DeepWalk

Bryan Perozzi,Rami Al-Rfou,Steven Skiena

©Team Deep Thought

Oliver Labsch

Zhenhao Li

Robert Müller

Florentin Schwarzer

Veronika Sonntag

19.07.2017

Outline

1. Introduction DeepWalk
2. DeepWalk – how does it work?
3. Implementation approaches
    3.1 DataSet
    3.2 DataStream
4. Lessons learned
5. Use case
6. Demo
7. Future ideas

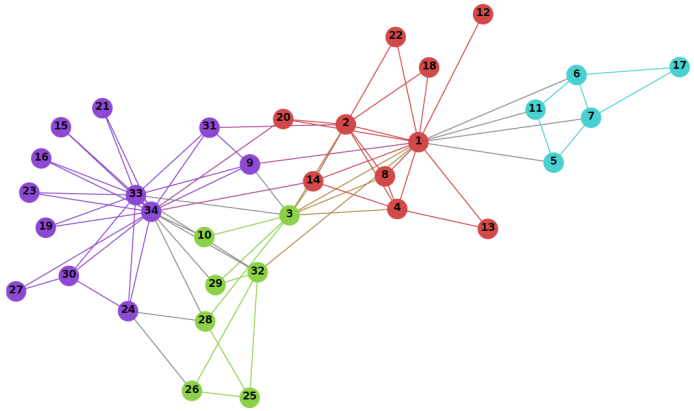
DeepWalk

- learning latent representation

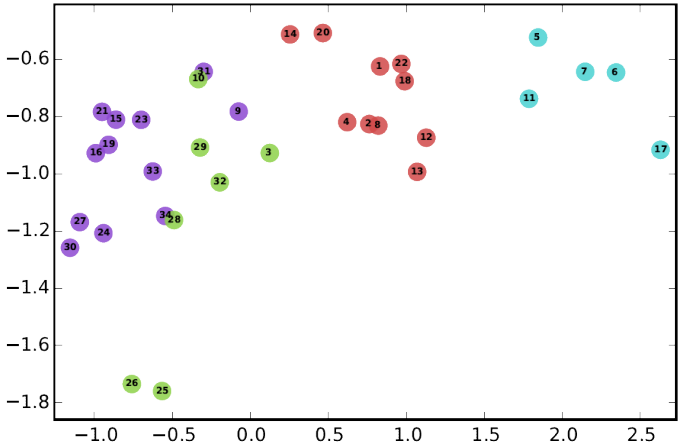
- similar nodes in graph
  → small distance in representation

- basis for further learning algorithms

- generaly consists of two parts:
  1. Random Walks
  2. Skipgram

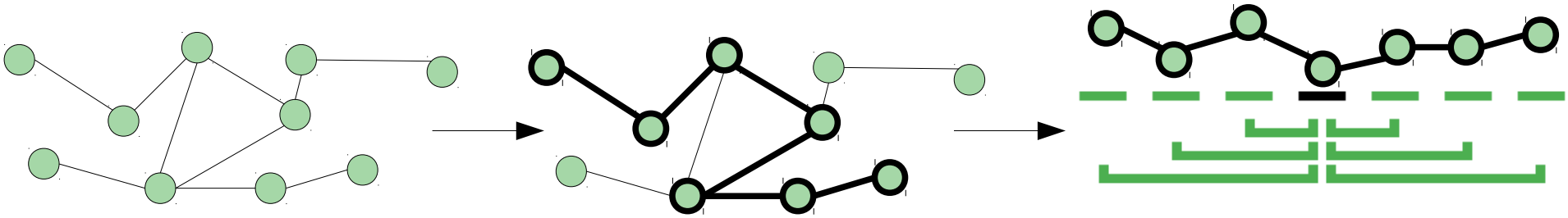


Input: graph (nodes and edges)



Output: Embeddings of the graph

Random Walk



1. Initial State

- Input: graph
- Output: focus-context pairs

2. Random Walk

- generate for each node several walks (γ)
- walklength for each node (walkLength)

3. Sliding Window

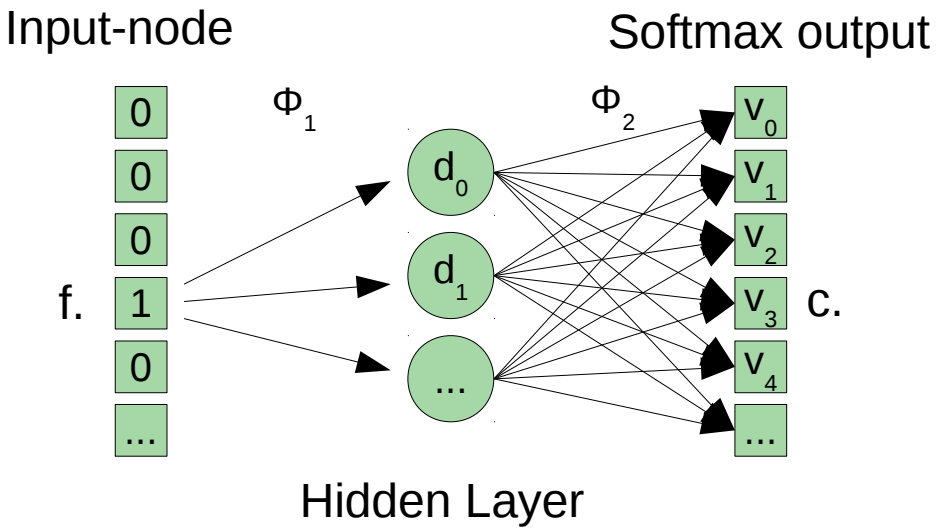
- generate focus-context pairs using sliding window (window size: ω)

Skipgram

- input: focus-context pair
- output: probability for each node to be context for given focus

- optimize weights ($\Phi_1 \Phi_2$) by
  - maximizing NN for focus-context pair
  - minimizing NN for randomly chosen negative samples

→ Embeddings: weights Φ_1

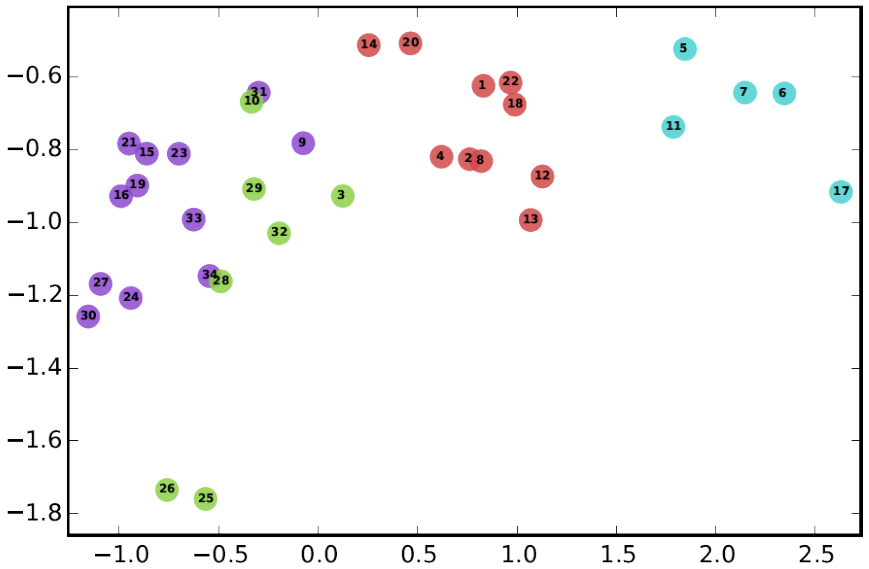


DataSet Approach

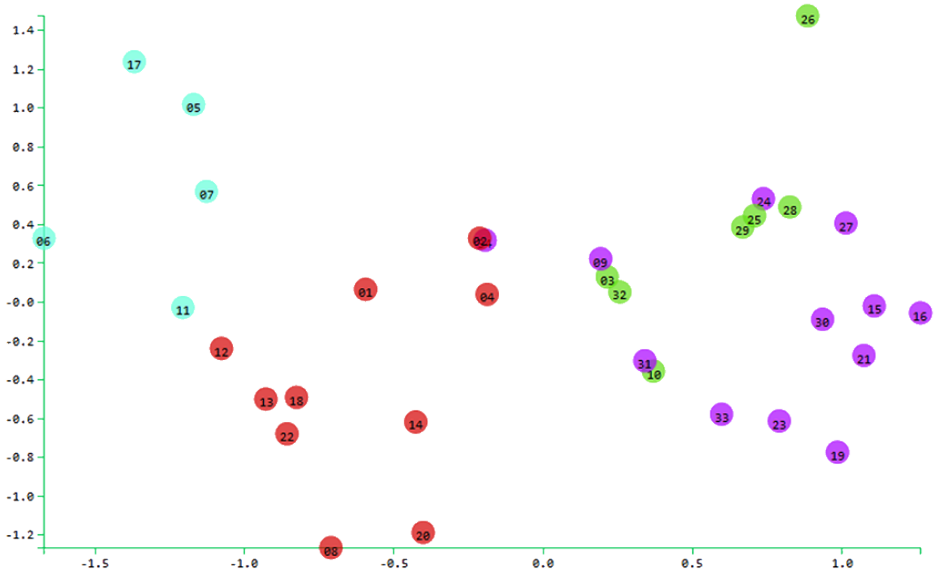
- Random Walks: DataSet API instead of Gelly

- Stochastic Gradient Descent is inherently sequential
  - → Mini-Batch Gradient Descent
  - → parallelism idea: Hogwild

- best results: 64 dim → PCA

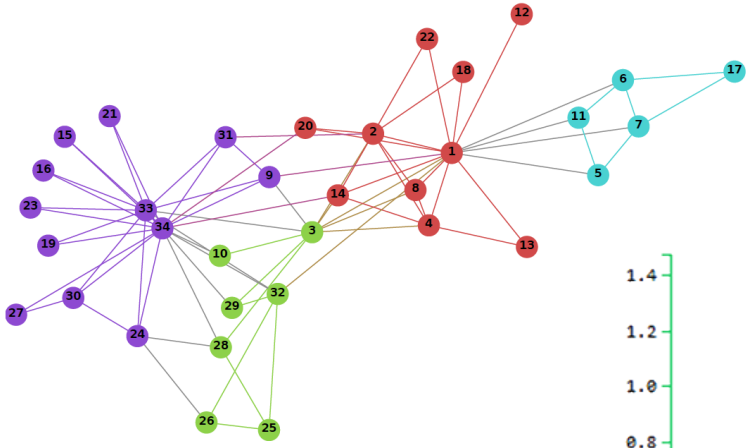


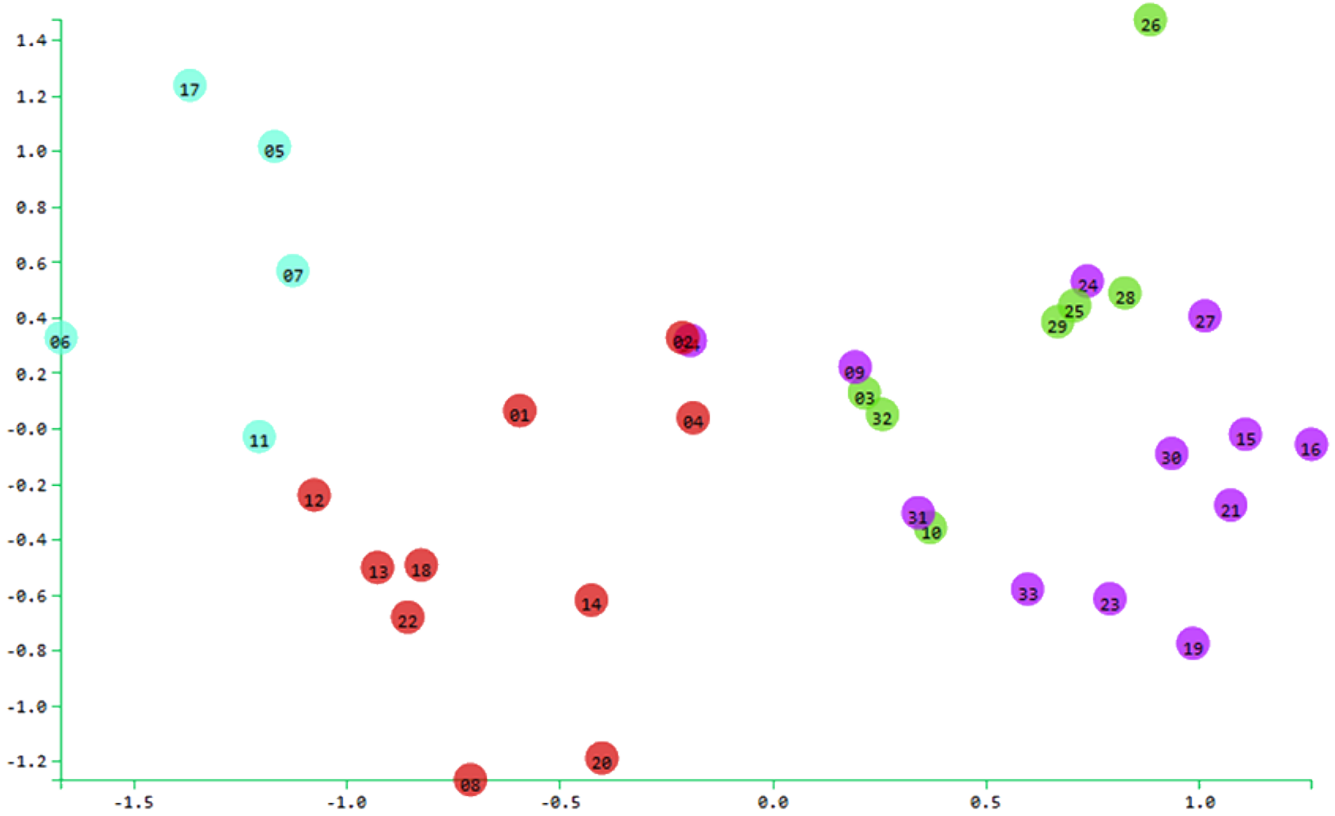
Authors‘ embeddings



Our embeddings

Our Embeddings





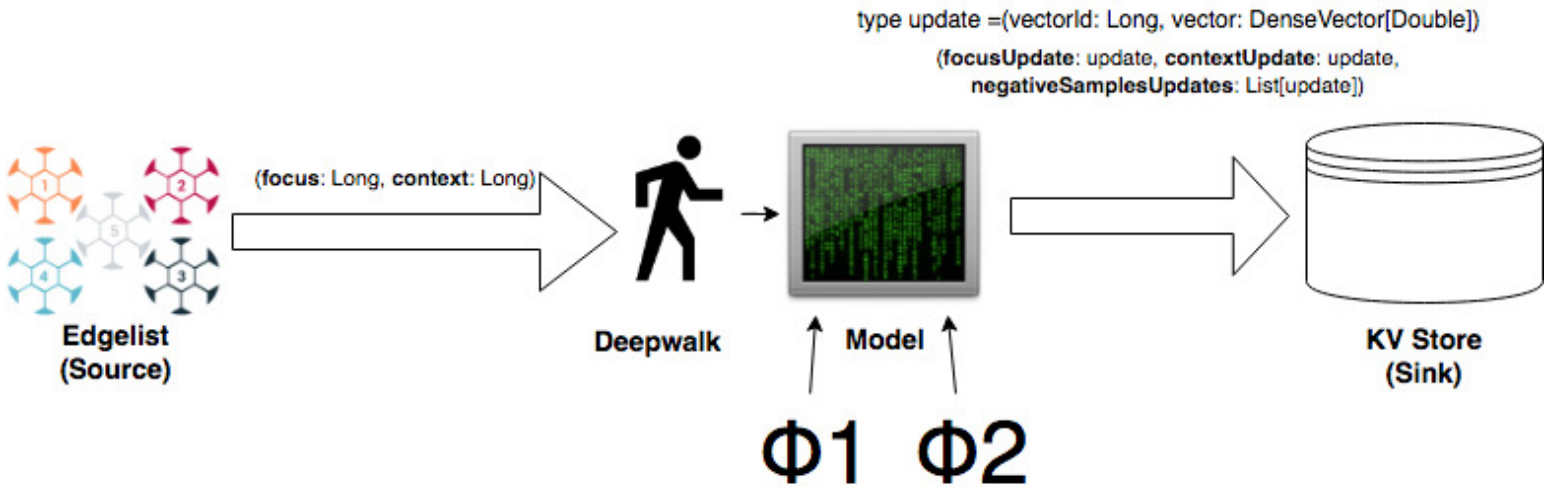
DataSet Approach: Problems

- Supervised learning task is sequential to a certain degree
  - → updating needs several steps

- no sampling operator in flink
  - → using java for-loop
  - → runtime problems

- strong dependence on choice of parameters
  - → #batches
  - → learning rate

Stream Approach



- Similar embeddings, less time
- Build model “on demand“
  → no initialization
- Due to possibly unknown number of edges use small fixed learning rate
- Access to model (Querable State) in Flink only possible during execution
  → KV Store
- Source and Sink easily exchangeable !
  → eg. Source: Socket, Dynamic Graph
  → eg. Sink: Dynamic Visualization (PCA)

Lessons Learned

- choice of parameters has large impact on results

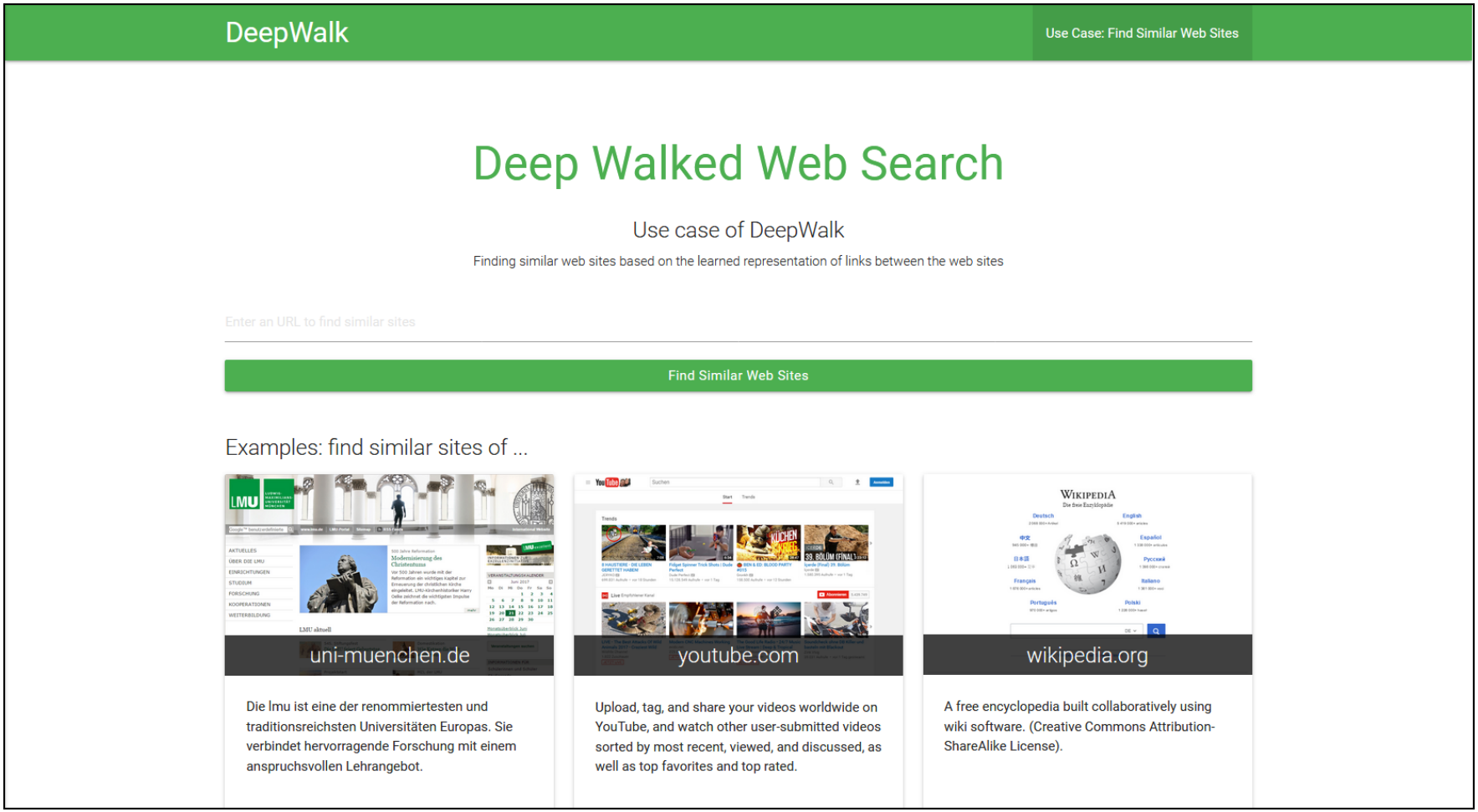
- MapReduce Paradigm different from „ordinary“ programming

- how to use tools and concepts like OpenNebula, Jira, Scrum, …

- how to work with scientific papers and apply the content practically

- Flink:
  - still rough around the edges
  - insufficiently documented
  - promising ideas not fully implemented (e.g. table API)
  - not the best choice for sequential algorithms

Use Case


DeepWalk
Use Case: Find Similar Web Sites
Deep Walked Web Search
Use case of DeepWalk
Finding similar web sites based on the learned representation of links between the web sites
Enter an URL to find similar sites
Find Similar Web Sites
Examples: find similar sites of ...
uni-muenchen.de
Die lmu ist eine der renommiertesten und traditionsreichsten Universitäten Europas. Sie verbindet hervorragende Forschung mit einem anspruchsvollen Lehrangebot.
youtube.com
Upload, tag, and share your videos worldwide on YouTube, and watch other user-submitted videos sorted by most recent, viewed, and discussed, as well as top favorites and top rated.
wikipedia.org
A free encyclopedia built collaboratively using wiki software. (Creative Commons Attribution-ShareAlike License).

Future Ideas

- streaming version for Random Walks

- extended use cases
  - → Twitter
  - → complete web graph

- further runtime optimization

- evaluation of parameter settings:
  - → Learning rate
  - → Random Walk parameters
  - → initialization of embeddings ($\Phi_1 \Phi_2$)

- evaluate the influence of the graph size
  - → nodes
  - → edges

Questions

Do you have questions?

BACKUP