Natural language processing

обзор задач и некоторых эмбеддингов

Computer science

+

Artificial intelligence

+

Linguistics


おはようございます。ジョーンズさん元気ですか。
GOOD MORNING, MR. JONES. HOW ARE YOU TODAY?
DECOSSE

More Deeper Application of NLP

Group 1	Group 2	Group 3
Cleanup, Tokenization	Information Retrieval and Extraction (IR)	Machine Translation
Stemming	Relationship Extraction	Automatic Summarization/ Paraphracing
Lemmatization	Named Entity Recognation (NER)	Natural Language Generation
Part of Speech Tagging	Sentiment Analysis/Sentance Boundary Dismbiguation	Reasoning over Knowledge Based
Query Expansion	World sense and Dismbiguation	Quation Answering System
Parsing	Text Similarity	Dialog System
Topic Segmentationand Recognation	Coreference Resolution	Image Captioning & other Multimodel Tasks
Morphological Degmentation (Word/Sentences)	Discourse Analysis	

Named entity recognition

Stanford Named Entity Tagger

Classifier: english.muc.7class.distsim.crf.ser.gz

Output Format: highlighted

Preserve Spacing: yes

Please enter your text here:

Partial invoice (€100,000, so roughly 40%) for the consignment C27655 we shipped on 15th August to London from the Make Believe Town depot. INV2345 is for the balance.. Customer contact (Sigourney) says they will pay this on the usual credit terms (30 days).

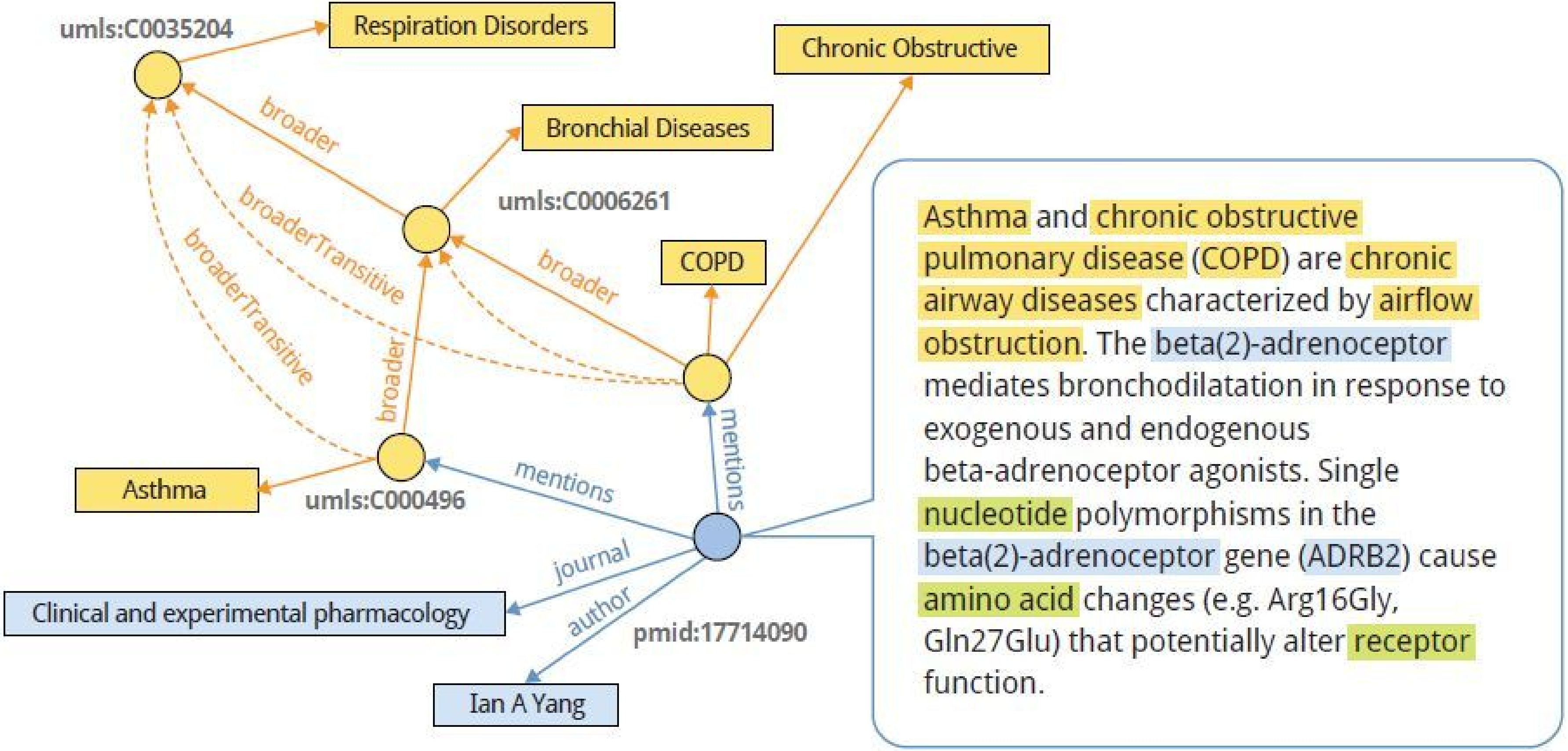
Submit Clear

atomic elements in text

Partial invoice (€100,000, so roughly 40%) for the consignment C27655 we shipped on 15th August to London from the Make Believe Town depot. INV2345 is for the balance.. Customer contact (Sigourney) says they will pay this on the usual credit terms (30 days).

Potential tags:
- LOCATION
- TIME
- PERSON
- ORGANIZATION
- MONEY
- PERCENT
- DATE

predefined categories

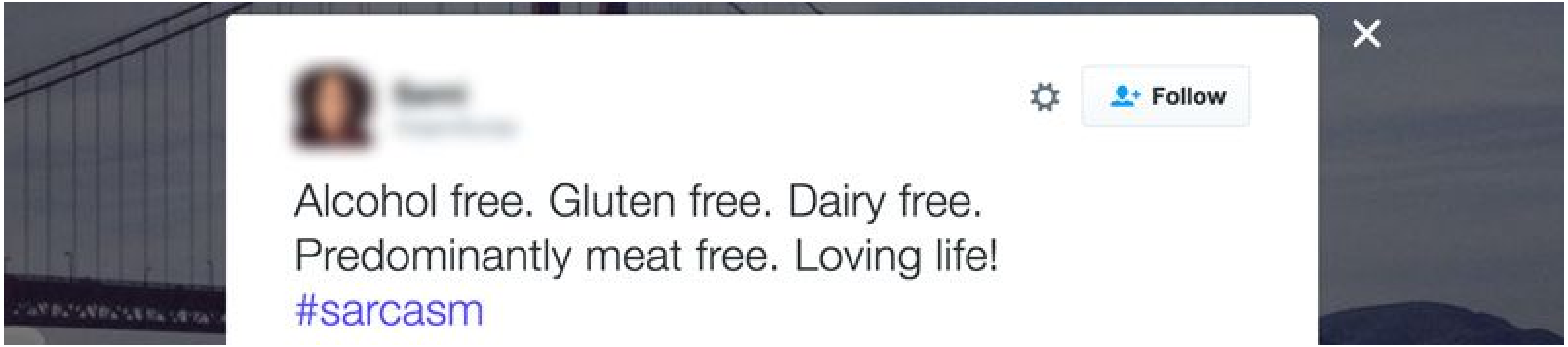


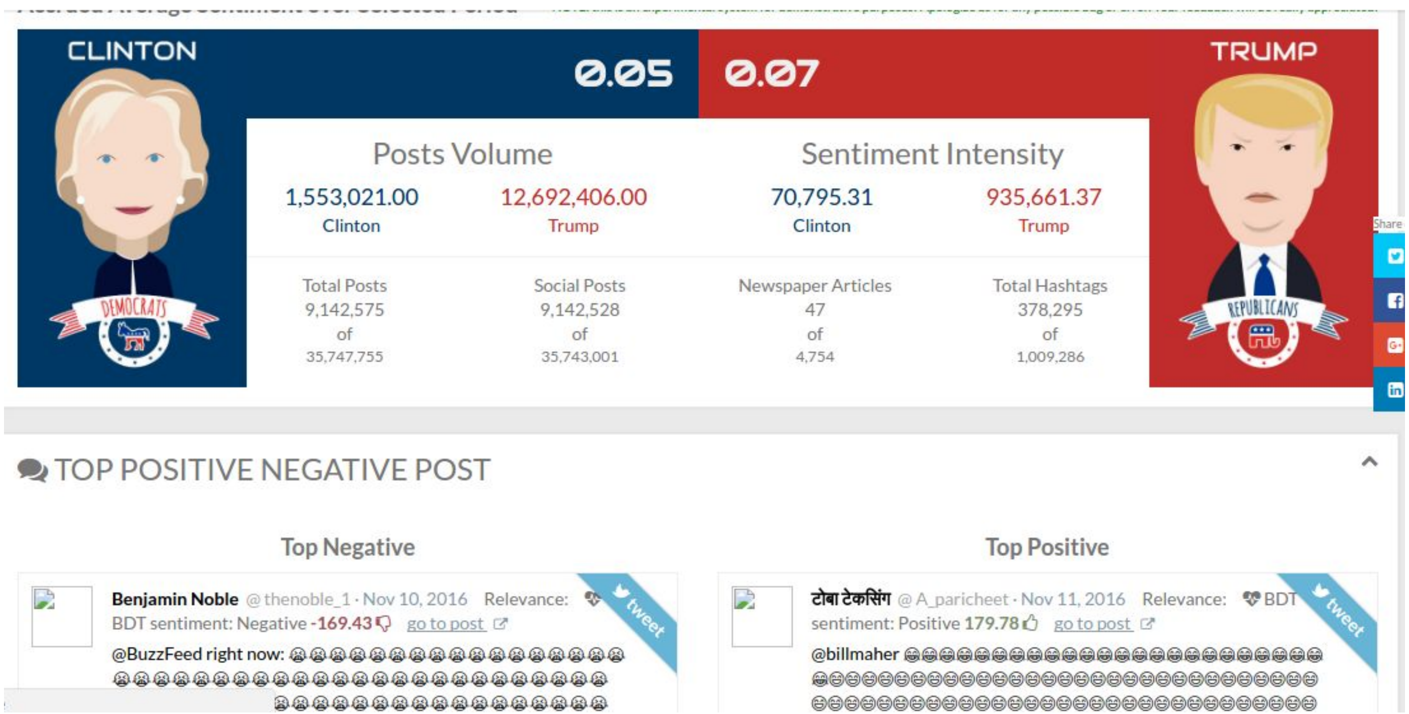
Ontotext's Semantic Biomedical Tagger | use knowledge bases and information extraction algorithms

Sentiment analysis

I like my life!

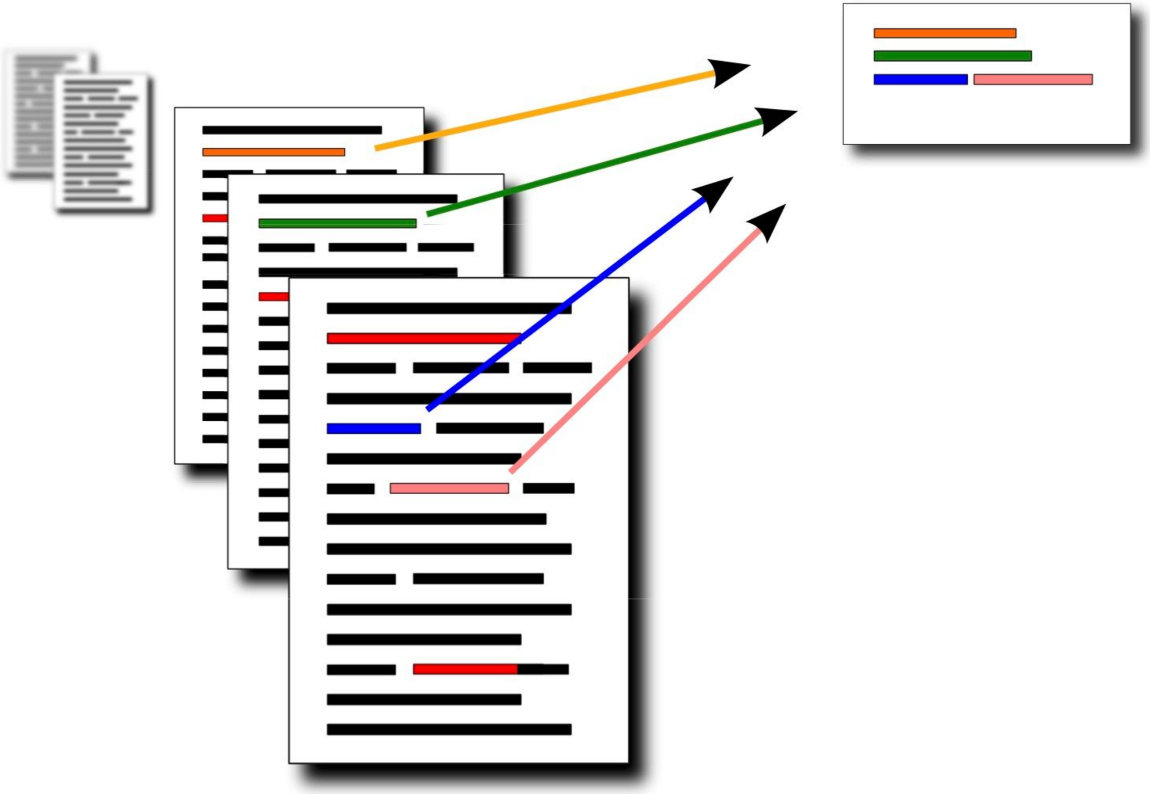
I do not dislike my life.





http://elections.bdt.systems/#/how/introductory

Automatic summarization

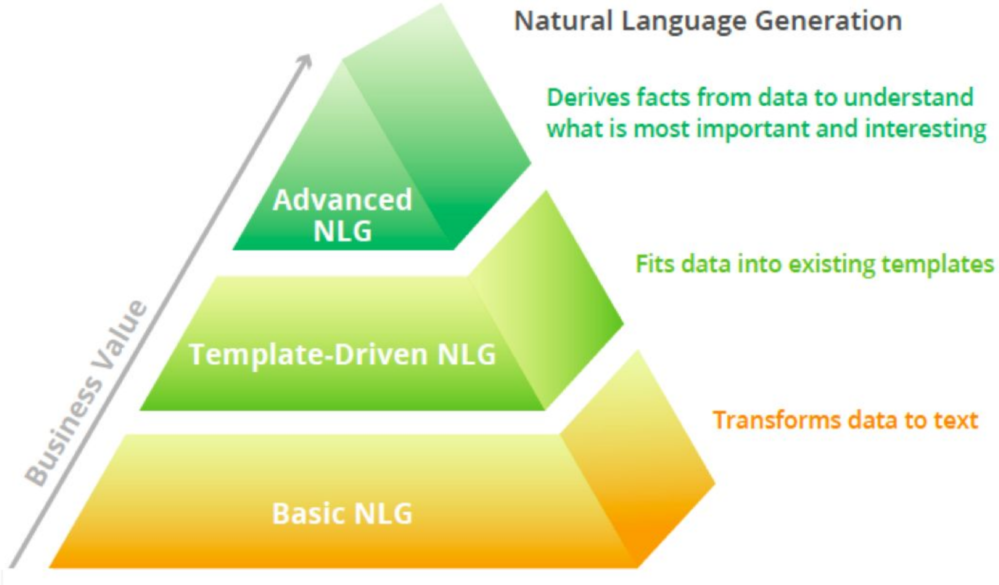


Sifting through lots of documents can be difficult and time consuming.

By extracting important sentences and creating comprehensive summaries, it’s possible to quickly assess whether or not a document is worth reading.

- calculating the word frequencies for the entire text document;
- the N most common words are stored and sorted;
- each sentence is then scored based on how many high frequency words it contains, with higher frequency words being worth more;
- the top X sentences are then taken, and sorted based on their position in the original text.

Natural language generation



[the Associated Press uses NLG] to create its corporate earnings reports.

word2vec

“words which are similar in meaning occur in similar contexts” (Rubenstein & Goodenough, 1965)

“words with similar meanings will occur with similar neighbors if enough text material is available” (Schutze & Pedersen, 1995)

Skip-gram

Given a specific word in the middle of a sentence (the input word), look at the words nearby and pick one at random. The network is going to tell us the probability for every word in our vocabulary of being the “nearby word” that we chose.

“nearby” ⇒ parameter “window size”

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. →

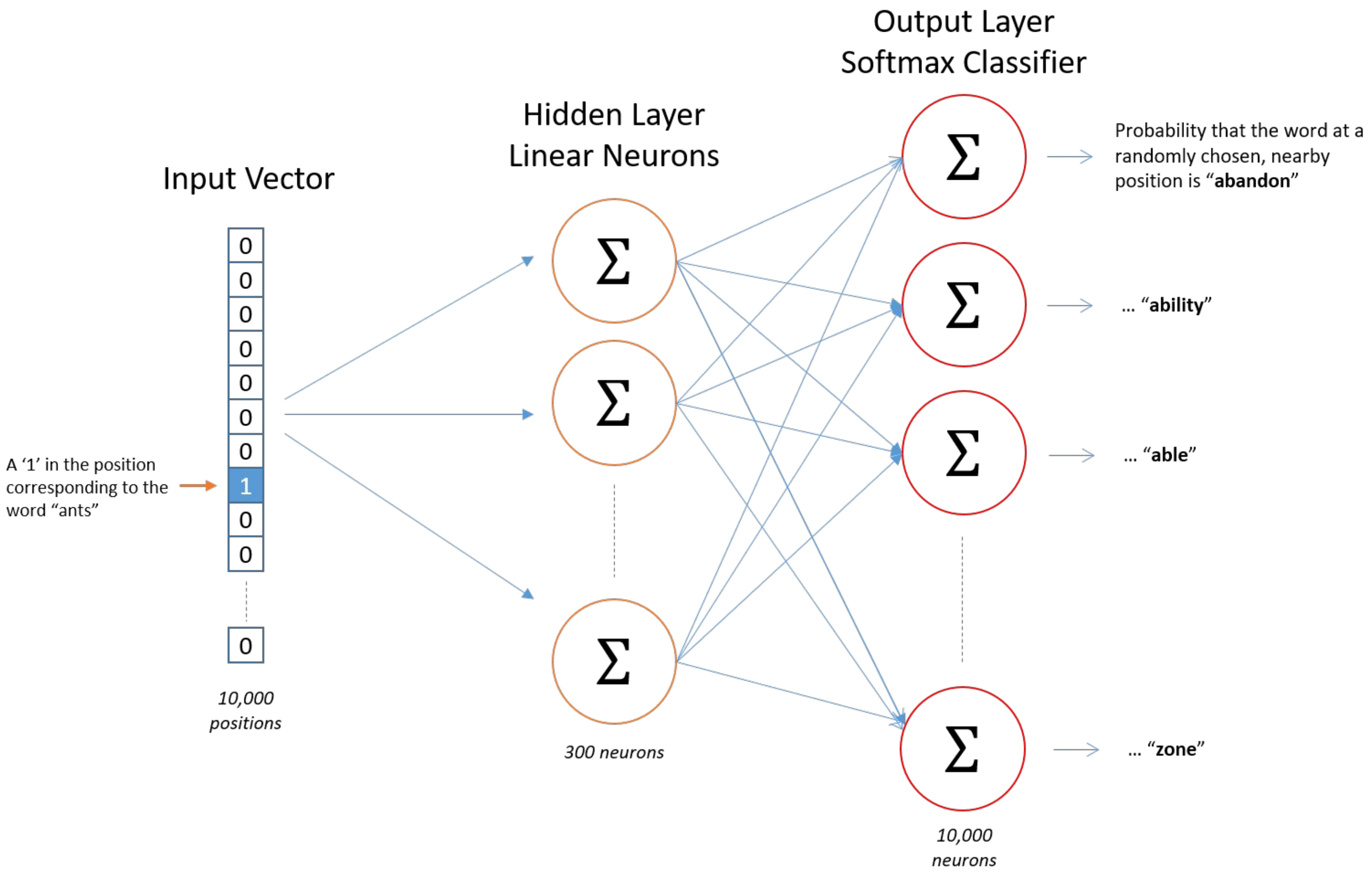
(quick, the)
(quick, brown)
(quick, fox)

The quick brown fox jumps over the lazy dog. →

(brown, the)
(brown, quick)
(brown, fox)
(brown, jumps)

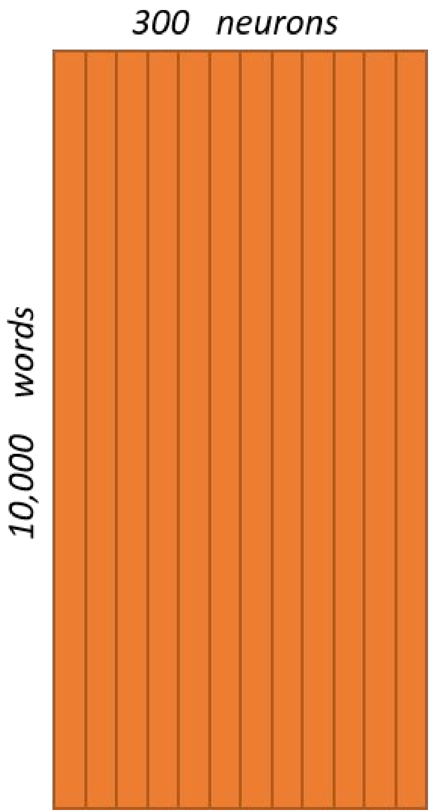
The quick brown fox jumps over the lazy dog. →

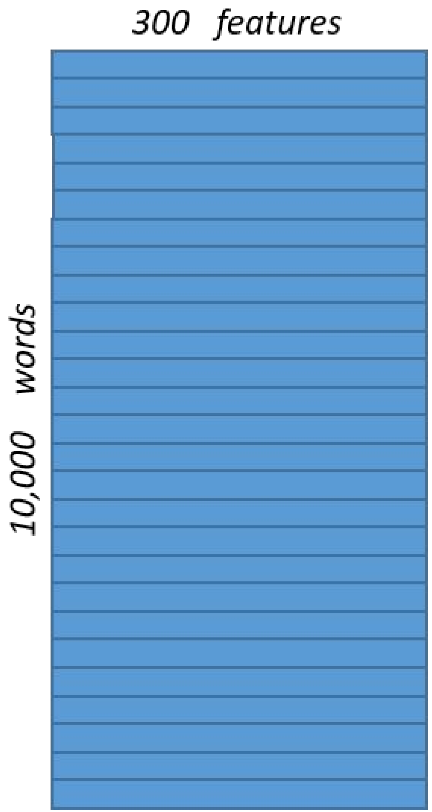
(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)

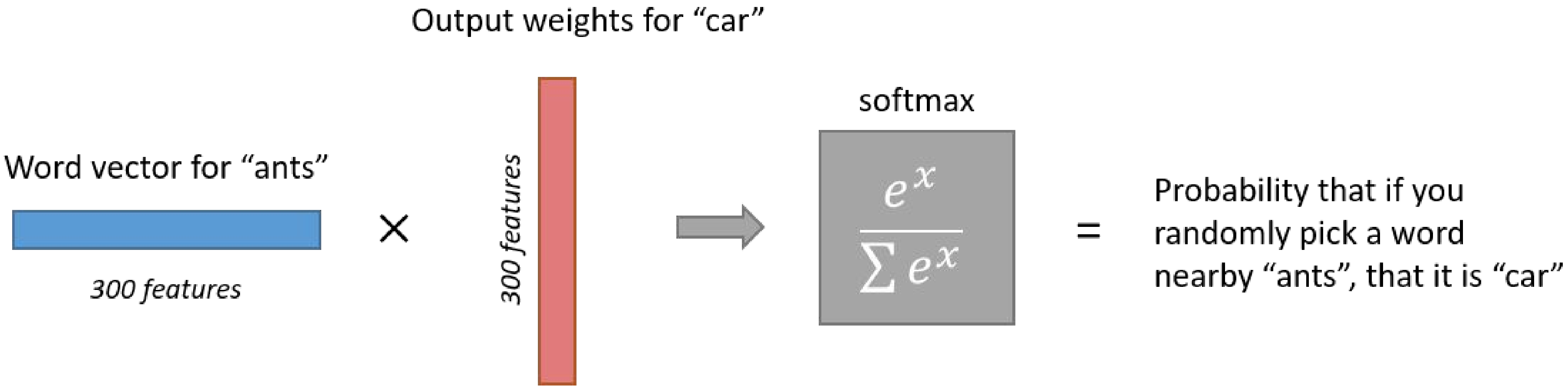

Output Layer
Softmax Classifier
Hidden Layer
Linear Neurons
Input Vector
0
0
0
0
0
0
0
1
0
0
0
A '1' in the position corresponding to the word "ants"
10,000 positions
300 neurons
10,000 neurons
Probability that the word at a randomly chosen, nearby position is "**abandon**"
... "**ability**"
... "**able**"
... "**zone**"

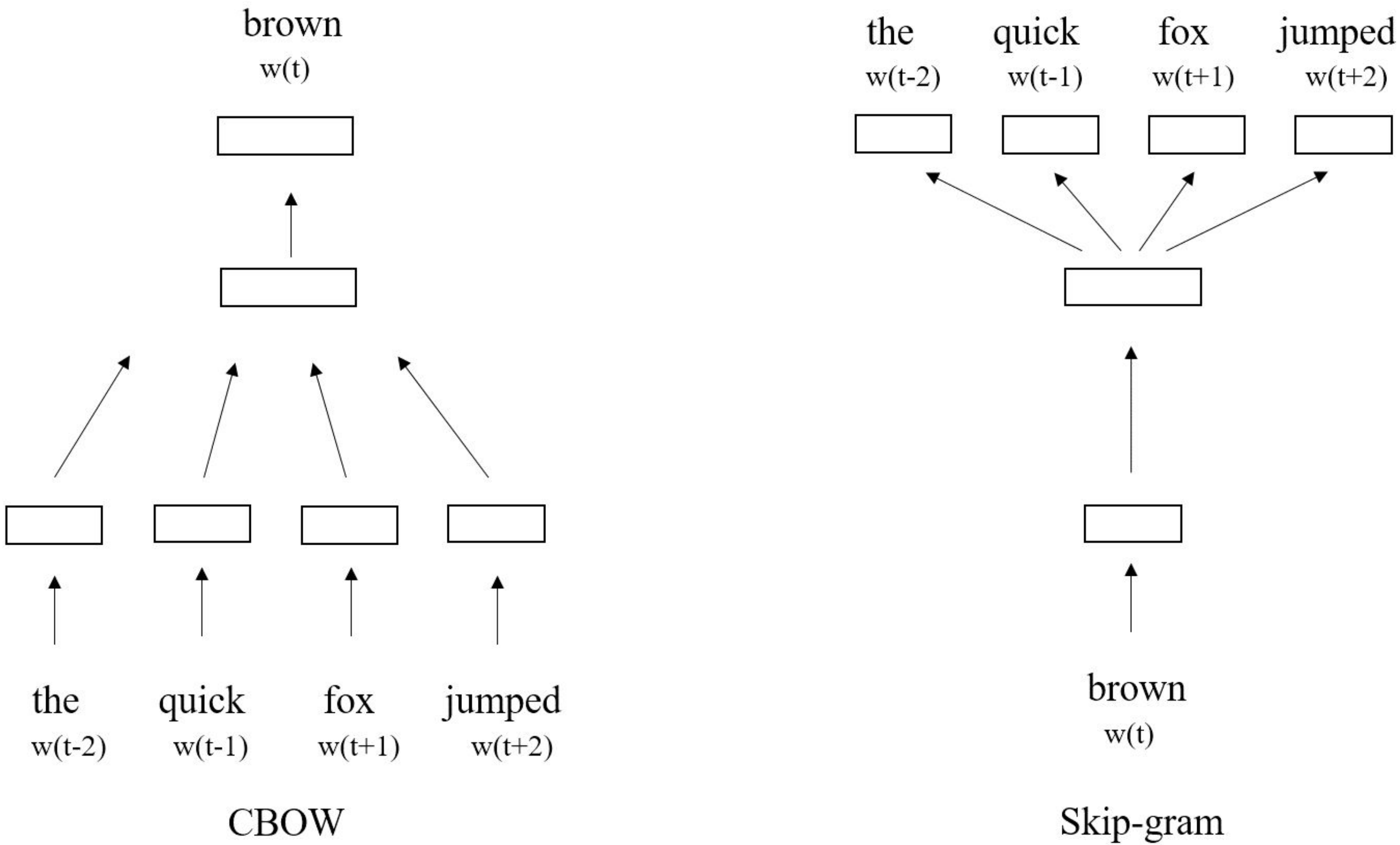
Hidden Layer
Weight Matrix

Word Vector
Lookup Table!






Output weights for “car”
Word vector for “ants”
300 features
×
300 features
softmax
$\frac{e^x}{\sum e^x}$
=
Probability that if you randomly pick a word nearby “ants”, that it is “car”


brown
w(t)
the
w(t-2)
quick
w(t-1)
fox
w(t+1)
jumped
w(t+2)
CBOW
the
w(t-2)
quick
w(t-1)
fox
w(t+1)
jumped
w(t+2)
brown
w(t)
Skip-gram

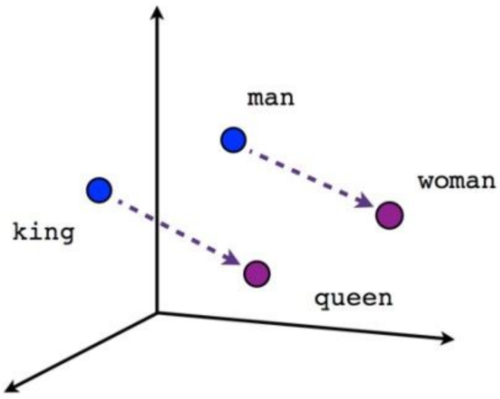
Continuous Bag of Words (CBOW)

scoring function

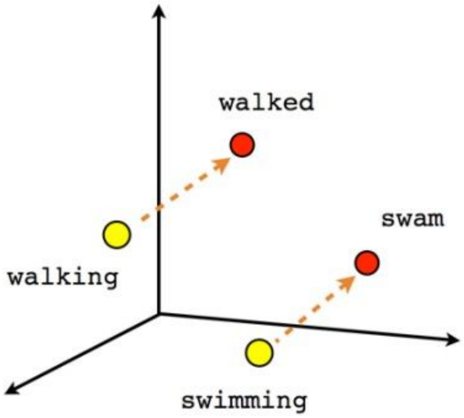
target word (vector)

$$P(w_o | w_c) = \frac{e^{s(w_o, w_c)}}{\sum_{w_i \in V} e^{s(w_i, w_c)}}$$

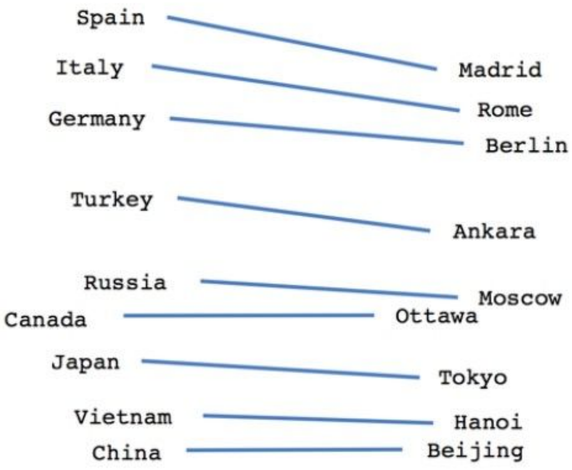
context vector



Male-Female



Verb tense



Country-Capital

Enriching Word Vectors with Subword Information

Each word w is represented as a bag of character n-gram.
We also include the word w itself in the set of its n-grams, to learn a representation for each word (in addition to character n-grams).

In practice, we extract all the n-grams for n greater or equal to 3 and smaller or equal to 6

We represent a word by the sum of the vector representations of its n-grams

https://arxiv.org/pdf/1607.04606.pdf

Why?

Popular models that learn such representations ignore the morphology of words, by assigning a distinct vector to each word. This is a limitation, especially for languages with large vocabularies and many rare words

https://www.upwork.com/hiring/for-clients/artificial-intelligence-and-natural-language-processing-in-big-data/

http://www.informit.com/articles/article.aspx?p=2265404

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

https://arxiv.org/pdf/1607.04606.pdf

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

https://blog.algorithmia.com/introduction-automatic-text-summarization/