Zie github repo Joost Bastings voor de updates (ook notebooks over networkX en T-NSE en Gensim)

**T-NSE**

Lijkt op PCA qua functionaliteit, maar kan ook non-lineaire input aan. (bijvoorbeeld een spiral)

**Hoe de data werkt:**

1e zin: ‘forces’ is het 6e woord en is dependent van het 7e woord. Van pos-tags rij, de 1e gebruiken.

Wat we gebruiken:

* Het getal van het hoeveelste woord het is. (0)
* Het woord zelf (niet de ‘stam’ die er naast staat) (1)
* De pos-tag (van de eerste kolom van pos-tags) (3)
* Het getal waar het de dependend van is, waar de arc vandaan komt (6)
* Het label dat er naast staat (7)

**Questions during lecture (and answers):**

**Hoe werken die word-embeddings, wat houdt al dat jargon in?**

* Make a one-hot vector with the size of the vocabulary :|V|. e.g. cat is [0,0,…1,0,…,0]
* Word embedding matrix W of D (=… )by |V|.

Deze word embedding matrix (W\_emb) kun je op drie manieren krijgen:

Van het internet halen (die embeddings zijn door ‘experts’ al getraind).   
Random beginnen en trainen  
Van internet halen én trainen

* Do the dotproduct W\*V\_cat = a vector of dimension D, which is called the word embedding

MLP(v\_cat) = f(W(W\_emb\*v\_cat)+b)

Voor pos-tag embeddings:

* Kan je op dezelfde manier trainen