Chapter one from "Speech and Language Processing" book

Definitions of Natural Language Processing:

Natural language processing

The goal of this field is to get computers to perform useful tasks involving human language, tasks like enabling

– human-machine communication, – improving human-human communication, or – simply doing useful processing of text or speech.

The sub-domain of artificial intelligence concerned with the task of developing programs possessing some capability of ‘understanding’ a natural language in order to achieve some specific goal.

Natural language processing

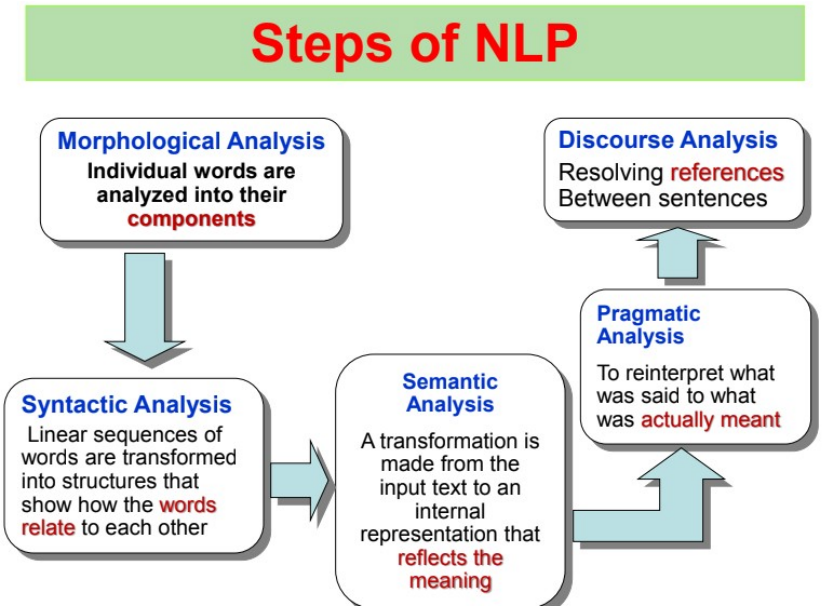
The goal of natural language processing is generally to build a representation of the text that adds structure to the unstructured natural language, by taking advantage of insights from linguistics. This structure can be syntactic in nature – capturing the grammatical relationships among constituents of the text – or more semantic – capturing the meaning conveyed by the text.

Computers use (analyze, understand, generate)

Steps of NLP

• Morphology: Concerns the way words are built up from smaller meaning bearing units

With lexical analysis, we divide a whole chunk of text into paragraphs, sentences, and words. It involves identifying and analyzing words' structure.



• Syntax: concerns how words are put together to form correct sentences and what structural role each word has

Syntactic analysis involves the analysis of words in a sentence for grammar and arranging words in a manner that shows the relationship among the words.

• Semantics: concerns what words mean and how these meanings combine in sentences to form sentence meanings

Semantic analysis draws the exact meaning for the words, and it analyzes the text meaningfulness.

• Pragmatics: concerns how sentences are used in different situations and how use affects the interpretation of the sentence – knowledge of the relationship of meaning to the goals and intentions of the speaker

Pragmatic analysis deals with overall communication and interpretation of language. It deals with deriving meaningful use of language in various situations.

• Discourse: concerns how the immediately preceding sentences affect the interpretation of the next sentence

Disclosure integration takes into account the context of the text. It considers the meaning of the sentence before it ends.

Parsing (Syntactic Analysis)

- Assigning a syntactic and logical form to an input sentence
  - – uses knowledge about word and word meanings (lexicon)
  - – uses a set of rules defining legal structures (grammar)

Ahmad ate the apple.

```
(S  (NP (NAME Ahmad))
    (VP  (V ate)
         (NP  (ART the)
              (N apple))))
```

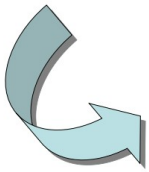
Word Sense Resolution

- Many words have many meanings or senses
- We need to resolve which of the senses of an ambiguous word is invoked in a particular use of the word
- I made her duck.
  - – made her a bird for lunch or
  - – made her move her head quickly downwards?

Ex. Morphological analysis

Surface form

I want to print Ali's .init file



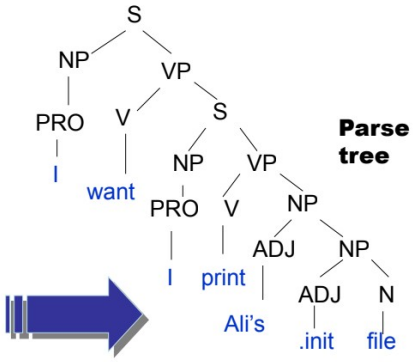
stems

I (pronoun)
want (verb)
to (prep)
to(infinitive)
print (verb)
Ali (noun)
's (possessive)
.init (adj)
file (noun)
file (verb)

Ex. Syntactic analysis

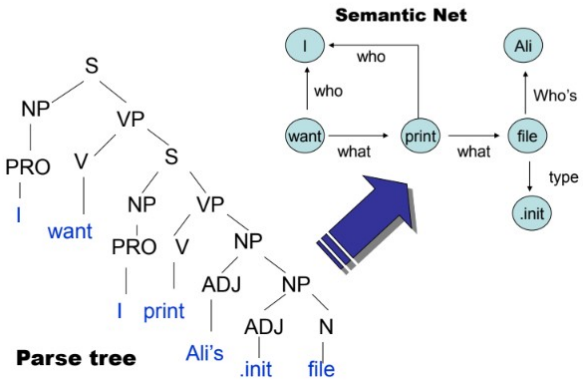
stems

I (pronoun)
want (verb)
to (prep)
to(infinitive)
print (verb)
Ali (noun)
's (possessive)
.init (adj)
file (noun)
file (verb)

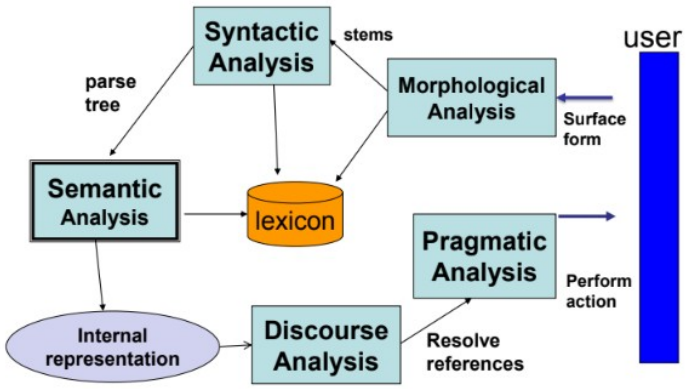


Parse tree

Semantic analysis



The steps of NLP



Syntactic and Morphological Analysis

Syntactic information about the text can be important to assist in resolving ambiguities and in establishing the appropriate relations among the words in a text. determining whether a word is a noun or a verb (or some other part of speech) can be useful. This is accomplished through tools that perform part of speech (POS) tagging. Then, identification of phrases in the text can be important, such as recognizing that a sequence of words forms a single conceptual unit (e.g. breast cancer, NF kappa beta inhibitor). A commonly used strategy for this is shallow parsing, which involves identifying coarse phrasal structures, such as noun phrases, without identifying the specific grammatical relationships among them. Deep parsing determines the full set of grammatical relations among words in a sentence, producing a complete parse tree to represent these relations.

The surface forms of words will vary depending on their syntactic usage in a sentence, for instance a noun appearing in plural form or a verb appearing in various tenses (regulated, regulating, regulates). Often, it is desirable to normalize such variation to a base form of the word in order to appropriately associate different occurrences of the same term. This is called morphological normalization and is often accomplished in practical NLP applications through stemming tools which strip off inflected word endings. The Porter algorithm, based on suffix stripping, is a popularly used strategy for stemming.

Information Extraction

Information extraction in general refers to the extraction of specific types of information from text, and normally formalized in a structured representation, such as an event template or a concept from an externally-defined ontology. It can refer to the association of particular strings of a text to a category of interest, for instance identifying protein names in a publication.

Named Entity Recognition

In the upper levels of Figure 1 we see annotations of ontology terms and gene/protein terms. Many of such terms correspond to named entities, that is, to objects that are generally referred to by name. This is in contrast to terms that correspond to processes or events, which normally require identification of higher-order relations. Examples of named entities in the biological domain that are often targeted for extraction are genes, diseases, chemicals, or experimental methods. Various methods exist for performing named entity recognition. The most basic approach is to compile a dictionary of the relevant names for a specific category of entities, and to perform a string match into the dictionary. Empirical methods based on supervised machine learning will often use a dictionary match as one feature of a model that also considers surrounding words, syntax, and other textual evidence to identify likely instances of terms from a particular category.

NLP vs NLU vs. NLG summary

Natural language processing (NLP) seeks to convert unstructured language data into a structured data format to enable machines to understand speech and text and formulate relevant, contextual responses. Its subtopics include natural language processing and natural language generation.

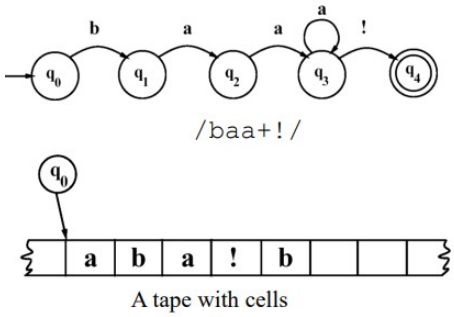
Natural language understanding (NLU) focuses on machine reading comprehension through grammar and context, enabling it to determine the intended meaning of a sentence.

Natural language generation (NLG) focuses on text generation, or the construction of text in English or other languages, by a machine and based on a given dataset.

Chapter 2 from "Speech and Language Processing" book

• An RE is one way of describing a FSA.

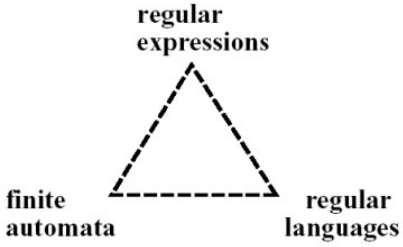
• An RE is one way of characterizing a particular kind of formal language called a regular language.



A tape with cells

	Input		
State	b	a	!
0	1	Ø	Ø
1	Ø	2	Ø
2	Ø	3	Ø
3	Ø	3	4
4:	Ø	Ø	Ø

The transition-state table

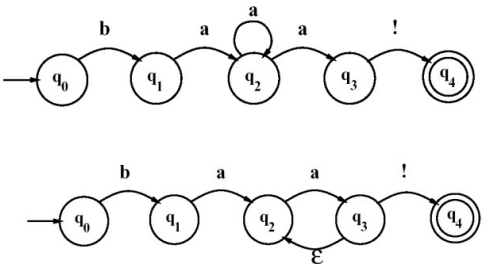


- Automaton (finite automaton, finite-state automaton (FSA))
- State, start state, final state (accepting state)

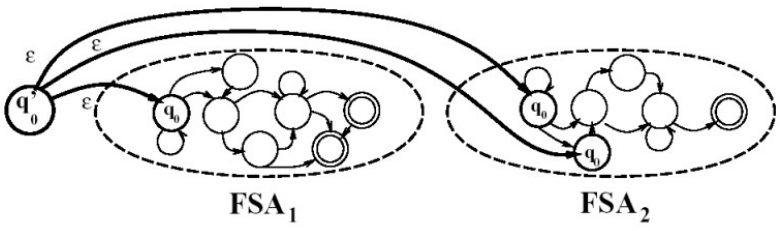
A finite automaton is formally defined by the following five parameters:

- Q: a finite set of N states $q_0, q_1, \ldots, q_N$
- Σ: a finite input alphabet of symbols
- q_0: the start state
- F: the set of final states, $F \subseteq Q$
- $\delta(q,i)$: the transition function or transition matrix between states. Given a state $q \in Q$ and input symbol $i \in \Sigma$, $\delta(q,i)$ returns a new state $q' \in Q$. δ is thus a relation from $Q \times \Sigma$ to Q;

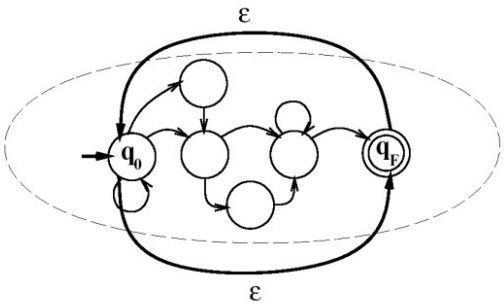
Non-Deterministic FSAs



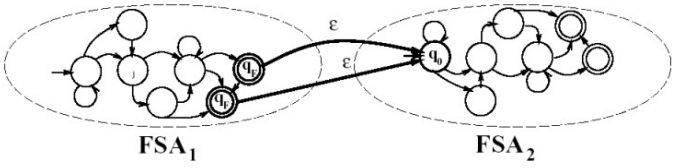
Primitive operations of RE (Concatenation, union and closure)



The union (|) of two FSAs



*The closure (Kleene *) of an FSAs*



The concatenation of two FSAs

Chapter 3. Morphology and Finite-State Transducers

The problem of recognizing that foxes breaks down into the two morphemes fox and -es is called morphological parsing

It takes two kinds of knowledge to correctly search for singulars and plurals of these forms: 1. Spelling rules tell us that English words ending in –y are pluralized by changing the -y to -i- and adding an -es.

Morphological rules tell us that fish has a null plural, and that the plural of goose is formed by changing the vowel. • Key Concept #2. Parsing means taking an input and producing some sort of structure for it.

Morphological parsing is necessary for more than just IR(Information Retrieval), but also

▪ Machine translation (to realize that the French words va and aller should both translate to forms of the English verb go.)

▪ Spelling checking (as we will see, it is morphological knowledge that will tell us that misclam and antiundoggingly are not words)

Finite-state transducers (FST)

FST is a type of FSA which maps between two sets of symbols. It is a two-tape automaton that recognizes or generates pairs of strings, one from each type. FST defines relations between sets of strings

Finite-state transducers for NLP :

● FST as recognizer – Takes a pair of strings and accepts or rejects them

● FST as generator – Outputs a pair of strings for a language

● FST as translator – Reads a string and outputs another string – Morphological parsing: letters (input); morphemes (output)

● FST as relater – Computes relations between sets

●Morphology and Morphological parsing: Breaking down words into components and building a structured representation.

Morphological analysis is an important task in NLP, the goal is to reduce words to its basic form. There are two possible ways of doing this:

Stemming is the process of reducing words to a base form (*e.g.*, "close" will be the root for "closed", "closing", "close", "closer" etc.). I does this based on rules, not a dictionary. In other words, the result may or may not be a real word, it is more like a prefix.
Lemmatization is the task of removing inflectional endings only and to return the base dictionary form of a word which is also known as a **lemma**. In other words, it produces a real word from a dictionary.

Tokenization or word segmentation separate out "words" (lexical entries) from running text expand abbreviated terms

Two broad classes of ways to form words from morphemes:

▪ Inflection: the combination of a word stem with a grammatical morpheme, usually resulting in a word of the same class as the original stem, and usually filling some syntactic function like agreement,
– Doesn't change the word class
– Usually produces a predictable meaning.

▪ Derivation: the combination of a word stem with a grammatical morpheme, usually resulting in a word of a different class, often with a meaning hard to predict exactly.

Inflectional Morphology

- In English, only nouns, verbs, and sometimes adjectives can be inflected, and the number of affixes is quite small.
- Inflections of nouns in English:
  - An affix marking **plural**,

Inflectional Morphology

- Verbal inflection is more complicated than nominal inflection.
  - English has three kinds of verbs:
    - **Main verbs**, *eat, sleep, impeach*
    - **Modal verbs**, *can will, should*
    - **Primary verbs**, *be, have, do*
  - Morphological forms of regular verbs:

stem	walk	merge	try	map
-s form	walks	merges	tries	maps
-ing principle	walking	merging	trying	mapping
Past form or *–ed* participle	walked	merged	tried	mapped

  - These regular verbs and forms are significant in the morphology of English because of their ***majority*** and being ***productive***.

Inflectional Morphology

- Morphological forms of irregular verbs

stem	eat	catch	cut
-s form	eats	catches	cuts
-ing principle	eating	catching	cutting
Past form	ate	caught	cut
–ed participle	eaten	caught	cut

Derivational Morphology

- **Nominalization** in English:
  - The formation of new nouns, often from verbs or adjectives

Suffix	Base Verb/Adjective	Derived Noun
-ation	computerize (V)	computerization
-ee	appoint (V)	appointee
-er	kill (V)	killer
-ness	fuzzy (A)	fuzziness

  - Adjectives derived from nouns or verbs

Suffix	Base Noun/Verb	Derived Adjective
-al	computation (N)	computational
-able	embrace (V)	embraceable
-less	clue (N)	clueless

Derivational Morphology

- Derivation in English is more complex than inflection because
  - Generally less productive
    - A nominalizing affix like *–ation* can not be added to absolutely every verb. *eatation*(*)
  - There are subtle and complex meaning differences among nominalizing suffixes. For example, *sincerity* has a subtle difference in meaning from *sincereness*.

Morphological parsing: parsing a word into stem and affixes and identifying the parts and their relationships.

3.2 Finite-State Morphological Parsing

- Parsing English morphology:

Input	Morphological parsed output
cats	`cat +N +PL`
cat	`cat +N +SG`
cities	`city +N +PL`
geese	`goose +N +PL`
goose	(`goose +N +SG`) or (`goose +V`)
gooses	`goose +V +3SG`
merging	`merge +V +PRES-PART`
caught	(`catch +V +PAST-PART`) or (`catch +V +PAST`)

Stems and morphological features

- *POS-tagger*

●N-grams
●N-gram Smoothing
●Markov Assumption(Independence Assumption)

8 (ish) traditional parts of speech ▪ Noun, verb, adjective, preposition, adverb, article, interjection, pronoun, conjunction, etc ▪ Called: parts-of-speech, lexical categories, word classes, morphological classes, lexical tags… ▪ Lots of debate within linguistics about the number, nature, and universality of these

- N noun *chair, bandwidth, pacing*
- V verb *study, debate, munch*
- ADJ adjective *purple, tall, ridiculous*
- ADV adverb *unfortunately, slowly*
- P preposition *of, by, to*
- PRO pronoun *I, me, mine*
- DET determiner *the, a, that, those*

POS Tagging

The process of assigning a part-of-speech or lexical class marker to each word in a collection.

WORD	tag
the	DET
koala	N
put	V
the	DET
keys	N
on	P
table	N

Why is POS Tagging Useful?

- First step of a vast number of practical tasks:
- **Speech synthesis**
  - How to pronounce "lead"?

INsult	inSULT
OBject	obJECT
OVERflow	overFLOW
DIScount	disCOUNT
CONtent	conTENT

- **Parsing**
  - Need to know if a word is an N or V before you can parse
- **Information extraction**
  - Finding names, relations, etc.
- **Machine Translation**

Two Methods for POS Tagging

1. Rule-based tagging (ENGTWOL):

• Start with a dictionary

• Assign all possible tags to words from the dictionary

• Write rules by hand to selectively remove tags

• Leaving the correct tag for each word

2. Stochastic 1. Probabilistic sequence models ▪ HMM (Hidden Markov Model) tagging ▪ MEMMs (Maximum Entropy Markov Models)

Independence Assumption

- This particular kind of independence assumption is called a **Markov assumption**
- So for each component in the product replace with the approximation (assuming a prefix of N)

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-N+1}^{n-1})$$

- Bigram version

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-1})$$

Problem

- Let's assume we're using N-grams
- How can we assign a probability to a sequence where one of the component n-grams has a value of zero
- Assume all the words are known and have been seen
  - Go to a lower order n-gram
  - Back off from bigrams to unigrams
  - Replace the zero with something else

Smoothing:

Add-One (Laplace)

- Make the zero counts 1.
- Rationale: They're just events you haven't seen yet.
- If you had seen them, chances are you would only have seen them once… so make the count equal to 1.

Add-one Smoothing

- For unigrams:
  - Add 1 to every word (type) count
  - Normalize by N (tokens) /(N (tokens) +V (types))
  - Smoothed count (adjusted for additions to N) is:

$$(c_i+1)\frac{N}{N+V}$$

  - Normalize by N to get the new unigram probability:

$$p_i^* = \frac{c_i+1}{N+V}$$

- For bigrams:
  - Add 1 to every bigram $c(w_{n-1}\, w_n) + 1$
  - Increase unigram count by vocabulary size $c(w_{n-1}) + V$

Add-one Smoothing

Add 1 to every N-gram count

- $P(w_n|w_{n-1}) = C(w_{n-1}w_n)/C(w_{n-1})$
- $P(w_n|w_{n-1}) = [C(w_{n-1}w_n) + 1] / [C(w_{n-1}) + V]$

Rule-based POS Tagging

One of the oldest techniques of tagging is rule-based POS tagging. Rule-based taggers use dictionary or lexicon for getting possible tags for tagging each word. If the word has more than one possible tag, then rule-based taggers use hand-written rules to identify the correct tag. Disambiguation can also be performed in rule-based tagging by analyzing the linguistic features of a word along with its preceding as well as following words. For example, suppose if the preceding word of a word is article then word must be a noun.

Stochastic POS Tagging

Another technique of tagging is Stochastic POS Tagging. Now, the question that arises here is which model can be stochastic. The model that includes frequency or probability (statistics) can be called stochastic. Any number of different approaches to the problem of part-of-speech tagging can be referred to as stochastic tagger.

Transformation-based Tagging

Transformation based tagging is also called Brill tagging. It is an instance of the transformation-based learning (TBL), which is a rule-based algorithm for automatic tagging of POS to the given text. TBL, allows us to have linguistic knowledge in a readable form, transforms one state to another state by using transformation rules.

It draws the inspiration from both the previous explained taggers − rule-based and stochastic. If we see similarity between rule-based and transformation tagger, then like rule-based, it is also based on the rules that specify what tags need to be assigned to what words. On the other hand, if we see similarity between stochastic and transformation tagger then like stochastic, it is machine learning technique in which rules are automatically induced from data.

Word Embedding	TF-IDF matrix
Multi dimensional vector which attempts to capture a words relationship to other words	Sparse matrix where each word maps to just a single value, captures no meaning
Often trained on large external corpus	Trained without external data
Must be applied to each word individually	Can be applied to each training document at once
More memory intensive	Less memory intensive
Ideal for problems involving a single word such as a word translation	Ideal for problems with many words and larger document files

Exercise:

Question One: *1Mark*

Suppose this is a small corpus:

```
<s> I am Maryam </s>
<s> Maryam I am </s>
<s> I do like English books </s>
<s> I do like teaching computer science in English Language</s>
```

Using the maximum likelihood estimate, find these probabilities:

a. P (I | <s>)
b. P (Maryam | <s>)
c. P (Do | I)
d. P (Books | English)

The Answer of Question One:

The Maximum Likelihood Estimate is: $P(W_n|W_{n-1}) = \frac{C(W_{n-1}W_n)}{C(W_{n-1})}$

a. $P(I|<s>) = \frac{3}{4} = 0.75$

b. $P(Maryam|<s>) = \frac{1}{4} = 0.25$

c. $P(Do|I) = \frac{2}{4} = 0.5$

d. $P(Books|English) = \frac{1}{2} = 0.5$

Question Four: *1.5 Mark*

Assign the correct part of speech (word class) to each word/token in the following sentences.

Note that: The table for Benn Treebank is attached for you in next papers of the quiz.

*It is a nice day.

*I am sitting in Mindy's restaurant putting on the gefillte fish, which is a dish I am very fond of

The Answer of Question Four:

a. It/PRP is/VBZ a/DT nice/JJ day/NN ./.
b. I/PRP am/VBP sitting/VBG in/IN Mindy/NNP 's/POS restaurant/NN putting/VBG on/RP the/DT gefillte/NN fish/NN ,/, which/WDT is/VBZ a/DT dish/NN I/PRP am/VBP very/RB fond/JJ of/RP

Question Two: *1Mark*

Compute the probabillity of the following sentences:

a. I want to eat lunch

b. I want to eat Chinese food

Note that: Some probability tables are attached in next papers of the quiz.

The Answer of Question Two:

a. P (I want to eat lunch) = $P(\text{I}|<s>)*P(\text{want}|\text{I})*P(\text{to}|\text{want})*P(\text{eat}|\text{to})*P(\text{lunch}|\text{eat})$

$= 0.25*0.32*0.65*0.26*0.06$

$= 8.112 \times 10^{-4}$

b. P (I want to eat Chiese food) = $P(\text{I}|<s>)*P(\text{want}|\text{I})*P(\text{to}|\text{want})*P(\text{eat}|\text{to})*P(\text{Chinese}|\text{eat})*P(\text{food}|\text{Chinese})$

$= 0.25*0.32*0.65*0.26*0.02*0.52$

$= 1.40608 \times 10^{-4}$

Question Three: *0.5 Mark*

If this is the equaion for bigram probability estimation

$$P(W_n|W_{n-1}) = \frac{C(W_{n-1}W_n)}{C(W_{n-1})}$$

Write out the equation for trigram probability estimation.

The Answer of Question Three:

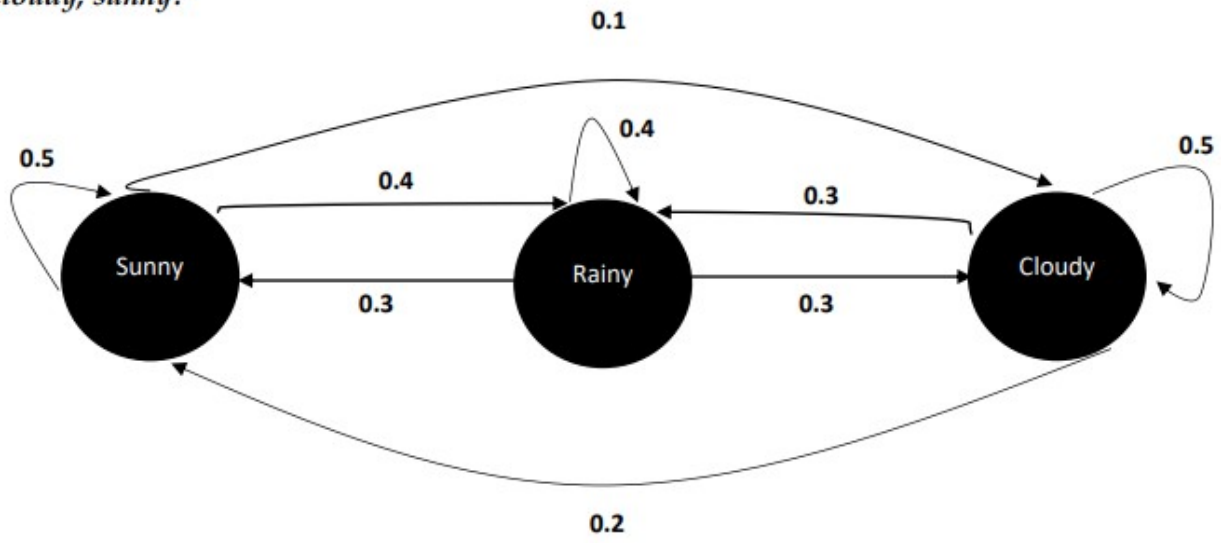
$$P(W_n|W_{n-1}, W_{n-2}) = \frac{C(W_{n-2}, W_{n-1}, W_n)}{C(W_{n-2}, W_{n-1})}$$

Question Five: *1 Mark*

Weather forecasting:

**Suppose tomorrow's weather depends on today's weather only.*

**Given today is sunny, what is the probability that the coming days are sunny, rainy, cloudy, cloudy, sunny?*



The Answer of Question Five:

The answer is : $(0.5)(0.4)(0.3)(0.5)(0.2) = 6 \times 10^{-3}$

From question 2.1 in EXCERCISES in the book:

Write regular expressions for the following languages:

a) The set of all strings from the alphabet a; b such that each a is immediately preceded and immediately followed by a b.

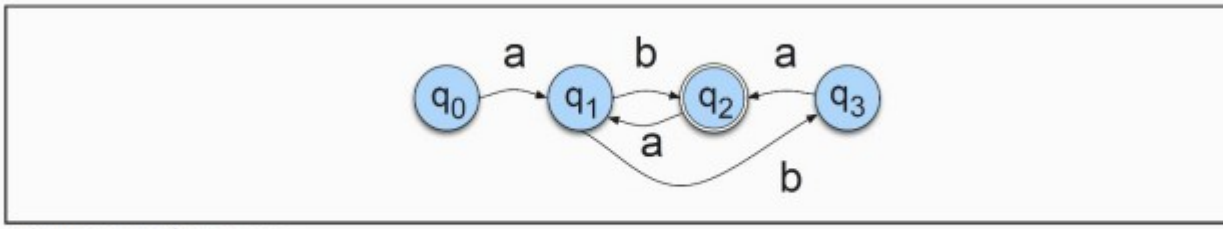
b) All strings that start at the beginning of the line with an integer and that end at the end of the line with a word.

Answer of Question Two:

a) (b + (ab+)+)?

b) ^\d +\b.*\b[a − zA − Z] + $

Write a regular expression for the language accepted by the NFSA in the following figure:



A mystery language.

Answer of Question Four:

(aba?) +

Question

Question: Which one of the following languages over the alphabet {0,1} is described by the regular expression?

(0+1)*0(0+1)*0(0+1)*

(A) The set of all strings containing the substring 00.

(B) The set of all strings containing at most two 0's.

(C) The set of all strings containing at least two 0's.

(D) The set of all strings that begin and end with either 0 or

Question: Which of the following languages is generated by given grammar?

S -> aS | bS | ∈

(A) $\{a^n b^m \mid n, m \geq 0\}$

(B) {w ∈ {a,b}* | w has equal number of a's and b's}

(C) $\{a^n \mid n \geq 0\} \cup \{b^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$

(D) {a, b}*

Question: The regular expression 0*(10*)* denotes the same set as:

(A) (1*0)*1*

(B) 0 + (0 + 10)*

(C) (0 + 1)* 10(0 + 1)*

(D) none of these

Question One: *2 Marks*

Find one tagging error in each of the following sentences that are tagged with the Penn Treebank tagset:

1.I/PRP need/VBP a/DT flight/NN from/IN Atlanta/NN

2.Does/VBZ this/DT flight/NN serve/VB dinner/NNS

3.I/PRP have/VB a/DT friend/NN living/VBG in/IN Denver/NNP

4.Can/VBP you/PRP list/VB the/DT nonstop/JJ afternoon/NN flights/NNS

Question One Answer:

1. Atlanta/**NNP**

2. dinner/**NN**

3. have/**VBP**

4. Can/**MD**

Question Two: *2 Marks*

Use the Penn Treebank tagset to tag each word in the following sentences from Damon Runyon's short stories. You may ignore punctuation.

1. It is a nice night.

2.This crap game is over a garage in Fifty-second Street. . .

3.He is a tall, skinny guy with a long, sad, mean-looking kisser, and a mournful voice.

4.. . .Nobody ever takes the newspapers she sells . . .

Question Two Answer:

1. It/**PRP** is/**VBZ** a/**DT** nice/**JJ** night/**NN** ./.

2. This/**DT** crap/**NN** game/**NN** is/**VBZ** over/**IN** a/**DT** garage/**NN** in/**IN** Fifty-second/**NNP** Street/**NNP**. . .

3. He/**PRP** is/**VBZ** a/**DT** tall/**JJ** ./, skinny/**JJ** guy/**NN** with/**IN** a/**DT** long/**JJ** ./, sad/**JJ** ./, mean-looking/**JJ** kisser/**NN** ./, and/**CC** a/**DT** mournful/**JJ** voice/**NN** ./.

4. . . .Nobody/**NN** ever/**RB** takes/**VBZ** the/**DT** newspapers/**NNS** she/**PRP** sells/**VBZ**. . .