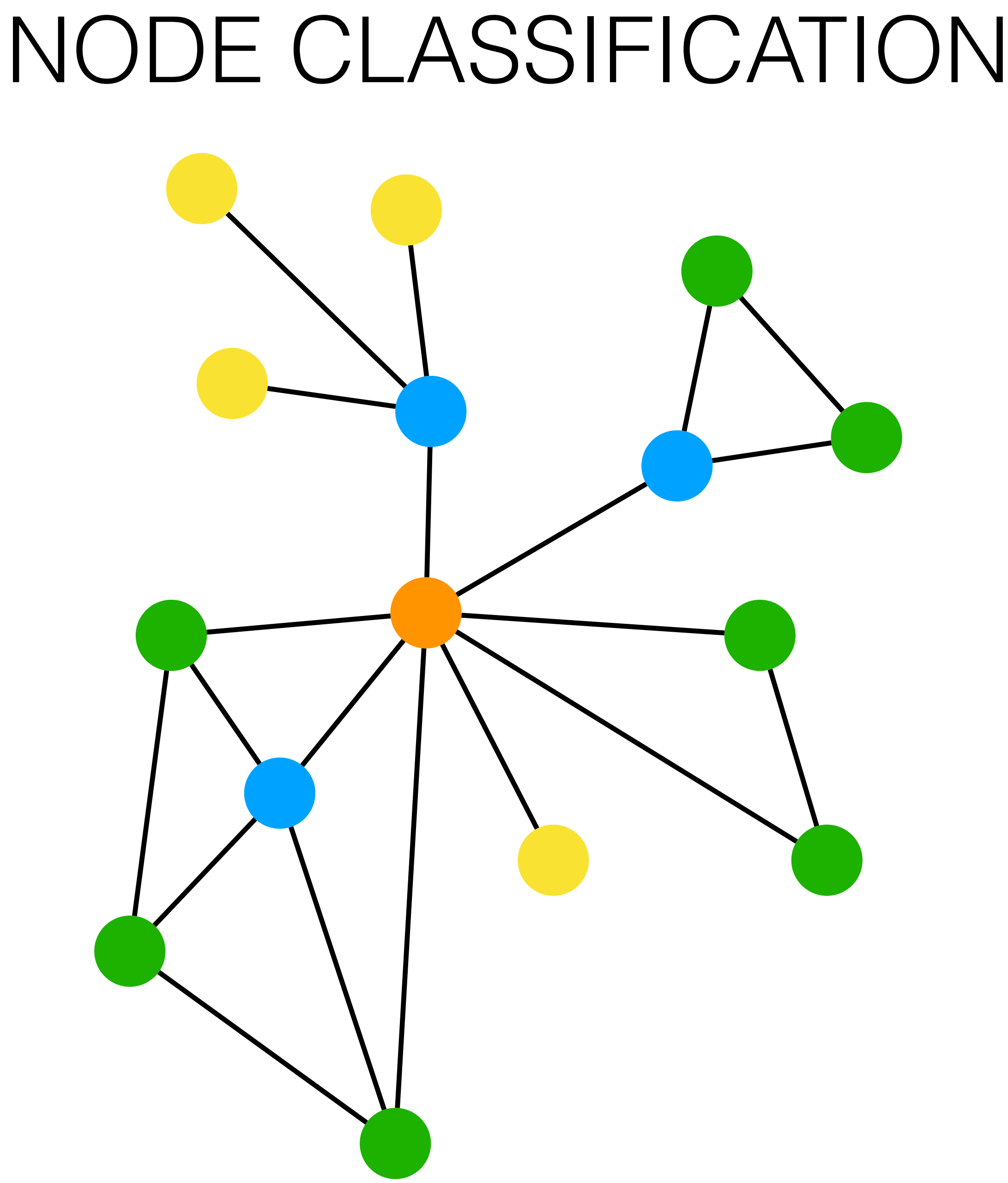
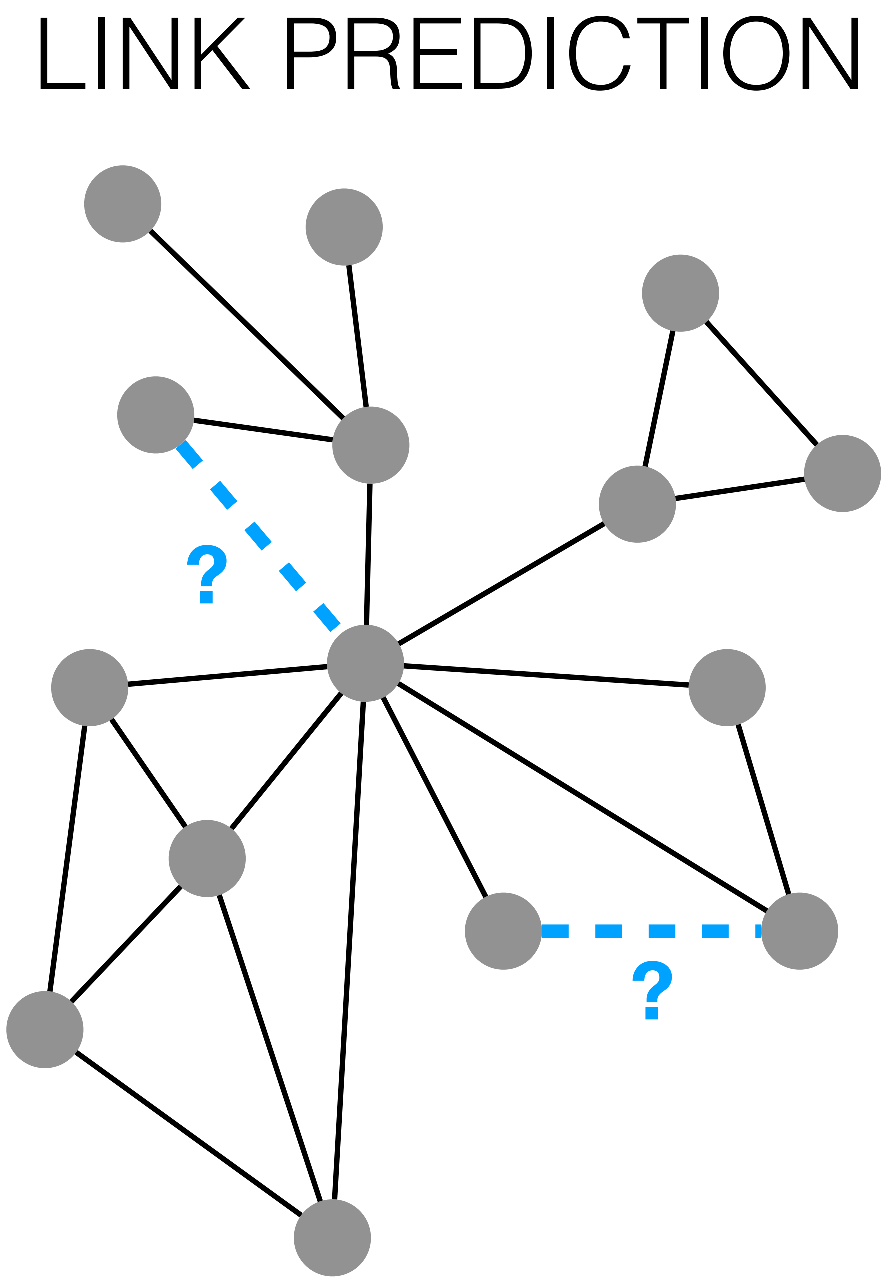
Feature Learning for Networks

ICPSR
Network Analysis I
Melanie Baybay

07/19/2017

Network Analysis:
From a Data Science Perspective





node2vec:
Feature Learning for Networks

Aditya Grover & Jure Leskovec (2016)
node2vec: Scalable Feature Learning for Networks

Goal :
To learn continuous representations for each node based on their relationships with other nodes

representation / feature learning : automatically discover representations for complex data that make it mathematically and computationally convenient to process

word2vec:
Feature Learning for Words

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013).
Efficient estimation of word representations in vector space.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013).
Distributed representations of words and phrases and their compositionality.

<u>Goal</u>: To learn continuous representations for words based on other words within the same context

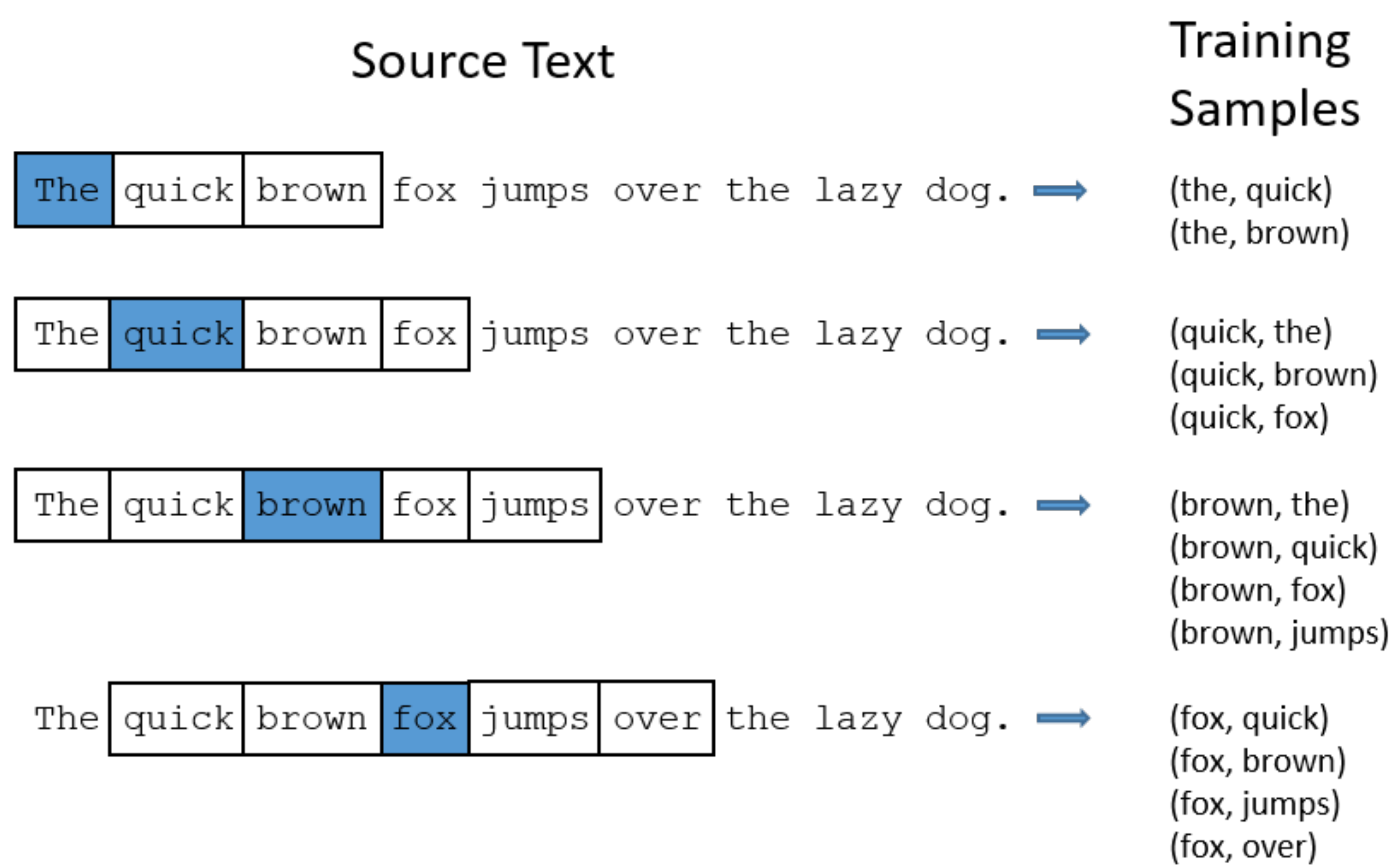
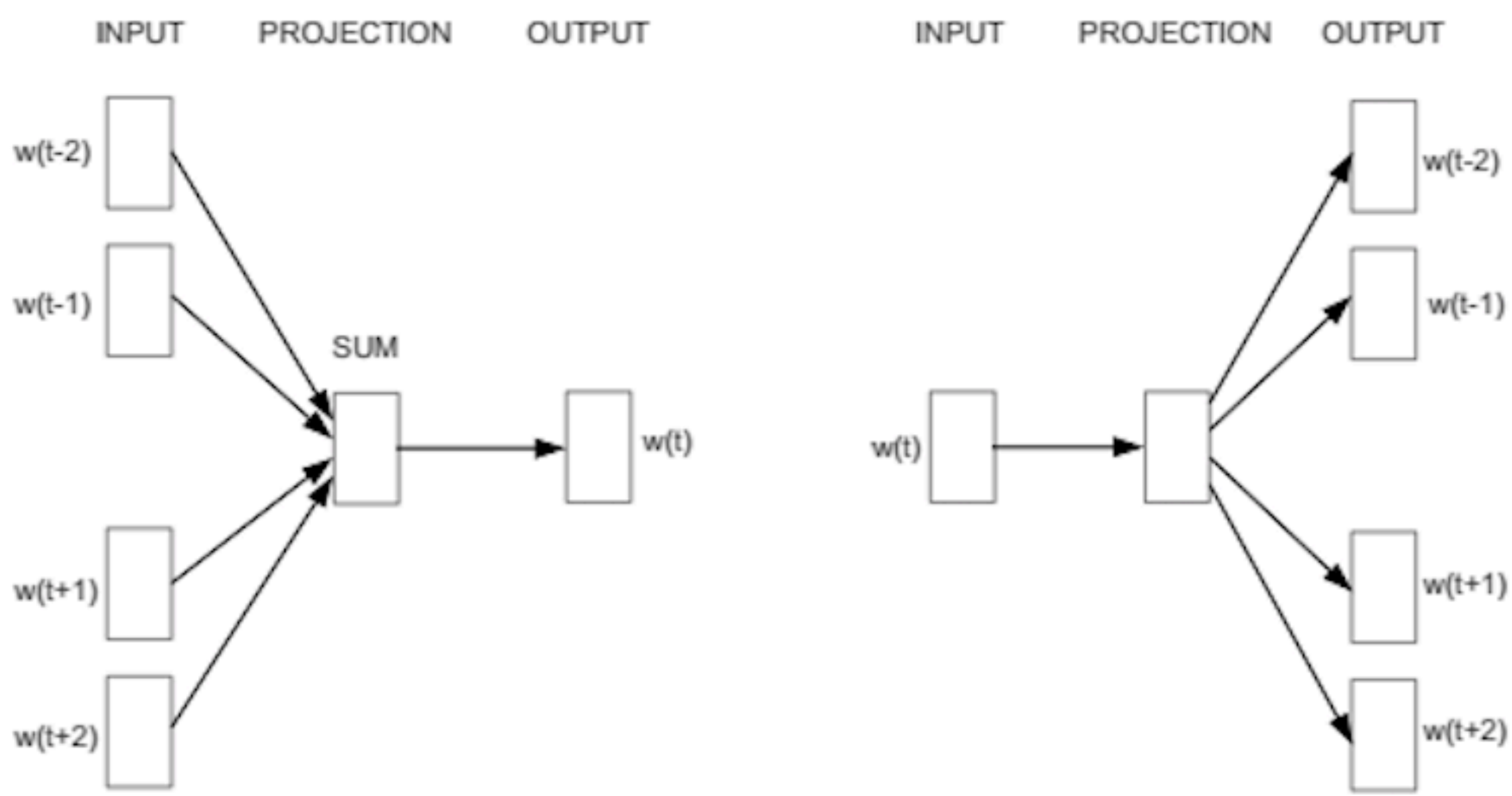


image: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

word2vec models



Continuous-Bag-of-Words

given the context,
predict the middle word

Skip-gram

given the middle word,
predict the context

word2vec results

Examples of learned relationships from
Google News text (~1.6B words):

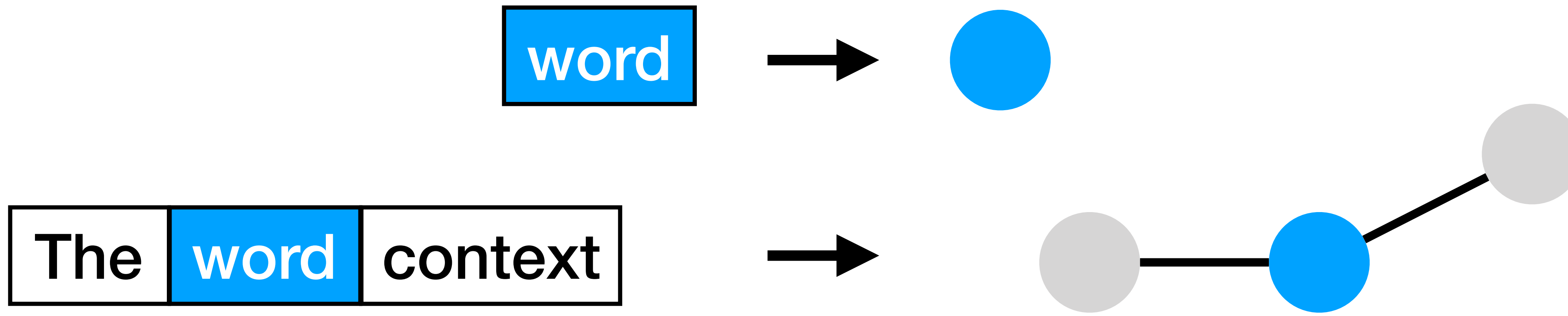
Table 8: *Examples of the word pair relationships, using the best word vectors from Table 4 (Skip-gram model trained on 783M words with 300 dimensionality).*

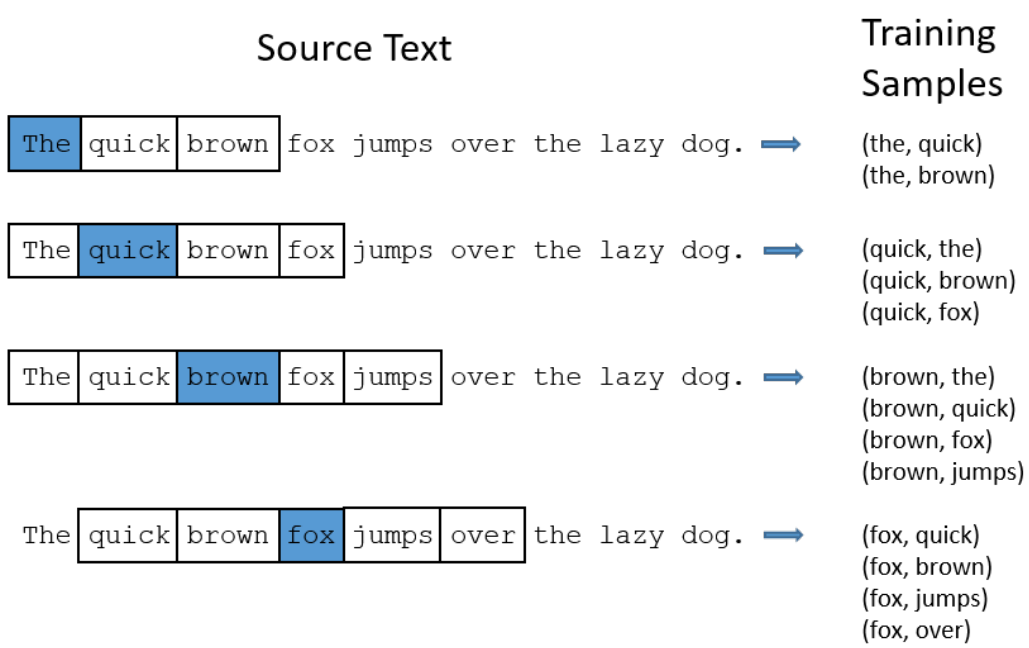
Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

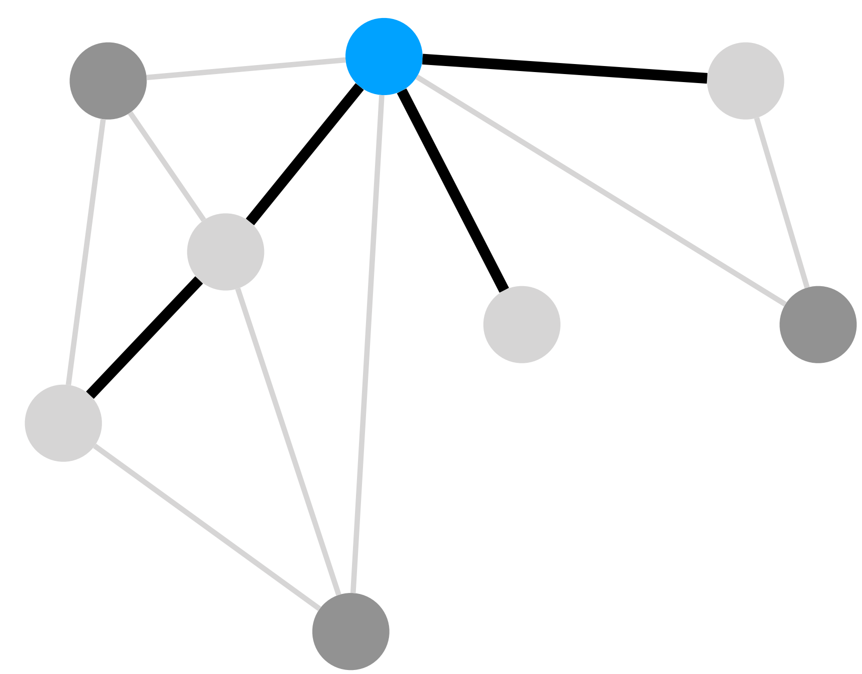
Paris - France + Italy = Rome

other amusing word2vec results:
https://deeplearning4j.org/word2vec#crazy

Feature Learning for Networks







<u>word2vec</u>: learn continuous representations for words based on other words within the same context

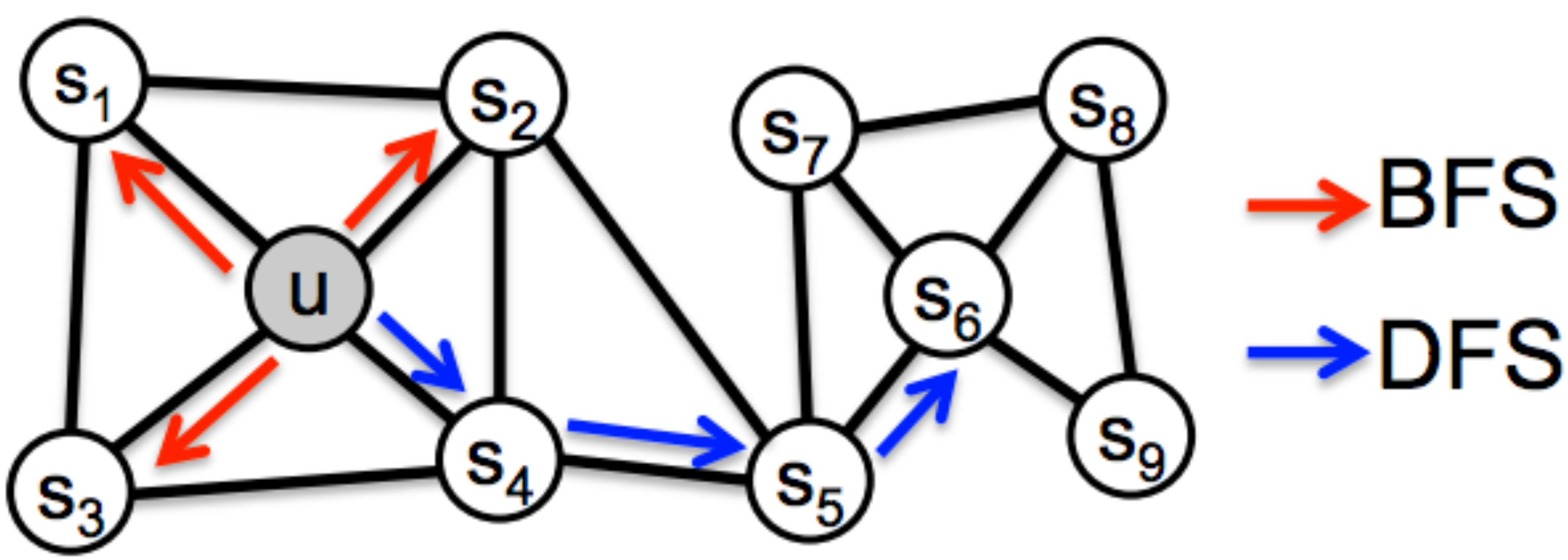
<u>node2vec:</u> learn continuous representations for each node based on their relationships with other nodes

node2vec:
Feature Learning for Networks

1. Generate “context" or node neighborhoods.

    - run biased random walk that uses transition probabilities to interpolate between Breadth First Search (BFS) and Depth First Search (DFS)

2. Implement the Skip-Gram model of word2vec to learn continuous representations for each node.



node2vec: Generating Node Neighborhoods

- select random walk parameters:
  - Return parameter, **p** : likelihood of immediately returning to a node
  - In-out parameter, **q** : bias random walk to differentiate between “inward“ and “outward” nodes

more BFS sampling:
Low p (< min(q, 1))
High q (> 1)

more DFS sampling:
High p (> max(q, 1))
Low q (< 1)

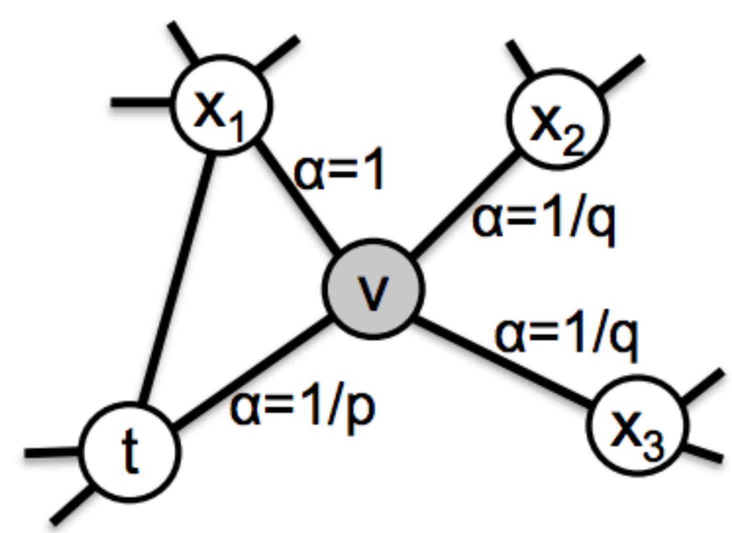
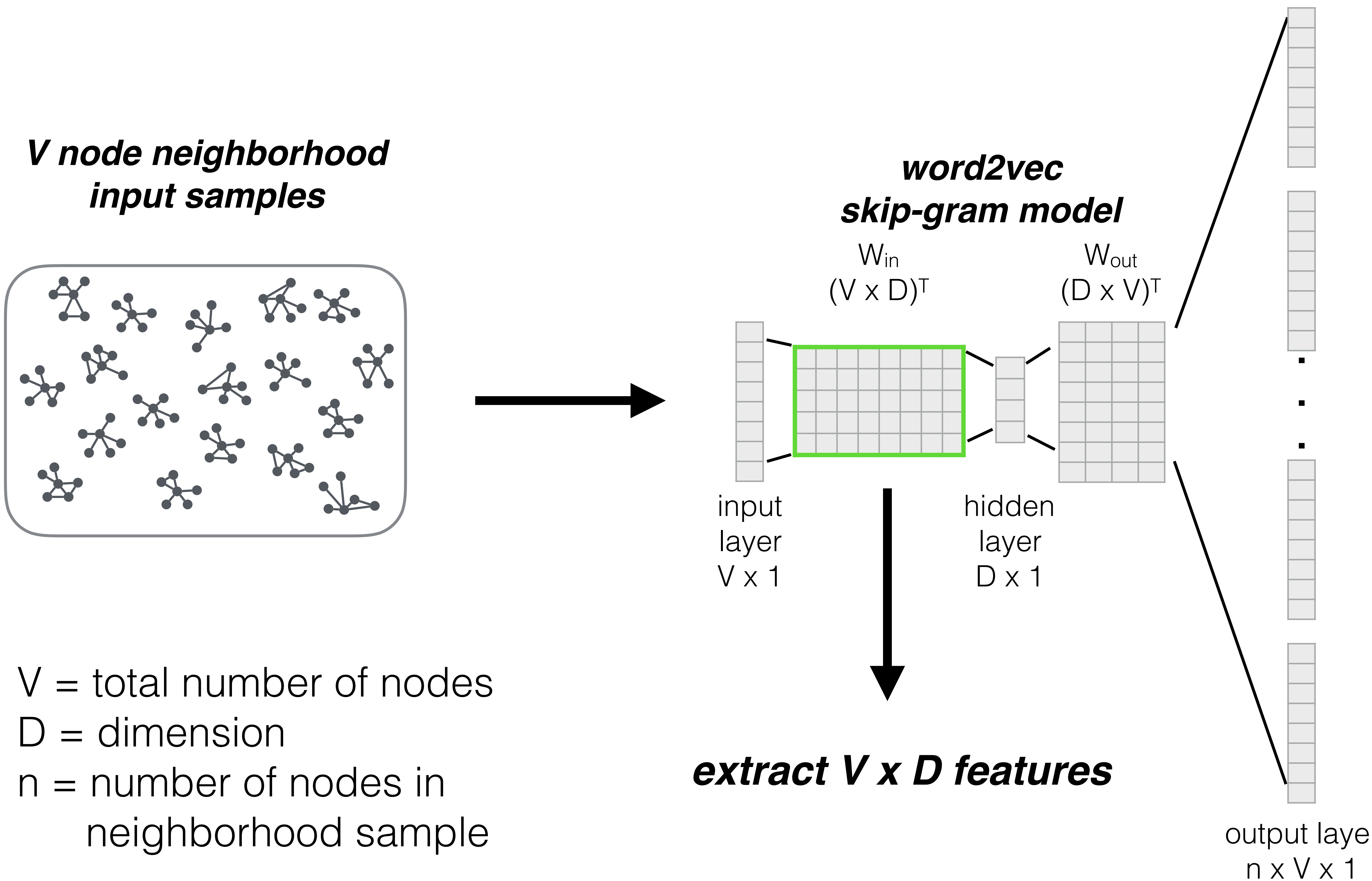


Figure 2: Illustration of the random walk procedure in *node2vec*. The walk just transitioned from t to v and is now evaluating its next step out of node v. Edge labels indicate search biases α.

node2vec:
Learn continuous representations



V = total number of nodes
D = dimension
n = number of nodes in
neighborhood sample

node2vec results: Node Classification

node2vec generated features for Les Miserable Network:

- **(top) p = 1, q = 0.5**
  more DFS exploration revealed homophily structure

- **(bottom) p = 1, q = 2**
  more BFS exploration identified structural equivalence between nodes

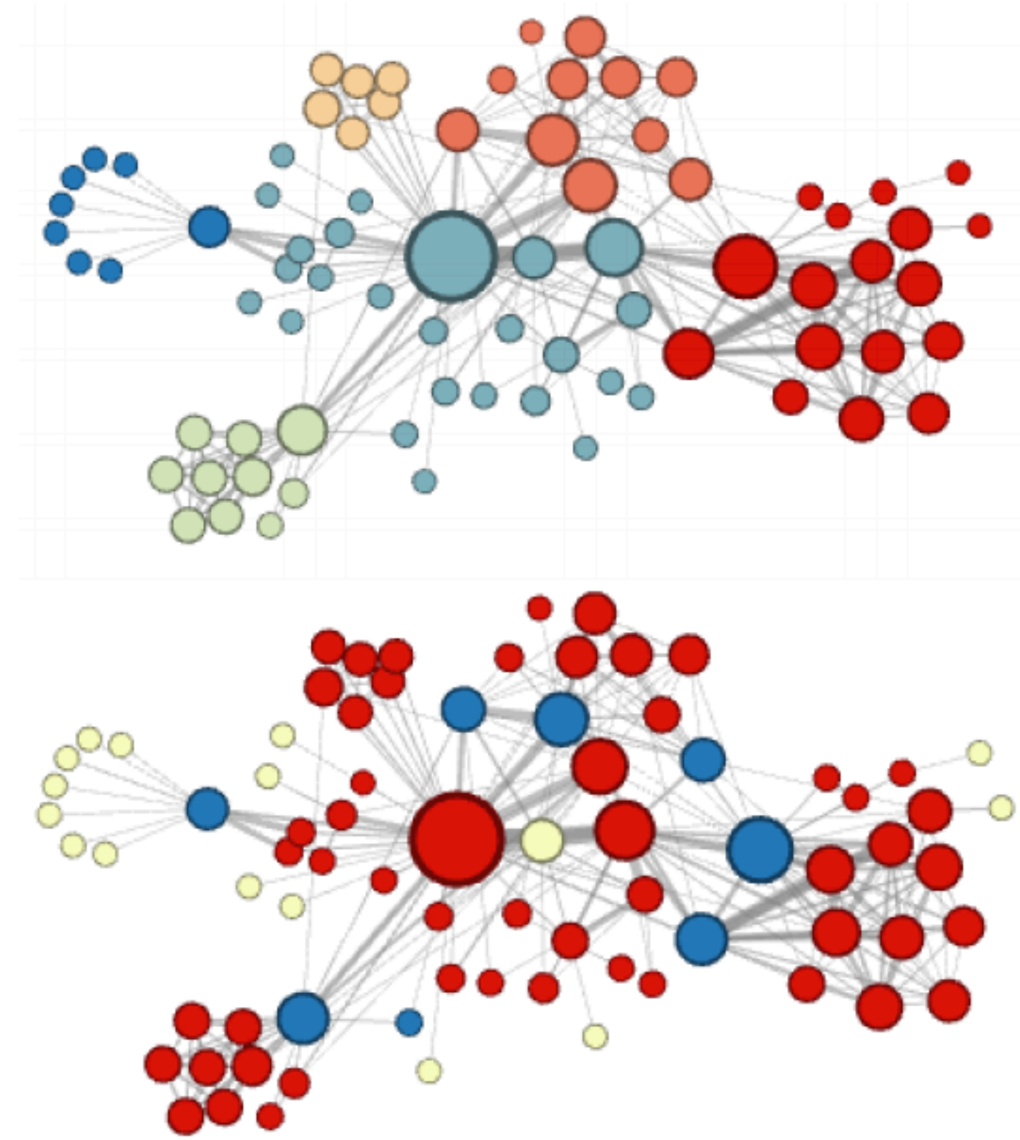


Figure 3: Complementary visualizations of Les Misérables coappearance network generated by *node2vec* with label colors reflecting homophily (top) and structural equivalence (bottom).

node2vec results: Link Prediction

Experiment:
Generate features for subnetworks of Facebook data, protein-protein interaction (PPI), and arXiv citations. Subnetworks contain all nodes, but 50% of edges are removed.

Task: Use the features to predict links between nodes

Op	Algorithm	Dataset		
		Facebook	PPI	arXiv
	Common Neighbors	0.8100	0.7142	0.8153
	Jaccard's Coefficient	0.8880	0.7018	0.8067
	Adamic-Adar	0.8289	0.7126	0.8315
	Pref. Attachment	0.7137	0.6670	0.6996
(a)	Spectral Clustering	0.5960	0.6588	0.5812
	DeepWalk	0.7238	0.6923	0.7066
	LINE	0.7029	0.6330	0.6516
	node2vec	0.7266	0.7543	0.7221
(b)	Spectral Clustering	0.6192	0.4920	0.5740
	DeepWalk	**0.9680**	0.7441	0.9340
	LINE	0.9490	0.7249	0.8902
	node2vec	**0.9680**	**0.7719**	**0.9366**
(c)	Spectral Clustering	0.7200	0.6356	0.7099
	DeepWalk	0.9574	0.6026	0.8282
	LINE	0.9483	0.7024	0.8809
	node2vec	0.9602	0.6292	0.8468
(d)	Spectral Clustering	0.7107	0.6026	0.6765
	DeepWalk	0.9584	0.6118	0.8305
	LINE	0.9460	0.7106	0.8862
	node2vec	0.9606	0.6236	0.8477

Table 4: Area Under Curve (AUC) scores for link prediction. Comparison with popular baselines and embedding based methods bootstapped using binary operators: (a) Average, (b) Hadamard, (c) Weighted-L1, and (d) Weighted-L2 (See Table 1 for definitions).

Resources

node2vec paper:
https://cs.stanford.edu/people/jure/pubs/node2vec-kdd16.pdf

word2vec papers:
https://arxiv.org/pdf/1301.3781.pdf

https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

word2vec skip-gram model explained :
http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

other amusing word2vec results :
https://deeplearning4j.org/word2vec#crazy

word2vec in R :
https://github.com/bmschmidt/wordVectors