NATURAL LANGUAGE ANALYSIS

LESSON 6: INTRODUCTION TO SEMANTIC ANALYSIS

OUTLINE

- What is Semantic?
- Semantic Analysis
  - Word level
  - Sentence level
  - Document level
- Text to Numbers
  - One Hot Vectors
  - Distributional Semantics
  - Neural Embeddings

WHAT IS SEMANTIC?

- Semantic is the meaning, interpretation of the words, signs and sentence structure.
- As you see in the figure, saying hello is different according to languages but meaning is the same.
- So semantic deals with the meaning of the things that is saved its behind.

WHAT IS SEMANTIC?

Semantic Analysis is a subfield of Natural Language Processing that attempts to understand the meaning of Natural Language. Understanding Natural Language might seem a straightforward process to us as humans. However, due to the vast complexity and subjectivity involved in human language, interpreting it is quite a complicated task for machines. Thus, in order to capture the meaning of the given text, the machines use simple quantitative tools at first such as letter or word orders, syntax, grammar and part of speech tags. But these tools are insufficient to extract meaning from the text.

WHAT IS SEMANTIC?

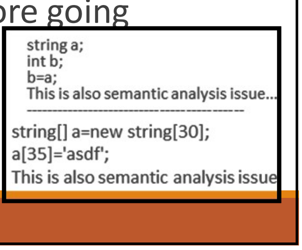
Although nowadays the contextualization approaches evaluate these together, there were two obvious approaches to texts in early semantic analysis studies: conceptual meaning and associative meaning.

- Semantic deals with conceptual meaning. This is also known as dictionary definition of the concept.
- Associative meaning is also known as Pragmatic and interest in the study of how context affects meaning.
- For conceptual meaning, **needle** means '**thin, sharp, steel instrument**'. But in associative meaning, needle ='painful'.

SEMANTIC ANALYSIS IN CS

There are **lexical** analysis, **syntax** analysis and **semantic** analysis phases in compiler design.

- In lexical analysis, the compiler checks the lexicons in the language and detects illegal inputs.
- In syntax analysis, using regular expressions of the language, it checks the syntax of each line in language, like variable definition, assignments, mathematical operations etc.
- Semantic analysis is the last step, catching all errors before going into machine level. For example, it checks its type while assign a value to a variable, and thus the error on the right window is found.



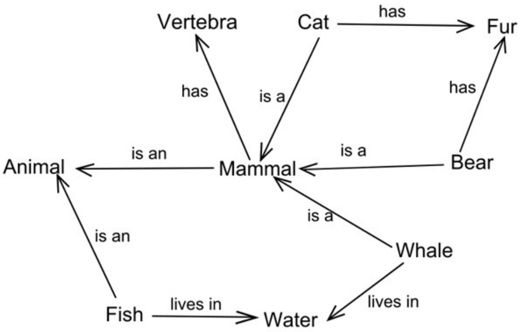
```
string a;
int b;
b=a;
This is also semantic analysis issue...
------------------------------------------------
string[] a=new string[30];
a[35]='asdf';
This is also semantic analysis issue
```

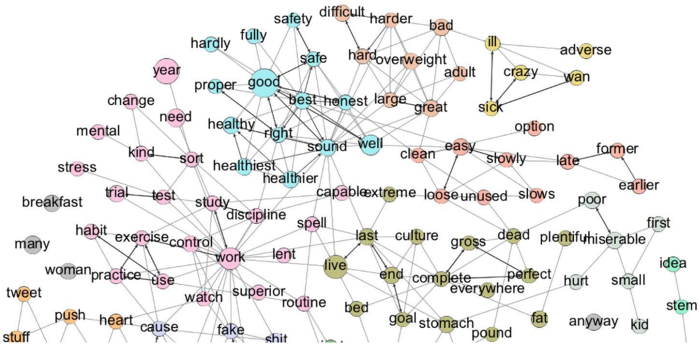
SEMANTIC ANALYSIS IN NLP

- Semantic analysis in the **word level** is generally done for the word sense disambiguation, semantic similarity or relatedness.
- Semantic analysis in the **sentence** or **short text level** is generally done to get similarity or relatedness of two given textual items, sentiment analysis, named entity recognition.
- Semantic analysis in the **document level** is usually done to get document similarity or relatedness, document classification, textual entailment, information retrieval, information extraction etc.

WORD LEVEL SEMANTIC

Semantic analysis at the word level is usually modeled on the relationships between the meanings of words.





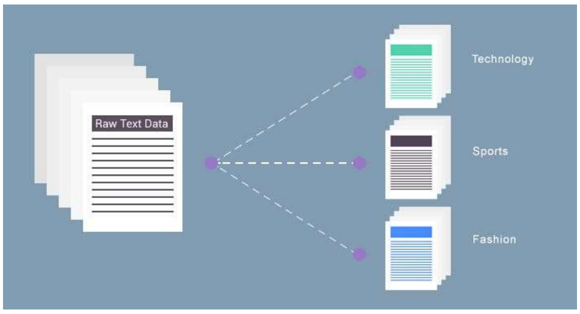
SENTENCE LEVEL SEMANTIC

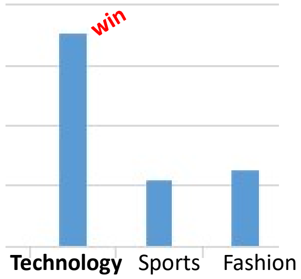
Sentence level semantics deals with the meaning of syntactic units larger than words, i.e. phrases, clauses, and sentences, and the semantic relationships between them. At this level, ambiguity is a little more difficult to resolve than word-by-word.

- Look at the dog using only one of your eyes.
- Look at the dog that only has one eye.

DOCUMENT LEVEL SEMANTIC

The simplest method for classifying documents is to count words matching with the given list of keywords for each topic. For example, let's identify three topics: technology, sports and fashion.



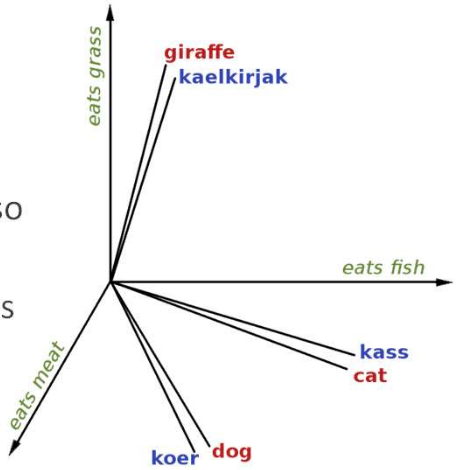


TEXT TO NUMBERS

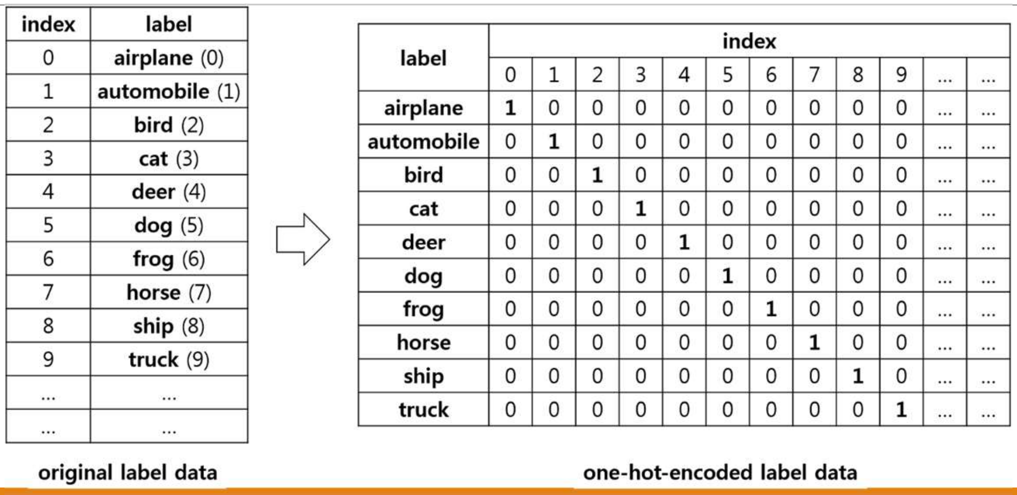
Machines are better at understanding numbers that accurately represent text rather than text.

Semantic analysis often applies to the process of converting text into numbers, also called **vectorization**.

There are different vectorization approaches in the literature. Let's move on from the most primitive to the most advanced.



ONE HOT VECTOR



index	label
0	**airplane** (0)
1	**automobile** (1)
2	**bird** (2)
3	**cat** (3)
4	**deer** (4)
5	**dog** (5)
6	**frog** (6)
7	**horse** (7)
8	**ship** (8)
9	**truck** (9)
...	...
...	...

original label data

label	**index**											
	0	1	2	3	4	5	6	7	8	9	...	...
airplane	**1**	0	0	0	0	0	0	0	0	0	...	...
automobile	0	**1**	0	0	0	0	0	0	0	0	...	...
bird	0	0	**1**	0	0	0	0	0	0	0	...	...
cat	0	0	0	**1**	0	0	0	0	0	0	...	...
deer	0	0	0	0	**1**	0	0	0	0	0	...	...
dog	0	0	0	0	0	**1**	0	0	0	0	...	...
frog	0	0	0	0	0	0	**1**	0	0	0	...	...
horse	0	0	0	0	0	0	0	**1**	0	0	...	...
ship	0	0	0	0	0	0	0	0	**1**	0	...	...
truck	0	0	0	0	0	0	0	0	0	**1**	...	...

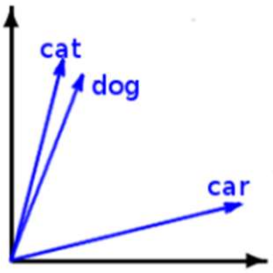
one-hot-encoded label data

DISTRIBUTIONAL SEMANTICS

Distributional semantics is an approach to both solving the sparsity problem and better modeling words with semantic relationships.

The terms semantic space models or vector space models are sometimes used instead of distributional semantics.

For example, when we look at the vectors on the right, it is more plausible that the terms "cat" and "dog" are closer to each other than the term "car".



DISTRIBUTIONAL SEMANTICS

“You shall know a word by the company it keeps!” (Firth, 1957)

Idea: Similar linguistic objects have similar contents or contexts.

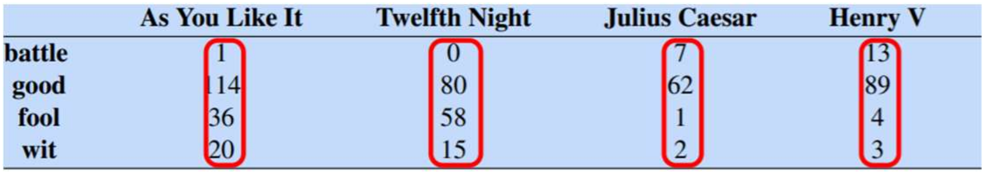
The main goal is to ensure that texts are represented by numbers using word distributions in a corpus.

We can generalize all these studies under two main headings:

- Term-Document Matrix
- Cooccurrence Matrix

DISTRIBUTIONAL SEMANTICS: Term-Document Matrix

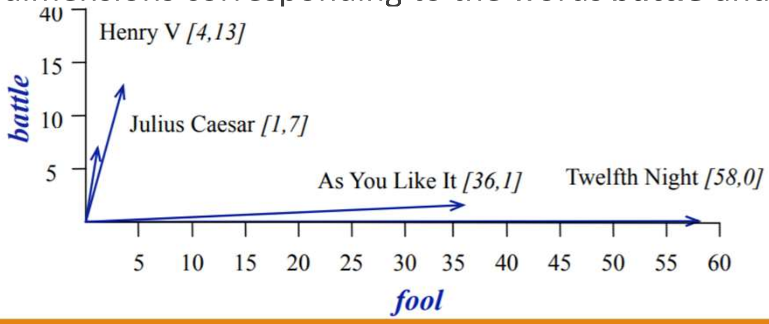
- **The term-document matrix** for four words in four Shakespeare plays. The red boxes show that each document is represented as a column vector of length four.
- We can think of the vector for a document as identifying a point in |Vector|-dimensional space; thus the documents in table above are points in 4-dimensional space.



	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

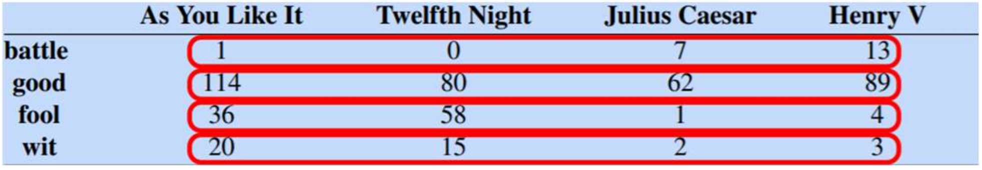
DISTRIBUTIONAL SEMANTICS: Term-Document Matrix

- Since 4-dimensional spaces are hard to display here,
- Shows a visualization in two dimensions; we've arbitrarily chosen the dimensions corresponding to the words ***battle*** and ***fool***.



DISTRIBUTIONAL SEMANTICS: Term-Document Matrix

Documents can also be represented as vectors in a vector space. Vector semantics can also be used to represent the meaning of words, by associating each word with a vector. The word vector is now a row vector rather than a column vector and hence the dimensions of the vector are different. The four dimensions of the vector for **fool**, (36,58,1,5) correspond to the four Shakespeare plays.



	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

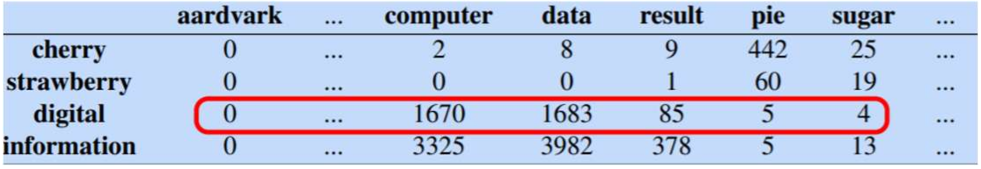
DISTRIBUTIONAL SEMANTICS: Cooccurrence Matrix

In term-document method, similar documents can have similar vectors, because similar documents tend to have similar words. This same principle applies to words: similar words can have similar vectors because they tend to occur in similar documents.

For this reason, to use cooccurrence matrix was needed by using more document. In order to show how it works, we will use a corpus with only one document. In this case, the cooccurrence term represents the number of times the two words appear in the that document. In smaller contexts, generally a window around the word is used such as 4 words to the left and 4 words to the right.

DISTRIBUTIONAL SEMANTICS: Cooccurrence Matrix

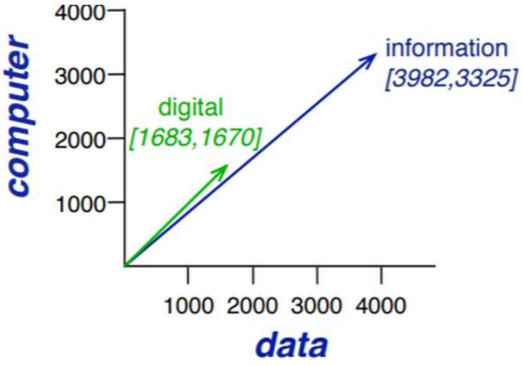
Co-occurrence vectors for four words, computed from the Brown corpus, showing only six of the dimensions. The vector for the word **digital** is outlined in red. Note that a real vector would have vastly more dimensions and thus be sparser.



	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

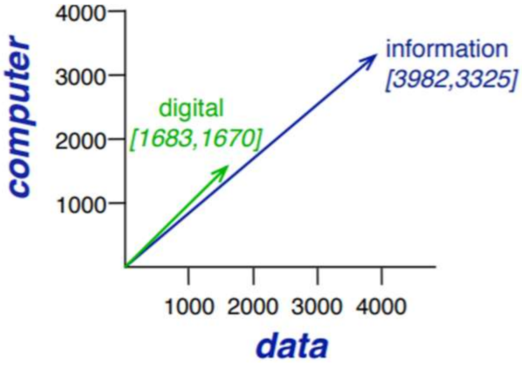
DISTRIBUTIONAL SEMANTICS: Cooccurrence Matrix

A spatial visualization of word vectors for **digital** and **information**, showing just two of the dimensions, corresponding to the words **data** and **computer**.



DISTRIBUTIONAL SEMANTICS: Cooccurrence Matrix

When we look at the words "Digital" and "Information" given in this chart, although the Euclidean distance between them seems quite high, the angular Cosine similarity seems quite good. What is your comment about the similarity of the words?



NEURAL EMBEDDINGS

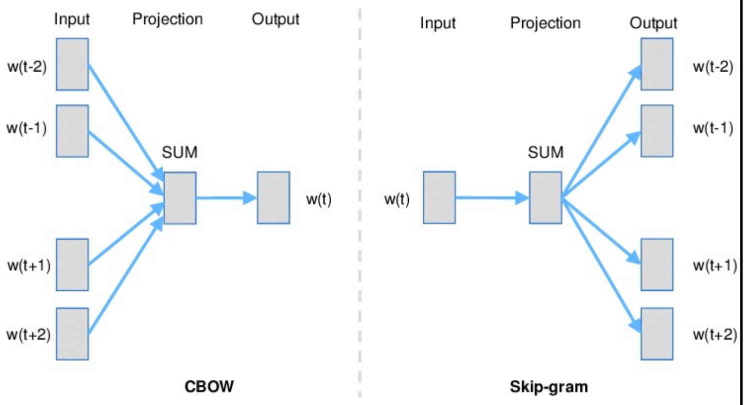
Recently, the computational power of next-generation neural systems and the hypothesis of distributional semantics have combined, resulting in much more capable text vectorization methods.

These studies, which have both high semantic representation and no sparsity problems, actually started with the Word2Vec method. Although many methods have been proposed, none of them have reached the popularity of the **Word2vec**, GloVe and FastText methods.

NEURAL EMBEDDINGS: Word2vec

The Word2vec model was introduced in 2013 with two approaches.

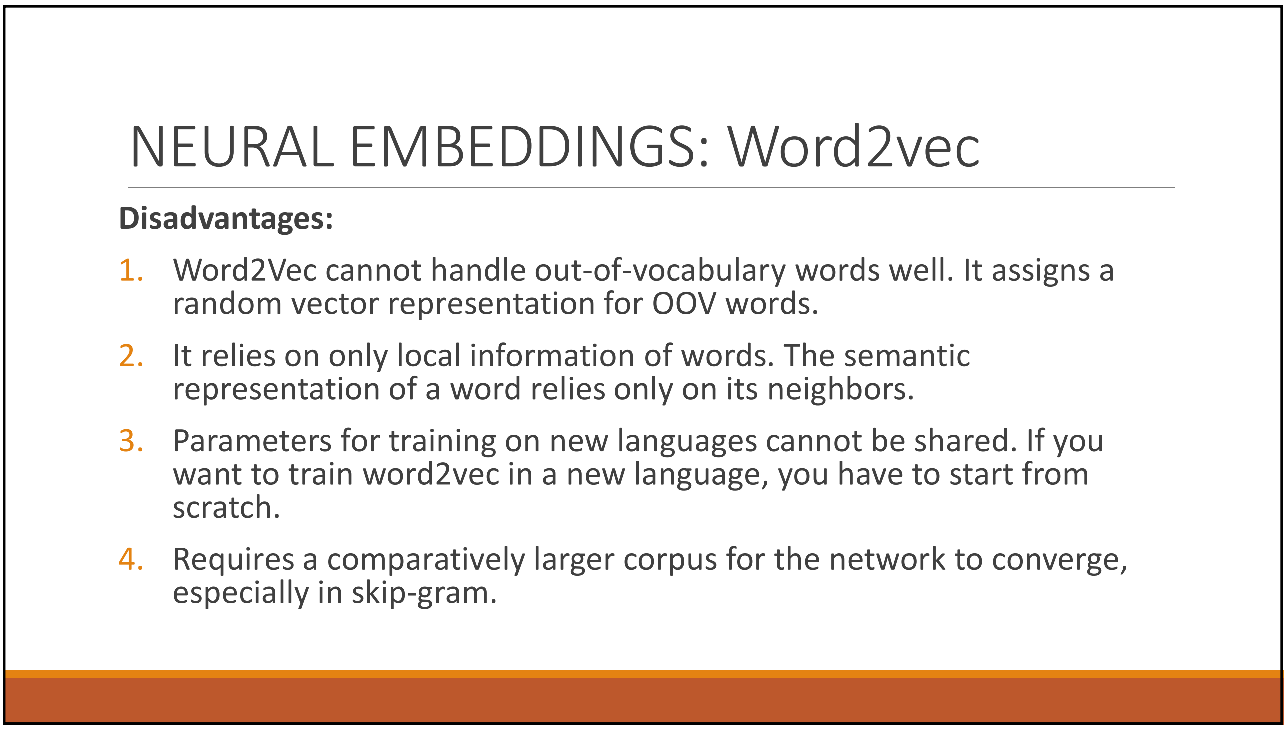
1. CBOW: The surrounding words are used as input to predict the middle word.
2. Skip-gram: Each word is used as input to predict surrounding words.



NEURAL EMBEDDINGS: Word2vec

Advantages:

1. Word2vec can capture relationships between different words including their syntactic & semantic relationships
2. The size of the embedding vector is small & flexible, unlike all the previous algorithms discussed where the size of embedding is proportional to vocabulary size
3. Since its unsupervised, human effort in tagging the data is less



NEURAL EMBEDDINGS: Word2vec

Disadvantages:

1. Word2Vec cannot handle out-of-vocabulary words well. It assigns a random vector representation for OOV words.
2. It relies on only local information of words. The semantic representation of a word relies only on its neighbors.
3. Parameters for training on new languages cannot be shared. If you want to train word2vec in a new language, you have to start from scratch.
4. Requires a comparatively larger corpus for the network to converge, especially in skip-gram.