Graduate Certificate in Intelligence Reasoning Systems

Cognitive Systems

NATURAL LANGUAGE COMPREHENSION AND PROCESSING

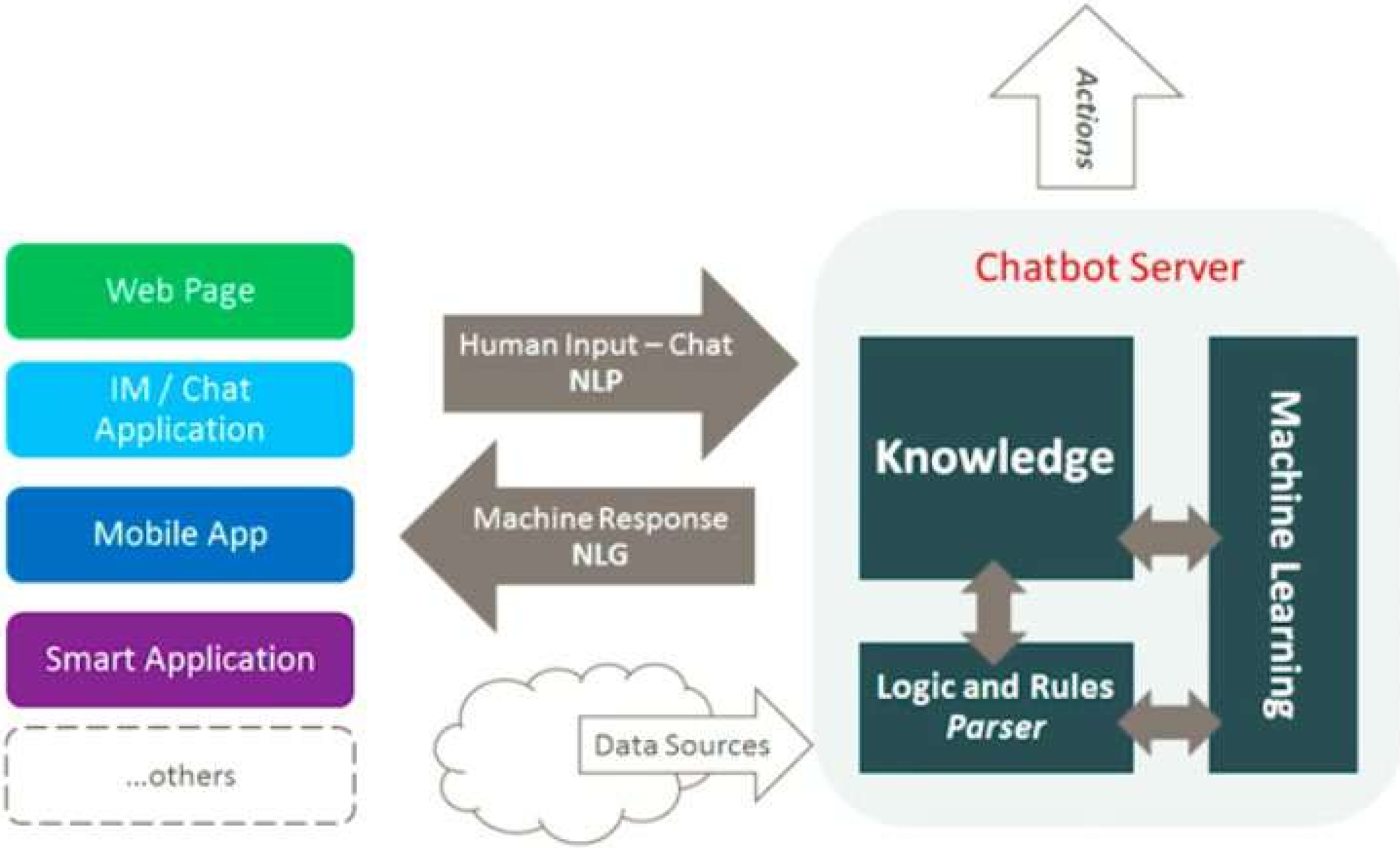
Dr. Fan Zhenzhen
zhenzhen@nus.edu.sg

Objective

- To review and identify appropriate NLP techniques and solutions for Cognitive Systems to process natural language inputs

- Topics:
  - Recap…
  - The need of natural language comprehension
  - Natural language processing techniques

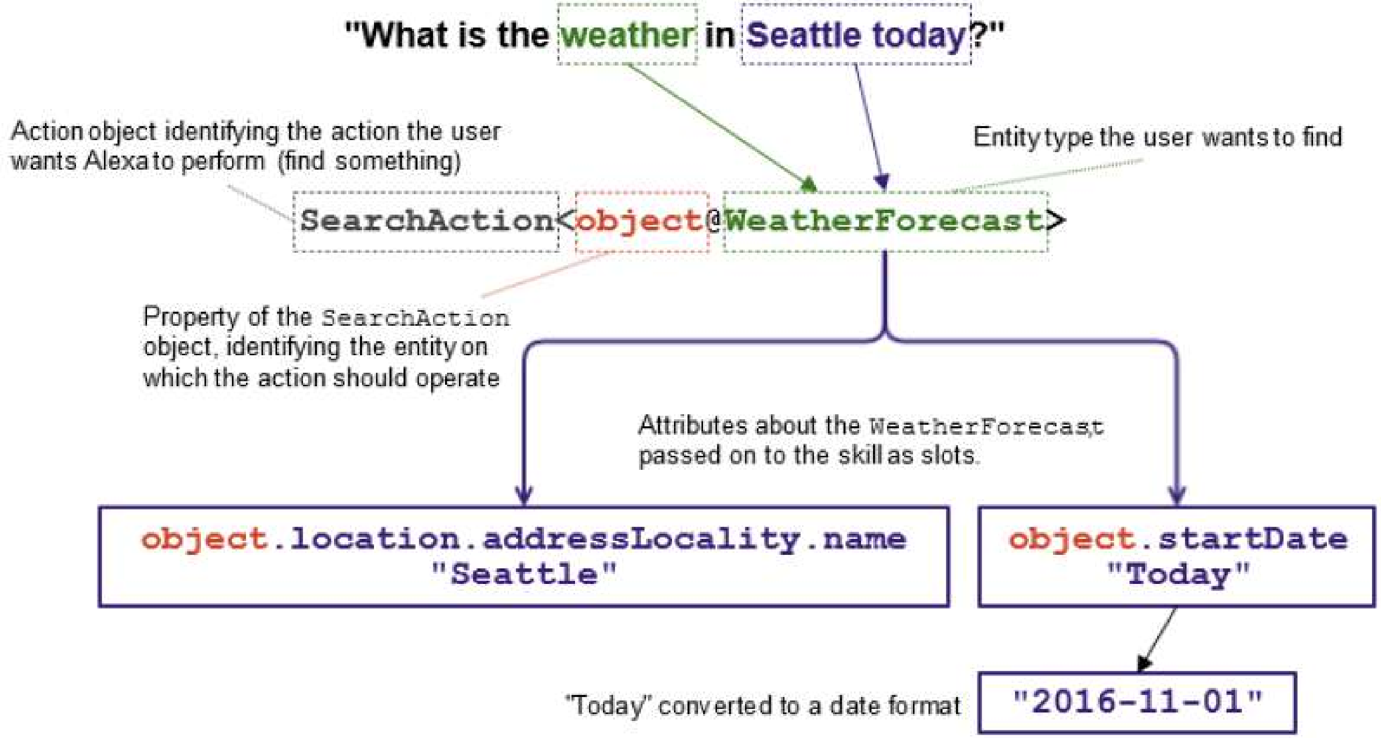
Recap…



https://medium.com/analytics-vidhya/building-a-simple-chatbot-in-python-using-nltk-7c8c8215ac6e

Recap…

- Natural language understanding is required for the system to understand the user's request
- To map user's utterances to **intents**, which may contain **slots**



Recap…

What is the weather in Seattle today?

What's the weather in London?

Tell me the temperature now.

Intent Detection

Slots Detection

SearchAction(WeatherForcast((Location 'Seattle'),(StartDate 'Today')))

NLP Tasks Required

- Named Entity Recognition
- Entity Labelling
- Entity Linking
- Co-reference Resolution

Information Extraction

- Intent Detection (Classification)
- Slot Detection (NER)

What is information extraction?

- The automatic extraction of (possibly pre-specified) information from natural language documents
  - Facts about types of entities, events, relationships

- The automatic population of a structured information source (template, or logical form) from natural language documents
  - Documents may be semi-structured (eg., patents), unstructured (e.g., websites) or free text (e.g., documents)

Concept vs. Named Entity vs. Information

- **Name Entity** = lowest level of recognition by an IE system
  – Normally recognized by dictionaries or rules
- **Concept** = rule or heuristic to create an abstraction
  – Sometimes called a “natural class” = different people at different times and in different places would refer to the same referent with that concept
  – “president of the United States” vs. “president of the United Kingdom”
- **Information** = words, named entities, concepts which fulfill a need
  – So if you have a question, and a phrase answers that question, then that phrase is an example of information
  – Information is often regular, i.e., with a pattern
    - Eg, information about a person = name, age, sex, address, hp#, …
    - Information about a company = name, address, stock symbol, Chairman, …

What is information extraction?

- The automatic extraction of (possibly pre-specified) information from natural language documents
  - Facts about types of entities, events, relationships

- The automatic population of a structured information source (template) from natural language documents (i.e., create a table!)
  - Documents may be semi-structured (eg., patents), unstructured (e.g., websites) or free text (e.g., documents)

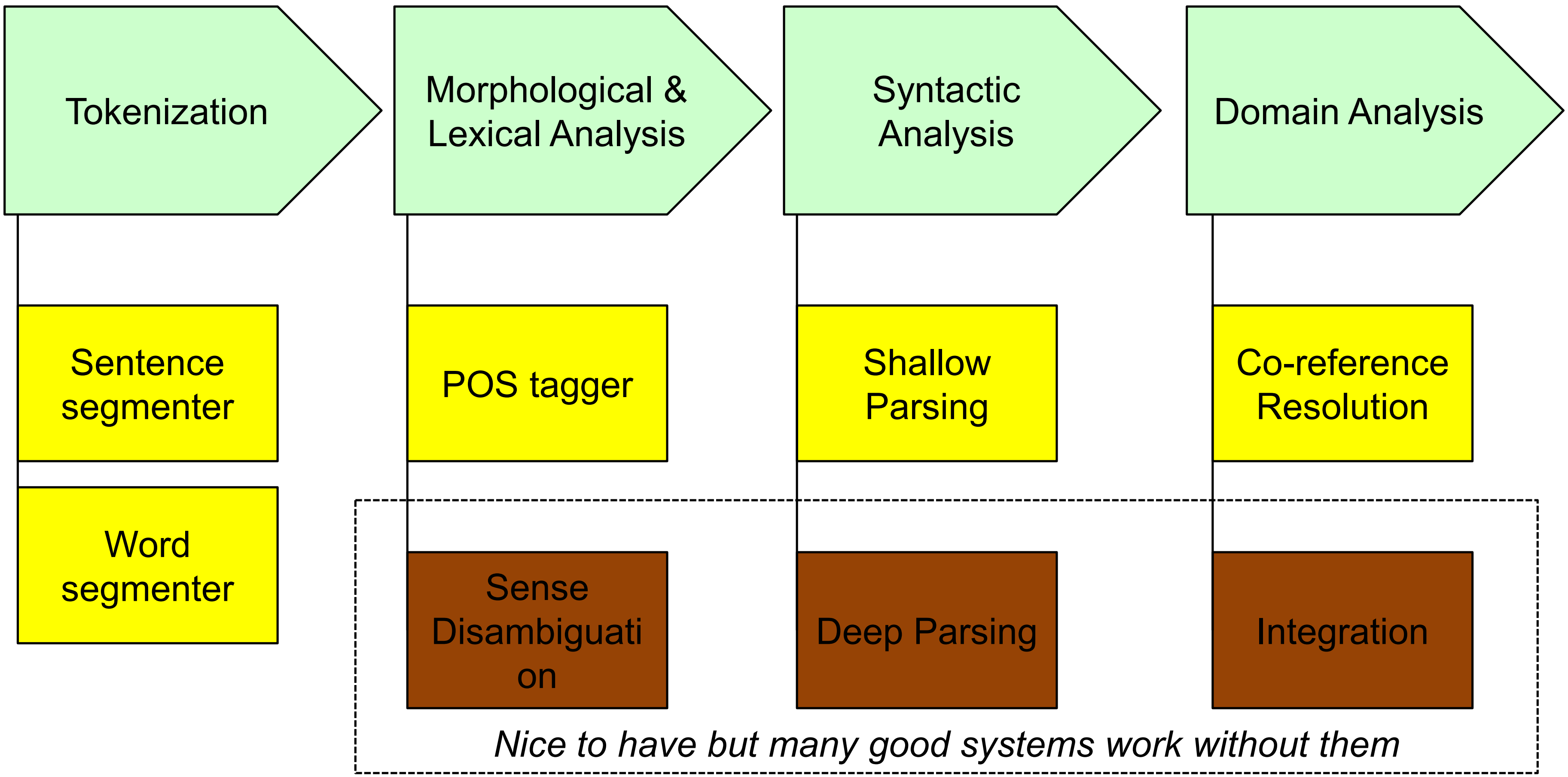
Types of IE systems

- Rule-based Systems
  - Hand-coded rules
    - Coded by linguists, with domain input
    - Iterative method based on document inspection
    - Slow but very good results
  - Induced (machine learning) rules
    - Fully machine learning
      - Given an annotated corpus, derive a basis set of rules that cover a pre-determined % of the annotated examples (and only the annotated examples)
      - Heuristic approach: one rule at a time!
    - Hybrid systems – machine learning to fine-tune the rules

Types of IE systems

- Statistics-based Systems
  - Start with a well-annotated corpus
  - Depending on the method (e.g., Hidden Markov Models), derive statistical rules to create a model that generates the examples
  - Advantages compared to Rule based systems
    - Language independent (within representational limits)
    - No linguistic or domain knowledge needed in the team
    - Relatively small effort in creating the models
  - Issues
    - The complexity moves to the corpus – must be well annotated and must cover the full space of possibilities
    - Requires <u>very large number of training examples</u> to get good results

Main components of an IE system



Tokenization

- To break a stream of characters into tokens

Great location with a little bit of history.

Great	location	with	a	little	bit	of	history

- This is done by identifying token delimiters
  - Whitespace characters such as ***space, tab, newline***
  - Punctuation characters like ***( ) < > ! ? “ ”***
  - Other characters ***. , : - ‘ ’*** etc.

Tokenization Challenges

- It seems simple, but…
  - . , : between numbers are part of the number

12.34	12,345	12:34

  - . can be part of an abbreviation or end of a sentence

U.S.A.	Dr.

  - ’ can be a closing internal quote, indicate a possessive, or be part of another token

My friend’s	isn’t

POS Tagging

- To determine POS or grammatical category of a term
  - Nouns, verbs, adjectives, adverbs, pronouns, determiners, prepositions, conjunctions, etc.
  - LDC Penn Tree Bank has 36 categories with detailed information, e.g.

CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential *there*
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative

UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun

POS Tagging

- Dictionary with word-POS correspondence is needed
- Challenge – POS disambiguation (words with >1 POS)
  - E.g. "*book*" can be a noun ("*my book*") or a verb ("*to book a room*")
- Example:
  - About six and a half hours later, Mr. Armstrong opened the landing craft's hatch, stepped slowly down the ladder and declared as he planted the first human footprint on the lunar crust: "That's one small step for man, one giant leap for mankind."

IN/ About CD/ six CC/ and DT/ a JJ/ half NNS/ hours RB/ later ,/ , NNP/ Mr. NNP/ Armstrong VBD/ opened DT/ the NN/ landing NN/ craft POS/ 's NN/ hatch ,/ , VBD/ stepped RB/ slowly IN/ down DT/ the NN/ ladder CC/ and VBD/ declared IN/ as PRP/ he VBD/ planted DT/ the JJ/ first NN/ human NN/ footprint IN/ on DT/ the NN/ lunar NN/ crust :/ : ``/ " DT/ That VBZ/ 's CD/ one JJ/ small NN/ step IN/ for NN/ man ,/ , CD/ one JJ/ giant NN/ leap IN/ for NN/ mankind ./ . "/ "

Generated by UIUC POS Tagger

POS Taggers

- Rule-based - e.g. Brill's tagger by Eric Brill
  - Error-driven transformation-based tagger
  - Initially assign the most frequent tag to each word, based on dictionary and morphological rules
  - Contextual rules are then applied repeatedly to correct any errors
- Stochastic taggers – e.g. CLAWS, Viterbi, Baum-Welch, etc.
  - based on Hidden Markov Models (HMMs) and n-gram probabilities
  - Manually tagged corpus is needed to estimate probabilities
- Many machine learning methods have also been applied
- Stanford's Statistical NLP website lists many free taggers

Shallow Parsing / Chunking

- To identify phrases in a text (noun phrases, verb phrases, and prepositional phrases, etc.)

- Example:
  - About six and a half hours later, Mr. Armstrong opened the landing craft's hatch, stepped slowly down the ladder and declared as he planted the first human footprint on the lunar crust: "That's one small step for man, one giant leap for mankind."

[NP About six and a half hours] [ADVP later] , [NP Mr. Armstrong] [VP opened] [NP the landing craft] [NP 's hatch] , [VP stepped] [ADVP slowly] [PP down] [NP the ladder] and [VP declared] [SBAR as] [NP he] [VP planted] [NP the first human footprint] [PP on] [NP the lunar crust] : "[NP That] [VP 's] [NP one small step] [PP for] [NP man] , [NP one giant leap] [PP for] [NP mankind] ."

Generated by UIUC chunker

Shallow Parsing / Chunking

- After morphological analysis and disambiguation, using information of lemmata, morphological information, and word order configuration
- Largely stochastic techniques based on probabilities derived from an annotated corpus
- Avoiding the complexity of full parsing, faster, more robust
- Useful in Information Extraction, Summary Generation, and Question Answering

Name Entity Recognition

- Recognition of particular types of proper noun phrases, specifically persons, organizations, locations, and sometimes money, dates, times, and percentages.
- Very useful in text mining applications, by turning verbose text data into a more compact structural form

[**LOC Houston**] , Monday, July 21 -- Men have landed and walked on the moon. Two [**MISC Americans**] , astronauts of [**ORG Apollo**] 11, steered their fragile four-legged lunar module safely and smoothly to the historic landing yesterday at 4:17:40 P.M., Eastern daylight time. [**PER Neil A. Armstrong**] , the 38-year-old civilian commander, radioed to earth and the mission control room here: "[**LOC Houston**] , [**ORG Tranquility Base**] here; the Eagle has landed."

Generated by UIUC NER system

Rule-based NER Systems

- Rule-based systems can and do work well
  - Corpus is relatively static (in terms of vocabulary, language structure, etc.)
  - Can be fast especially in well-defined limited domains (compared to annotating training examples)
- A typical rule-based system comprises
  - Set of rules
  - Policies to control when and how (multiple) rules are applied, e.g., order, looping.

What does a rule look like?

- Lexical pattern matching
- Form:
  - Match(pattern) then Do(action)

from gate.ac.uk

```
Rule: Company1
  ( ( {Token.orthography == upperInitial} )+
      {Lookup.kind == companyDesignator}
  ):match
 -->
   :match.NamedEntity = { kind=company, rule="Company1" }
```

When to use statistics based systems?

- Many top performing systems are statistics based
  - Machine learning (ML) on very large corpora is state-of-the-art
- Annotation based corpora for training
  - You have a well annotated corpora with many features
  - Various ML techniques from simple to sophisticated
  - Relatively homogeneous real data (not training data) in any given domain. Note that models don't transfer well across domains
  - You don't have domain or language resources in that area

Simple model is at token level

- Text is a linear sequence of tokens (such as words)
- Token boundaries can be fairly easily derived in some languages, e.g., space & punctuation for English, but much harder for others, e.g., Chinese
- Simple tokenization
  - Dictionary based
  - Colocation frequencies (see next page)
- Alternatives
  - Ignore multi-unit tokens
  - Bi-grams, tri-grams, multi-grams

Popular models

- Hidden Markov Models (HMM)
  - Simple, joint probability
- Conditional Random Fields (CRF)
  - Conditional probability
  - Considers features of current token, and of preceding *n* tokens (window=*n*)
- Similarity algorithms
  - Measure distance of group of words to a dictionary list
  - Works especially well for jargon and other terminology
- Support Vector Machines (SVM)
  - Training method for standard perceptron
  - Optimize the points to determine the hyperplane dividing the positive training samples from the negative ones

What is coreference?

- Coreference resolution
  - Determine relationship between entities which are related
    - Identity relation (morning star vs. evening star)
    - Whole-part relation
  - Simple version
    - Determine entities which have the same referent
      - Anaphora (Pronouns)
      - Proper names, proper nouns, noun phrases,…
    - Definite descriptions (may be time dependent)
      - Usain Bolt & "the fastest man in the world"

Co-reference Examples

- Anaphora
  - The elephant stepped on the rabbit and it died.
  - The elephant stepped on the landmine and it died.
- Proper nouns
  - John Smith and Mary Brown were married this morning. The groom was dressed in a white tuxedo while the bride was…
- Definite descriptions
  - Usain Bolt has won the Olympic 100m gold medal. The fastest man in the world successfully defended his title last night.

Intent Classification

- Using labelled data to build machine learning models that can classify input into intent classes (supervised learning)
- SVM/NB/LR/DT/KNN
- Pre-process the input text into features (vector model)

	amazing	service	lost	glamour	disappoint	brilliant	super	expensive	noisy	…
Doc1	1	1	0	0	0	1	0	0	0	
Doc2	0	0	1	1	1	0	0	1	0	
Doc3	0	0	0	1	0	0	1	0	0	
Doc4	0	0	0	0	2	0	0	1	1	
…										

Typical Pre-processing

- Tokenization
- Case normalisation (to lowercase)
- Lemmatization/stemming
  - To reduce the words to its root form
  - E.g. classes -> class, ran -> run , production -> produce
- Punctuation removal
- Stopword removal
  - To remove extremely common words (with little meaning) like functional words (the, a, of…)

Indexing

- Many text mining applications are based on vector representation of documents (term-document matrix) using “bag-of-words” approach

$$
\begin{array}{cc}
 & \begin{array}{cccc} T_1 & T_2 & \cdots & T_t \end{array} \\
\begin{array}{c} D_1 \\ D_2 \\ \vdots \\ \vdots \\ D_n \end{array} &
\begin{pmatrix}
w_{11} & w_{21} & \cdots & w_{t1} \\
w_{12} & w_{22} & \cdots & w_{t2} \\
\vdots & \vdots & & \vdots \\
\vdots & \vdots & & \vdots \\
w_{1n} & w_{2n} & \cdots & w_{tn}
\end{pmatrix}
\end{array}
$$

T: term
D: document
w: weight of the term

- Usually only content words (adjectives, adverbs, nouns, and verbs) are used as vector features.

Term Weighting

- Binary
  - 0 or 1, simply indicating whether a word has occurred in the document (but that's not very helpful).
- Frequency-based
  - *term frequency*, the frequency of words in the document, which provides additional information that can be used to contrast with other documents.

	amazing	service	lost	glamour	disappoint	brilliant	super	expensive	noisy	...
Doc1	1	1	0	0	0	1	0	0	0	
Doc2	0	0	1	1	1	0	0	1	0	
Doc3	0	0	0	1	0	0	1	0	0	
Doc4	0	0	0	0	2	0	0	1	1	
...										

tf-idf Indexing

- To modify the frequency of a word in a document by the perceived importance of the word(the *inverse document frequency)*, widely used in information retrieval
  - When a word appears in many documents, it's considered unimportant.
  - When the word is relatively unique and appears in few documents, it's important.

$$\boldsymbol{tf\text{-}idf_{t,d} = tf_{t,d} * idf_t}$$

$$idf_t = \log \frac{N}{df_t}$$

- $tf_{t,d}$: term frequency – number of occurrences of term t in document d
- idf_t : inverted document frequency of term t

N : the total number of documents in the corpus

df_t : the document frequency of term t, i.e., the number of documents that contain the term.

tf-idf Indexing – An Example

TERM VECTOR MODEL BASED ON $w_i = tf_i * IDF_i$												
Query, Q: "gold silver truck" D_1: "Shipment of gold damaged in a fire" D_2: "Delivery of silver arrived in a silver truck" D_3: "Shipment of gold arrived in a truck" D = 3; IDF = log(D/df_i)												
		Counts, tf_i						Weights, $w_i = tf_i * IDF_i$				
Terms	Q	D_1	D_2	D_3	df_i	D/df_i	IDF_i	Q	D_1	D_2	D_3	
a	0	1	1	1	3	3/3 = 1	0	0	0	0	0	
arrived	0	0	1	1	2	3/2 = 1.5	0.1761	0	0	0.1761	0.1761	
damaged	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0	
delivery	0	0	1	0	1	3/1 = 3	0.4771	0	0	0.4771	0	
fire	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0	
gold	1	1	0	1	2	3/2 = 1.5	0.1761	0.1761	0.1761	0	0.1761	
in	0	1	1	1	3	3/3 = 1	0	0	0	0	0	
of	0	1	1	1	3	3/3 = 1	0	0	0	0	0	
silver	1	0	2	0	1	3/1 = 3	0.4771	0.4771	0	0.9542	0	
shipment	0	1	0	1	2	3/2 = 1.5	0.1761	0	0.1761	0	0.1761	
truck	1	0	1	1	2	3/2 = 1.5	0.1761	0.1761	0	0.1761	0.1761	

Note that in this example, stopwords and very common words are not removed, and terms are not reduced to root terms.

http://www.miislita.com/term-vector/term-vector-3.html

Cosine Similarity

- A similarity measure between two vectors (input and candidate response)
- by measuring the cosine of the angle between t

$$Sim(D_i, D_j) = \frac{D_i \bullet D_j}{|D_i| * |D_j|} = \frac{\sum_k w_{ki} w_{kj}}{\sqrt{\sum_k w_{ki}^2 \sum_k w_{kj}^2}}$$

- Example: Given 3 document vectors shown here

D_1	D_2	D_3
0	0	0
0	0	0.1761
0	0.4771	0
0	0	0.4771
0	0.4771	0
0.1761	0.1761	0
0	0	0
0	0	0
0.4771	0	0.9542
0	0.1761	0
0.1761	0	0.1761

$$|D_1| = \sqrt{0.1761^2 + 0.4771^2 + 0.1761^2} = \sqrt{0.2896} = 0.5382$$

$$|D_2| = \sqrt{0.4771^2 + 0.4771^2 + 0.1761^2 + 0.1761^2} = \sqrt{0.5173} = 0.7192$$

$$|D_3| = \sqrt{0.1761^2 + 0.4771^2 + 0.9542^2 + 0.1761^2} = \sqrt{1.2001} = 1.0955$$

$Sim(D_1, D_2)$ = (0.1761*0.1761)/(0.5382*0.7192)=0.0801

$Sim(D_1, D_3)$ = (0.4771*0.9542+0.1761*0.1761)/(0.5382*1.0955)=0.8246