Search Powered by Deep Learning

- From Content Similarity to Semantic Similarity

Infosys® | Building Tomorrow's Enterprise

Speakers

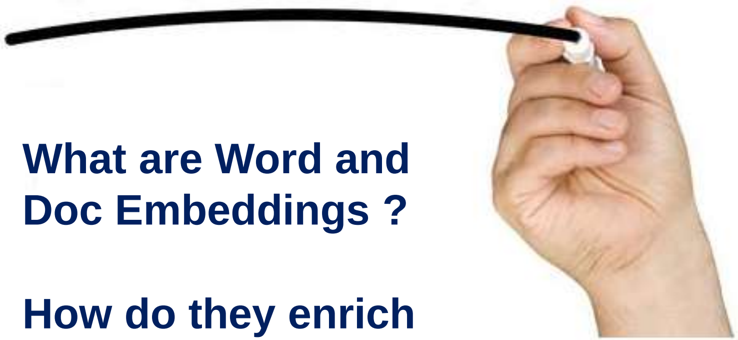
John Kuriakose
Principal Architect
Big Data & Analytics - IIP

Ashish Vishwas Kaduskar
Architect
Big Data & Analytics - IIP

Debanjan Mahata
Senior Research Associate
Big Data & Analytics - IIP

Infosys Information Platform (IIP)

Agenda



- What are Word and Doc Embeddings ?
- How do they enrich Search Applications ?
- How to build them for a search application ?
- How to integrate them in a search application ?

Searching across Research Articles

recurrent neural networks

1. Recognizing recurrent neural networks (rRNN): Bayesian inference for recurrent neural networks
["Bitzer, Sebastian","Kiebel, Stefan J."] - Fri Jan 20 00:00:00 UTC 2012
["recurrent neural networks","bayesian inference","computational neuroscience","machine learning applications","rnn","nonlinear function","brain...
Recurrent neural networks (RNNs) are widely used in computational neuroscience and machine learning applications. In an RNN, each neuron computes its output as a nonlinear function of its integrated input. While the...

2. Conversion of Artificial Recurrent Neural Networks to Spiking Neural Networks for Low-power Neuromorphic Hardware
["Diehl, Peter U.","Zarrella, Guido","Cassidy, Andrew","Pedroni, Bruno U.","Neftci, Emre"] - Sat Jan 16 00:00:00 UTC 2016
["rnn","artificial recurrent neural networks","low-power neuromorphic hardware","neuromorphic low-power systems","significant momentum","recurrent...
In recent years the field of neuromorphic low-power systems that consume orders of magnitude less power gained significant momentum. However, their wider use is still hindered by the lack of algorithms that can harness the...

3. Sequence Modeling using Gated Recurrent Neural Networks
Pezeshki, Mohammad - Thu Jan 01 00:00:00 UTC 2015
["gated recurrent neural networks","recurrent neural networks","human motion data","next immediate data point","recently proposed gated recurrent units","promisi...
In this paper, we have used Recurrent Neural Networks to capture and model

Ingestion

- Data Preprocessing
- Indexing

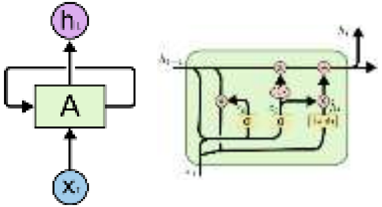
Enrichment

- Training different word embedding models

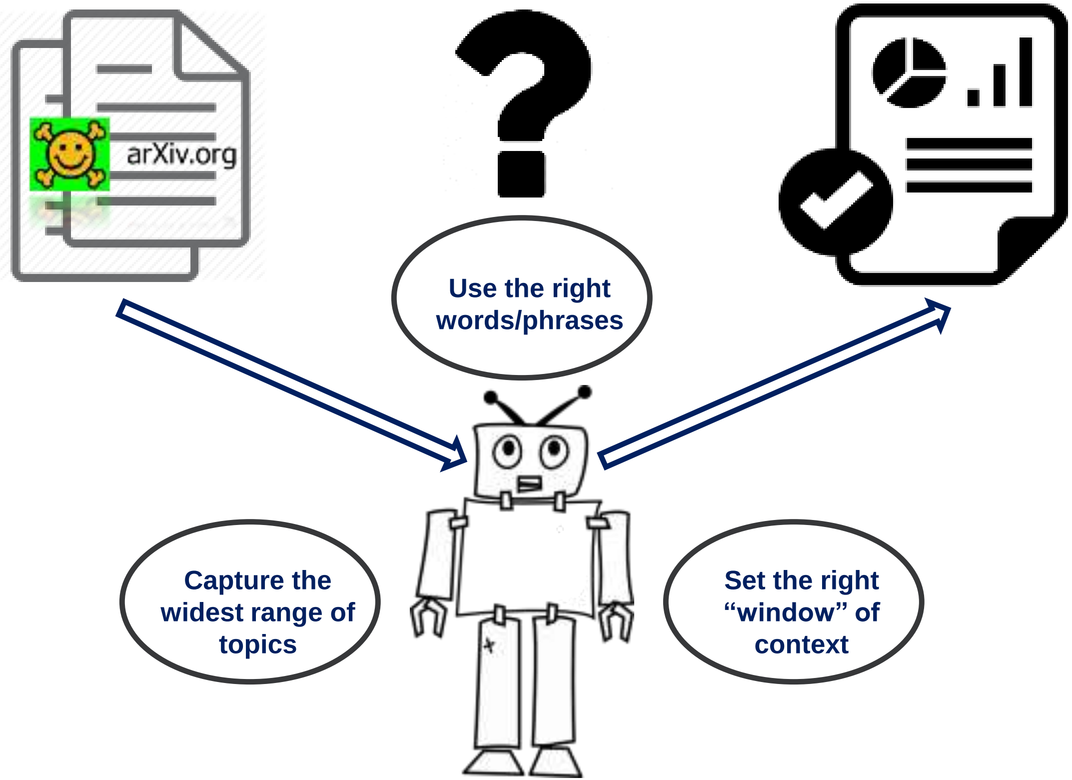
Content Selection

- Query Expansion
- Similar Article Recommendation
- Keyword Extraction

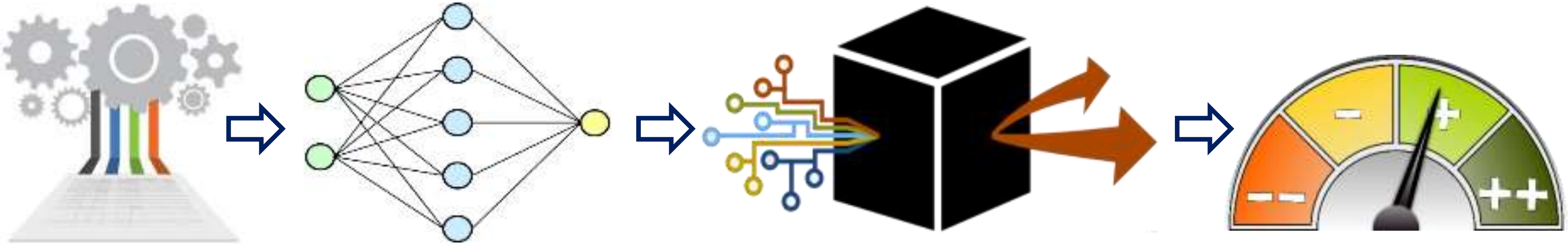
Humans Vs Machines



1147001 Research Abstracts



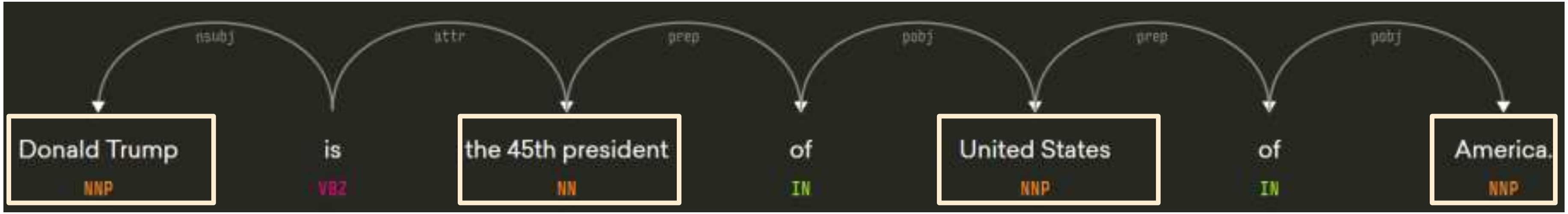
Building the Models

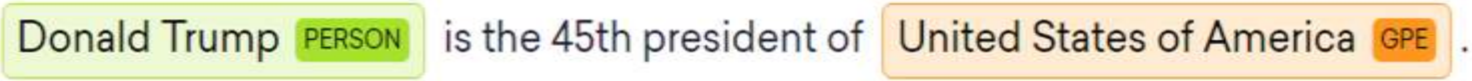


Preprocessing Training Prediction Evaluation

gensim
topic modelling for humans

Preprocessing





Sentence splitting

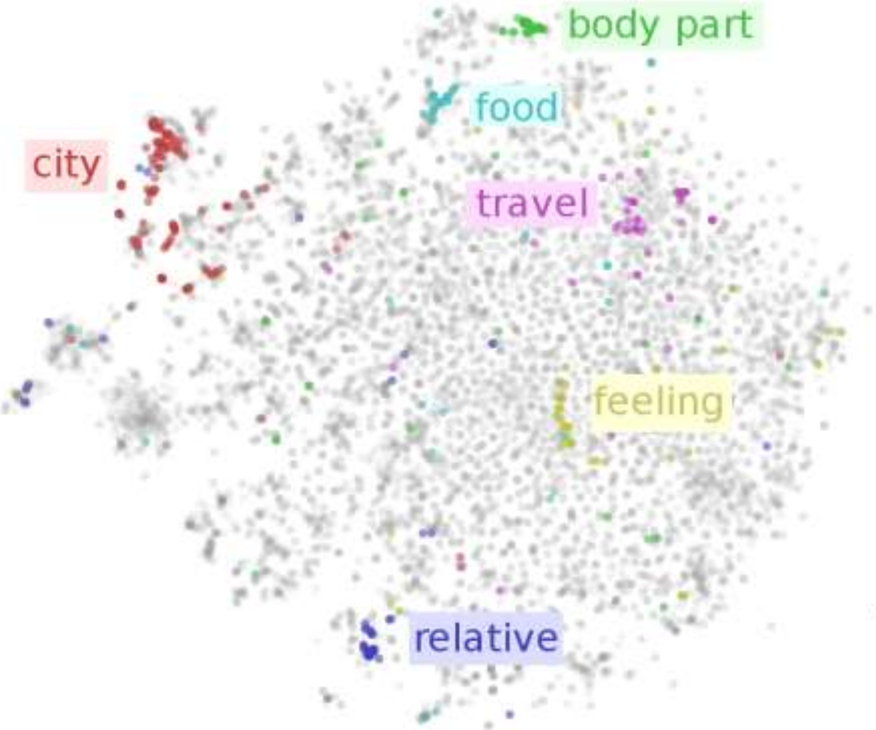
Phrase Tokenization

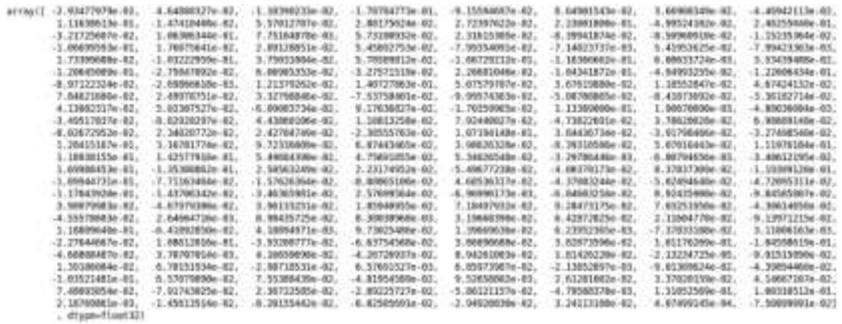
Removal of strings containing only numeric characters

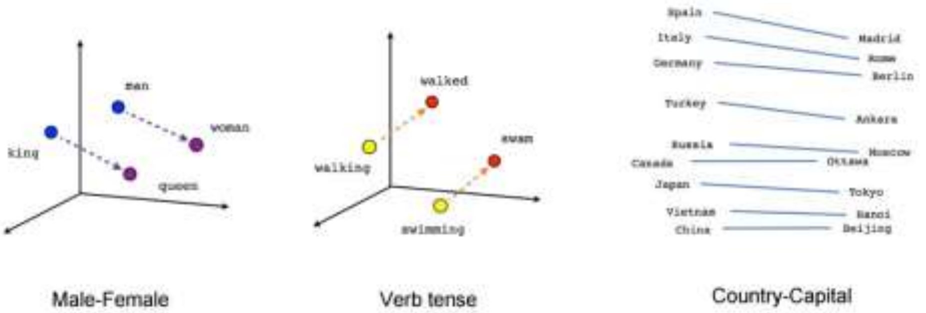
Removal of functional words like ‘accordingly’, ‘although’, etc

Removal of named entities of types “DATE”, “TIME”, “PERCENT”, “MONEY”, “QUANTITY”, “ORDINAL”, “CARDINAL”

Word Embeddings



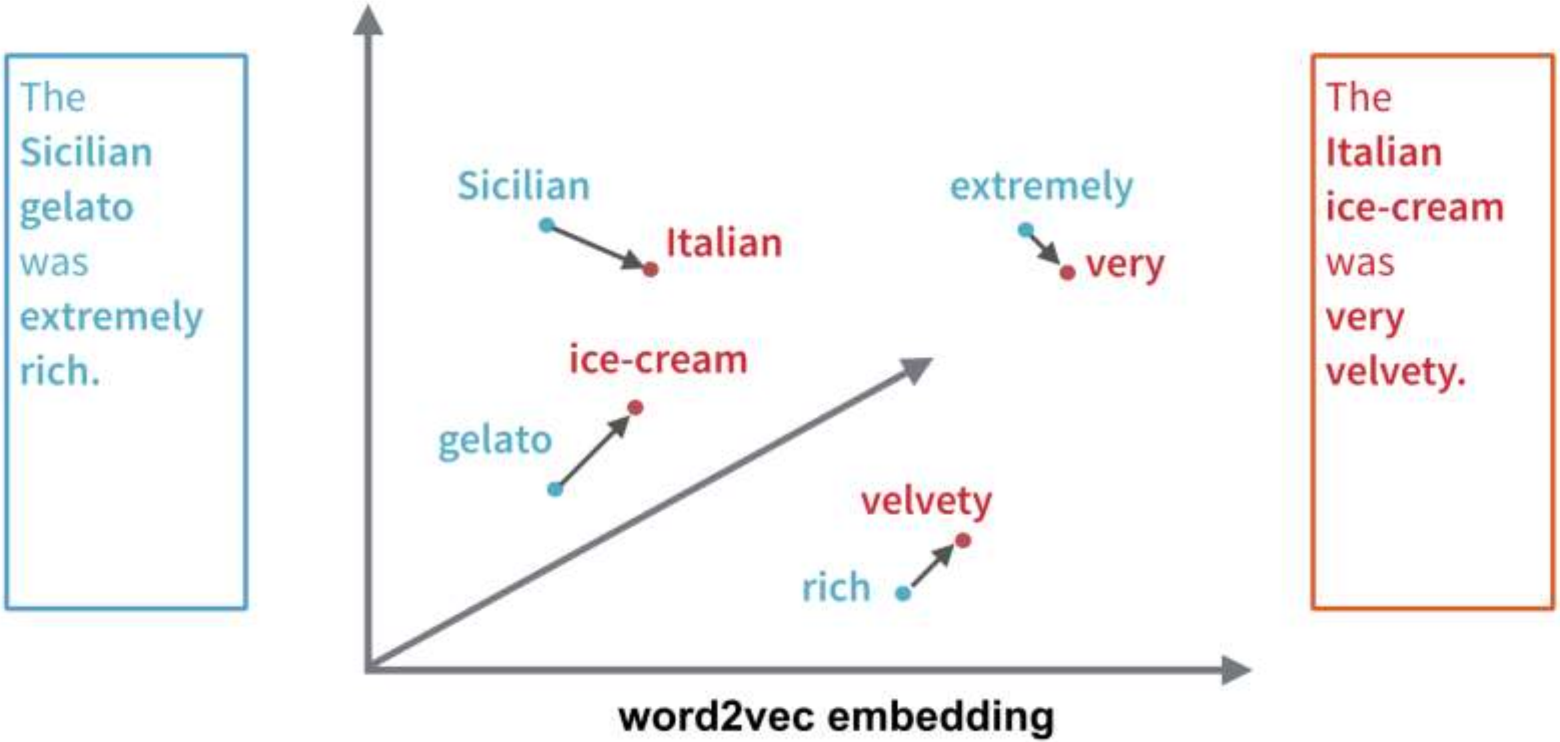




Source: http://sebastianruder.com/word-embeddings-1/index.html

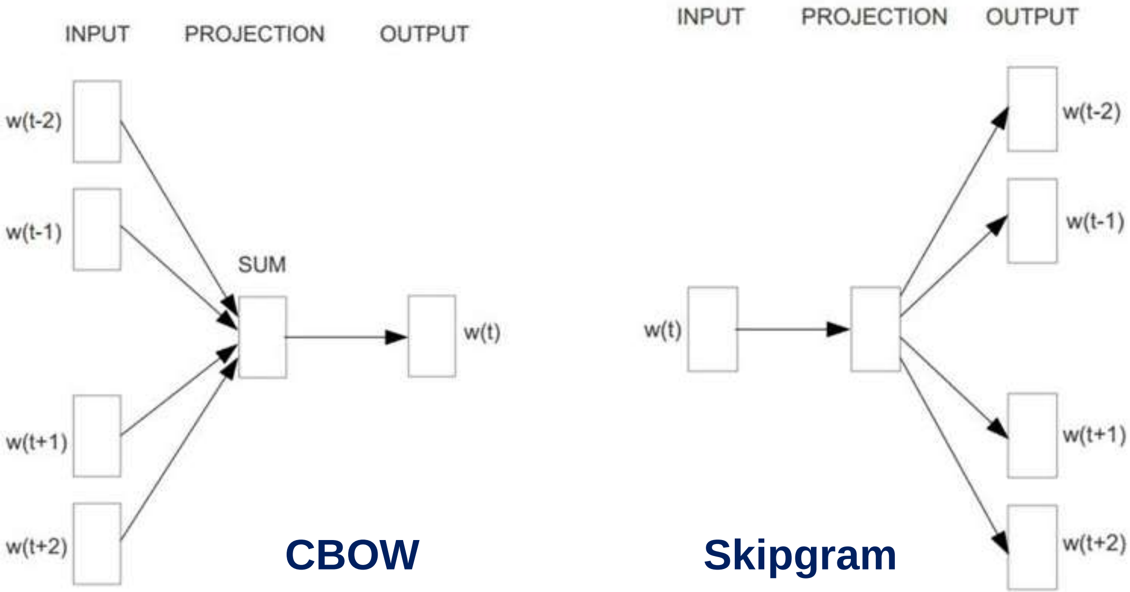
Mikolov et al, (2013a).

Semantic Similarity between Words



Kusner et al (2015)

Word2Vec



CBOW **Skipgram**

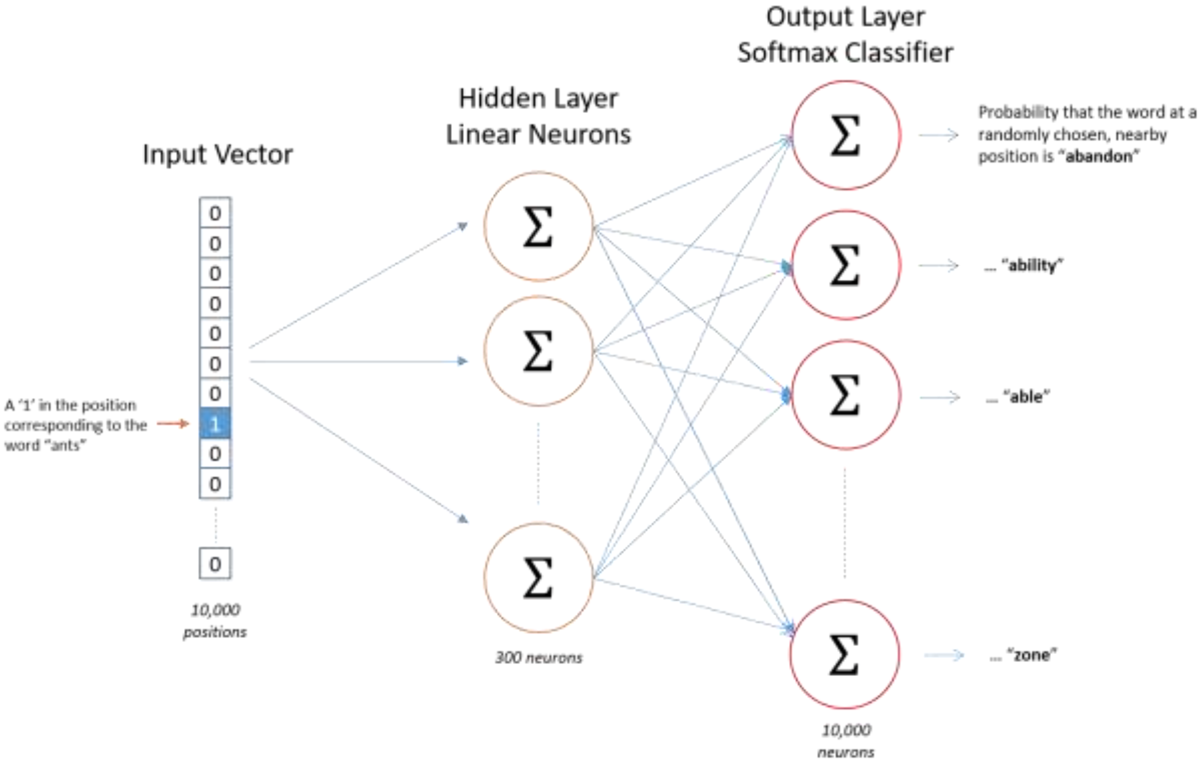
Mikolov et al, (2013a).

Word2Vec Training

Source Text	Training Samples
[**The** quick brown] fox jumps over the lazy dog. ⟹	(the, quick) (the, brown)
[The **quick** brown fox] jumps over the lazy dog. ⟹	(quick, the) (quick, brown) (quick, fox)
[The quick **brown** fox jumps] over the lazy dog. ⟹	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The [quick brown **fox** jumps over] the lazy dog. ⟹	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

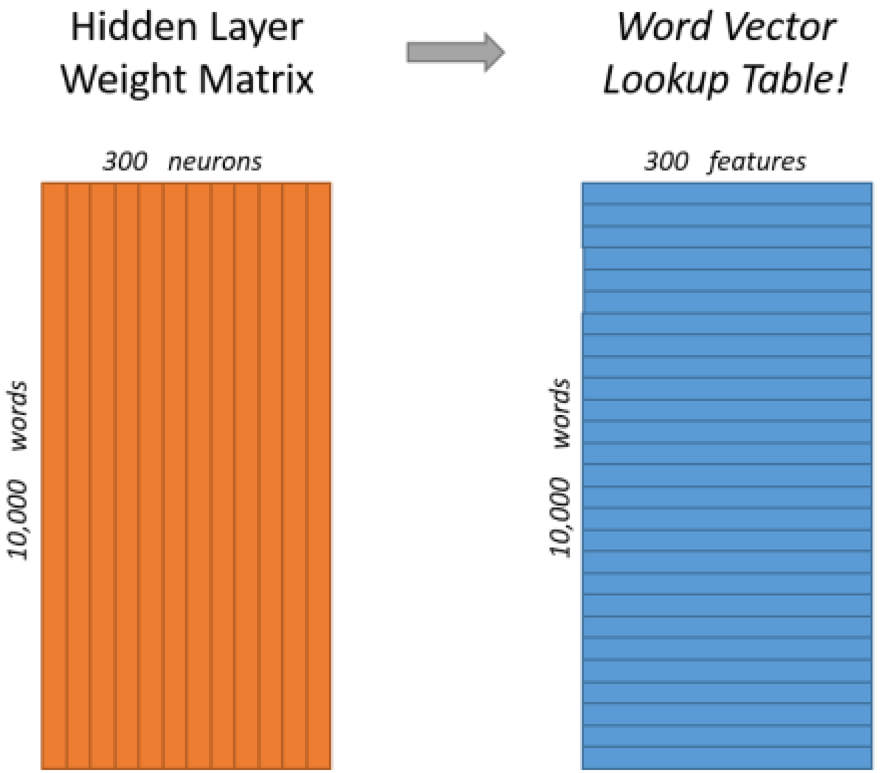
Source: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Word2Vec Training



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

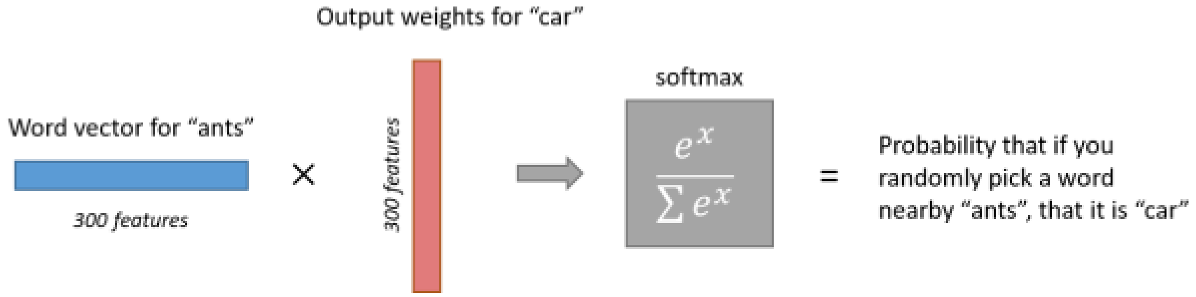
Word2Vec Training



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

Word2Vec Training

$$\begin{bmatrix} 0 & 0 & 0 & \boxed{1} & 0 \end{bmatrix} \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ \boxed{10} & \boxed{12} & \boxed{19} \\ 11 & 18 & 25 \end{bmatrix} = \begin{bmatrix} 10 & 12 & 19 \end{bmatrix}$$



Negative Sampling

$$\log\sigma({v'_{w0}}^{T} v_{wI}) + \sum_{i=1}^{k} \mathbb{E}_{wi} \sim P_n(w)[\log\sigma({v'_{w0}}^{T} v_{wI})]$$

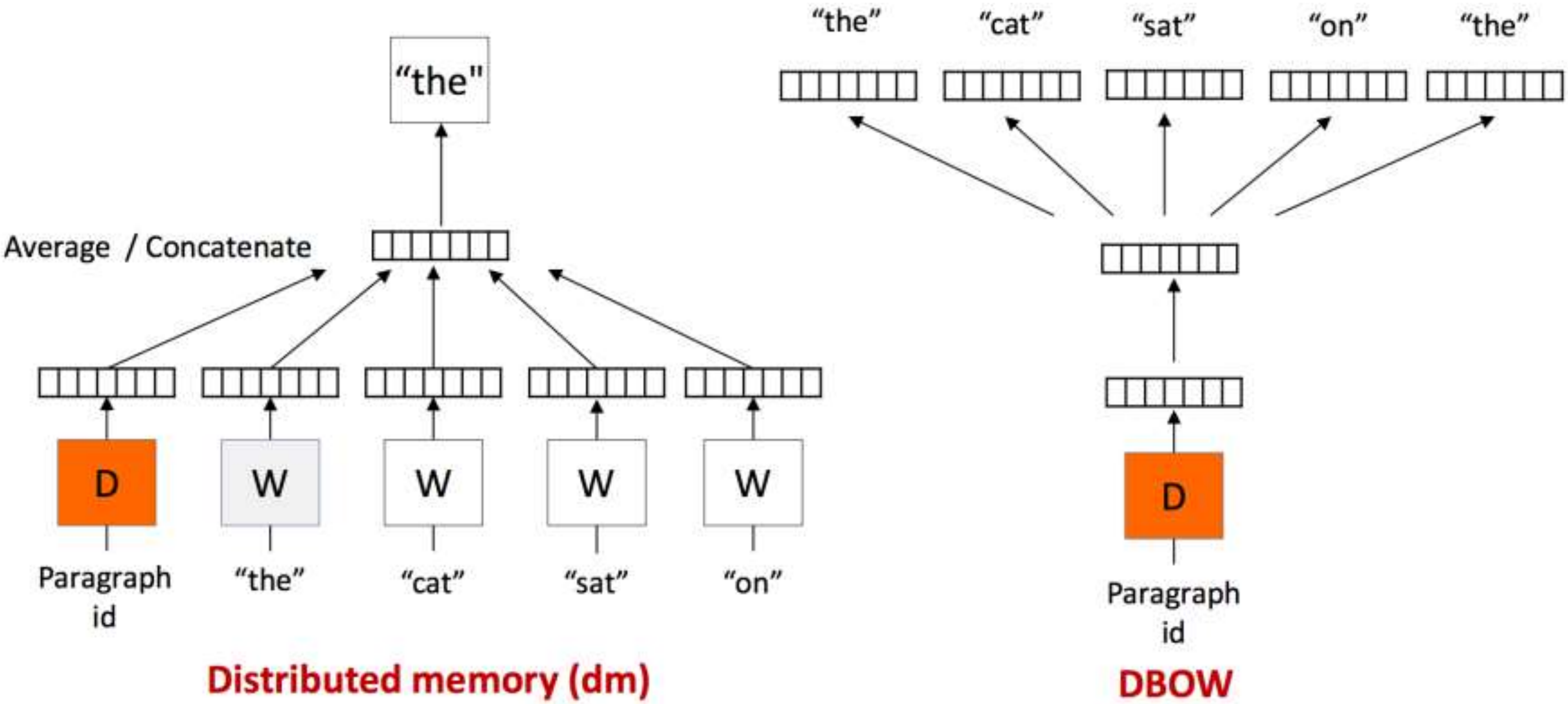
$$P_n = U(s)^{3/4}/Z$$

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

Fasttext

- Very Similar to Word2vec
- Primary Difference
  - Takes into account the internal structure of words while learning word representations
  - The resultant word vector is a combination of vectors for its constituent character ngrams
- Extremely fast training
- Very good for morphologically rich languages
- Takes into account both “Semantic as well as Syntactic Similarity”
- Very good performances in Syntactic Similarity tasks

Doc2Vec

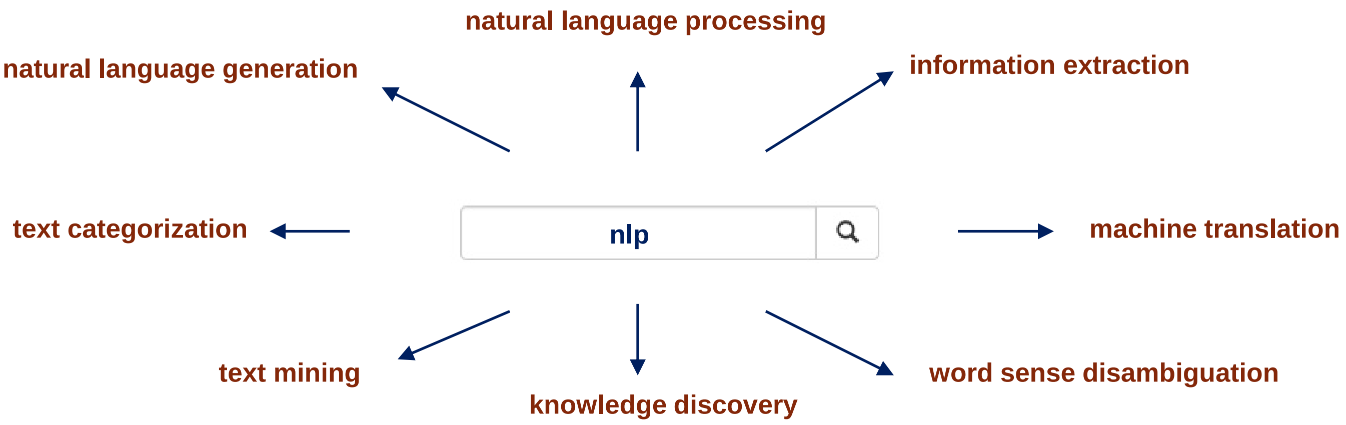


Mikolov et al. (2014)

Training Parameters

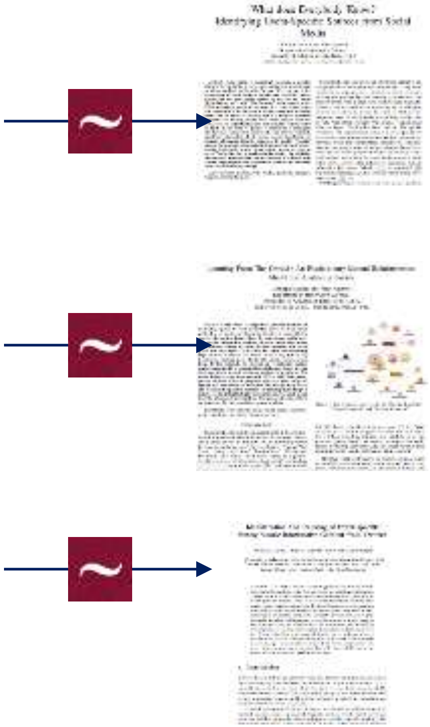
Word2vec	Fasttext	Doc2Vec
• **Skipgram** • **Negative Sampling** • **Dimensions =** 1000 • **Context Window Size =** 5 • **Learning Rate =** 0.025 • **Trained on unigrams and phrases**	• **Skipgram** • **Negative Sampling** • **Dimensions =** 1000 • **Max Length of Char Ngrams =** 6 • **Min Length of Char Ngrams =** 3 • **Learning Rate =** 0.05 • **Trained on unigrams and phrases**	• **Distributed Memory** • **Dimensions =** 1000 • **Window Size =** 10 • **Epochs =** 10 • **Initial Learning Rate =** 0.025 • **Trained on unigrams and phrases**

Query Expansion



- Getting rid of thesaurus based or dictionary based query expansion.
- How many phrases to use for expansion ?
- Which model ?
- How to integrate the trained models with a search application for query expansion ?

Recommending Similar Research Articles



Content Similarity

Vs

Semantic Similarity

Ranked Keyphrase Extraction

Supervised	Unsupervised

• Known to give better results
• Drawbacks
  – Domain Specific
  – Training and Tuning of the models for generalization
  – Intelligent Feature Engineering
• Examples: KEA, MAUI
	• Known to give worse results than supervised in domain specific tasks
	• Domain independent
	• No need of feature engineering
	• Algorithm determines relationships between candidates for identifying the keyphrases
	• Examples: TextRank, RAKE

Ranked Keyphrase Extraction

Research Article → Text Preprocessing → Candidate Phrase Selection → Supervised Model

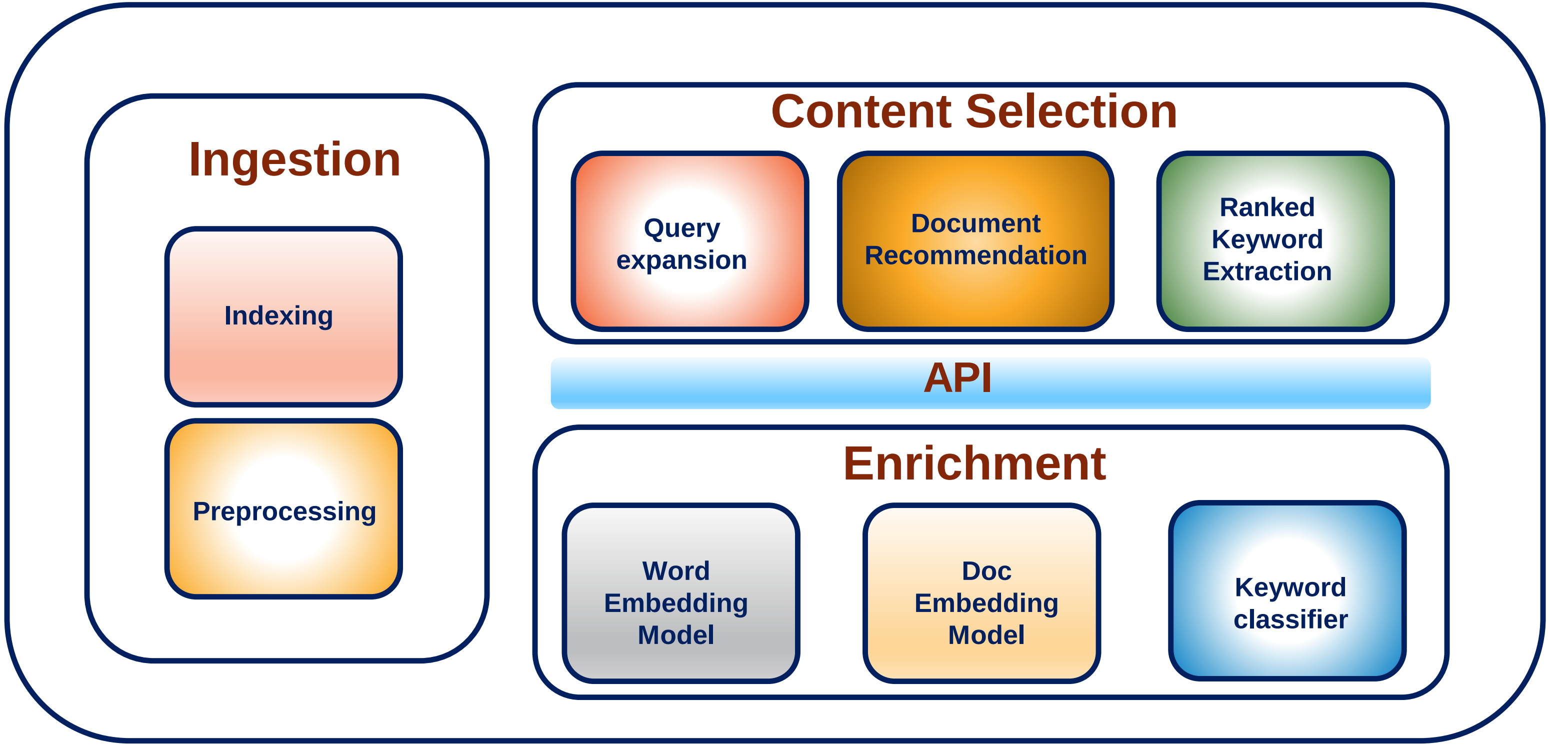
Supervised Model → Initialize Candidate Phrases with Scores from Classifier → Build Strongly Connected Graph between the Candidate Phrases → Ranked Keyphrases

Word2Vec Model

Phrase 1
Phrase 2
●
●
●
Phrase K

Ranked Keyphrases

Integration




Start
DEMO

References

1. Sebastian Rudder, “**On Word Embeddings**”; http://sebastianruder.com/tag/word-embeddings/index.html
2. Christopher Olah, “**Colah’s Blog**”, http://colah.github.io/
3. Chris McCormick, “**Word2Vec Tutorial – The Skip-Gram Model**”, http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
4. Le, Quoc V., and Tomas Mikolov. "**Distributed Representations of Sentences and Documents.**" ICML. Vol. 14. 2014.
5. Mikolov, Tomas, et al. "**Distributed representations of words and phrases and their compositionality.**" Advances in neural information processing systems. 2013.
6. Bojanowski, Piotr, et al. "**Enriching word vectors with subword information.**" arXiv preprint arXiv:1607.04606 (2016).
7. Rong, Xin. "**word2vec parameter learning explained**." arXiv preprint arXiv:1411.2738 (2014).


Q&A

thank you!