td>43.04</td>
<td>43.35</td>
</tr>
<tr>
<td></td>
<td>LINE (1st+2nd)</td>
<td><strong>39.01</strong></td>
<td><strong>41.89</strong></td>
<td><strong>43.14</strong></td>
<td><strong>44.04</strong></td>
<td><strong>44.62</strong></td>
<td><strong>45.06</strong></td>
<td><strong>45.34</strong></td>
<td><strong>45.69</strong></td>
<td><strong>45.91</strong></td>
<td><strong>46.08</strong></td>
</tr>
</tbody>
</table>

Significantly outperforms DeepWalk at the: **0.01 and * 0.05 level, paired t-test.**

- **Youtube dataset**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Algorithm</th>
<th>1%</th>
<th>2%</th>
<th>3%</th>
<th>4%</th>
<th>5%</th>
<th>6%</th>
<th>7%</th>
<th>8%</th>
<th>9%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macro-F1</td>
<td>GF</td>
<td>7.38</td>
<td>8.44</td>
<td>9.35</td>
<td>9.80</td>
<td>10.38</td>
<td>10.79</td>
<td>11.21</td>
<td>11.55</td>
<td>11.81</td>
<td>12.08</td>
</tr>
<tr>
<td></td>
<td>DeepWalk</td>
<td>28.39</td>
<td>30.96</td>
<td>32.28</td>
<td>33.43</td>
<td>33.92</td>
<td>34.32</td>
<td>34.83</td>
<td>35.27</td>
<td>35.54</td>
<td>35.86</td>
</tr>
<tr>
<td></td>
<td>DeepWalk (256dim)</td>
<td>28.95</td>
<td>31.79</td>
<td>33.16</td>
<td>34.42</td>
<td>34.93</td>
<td>35.44</td>
<td>35.99</td>
<td>36.41</td>
<td>36.78</td>
<td>37.11</td>
</tr>
<tr>
<td></td>
<td>LINE (1st)</td>
<td>28.74</td>
<td>31.24</td>
<td>32.26</td>
<td>33.05</td>
<td>33.29</td>
<td>33.60</td>
<td>33.88</td>
<td>34.18</td>
<td>34.33</td>
<td>34.47</td>
</tr>
<tr>
<td></td>
<td>LINE (2nd)</td>
<td>17.06</td>
<td>21.73</td>
<td>25.28</td>
<td>27.36</td>
<td>28.50</td>
<td>29.59</td>
<td>30.43</td>
<td>31.14</td>
<td>31.81</td>
<td>32.32</td>
</tr>
<tr>
<td></td>
<td>LINE (1st+2nd)</td>
<td><strong>29.85</strong></td>
<td><strong>31.93</strong></td>
<td><strong>33.96</strong></td>
<td><strong>35.48</strong></td>
<td><strong>36.25</strong></td>
<td><strong>36.90</strong></td>
<td><strong>37.48</strong></td>
<td><strong>38.10</strong></td>
<td><strong>38.4</strong></td>
<td><strong>38.85</strong></td>
</tr>
</tbody>
</table>

Significantly outperforms DeepWalk at the: **0.01 and * 0.05 level, paired t-test.**
Outline

- Learning Embeddings in Networks and Text
  - LINE: Large-scale Information Network Embedding
  - Biological Relationship Discovery with Network Embedding
- LAKI: Representing Documents via Latent Keyphrase Inference
- MetaPAD: Meta Pattern Discovery from Massive Text Corpora
- TextCube, EventCube and CaseOLAP
- Summary
Biological Relationship Extraction

- Problem description: Automatic extraction of relationships between different biological entities from biological research papers
  - Examples
    - Gene – Disease; Drug - Disease; Drug - Pathway; Drug - Target gene
  - Challenges
    - Entity detection
      - Most biological entities consist of several words
        - E.g., Non-small Cell Lung Cancer, Acute Myeloid Leukemia
    - Sparsity
      - Most biological entities co-occur only a few times in research papers
      - Most relationships are not explicitly described in papers
    - Few Labeled data
Overview of Embedding-Based Methods

- Goal: Find low-dimensional vector representation for entities and similar entities should have similar representations
- Data: Raw text or co-occurrence network
- Hypothesis: Entities which share many neighbors turn to be similar to each other
- Observation: Most entities co-occur only few times in papers, but they always share many neighbors
- Examples on word co-occurrence network
  - Word similarity: San Francisco
  - Word relation: $A: B \approx C: ?$
  - $\text{Argmax}_X \text{Sim}(\vec{B} - \vec{A} + \vec{C}, \vec{X})$
Framework

- Construct a biological entity list
  - SegPhrase
  - Biological vocabulary
- Detect and extract biological entities from biological text (research papers), using
  - SegPhrase
  - Maximum Matching
- Construct a co-occurrence network between words and biological entities
- Learn embedding vector for each entity by using a network embedding technique LINE (i.e., Run LINE-2nd to learn entity embeddings)
  - Entities share many neighbors in the co-occurrence network tend to be similar
- Given seed entity pair (a, b) and a query entity x, return an entity y according to their embedding vectors, so that the relation between x and y is similar to the relation between a and b.
Key Property to Learn Embedding & Experiments

- Key Property to Learn Embedding
  - The lines between genes and diseases are parallel
  - Given a seed pair \((A, B)\) and a query \(X\), we can find an entity \(Y\) which satisfies
    \[(A, B) \approx (X, Y)\]
    \[Y = \text{Argmax}/\text{sim}(B - A^\top + X, Y)\]

- Experimental Settings
  - Sample 10% Pubmed abstracts
  - Detect phrases by using a 200K phrase list
  - Build a co-occurrence network for all words and phrases
  - Learn entity embedding from the co-occurrence network
Experimental Results: Find Related Entities

- **t-cell**

<table>
<thead>
<tr>
<th>Word</th>
<th>Cosine distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>b_cell</td>
<td>0.899536</td>
</tr>
<tr>
<td>t-lymphocyte</td>
<td>0.897891</td>
</tr>
<tr>
<td>nk-cell</td>
<td>0.874473</td>
</tr>
<tr>
<td>b-cell</td>
<td>0.871003</td>
</tr>
<tr>
<td>natural_killer_cell</td>
<td>0.857540</td>
</tr>
<tr>
<td>t_lymphocyte</td>
<td>0.855409</td>
</tr>
<tr>
<td>nk_cell</td>
<td>0.838348</td>
</tr>
<tr>
<td>b_lymphocyte</td>
<td>0.826705</td>
</tr>
<tr>
<td>cd8</td>
<td>0.825255</td>
</tr>
<tr>
<td>b_lymphocyte</td>
<td>0.818053</td>
</tr>
</tbody>
</table>

- **Leukemia**

<table>
<thead>
<tr>
<th>Word</th>
<th>Cosine distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>leukaemia</td>
<td>0.968356</td>
</tr>
<tr>
<td>lymphoblastic</td>
<td>0.907420</td>
</tr>
<tr>
<td>leukemias</td>
<td>0.869021</td>
</tr>
<tr>
<td>myelocytic</td>
<td>0.860447</td>
</tr>
<tr>
<td>acute_myelogenous_leukemia</td>
<td>0.845294</td>
</tr>
<tr>
<td>myelogenous</td>
<td>0.843615</td>
</tr>
<tr>
<td>ph1+</td>
<td>0.831842</td>
</tr>
<tr>
<td>philadelphia-positive</td>
<td>0.830873</td>
</tr>
<tr>
<td>myelomonocytic</td>
<td>0.830090</td>
</tr>
<tr>
<td>acute_myeloid_leukemia</td>
<td>0.830058</td>
</tr>
</tbody>
</table>

- **Doxorubicin**

<table>
<thead>
<tr>
<th>Word</th>
<th>Cosine distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adriamycin</td>
<td>0.921346</td>
</tr>
<tr>
<td>Paclitaxel</td>
<td>0.906449</td>
</tr>
<tr>
<td>Epirubicin</td>
<td>0.904076</td>
</tr>
<tr>
<td>Etoposide</td>
<td>0.889664</td>
</tr>
<tr>
<td>Dox</td>
<td>0.887675</td>
</tr>
<tr>
<td>Daunorubicin</td>
<td>0.883601</td>
</tr>
<tr>
<td>Mitoxantrone</td>
<td>0.883223</td>
</tr>
<tr>
<td>Cisplatin</td>
<td>0.881845</td>
</tr>
<tr>
<td>Cddp</td>
<td>0.872368</td>
</tr>
<tr>
<td>Docetaxel</td>
<td>0.847687</td>
</tr>
</tbody>
</table>

- **Tumor Suppressor Gene**

<table>
<thead>
<tr>
<th>Word</th>
<th>Cosine distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>TumorSuppressor</td>
<td>0.900334</td>
</tr>
<tr>
<td>TumorSuppressor</td>
<td>0.883250</td>
</tr>
<tr>
<td>Suppressor_gene</td>
<td>0.853865</td>
</tr>
<tr>
<td>Tsgs</td>
<td>0.812559</td>
</tr>
<tr>
<td>Dbcrr1</td>
<td>0.810981</td>
</tr>
<tr>
<td>Mmac1</td>
<td>0.809297</td>
</tr>
<tr>
<td>Mts1</td>
<td>0.806708</td>
</tr>
<tr>
<td>TumourSuppressor</td>
<td>0.803866</td>
</tr>
<tr>
<td>Smarca4</td>
<td>0.789171</td>
</tr>
<tr>
<td>Cdkn2a</td>
<td>0.788514</td>
</tr>
<tr>
<td>Relation</td>
<td>Seed Pair</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Gene-Disease</td>
<td>Breast Cancer, BRCA1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Drug-Disease</td>
<td>BRCA1, Breast Cancer</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Leukemia, Doxorubicin</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Doxorubicin, Leukemia</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Relation: Drug-Target Gene

- Very difficult as there are too many genes
- Use (Camptothecin: top1) as seed pair
- Metric: rank of the first-hit target gene
- Compare with a paper published on CELL

<table>
<thead>
<tr>
<th>Drug</th>
<th>Embedding</th>
<th>CELL Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>cycloheximide</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>doxorubicin hydrochloride</td>
<td>1998</td>
<td>1000</td>
</tr>
<tr>
<td>etoposide</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>geldanamycin</td>
<td>426</td>
<td>5</td>
</tr>
<tr>
<td>methotrexate</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>monastrol</td>
<td>1116</td>
<td>1000</td>
</tr>
<tr>
<td>rapamycin</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>trichostatin</td>
<td>7</td>
<td>100</td>
</tr>
</tbody>
</table>
Future Work

- Use more context information
  - By using topic assignment of words as extra features, the performance on word analogy task is improved
- Use biological networks
  - Well constructed with little noise
- Exploring multi-typed information networks
  - Integration of expert-provided type information and automated type extraction (e.g., ClusType)
  - Exploring meta-path
- Add a regularization term to the model
Outline

- Learning Embeddings in Networks and Text
  - LINE: Large-scale Information Network Embedding
  - Biological Relationship Discovery with Network Embedding
  - LAKI: Representing Documents via Latent Keyphrase Inference
  - MetaPAD: Meta Pattern Discovery from Massive Text Corpora
  - TextCube, EventCube and CaseOLAP
- Summary
Jialu Liu, Xiang Ren, Jingbo Shang, Taylor Cassidy, Clare Voss and Jiawei Han, "Representing Documents via Latent Keyphrase Inference", WWW'16

Document Representation

A document can be represented by
- A set of works, topics, KB concepts, Keyphrases, ...

Words:
dbscan, methods, clustering, process, ...

Topics:
[k-means, clustering, clusters, dbscan, ...]
[clusters, density, dbscan, clustering, ...]
[machine, learning, knowledge, mining, ...]

Knowledge base concepts:
data mining: /m/0blvg
clustering analysis: /m/031f5p
dbscan: /m/03cg_k1

Document keyphrase:
dbscan: [dbscan, density, clustering, ...]
clustering: [clustering, clusters, partition, ...]
data mining: [data mining, knowledge, ...]
Document Representation: Traditional Methods

- Bag-of-Words or Bag-of-Phrases
  - Cons: Sparse on short texts
- Topic models [LDA]
  - Each **topic** is a distribution over words; each **document** is a mixture of corpus-wide topics
  - Cons: Difficult for human to infer topic semantics

<table>
<thead>
<tr>
<th></th>
<th>doc1</th>
<th>doc2</th>
<th>doc3</th>
</tr>
</thead>
<tbody>
<tr>
<td>like</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>football</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>John</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>likes</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>basketball</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

![Diagram showing document representation](image_url)

**Seeking Life’s Bare (Genetic) Necessities**

COLD SPRING HARBOR, NEW YORK—
Ever wonder why humans need DNA? Despite lack of an obvious function, DNA is essential for life, even in silly organisms that lack a nucleus.

The key to understanding DNA’s role lies in its ability to store genetic information. DNA is a master blueprint that dictates how organisms grow, develop, and function.

In this issue, we explore different approaches to uncovering DNA’s roles. From ancient fossils to modern organisms, each approach provides unique insights into the fundamental nature of life.

In the first study, researchers compared DNA sequences from different species. They discovered that despite vast evolutionary distances, certain DNA motifs are conserved across species, suggesting a common ancestor for life.

In the second study, scientists used computational methods to predict gene functions based on DNA sequence analysis. This approach helped identify new biological pathways in obscure organisms.

Finally, a third study investigated the role of DNA in shaping the physical properties of cells. It revealed that DNA structure can influence cell shape and function, providing new insights into the interplay between genetics and morphology.

These studies highlight the importance of DNA in all forms of life and underscore the need for continued research to unlock its mysteries.

*Image credit: Courtesy of the DNA Research Institute.*

---

28
Document Representation: Concept-based models [ESA]

- Concept-based models [ESA]
- Cons: Low coverage of concepts in human-curated knowledge base

Every Wikipedia article represents a concept

Panthera

From Wikipedia, the free encyclopedia

Panthera is a genus of the family Felidae that contains four well-known living species: the lion, tiger, jaguar, and leopard. The genus comprises about half of the big cats. One meaning of the word panther is to designate any of this family. Only these four cat species have the anatomical changes enabling them to roar. The primary reason for this was assumed to be the incomplete ossification of the hyoid bone. However, new studies show that the ability to roar is due to other morphological features, especially of the larynx. The snow leopard, *Uncia unica*, which is sometimes included within Panthera, does not roar. Although it has an incomplete ossification of the hyoid bone, it lacks the special morphology of the larynx, which is typical for lions, tigers, jaguars and *Leopards*.

Species and subspecies [edit]

<table>
<thead>
<tr>
<th>Kingdom</th>
<th>Animalia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phylum</td>
<td>Chordata</td>
</tr>
<tr>
<td>Class</td>
<td>Aves</td>
</tr>
<tr>
<td>Order</td>
<td>Carnivora</td>
</tr>
<tr>
<td>Family</td>
<td>Felidae</td>
</tr>
<tr>
<td>Genus</td>
<td>Panthera</td>
</tr>
</tbody>
</table>

Article words are associated with the concept (TF.IDF), which help infer concepts from document
Word/document embedding models [word2vec]

Cons: Difficult to explain what each dimension means
Document Representation Using Keyphrases

- Use quality phrases as the entries in the vector and identify document keyphrases (subset of quality phrases) by evaluating relatedness between (doc, quality phrase).

- Unsupervised model

- Challenges
  - Where to get quality phrases from a given corpus?
  - Mining Quality Phrases from Massive Text Corpora [SIGMOD15]
  - How to identify document keyphrases?
  - Can be latent mentions
  - Relatedness scores
  - How to deal with relationship between quality phrases?
Document Representation Using Keyphrases: General Ideas

- How to identify document keyphrases?
  - Powered by Bayesian Inference on “Quality Phrase Silhouette”
  - Quality Phrase Silhouette: Topic centered on quality phrase
    - “Reverse” topic models
    - “Pseudo content” for quality phrase

- How to deal with relationship between quality phrases?
  - Phrases are interconnected as a Directed Acyclic Graph
Framework for Latent Keyphrase Inference (LAKI)

Phrase Mining

- data mining
- text mining
- clustering
- kernel k-means
- dbscan

Offline:

DBSCAN / is / a / method / for / clustering / in / process / of / knowledge discovery. DBSCAN / was / proposed / by …

Segmentation

Online:

Kernel k-means 1
kernel kmeans 1
correlation 0.65
kernel 0.55
rbf kernel 0.5

Dbscan 1
density 0.8
clustering 0.6
dense regions 0.3
shape 0.25

Data mining 1
knowl. discov. 1
kdd 0.67
clustering 0.6
text mining 0.6

Knowledge Discovery
KDD
Dbscan
Clustering
Data
Kernel k-means

Document Keyphrase Inference

Document Representation

Knowledge Discovery
Data
Clustering
Density-based
Clustering
LAKI: Deriving Quality Phrase Silhouette

- Learning Hierarchical Bayesian Network (DAG)

Task 1: Model Learning: Learning link weights
Task 2: Structure Learning: Learning network structure
Deriving Quality Phrase Silhouette Task 1: Model Learning Given Structure

- Use Z to represent K (quality phrases) and T (content units)
- Noisy-OR
  - A parent node is easier to activate its children when the link weight is larger
  - A child node is influenced by all its parents

\[ p(Z_j = 1 | Pa(Z_j)) = 1 - \exp(-W_{0j} - \sum_i W_{ij} \mathbb{1}_{Pa_j^i}) \]

- Maximum Likelihood Estimation
- Training data: Documents
- Expectation-step: For each document, collect sufficient statistics
  - Link firing (Parent, child both being activated) probability
  - Node activation probability
- Maximization-step: Update link weight

Toy example

- Fully observed content units
- Partially observed document keyphrases

\[ L(D) = \sum_{d=1}^{N} \log \sum_{k \in \Omega(d)} p(K = k, T = t^{(d)}) \]
Deriving Quality Phrase Silhouette Task 2: Structure Learning

- Structure Learning
  - Quality phrases are connected to content units
    - Help infer document keyphrases from content units
  - Quality phrases are interconnected
    - Help infer document keyphrases from other keyphrases
- A Heuristic Approach
  - Data-Driven, DAG, similar to ontology
  - Heuristic:
    - Two nodes are connected only
      - Closely Related: word2vec
      - Co-occur frequently
    - Links are always pointing to less frequent nodes
  - Works well in practice
LAKI: Inference

- When do we need inference?
  - Expectation step in model learning
  - New documents

- Why is it slow?
  - NP hard to compute posterior probability for Noisy-Or networks

- Method: Approximate inference instead
  - Pruning irrelevant nodes using an efficient scoring function
  - Gibbs sampling
LAKI: Experiment Setting

- Two text-related tasks to evaluate document representation quality
  - Phrase relatedness
  - Document classification
- Two datasets:
- Methods:
  - **ESA** (Explicit Semantic Analysis)
  - **KBLINK** uses link structure in Wikipedia
  - **BoW** (bag-of-words)
  - **ESA-C**: extends ESA by replacing Wiki with domain corpus
  - **LSA** (Latent Semantic Analysis)
  - **LDA** (Latent Dirichlet Allocation)
  - **Word2Vec** is a neural network computing word embeddings
  - **EKM** uses explicit keyphrase detection

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Docs</th>
<th>#Words</th>
<th>Content type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Academia</td>
<td>0.43M</td>
<td>28M</td>
<td>title &amp; abstract</td>
</tr>
<tr>
<td>Yelp</td>
<td>0.47M</td>
<td>98M</td>
<td>review</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Semantic Space</th>
<th>Input Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESA</td>
<td>KB concepts</td>
<td>KB</td>
</tr>
<tr>
<td>KBLINK</td>
<td>KB concepts</td>
<td>KB</td>
</tr>
<tr>
<td>BoW</td>
<td>Words</td>
<td>-</td>
</tr>
<tr>
<td>ESA-C</td>
<td>Documents</td>
<td>Corpus</td>
</tr>
<tr>
<td>LSA</td>
<td>Topics</td>
<td>Corpus</td>
</tr>
<tr>
<td>LDA</td>
<td>Topics</td>
<td>Corpus</td>
</tr>
<tr>
<td>Word2Vec</td>
<td>-</td>
<td>Corpus</td>
</tr>
<tr>
<td>EKM</td>
<td>Explicit Keyphrases</td>
<td>Corpus</td>
</tr>
<tr>
<td>LAKI</td>
<td>Latent Keyphrases</td>
<td>Corpus</td>
</tr>
</tbody>
</table>
LAKI: Experimental Results

- **Phrase Relatedness Correlation**

<table>
<thead>
<tr>
<th>Method</th>
<th>Academia (w/ phrase)</th>
<th>Yelp (w/ phrase)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESA</td>
<td>37.61 (-)</td>
<td>46.56 (-)</td>
</tr>
<tr>
<td>KBLINK</td>
<td>36.37 (-)</td>
<td>35.94 (-)</td>
</tr>
<tr>
<td>BoW</td>
<td>48.05 (45.60)</td>
<td>51.26 (45.97)</td>
</tr>
<tr>
<td>ESA-C</td>
<td>39.75 (42.20)</td>
<td>49.13 (54.51)</td>
</tr>
<tr>
<td>LSA</td>
<td>72.50 (79.22)</td>
<td>66.55 (78.57)</td>
</tr>
<tr>
<td>LDA</td>
<td>77.27 (80.52)</td>
<td>75.55 (82.65)</td>
</tr>
<tr>
<td>EKM</td>
<td>45.46</td>
<td>40.57</td>
</tr>
<tr>
<td>LAKI</td>
<td><strong>84.42</strong></td>
<td><strong>90.58</strong></td>
</tr>
</tbody>
</table>

- **Document Classification**

<table>
<thead>
<tr>
<th>Method</th>
<th>Academia (w/ phrase)</th>
<th>Yelp (w/ phrase)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESA</td>
<td>0.4320 (-)</td>
<td>0.4567 (-)</td>
</tr>
<tr>
<td>KBLINK</td>
<td>0.1878 (-)</td>
<td>0.4179 (-)</td>
</tr>
<tr>
<td>ESA-C</td>
<td>0.4905 (0.5243)</td>
<td>0.4655 (0.5029)</td>
</tr>
<tr>
<td>LSA</td>
<td>0.5877 (0.6383)</td>
<td>0.6700 (0.7229)</td>
</tr>
<tr>
<td>LDA</td>
<td>0.3610 (0.5391)</td>
<td>0.3928 (0.5405)</td>
</tr>
<tr>
<td>Word2Vec</td>
<td>0.6674 (0.7281)</td>
<td>0.7143 (0.7419)</td>
</tr>
<tr>
<td>LAKI</td>
<td><strong>0.7504</strong></td>
<td><strong>0.7609</strong></td>
</tr>
</tbody>
</table>

- **Time Complexity**
### Case Study

- **Query on phrases**
  - **Academia**
  - **Yelp**

- **Query on short documents (paper titles or sentences)**
  - **Academia**
  - **Yelp**

<table>
<thead>
<tr>
<th>Query</th>
<th>LDA</th>
<th>BOA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Keyphrases</strong></td>
<td>linear discriminant analysis, latent dirichlet allocation, topic models, topic modeling, face recognition, lda, latent dirichlet, generative model, topic, subspace models, ...</td>
<td>boa steakhouse, bank of america, strip steak, agnolotti, credit card, santa monica, restaurants, wells fargo, steakhouse, prime rib, bank, vegas, las vegas, cash, cut, dinner, bank, money, ...</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>LDA topic</td>
<td>BOA steak</td>
</tr>
<tr>
<td><strong>Keyphrases</strong></td>
<td>latent dirichlet allocation, topic, topic models, topic modeling, probabilistic topic models, latent topics, topic discovery, generative model, mixture, text mining, topic distribution, plsi, ...</td>
<td>steak, strip steak, boa steakhouse, steakhouse, ribeye, craft steak, santa monica, medium rare, prime, veggies, entrees, potatoes, french fries, filet mignon, mashed potatoes, texas roadhouse, ...</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>SVM</td>
<td>deep dish pizza</td>
</tr>
<tr>
<td><strong>Keyphrases</strong></td>
<td>support vector machines, svm classifier, multi class, training set, margin, k-nn, classification problems, kernel function, multi class svm, multi class support vector machine, support vector, ...</td>
<td>deep dish pizza, chicago, deep dish, amore taste of chicago, amore, pizza, oreogano, chicago style, chicago style deep dish pizza, thin crust, windy city, slice, pan, oven, pepperoni, hot dog, ...</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>Mining Frequent Patterns without Candidate Generation</td>
<td>I am a huge fan of the All You Can Eat Chinese food buffet.</td>
</tr>
<tr>
<td><strong>Keyphrases</strong></td>
<td>mining frequent patterns, candidate generation, frequent pattern mining, candidate, prune, fp growth, frequent pattern tree, apriori, subtrees, frequent patterns, candidate sets, ...</td>
<td>all you can eat, chinese food, buffet, chinese buffet, dim sum, orange chicken, chinese restaurant, asian food, asian buffet, crab legs, lunch buffet, fan, salad bar, all you can drink, ...</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>Text mining, also referred to as text data mining, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text. High-quality information is typically derived through means such as statistical pattern learning.</td>
<td>It’s the perfect steakhouse for both meat and fish lovers. My table guest was completely delirious about his Kobe Beef and my lobster was perfectly cooked. Good wine list, they have a lovely Sancerre! Professional staff, quick and smooth.</td>
</tr>
<tr>
<td><strong>Keyphrases</strong></td>
<td>text analytics, text mining, patterns, text, textual data, topic, information, text documents, information extraction, machine learning, data mining, knowledge discovery, ...</td>
<td>kobe beef, fish lovers, steakhouse, sancerre, wine list, guests, perfectly cooked, lobster, staff, meat, fillet, fish, lover, seafood, ribeye, filet, sea base, risotto, starter, scallops, steak, beef, ...</td>
</tr>
</tbody>
</table>
Outline

- Learning Embeddings in Networks and Text
  - LINE: Large-scale Information Network Embedding
  - Biological Relationship Discovery with Network Embedding
- LAKI: Representing Documents via Latent Keyphrase Inference
- MetaPAD: Meta Pattern Discovery from Massive Text Corpora
- TextCube, EventCube and CaseOLAP
- Summary
MetaPAD: Meta Pattern-driven Attribute Discovery from Massive Text Corpora

- Meng Jiang, Jingbo Shang, Xiang Ren, Taylor Cassidy, Lance Kaplan, Timothy Hanratty, and Jiawei Han, “MetaPAD: Meta Pattern-driven Attribute Discovery from Massive Text Corpora”, submitted in 2017

- Motivation: Given a sentence in a large corpus, “President Blaise Compaoré’s government of Burkina Faso was founded...”, ...

- We may find:

  - Attribute Discovery: Two tasks

    - Task 1: \( \langle \text{entity, attribute name, attribute value} \rangle \)
      - \( \langle \text{Burkina Faso, president, Blaise Compaoré} \rangle \)
      - \( \langle \text{Burkina Faso, population, 17 million} \rangle \)
      - \( \langle \text{Blaise Compaoré, age, 65} \rangle \)

    - Task 2: \( \langle \text{entity type, attribute name} \rangle \)
      - \( \langle \$\text{COUNTRY}, \text{president} \rangle \)
      - \( \langle \$\text{COUNTRY}, \text{population} \rangle \)
      - \( \langle \$\text{PERSON}, \text{age} \rangle \)

  - Instance-level
  - Type-level
Previous Work on Finding E-A-V and Typed Patterns

- Task 1: Finding E-A-V at the Instance Level
  - Stanford OpenIE [ACL’15], AI²’s Open IE-Ollie [EMNLP’12]
    - Learn syntactic and lexical patterns of expressing relations
  - Input: “President Blaise Compaoré’s government of Burkina Faso was founded...”
  - Output: ⟨President Blaise Compaoré, have, government of Burkina Faso⟩

- Task 2: Finding Typed Patterns
  - Google’s Biperpedia+ARI [VLDB’14, WWW’16], ReNoun [EMNLP’15]:
    - “Barack Obama, President of U.S.,” → “O, A of S,”, “S A O”

Query log: Highly constrained and unavailable

- Input: “…Sunday night, Burkina Faso…” and the “A, E” pattern
- Output: ⟨$COUNTRY, Sunday night⟩
Our Meta-Pattern Methodology

Generate patterns with massive instances in the data

Meta patterns:

- (1) “President Blaise Compaoré’s government of Burkina Faso was founded…”
- (2) “President Barack Obama’s government of U.S. claimed that…”
- (3) “U.S. President Barack Obama visited…”

Meta pattern segmentation

Adjust types for appropriate granularity

Joint extraction

Generate massive triples by matching the meta patterns

Group synonymous patterns by massive triples

No heavy annotation required
No domain knowledge required
No query log required if we can recognize and type the entities in the same manner...

⟨Burkina Faso, {president}, Blaise Compaoré⟩
⟨U.S., {president}, Barack Obama⟩
Pattern Discovery by Phrase Mining and Entity Typing

“President Blaise Compaoré’s government of Burkina Faso was founded ...”

Phrase mining (SegPhrase and AutoPhrase)

“president blaise_compaoré’s government of burkina_faso was founded ...”

Entity recognition and typing with Distant Supervision (ClusType)

“president $PERSON’s government of $LOCATION was founded ...”

Fine-grained typing (PLE by Ren et al. KDD’16)

“president $PERSON.POLITICIAN’s government of $LOCATION.COUNTRY was founded ...”
Meta-Pattern Quality Assessment and Segmentation

A rich set of features:

- Frequency
- Concordance: “$PERSON’s wife”
- Completeness: “$COUNTRY president” vs. “$COUNTRY president $POLITICIAN”
- Informativeness: “$PERSON and $PERSON” vs. “$PERSON’s wife, $PERSON”

Regression Q(.): random forest with only 300 labels

president $PERSON.POLITICIAN ’s government of $LOCATION.COUNTRY
Grouping Synonymous Patterns

\[ \langle \text{COUNTRY}, \text{president}, \text{POLITICIAN}\rangle \]

\text{COUNTRY president POLITICIAN}
\text{president POLITICIAN of COUNTRY}

\langle \text{United States, Barack Obama}\rangle

\langle \text{Barack Obama, 55}\rangle

\langle \text{Justin Trudeau, 43}\rangle

\[ \langle \text{PERSON}, \{\text{age, -year-old}\}, \text{DIGIT}\rangle \]

\text{PERSON, DIGIT,}
\text{PERSON’s age is DIGIT}
\text{PERSON, a DIGIT-year-old}
Adjusting Types in Meta Patterns for Appropriate Granularity

$PERSON, $Digit,

$PERSON's age is $Digit

$PERSON, a $Digit-year-old

$COUNTRY president $POLITICIAN

president $POLITICIAN of $COUNTRY
## Results: Patterns, Entities and Attribute Values in News Corpus

<table>
<thead>
<tr>
<th>Meta patterns</th>
<th>Entity</th>
<th>Attribute value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{COUNTRY President }$\text{POLITICIAN}</td>
<td>United States</td>
<td>Barack Obama</td>
</tr>
<tr>
<td>$\text{COUNTRY's president }$\text{POLITICIAN}</td>
<td>Russia</td>
<td>Vladimir Putin</td>
</tr>
<tr>
<td>$\text{President }$\text{POLITICIAN of }$\text{COUNTRY}$</td>
<td>France</td>
<td>Francois Hollande</td>
</tr>
<tr>
<td>$\text{POLITICIAN's government of }$\text{COUNTRY}$</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>Burkina Faso</td>
<td>Blaise Compaoré</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Meta patterns</th>
<th>Entity</th>
<th>Attribute value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{COMPANY CEO }$\text{PERSON}</td>
<td>Apple</td>
<td>Tim Cook</td>
</tr>
<tr>
<td>$\text{COMPANY chief executive }$\text{PERSON}</td>
<td>Facebook</td>
<td>Mark Zuckerberg</td>
</tr>
<tr>
<td>$\text{PERSON, the }$\text{COMPANY CEO,}$</td>
<td>Hewlett-Packard</td>
<td>Carly Fiorina</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$\text{COMPANY former CEO }$\text{PERSON}$</td>
<td>Infor</td>
<td>Charles Phillips</td>
</tr>
<tr>
<td>$\text{PERSON, the }$\text{COMPANY former CEO,}$</td>
<td>Afghan Citadel</td>
<td>Roya Mahboob</td>
</tr>
</tbody>
</table>
### Patterns and Entities Found in Medical Science Corpus

<table>
<thead>
<tr>
<th>Meta patterns</th>
<th>Entity</th>
<th>Attribute value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{TREATMENT}$ was used to treat $\text{DISEASE}$</td>
<td>zoledronic acid therapy</td>
<td>Paget’s disease of bone</td>
</tr>
<tr>
<td>$\text{DISEASE}$ using the $\text{TREATMENT}$</td>
<td>bisphosphonates</td>
<td>osteoporosis</td>
</tr>
<tr>
<td>$\text{TREATMENT}$ has been used to treat $\text{DISEASE}$</td>
<td>calcitonin</td>
<td>Paget’s disease of bone</td>
</tr>
<tr>
<td>$\text{TREATMENT}$ of patients with $\text{DISEASE}$</td>
<td>calcitonin</td>
<td>osteoporosis</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meta patterns</td>
<td>Entity</td>
<td>Attribute value</td>
</tr>
<tr>
<td>$\text{BACTERIA}$ was resistant to $\text{ANTIBIOTICS}$</td>
<td>corynebacterium striatum BM4687</td>
<td>gentamicin</td>
</tr>
<tr>
<td>$\text{BACTERIA}$ are resistant to $\text{ANTIBIOTICS}$</td>
<td>corynebacterium striatum BM4687</td>
<td>tobramycin</td>
</tr>
<tr>
<td>$\text{BACTERIA}$ is the most resistant to $\text{ANTIBIOTICS}$</td>
<td>methicillin-susceptible S aureus</td>
<td>vancomycin</td>
</tr>
<tr>
<td>$\text{BACTERIA}$, particularly those resistant to $\text{ANTIBIOTICS}$</td>
<td>multidrug-resistant enterobacteriaceae</td>
<td>gentamicin</td>
</tr>
</tbody>
</table>
## Comparative Experimental Results

<table>
<thead>
<tr>
<th>F1 score</th>
<th>(&lt;\text{entity type, attribute name}&gt;)</th>
<th>(&lt;\text{entity, attribute name, attribute value}&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baselines</td>
<td>Stanford’s OpenIE: 0.035</td>
<td>AI2’s Ollie: 0.131</td>
</tr>
<tr>
<td></td>
<td>Biperpedia: 0.324</td>
<td>Google’s ReNoun: 0.309</td>
</tr>
<tr>
<td>+Segmentation</td>
<td>+40.0%</td>
<td>+19.4%</td>
</tr>
<tr>
<td>+Type Adjustment</td>
<td>+6.5%</td>
<td>+15.0%</td>
</tr>
<tr>
<td>+Synonymous</td>
<td>+2.6%</td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>0.495 relatively +52.9%</td>
<td>0.424 relatively +37.3%</td>
</tr>
</tbody>
</table>
Outline

- Learning Embeddings in Networks and Text
  - LINE: Large-scale Information Network Embedding
  - Biological Relationship Discovery with Network Embedding
- LAKI: Representing Documents via Latent Keyphrase Inference
- MetaPAD: Meta Pattern Discovery from Massive Text Corpora
- TextCube, EventCube and CaseOLAP
- Summary
Multi-Dimensional Text Cube

- Numerical data cube (each cell is a numerical value) has been extensively studied
  - Measures: Numerical aggregations as \textit{sum} & \textit{avg}.
- Text cube: Each cell contains a set of documents (e.g., Apple, TV, 2016+)
  - There is an imminent need to do OLAP analysis on text cubes

Dimensions:
- \textit{Brand}: Apple, Samsung, Huawei...
- \textit{Product}: phone, tablet, TV, laptop...
- \textit{Field}: IR, Machine Learning, NLP...

Text:
- \textit{Aviation Safety Reports from NASA}
- \textit{Product Reviews from Amazon}
- \textit{Research Papers from DBLP}
Multi-dimensional Text Cube with Queries & Hierarchies

Queries:
*Point Query*: one specific cell
*Plane Query*: a set of cells

- Query “United States presidential election”
- Query “Illinois economy”
- Query “Illinois sports”

### Topic dimension
- Politics
- Economy
- Sports
- The Queen
- Stock price
- Inflation
- GDP
- Basketball
- Football

### Time dimension
- 7pm
- 8pm
- 9pm

### Location dimension
- Champaign
- Chicago
- New York City
- San Francisco
- Los Angeles
- Moscow
- Beijing
- Illinois
- New York
- California
- Russia
- China
- United States

+ dimension \(k, k+1\) ……
Exploration of Text Cube—Semantic Analysis

- EventCube [KDD’13 demo]: Point Query
  - Simple summary to support keyword/document search
- CASeOLAP [EngBul’16]: Plane Query
  - Comparative summary/mining

Cube Structure

Text Data

Slice
Roll-up
Drill-down
Dice

Textual Analysis

Structural Analysis
Multiple functions supported by EvenCube

- **Contextual Search**
- **Hierarchical Dimension Selection**: support multiple choices
- **Similar Document Finding**: based on Contextual Search
- **Keyword Frequency Distribution**
- **Multi-gram Summarization**

<table>
<thead>
<tr>
<th>Query 1</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>assign altitude</td>
<td>13.5010</td>
</tr>
<tr>
<td>altitud deviat</td>
<td>13.1992</td>
</tr>
<tr>
<td>cross restrict</td>
<td>12.7517</td>
</tr>
<tr>
<td>altitud restrict assign</td>
<td>12.2687</td>
</tr>
<tr>
<td>altitud excur</td>
<td>11.6587</td>
</tr>
<tr>
<td>temporail flight restrict</td>
<td>11.0593</td>
</tr>
<tr>
<td>cross restrict cross</td>
<td>10.8610</td>
</tr>
<tr>
<td>altitud assign</td>
<td>10.3631</td>
</tr>
<tr>
<td>altitud restrict</td>
<td>10.2753</td>
</tr>
<tr>
<td>cross altitud</td>
<td>9.9490</td>
</tr>
<tr>
<td>cross restrict cross 90</td>
<td>9.2516</td>
</tr>
<tr>
<td>altitud captur</td>
<td>9.0427</td>
</tr>
<tr>
<td>4000 feet</td>
<td>9.0394</td>
</tr>
<tr>
<td>5000 feet</td>
<td>8.6173</td>
</tr>
<tr>
<td>50000 feet</td>
<td>8.4111</td>
</tr>
</tbody>
</table>

**Detail Info**

we were given a crossing restrict of 9000 feet at krenna; but crossed krenna at 10000 feet, both the first officer pilot flying and i missed the intersection distance measuring equipment and thought we had 6 more miles than we actually did to meet the restrict center asked if we would be able to make the restr; and we said yes; thinking we had the necessary distance about that time; center gave us a radio frequency change; and we realized we were at 10000 feet at krenna; instead of 9000 feet, this problem was strictly a result of misreading the approach place; and i believe the problem can be avoided in the future. 

**Problem Area**

Select Some Options

**Target Dimensions**

- **B737 Undifferentiated or Other**
- **Airport**

Close
CASE (Context-Aware SEmantic) OLAP

- A cell has comparative context
- Comparative study is meaningful
  - Given a query <China, Economy>
- Target documents have frequent phrases
- Be specific to “China”+“Economy”
Design Question I: Which Comparative Groups to Pick?

- Option 1: User-specified (too much burden to users): undesirable
- Option 2: **Sibling cells** in every dimension (comparable cells)
Design Question: How to Score Important Phrases?

- Three ingredients
  - **Integrity**: meaningful, high-quality phrase
    - Using SegPhrase as score (>0.7)
  - **Popularity**: large # of occurrences in the cell
  - **Distinctness**: distinguish the target cell from context cells
    - A key to have a crisp definition
  - Combining with geometric mean:

\[
s_{\text{int}}(p, c) = \frac{\log(\text{tf}(p, c) + 1)}{\log \text{cntP}(c)} \quad (2)
\]

\[r(p, c) = \sqrt[3]{s_{\text{int}}(p, c) \cdot s_{\text{pop}}(p, c) \cdot s_{\text{dist}}(p, c)} \quad (1)
\]
How to Find or Evaluate Distinct Phrases in a Cell?

- Judge if a phrase \( p \) is distinct in cell \( c \): Transform it into a dual problem
  - Original problem: Find distinctive phrases for cell \( c \), compared to sibling cells
  - Transformed problem: Classify phrases into one of the most relevant cell
- For a distinct phrase \( p \), if we measure relevance(\( p, c \)) for all \( c \)
  - \( \text{rel}(p, c^*) >> \text{rel}(p, \text{sibling}) \)
- Adopt Softmax function as

\[
disti(p, c) = \frac{1}{1 + \sum_{c' \in S \cup \{c\}, p \in c'} e^{\text{rel}(p, c')}}
\]  

(4)
How to Design Relevance Score for a Phrase to a Cell?

- Normalized Term Frequency
- Treat each cell as a super document
- Apply BM25

\[
ntf(p, c) = \frac{tf(p, c) \cdot (k_1 + 1)}{tf(p, c) + k_1 \cdot (1 - b + b \cdot \frac{cntP(c)}{avgCP(c)})}
\]  

(5)

- Normalized Document Frequency

\[
dnf(p, c) = \frac{\log(1 + df(p, c))}{\log(1 + maxDF(c))}
\]  

(6)

Balance cell size

- Guarantee spread out!

- Combine:

\[
rel(p, c) = ndf(p, c) \cdot ntf(p, c)
\]  

(7)
Experiments

- Data:
  - 4,785,990 news articles
  - Six dimensions

- Methods
  - **MCX**: based ratio to global background
  - **SegPhrase**: Using only integrity score
  - **MCX+Seg**: SegPhrase + MCX
  - **TF-IDF**: Distinctness by IDF, Popularity by TF
  - **RP (No INT)**: ablation without integrity
  - **RP (No POP)**: ablation without popularity
  - **RP (No DIS)**: ablation without distinctness
  - **RP**: our proposed method

Figure 7: Phrase assignment accuracy comparison to baselines
Figure 8: Phrase assignment accuracy comparison to ablations

Effectiveness

(a) Time-space balance of 4-Dim Cube
(b) Time-space balance of 6-Dim Cube

Efficiency
Effectiveness of CaseOLAP on Real-World Datasets

Distinct phrases on 2016 news data

<table>
<thead>
<tr>
<th>(US, Gun Control)</th>
<th>(US, Immigration)</th>
<th>(US, Domestic Politics)</th>
<th>(US, Law and Crime)</th>
<th>(US, Military)</th>
</tr>
</thead>
<tbody>
<tr>
<td>gun laws</td>
<td>immigration debate</td>
<td>gun laws</td>
<td>district attorney</td>
<td>sexual assault in the military</td>
</tr>
<tr>
<td>the national rifle association</td>
<td>border security</td>
<td>insurance plans</td>
<td>shot and killed</td>
<td>military prosecutors</td>
</tr>
<tr>
<td>gun rights</td>
<td>guest worker program</td>
<td>background check</td>
<td>federal court</td>
<td>armed services committee</td>
</tr>
<tr>
<td>background check</td>
<td>immigration legislation</td>
<td>health coverage</td>
<td>life in prison</td>
<td>armed forces</td>
</tr>
<tr>
<td>gun owners</td>
<td>undocumented immigrants</td>
<td>tax increases</td>
<td>death row</td>
<td>defense secretary</td>
</tr>
<tr>
<td>assault weapons ban</td>
<td>overhaul of the nation's immigration laws</td>
<td>the national rifle association</td>
<td>grand jury</td>
<td>military personnel</td>
</tr>
<tr>
<td>mass shootings</td>
<td>legal status</td>
<td>assault weapons ban</td>
<td>department of justice</td>
<td>sexually assaulted</td>
</tr>
<tr>
<td>high capacity magazines</td>
<td>path to citizenship</td>
<td>immigration debate</td>
<td>child abuse</td>
<td>for morale</td>
</tr>
<tr>
<td>gun legislation</td>
<td>immigration status</td>
<td>the federal exchange</td>
<td>plea deal</td>
<td>private manning</td>
</tr>
<tr>
<td>gun control advocates</td>
<td>immigration reform</td>
<td>medicaid program</td>
<td>second degree murder</td>
<td>pentagon officials</td>
</tr>
</tbody>
</table>

PubMed Abstracts: Distinct relationships between subcategories of cardiovascular diseases and proteins

Table 2: Top representative phrases for 6 cardiac diseases
More on Real-World Case Study: BioData Analysis

- CASeOLAP found top-k representative proteins for each Cardiovascular Disease
- Protein in IHD & CVA has similar pattern to reveal inflammatory function
- Amyloid beta A4 appears to be consistent over 6 groups
- Galactosidase had a high score of (5.903), suggesting that glycoproteins and lipids are highly relevant
## More on Real-World Cases: Protest News

<table>
<thead>
<tr>
<th>Dataset: PROTEST NEWS ARTICLES</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th># Documents</th>
<th># Dimensions</th>
<th>Avg. Word Count</th>
<th>Text Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>10K</td>
<td>6</td>
<td>~500</td>
<td>News</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dimension</th>
<th># Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident</td>
<td>111</td>
<td>Individual protest and attach incidents</td>
</tr>
<tr>
<td>Location</td>
<td>20</td>
<td>Countries that the protests happened</td>
</tr>
<tr>
<td>Type of Protest</td>
<td>6</td>
<td>Six different types of protest such as Demonstration</td>
</tr>
<tr>
<td>Demands of Protest</td>
<td>4</td>
<td>Four different types of protest such as Political and Environmental</td>
</tr>
<tr>
<td>Protester</td>
<td>10</td>
<td>Different protest groups such as students and political opposition</td>
</tr>
<tr>
<td>Time</td>
<td>48</td>
<td>Different months spanning from Jan 2009 to Dec 2015</td>
</tr>
</tbody>
</table>

- CaseOLAP helps find top-k representative phrases and helps automatically generate representative video captions.
Outline

- Learning Embeddings in Networks and Text
  - LINE: Large-scale Information Network Embedding
  - Biological Relationship Discovery with Network Embedding
  - LAKI: Representing Documents via Latent Keyphrase Inference
  - MetaPAD: Meta Pattern Discovery from Massive Text Corpora
  - TextCube, EventCube and CaseOLAP

- Summary
Summary

- A promising integrated approach for text analysis
  - Text mining + embedding + information network analysis
- Embedding for text analysis
  - LINE + PTE + information network approach (e.g., meta path-guided analysis)
- Representing documented by in-depth semantic analysis
  - LAKI and beyond
- MetaPAD: Meta Pattern Discovery from Massive Text Corpora
  - Facilitate information extraction from massive text using meta-patterns
- CaseOLAP: Distinctive analysis of documents in multi-dimensional space
  - Many more to be studied
References

- JianTang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei, “LINE: Large-scale information network embedding”, WWW'15
- JianTang, Meng Qu, and Qiaozhu Mei, “PTE: Predictive text embedding through large-scale heterogeneous text networks”, KDD'15
- Jialu Liu, Xiang Ren, Jingbo Shang, Taylor Cassidy, Clare Voss and Jiawei Han, “Representing Documents via Latent Keyphrase Inference”, WWW'16
- Meng Jiang, Jingbo Shang, Taylor Cassidy, Xiang Ren, Lance M. Kaplan, Timothy P. Hanratty, and Jiawei Han, “MetaPAD: Meta Pattern Discovery from Massive Text Corpora”, submitted for publication, 2017
- Alon Halevy, Natalya Noy, Sunita Sarawagi, Steven Euijiong Whang, and Xiao Yu, “Discovering structure in the universe of attribute names”, WWW’16
- Fangbo Tao, Honglei Zhuang, Chi Wang Yu, Qi Wang, Taylor Cassidy, Lance Kaplan, Clare Voss, Jiawei Han, “Multi-Dimensional, Phrase-Based Summarization in Text Cubes", Data Eng. Bulletin 39(3), Sept. 2016
Text Representation

Learning meaningful text representations is important for various machine learning tasks

Bag of words:
- Sparsity
- Ignore the relatedness between different words

Text Classification
Text Clustering
Retrieval
Recent Studies

- **Distributed Representations**
  - Embed text into a low-dimensional space
  - Word2vec, Paragraph Vector
  - **Strength:**
    - Low-dimensional vectors; similar texts have similar vectors; efficient
  - **Weakness:**
    - Totally unsupervised; Can’t guide the training

- **Convolutional Neural Network**
  - Used for text classification
  - The hidden layer can be used for text representation
  - **Strength:** High accuracy
  - **Weakness:** Totally supervised;
    - Slow to train; Training is very tricky
Effective Text Representations

- Desired Features for Effective Text Representation
  - Semi-supervised
    - Use labeled data to guide the training and boost the performance
    - Learn from massive unlabeled text data
  - Low-dimensional vector representations & Efficient
- How to realize such an effective representation?
  - A prerequisite of learning good text representations is to train effective word representations
  - Texts consist of words: To understand a text, we should first know the words well
  - Learn effective word representations
    - Distributional Hypothesis
      - words that occur in the same contexts tend to have similar meanings
      - Use multiple contexts to train word representations
The PTE Method

- Three different contexts:
  - Unsupervised: word document
  - Supervised: label
- Use Heterogeneous Information Networks to represent the co-occurrence relationship between words and contexts

(a) word-word network
(b) word-document network
(c) word-label network

Heterogeneous text network
Learn Word Embedding from Hetero. Info. Networks

- For a single context:
  \[ O_c = \sum_{w,c} \hat{P}_{wc} \log \sigma(\mathbf{w} \cdot \mathbf{c}) + K \sum_{w,c} \hat{P}_w \hat{P}_c \log \sigma(-\mathbf{w} \cdot \mathbf{c}) \]

- For all three contexts:
  \[ O_{pte} = O_{word} + O_{doc} + O_{label} \]

**Algorithm 1: Joint training.**

Data: \( G_{ww}, G_{wd}, G_{wl} \), number of samples \( T \), number of negative samples \( K \).

Result: word embeddings \( \mathbf{w} \).

while \( \text{iter} \leq T \) do
  - sample an edge from \( E_{ww} \) and draw \( K \) negative edges, and update the word embeddings;
  - sample an edge from \( E_{wd} \) and draw \( K \) negative edges, and update the word and document embeddings;
  - sample an edge from \( E_{wl} \) and draw \( K \) negative edges, and update the word and label embeddings;
end

**Algorithm 2: Pre-training + Fine-tuning.**

Data: \( G_{ww}, G_{wd}, G_{wl} \), number of samples \( T \), number of negative samples \( K \).

Result: word embeddings \( \mathbf{w} \).

while \( \text{iter} \leq T \) do
  - sample an edge from \( E_{ww} \) and draw \( K \) negative edges, and update the word embeddings;
  - sample an edge from \( E_{wd} \) and draw \( K \) negative edges, and update the word and document embeddings;
end

while \( \text{iter} \leq T \) do
  - sample an edge from \( E_{wl} \) and draw \( K \) negative edges, and update the word and label embeddings;
end
Taking average of word embedding

For \( d = [w_1, w_2, \ldots, w_n] \)

\[
\hat{d} = \frac{1}{n} \sum_{k=1}^{n} \vec{w}_k
\]

Ignore word order

Efficient and effective in most cases

Using Recursive NN or Recurrent NN

Consider word order

Too slow
Experiments: Setup and Results

- **Datasets**

- **Task:**
  - Text classification: Metric: Macro-F1 Micro-F1 (accuracy)
  - Baselines: BOW, Skip-Gram, PVDM, PVDBOW, CNN

### Table 1: Statistics of the Data Sets

<table>
<thead>
<tr>
<th>Name</th>
<th>20NG</th>
<th>WIKI</th>
<th>IMDB</th>
<th>CORPORATE</th>
<th>Economics</th>
<th>GOVERNMENT</th>
<th>MARKET</th>
<th>DBLP</th>
<th>MR</th>
<th>TWITTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long Documents</td>
<td>19,314</td>
<td>1,911,617*</td>
<td>25,000</td>
<td>245,650</td>
<td>77,242</td>
<td>138,990</td>
<td>132,040</td>
<td>61,479</td>
<td>7,108</td>
<td>800,000</td>
</tr>
<tr>
<td>Short Documents</td>
<td>89,039</td>
<td>913,881</td>
<td>71,381</td>
<td>141,740</td>
<td>65,254</td>
<td>139,960</td>
<td>64,049</td>
<td>22,270</td>
<td>17,376</td>
<td>405,994</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>Avg.Length</th>
<th>#Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>89,039</td>
<td>305.77</td>
<td>20</td>
</tr>
<tr>
<td>Test</td>
<td>913,881</td>
<td>672.56</td>
<td>7</td>
</tr>
</tbody>
</table>

*In the Wiki data, only 42,000 documents are labeled.*
## Experimental Results

- Results on Long and short documents

### Table 2: Results of text classification on long documents.

<table>
<thead>
<tr>
<th>Type</th>
<th>Algorithm</th>
<th>20NG Micro-F1</th>
<th>20NG Macro-F1</th>
<th>Wikipedia Micro-F1</th>
<th>Wikipedia Macro-F1</th>
<th>IMDB Micro-F1</th>
<th>IMDB Macro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word</td>
<td>BOW</td>
<td>80.88</td>
<td>79.30</td>
<td>79.95</td>
<td>80.03</td>
<td>86.54</td>
<td>86.54</td>
</tr>
<tr>
<td></td>
<td>Skip-gram</td>
<td>70.62</td>
<td>68.99</td>
<td>75.80</td>
<td>75.77</td>
<td>85.34</td>
<td>85.34</td>
</tr>
<tr>
<td></td>
<td>PVDBOW</td>
<td>75.13</td>
<td>73.48</td>
<td>76.68</td>
<td>76.75</td>
<td>86.76</td>
<td>86.76</td>
</tr>
<tr>
<td></td>
<td>PVDM</td>
<td>61.03</td>
<td>56.46</td>
<td>72.96</td>
<td>72.76</td>
<td>82.33</td>
<td>82.33</td>
</tr>
<tr>
<td></td>
<td>LINE($G_{uw}$)</td>
<td>72.78</td>
<td>70.95</td>
<td>77.72</td>
<td>77.72</td>
<td>86.16</td>
<td>86.16</td>
</tr>
<tr>
<td></td>
<td>LINE($G_{ud}$)</td>
<td>79.73</td>
<td>78.40</td>
<td>80.14</td>
<td>80.13</td>
<td>89.14</td>
<td>89.14</td>
</tr>
<tr>
<td></td>
<td>LINE($G_{uw} + G_{ud}$)</td>
<td>78.74</td>
<td>77.39</td>
<td>79.91</td>
<td>79.94</td>
<td>89.07</td>
<td>89.07</td>
</tr>
<tr>
<td>Unsupervised Embedding</td>
<td>CNN</td>
<td>78.85</td>
<td>78.29</td>
<td>79.72</td>
<td>79.77</td>
<td>86.15</td>
<td>86.15</td>
</tr>
<tr>
<td></td>
<td>CNN(pretrain)</td>
<td>80.15</td>
<td>79.43</td>
<td>79.25</td>
<td>79.32</td>
<td>89.00</td>
<td>89.00</td>
</tr>
<tr>
<td></td>
<td>PTE($G_{wl}$)</td>
<td>82.70</td>
<td>81.97</td>
<td>79.00</td>
<td>79.02</td>
<td>85.98</td>
<td>85.98</td>
</tr>
<tr>
<td></td>
<td>PTE($G_{uw} + G_{wl}$)</td>
<td>83.90</td>
<td>83.11</td>
<td>81.65</td>
<td>81.62</td>
<td>89.14</td>
<td>89.14</td>
</tr>
<tr>
<td></td>
<td>PTE($G_{ud} + G_{wl}$)</td>
<td><strong>84.39</strong></td>
<td><strong>83.64</strong></td>
<td>82.29</td>
<td>82.27</td>
<td>89.76</td>
<td>89.76</td>
</tr>
<tr>
<td></td>
<td>PTE(pretrain)</td>
<td>82.86</td>
<td>82.12</td>
<td>79.18</td>
<td>79.21</td>
<td>86.28</td>
<td>86.28</td>
</tr>
<tr>
<td></td>
<td>PTE(joint)</td>
<td>84.20</td>
<td>83.39</td>
<td><strong>82.51</strong></td>
<td><strong>82.49</strong></td>
<td><strong>89.80</strong></td>
<td><strong>89.80</strong></td>
</tr>
</tbody>
</table>

### Table 4: Results of text classification on short documents.

<table>
<thead>
<tr>
<th>Type</th>
<th>Algorithm</th>
<th>DBLP Micro-F1</th>
<th>DBLP Macro-F1</th>
<th>MR Micro-F1</th>
<th>MR Macro-F1</th>
<th>Twitter Micro-F1</th>
<th>Twitter Macro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word</td>
<td>BOW</td>
<td>75.28</td>
<td>71.59</td>
<td>71.90</td>
<td>71.90</td>
<td>75.27</td>
<td>75.27</td>
</tr>
<tr>
<td></td>
<td>Skip-gram</td>
<td>73.08</td>
<td>68.92</td>
<td>67.05</td>
<td>67.05</td>
<td>73.02</td>
<td>73.00</td>
</tr>
<tr>
<td></td>
<td>PVDBOW</td>
<td>67.19</td>
<td>62.46</td>
<td>67.78</td>
<td>67.78</td>
<td>71.29</td>
<td>71.18</td>
</tr>
<tr>
<td></td>
<td>PVDM</td>
<td>37.11</td>
<td>34.38</td>
<td>58.22</td>
<td>58.17</td>
<td>70.75</td>
<td>70.73</td>
</tr>
<tr>
<td></td>
<td>LINE($G_{uw}$)</td>
<td>73.98</td>
<td>69.92</td>
<td>71.07</td>
<td>71.06</td>
<td>73.19</td>
<td>73.18</td>
</tr>
<tr>
<td></td>
<td>LINE($G_{ud}$)</td>
<td>71.50</td>
<td>67.23</td>
<td>69.25</td>
<td>69.24</td>
<td>73.19</td>
<td>73.19</td>
</tr>
<tr>
<td></td>
<td>LINE($G_{uw} + G_{ud}$)</td>
<td>74.22</td>
<td>70.12</td>
<td>71.13</td>
<td>71.12</td>
<td>73.84</td>
<td>73.84</td>
</tr>
<tr>
<td>Unsupervised Embedding</td>
<td>CNN</td>
<td>76.16</td>
<td>73.08</td>
<td>72.71</td>
<td>72.69</td>
<td><strong>75.97</strong></td>
<td><strong>75.96</strong></td>
</tr>
<tr>
<td></td>
<td>CNN(pretrain)</td>
<td>75.39</td>
<td>72.28</td>
<td>68.96</td>
<td>68.87</td>
<td>75.92</td>
<td>75.92</td>
</tr>
<tr>
<td></td>
<td>PTE($G_{wl}$)</td>
<td>76.45</td>
<td>72.74</td>
<td>73.44</td>
<td>73.42</td>
<td>73.92</td>
<td>73.91</td>
</tr>
<tr>
<td></td>
<td>PTE($G_{uw} + G_{wl}$)</td>
<td>76.80</td>
<td>73.28</td>
<td>72.93</td>
<td>72.92</td>
<td>74.93</td>
<td>74.92</td>
</tr>
<tr>
<td></td>
<td>PTE($G_{ud} + G_{wl}$)</td>
<td><strong>77.46</strong></td>
<td><strong>74.03</strong></td>
<td>73.13</td>
<td>73.11</td>
<td>75.61</td>
<td>75.61</td>
</tr>
<tr>
<td></td>
<td>PTE(pretrain)</td>
<td>76.53</td>
<td>72.94</td>
<td>73.27</td>
<td>73.24</td>
<td>73.79</td>
<td>73.79</td>
</tr>
<tr>
<td></td>
<td>PTE(joint)</td>
<td>77.15</td>
<td>73.61</td>
<td><strong>73.58</strong></td>
<td><strong>73.57</strong></td>
<td>75.21</td>
<td>75.21</td>
</tr>
</tbody>
</table>
Performance & Visualization Results

- **Performance comparison on percentage of labeled data**
  - (a) 20NG
  - (b) DBLP

- **Visualization results**
  - (a) Train(LINE($G_{wd}$))
  - (b) Train(PTE($G_{wd}$))
  - (c) Test(LINE($G_{wd}$))
  - (d) Test(PTE($G_{wd}$))