自然语言处理

week-1

凌震华

2024年2月29日

- 自然语言处理概述与课程介绍
- 正则表达式 (Regular Expression)
- 有限状态自动机 (Finite State Automata)
- 分词 (Word Segmentation)

自然语言处理概念

自然语言处理

- 用计算机对自然语言的形、音、义等信息进行处理，即对字、词、句、篇章的输入、输出、识别、分析、理解、生成等的操作和加工。

两个流程

- 自然语言理解&自然语言生成

自然语言理解

- 计算机能够理解自然语言文本的意义

自然语言生成

- 计算机能够以自然语言文本来表达给定的意图

自然语言处理表现形式

- 机器翻译、文本摘要、文本分类、文本校对、信息抽取、语音合成、语音识别等。

NLP发展历程

20世纪50年代-70年代	• 图灵测试的提出——自然语言处理思想的开端 • 基于规则的方发——理性主义思潮
20世纪70年代以后	• 语料库不断丰富 • 基于统计的方法——IBM华生实验室起了推动作用 • 理性主义思潮向经验主义思潮过渡
2008年至今	• 深度学习与自然语言处理相结合

NLP技术分类

基础技术

- 词法与句法分析
- 语义分析
- 语篇分析
- 知识图谱
- 语言认知模型
- 语言知识表示与深度学习

应用技术

- 机器翻译
- 信息检索
- 情感分析
- 自动问答
- 自动文摘
- 信息抽取
- 信息推荐与过滤
- 文本分类与聚类
- 文字识别

NLP基础技术

词法分析

- 主要任务：词性标注和词义标注
- 词性标注方法：基于规则和基于统计

句法分析

- 主要任务：判断句子的句法结构和成分，明确各成分的相互关系 分类：完全句法分析、浅层句法分析
- 策略：“先句法后语义”、“句法语义 一体化”（占主流）

语义分析

- 根据句子的句法结构和句子中每 个实词的词义推导出来能够反映 这个句子意义的某种形式化表示

语用分析

- 人对语言的具体运用，是对自然语言的深层理解

篇章分析

- 对段落和整篇文章进行理解和分析

NLP应用技术（一）

机器翻译

概念

- 通过特定的计算机程序将一种书写形式或声音形式的自然语言，翻译成另一种书写形式或 声音形式的自然语言

方法

- 基于理性的研究方法—基于规则的方法
- 基于经验的研究方法—基于统计的方法
- 与深度学习相结合

应用

- 亚马逊的Alexa、苹果的Siri、微软的Cortana等
- 科大讯飞的晓译翻译机

NLP应用技术（二）

信息检索

概念

- 从相关文档集合中查找用户所需信息的过程

原理

- “存”：对信息进行收集、标引、描述、组织，进行有序的存放
- “取”：按照某种查询机制从有序存放的信息集合（数据库）中找出用户所需信 息或获取其线索
- 检索成功：将用户输入的检索关键词与数据库中的标引词进行对比，二者匹配成功时检索成功
- 检索结果按照与提问词的关联度输出，供用户选择，用户采用“关键词查询+选择性浏览”的交互方式获取信息

NLP应用技术（三）

情感分析

概念

- 通过计算技术对文本的主客观性、观点、情绪、极性的挖掘和分析，对文本的情 感倾向做出分类判断

应用

- 评论机制的App中应用较为广泛
- 互联网舆情分析中情感分析起着举足轻重的作用
- 选举预测、股票预测等领域

NLP应用技术（四）

自动问答

概念

- 利用计算机自动回答用户所提出的问题以满足用户知识需求的任务

分类

- 检索式问答：通过检索和匹配回答问题，推理能力较弱
- 知识库问答：web2.0的产物，用户生成内容是其基础，Yahoo！Answer、百度知道等是典型代表
- 社区问答：正在逐步实现知识的深层逻辑推理

工作流程

- 首先要正确理解用户所提出的问题，
- 抽取其中关键的信息，在已有的语料库或者知识库中进行检索、匹配，
- 将获取的答案反馈给用户

NLP应用技术（五）

自动文摘

概念

- 运用计算机技术，依据用户需求从源文本中提取最重要的信息内容，进行精简、 提炼和总结，最后生成一个精简版本

特点

- 压缩性
- 内容完整性
- 可读性

分类

- 基于统计的机械式文摘：简单容易实现，是目前主要被采用的方法，但是结果不 尽如人意
- 基于意义的理解式文摘：建立在对自然语言的理解的基础之上的，接近于人提取摘要的方法，难度较大

NLP应用技术（六）

社会计算

概念

- 在互联网的环境下，以现代信息技术为手段，以社会科学理论为指导，帮助人们分析社会关 系，挖掘社会知识，协助社会沟通，研究社会规律，破解社会难题

社会媒体

- 文本属性：草根性，字数少、噪声大、书写随意、实时性强
- 社会属性：社交性，在线、交互
- 典型社会媒体：Twitter、Facebook、微信、微博

应用

- 金融市场采用社会计算方法探索金融风险和危机的动态规律
- 社会安全：把握舆情、引导舆论
- 军事方面：许多国家加大投入力度扶持军事信息化的发展

NLP应用技术（七）

信息抽取

概念

- 从文本中抽取出特定的事实信息。这些被抽取出来的信息通常以结构化的形式直 接存入数据库，可以供用户查询及进一步分析使用，为之后构建知识库、智能问 答等提供数据支撑

原理

- 利用自然语言处理的技术，包括命名实体识别、句法分析、篇章分析与推理以及 知识库等，对文本进行深入理解和分析完成信息抽取工作

应用

- 信息抽取技术对于构建大规模的知识库有着重要的意义，但是目前由于自然语言 本身的复杂性、歧义性等特征，而且信息抽取目标知识规模巨大、复杂多样等问 题，使得信息抽取技术还不是很完善

NLP业界发展

自然语言处理

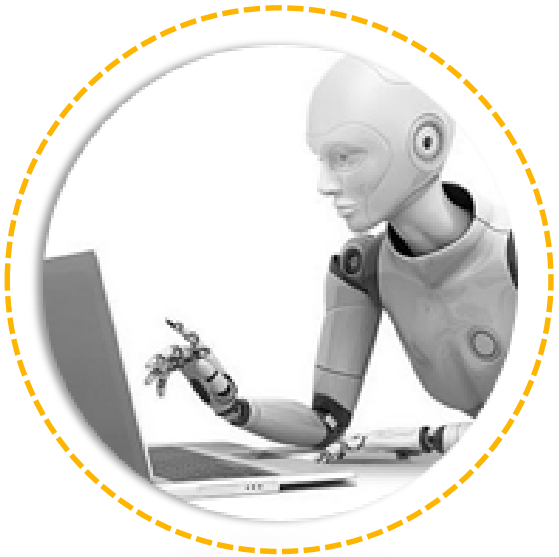
人工智能三个阶段

计算智能

能存会算

感知智能

能听会说、能看会认



认知智能

能理解会思考

“得语言者得天下”

- 语义理解
- 知识表示
- 联想推理
- 智能问答
- 自主学习

“深度学习的下一个大的进展应该是让神经网络真正理解文档的内容”

—— Geoffrey Hinton

“深度学习的下一个前沿课题是自然语言理解

——Yann LeCun

“如果给我10亿美金，我会用这10亿美金建造一个NASA级别的自然语言处理研究项目”

—— Michael I. Jordan

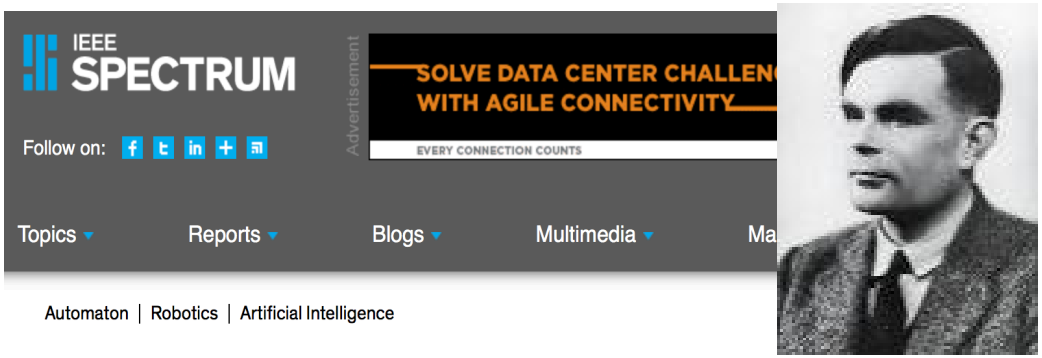
“下一个十年，懂语言者得天下”

——沈向洋

自然语言理解（NLU）也被称为“人工智能皇冠上的一颗明珠”！

“得语言者得天下”

- **Winograd Schema Challenge(WSC)**
  - 基于自然语言的常识推理评测任务
  - 取代图灵测试用于评估机器的智能水平



  - 示例

爸爸没法举起他的儿子，因为他很重。

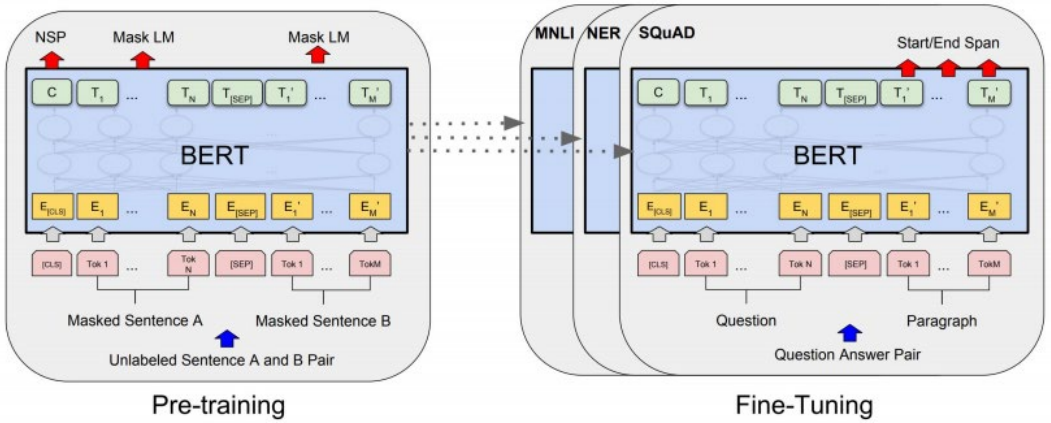
问：谁重？ 答：儿子

爸爸没法举起他的儿子，因为他很虚弱。

问：谁虚弱？ 答：爸爸

预训练语言模型

- BERT(**B**idirectional **E**ncoder **R**epresentations from **T**ransformers) (Google 2018)



GLUE Results

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$BERT_{BASE}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$BERT_{LARGE}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

- GPT-3 (**G**enerative **P**re-trained **T**ransformer 3) (OpenAI 2020)

机器之心 原创
2020/05/30 12:15

机器之心编辑部
报道

1750亿参数，史上最大AI模型GPT-3上线：不仅会写文章、答题，还懂数学

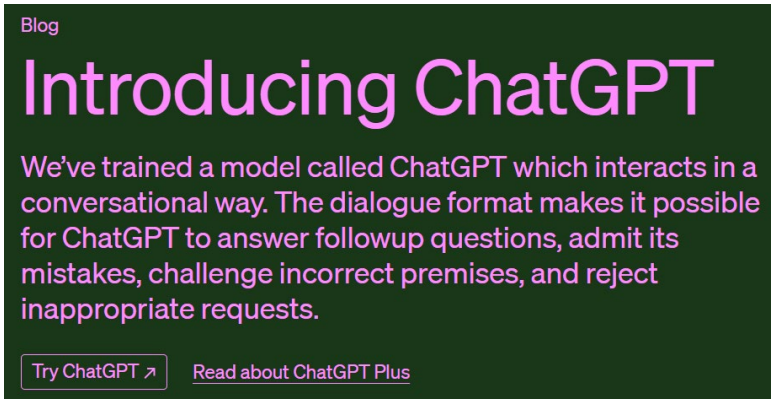
时隔一年，OpenAI 放出的预训练语言模型 GPT-3 再次让人刮目相看。

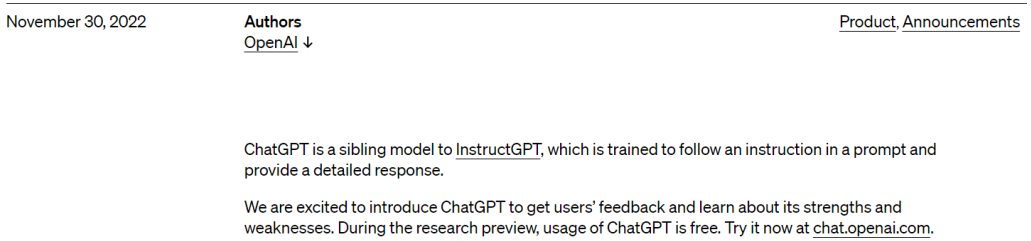
「我们训练了 GPT-3，一种具有 1750 亿参数的自回归语言模型，这个数字比以往任何非稀疏语言模型都多 10 倍。我们在 few-shot 情况下测试了它的性能。」

```
Article:  After two days of intense debate, the United Methodist Church
has agreed to a historic split - one that is expected to end in the
creation of a new denomination, one that will be "theologically and
socially conservative," according to The Washington Post.  The majority of
delegates attending the church's annual General Conference in May voted to
strengthen a ban on the ordination of LGBTQ clergy and to write new rules
that will "discipline" clergy who officiate at same-sex weddings.  But
those who opposed these measures have a new plan:  They say they will form a
separate denomination by 2020, calling their church the Christian Methodist
denomination.
```

在 OpenAI 的测试中，人类评估人员也很难判断出这篇新闻的真假，检测准确率仅为 12%。

大语言模型(Large LM，LLM)





- 海量训练数据
- 复杂神经网络
- 大量计算资源
- 通用语言理解与生成
- 上下文理解
- 可适应性和微调

课程内容

NLP基础技术

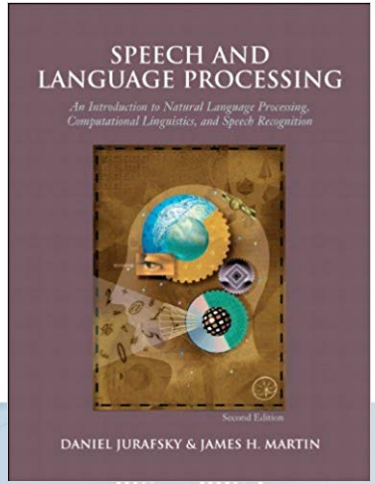
- 正则表达式与自动机
- 语言模型
- 词性标注
- 句法剖析
- 统计剖析
- 向量语义
- 词汇语义
- 词义排岐

NLP应用技术

- 信息抽取
- 问答
- 信息检索
- 摘要
- 对话系统
- 机器翻译

课程信息

- 课程讲座：12周
- 课程信息：QQ群 851237125
- 教材
  - 《自然语言处理综论》（第二版），冯志伟/孙乐 译，中国工信出版集团，2018.
  - Martin J H, Jurafsky D. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition[M]. Pearson/Prentice Hall, 2009.
  - https://web.stanford.edu/~jurafsky/slp3/



2024自然语言处理

群号: 851237125

扫一扫二维码，加入群聊

QQ

考核方法

- 考核方法
  - 期末考试　70%
  - 阅读报告　20%
  - 出勤等其他　10%

联系方式

- 凌震华
  - zhling@ustc.edu.cn
  - 西区科技实验楼语音及语言信息处理国家工程实验室503室 / 高新区1号学科楼B507室
- 助教：闫世奇
  - yansiki@mail.ustc.edu.cn
  - 高新校区信智楼C501室

- 自然语言处理概述与课程介绍
- 正则表达式 (Regular Expression)
- 有限状态自动机 (Finite State Automata)
- 分词 (Word Segmentation)

Regular Expressions and Text Searching

- Everybody does it
  - Emacs, vi, grep, Perl, Python, Word, etc..

- A regular expression (正则表达式) is a formula in a special language that specifies simple classes of strings

- Regular expression search requires a pattern (模式) that we want to search for a corpus (语料库) of texts to search through

Disjunctions (析取)

- Letters inside square brackets [ ]

Pattern	Matches
`[wW]oodchuck`	Woodchuck, woodchuck
`[1234567890]`	Any digit

- Ranges [A-Z]

Pattern	Matches	
`[A-Z]`	An upper case letter	`Drenched Blossoms`
`[a-z]`	A lower case letter	`my beans were impatient`
`[0-9]`	A single digit	`Chapter 1: Down the Rabbit Hole`

Negation (否定) in Disjunction

- Negations [^Ss]
  - Carat means negation only when first in []

Pattern	Matches	
[^A-Z]	Not an upper case letter	Oyfn pripetchik
[^Ss]	Neither 'S' nor 's'	I have no exquisite reason"
[e^]	Either e or ^	Look here
a^b	The pattern a carat b	Look up a^b now

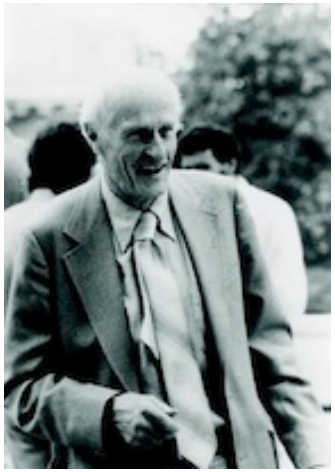
More Disjunction

- Woodchucks is another name for groundhog!
- The pipe | for disjunction

Pattern	Matches
`groundhog|woodchuck`	
`yours|mine`	yours mine
`a|b|c`	= [abc]
`[gG]roundhog|[Ww]oodchuck`	

? * + .

Pattern	Matches	
colou?r	Optional previous char	color colour
oo*h!	0 or more of previous char	oh! ooh! oooh! ooooh!
o+h!	1 or more of previous char	oh! ooh! oooh! ooooh!
baa+		baa baaa baaaa baaaaa
beg.n		begin begun begun beg3n



Stephen C Kleene

Kleene *, Kleene +

Anchors (锚号) ^ $

- Anchors are special characters that anchor REs to particular places in a string

Pattern	Matches
^[A-Z]	Palo Alto
^[^A-Za-z]	1 "Hello"
\.$	The end.
.$	The end? The end!

Example

- Find me all instances of the word "the" in a text.

`the`

Misses capitalized examples

`[tT]he`

Incorrectly returns `other` or `theology`

`[^a-zA-Z][tT]he[^a-zA-Z]`

Errors

- The process we just went through was based on two fixing kinds of errors
  - Matching strings that we should not have matched (there, then, other)
    - False positives (Type I)
  - Not matching things that we should have matched (The)
    - False negatives (Type II)

Errors

- We'll be telling the same story for many tasks, all semester. Reducing the error rate for an application often involves two antagonistic efforts:
  - Increasing accuracy, or precision, (minimizing false positives)
  - Increasing coverage, or recall, (minimizing false negatives)

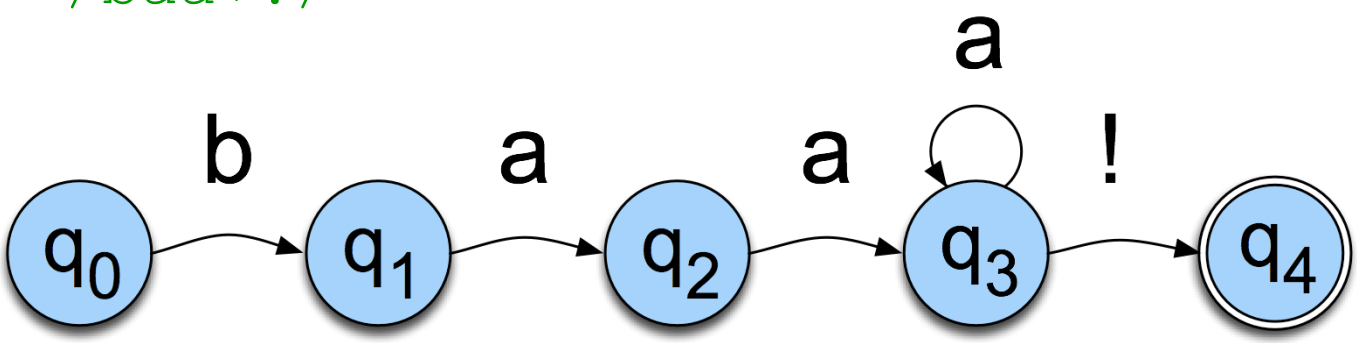
- 自然语言处理概述与课程介绍
- 正则表达式 (Regular Expression)
- 有限状态自动机 (Finite State Automata)
- 分词 (Word Segmentation)

Finite State Automata (有限状态自动机)

- Regular expressions can be viewed as a textual way of specifying the structure of finite-state automata
- Both regular expressions and finite state antomata can be used to describe regular language (正则语言)
- Regular language is a particular kind of formal language (形式语言)

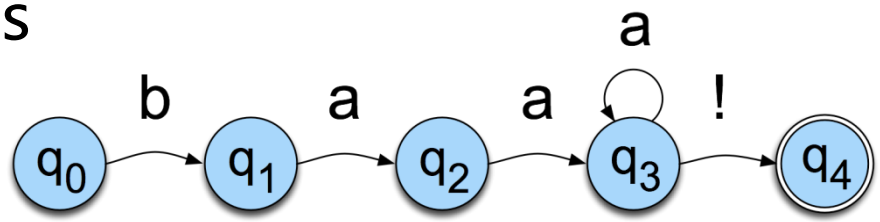
FSAs as Graphs

- Let's start with the sheep language from the text
  - /baa+!/



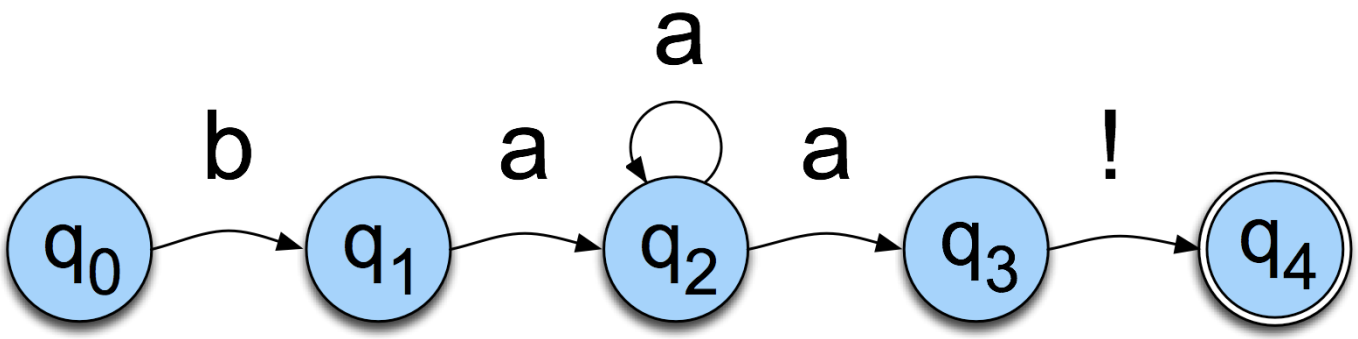
Sheep FSA

- We can say the following things about this machine
  - It has 5 states
  - b, a, and ! are in its alphabet
  - q0 is the start state
  - q4 is an accept state
  - It has 5 transitions



But note

- There are other machines that correspond to this same language



- More on this one later

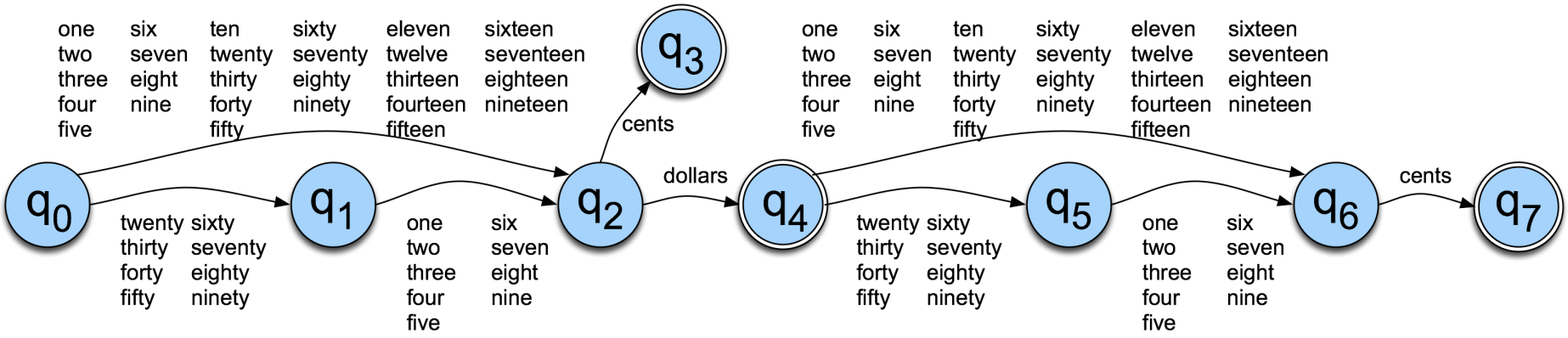
More Formally

- You can specify an FSA by enumerating the following things.
  - The set of states (状态): Q
  - A finite alphabet (字母表): Σ
  - A start state (初始状态)
  - A set of accept/final states (终极状态)
  - A transition function (转移函数) that maps QxΣ to Q

About Alphabets

- Don't take that word to narrowly; it just means we need a finite set of symbols in the input.
- These symbols can and will stand for bigger objects that can have internal structure.

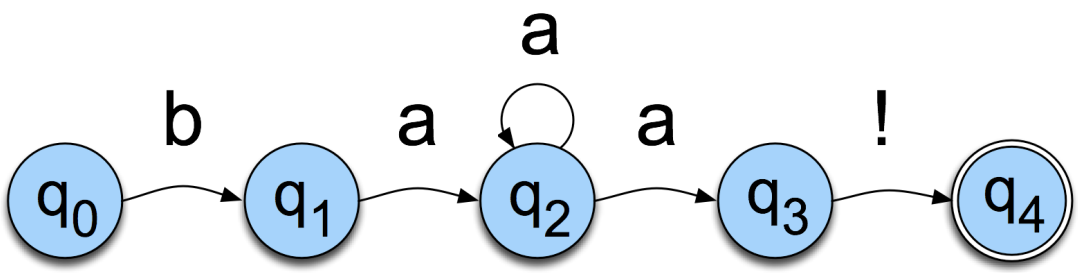
Dollars and Cents



Yet Another View

- Represent an automation with a state-transition table (状态转移表)

State	b	a	!	e
0	1			
1		2		
2		2,3		
3			4	
4				



Formal Languages

- Formal Languages (形式语言) are sets of strings composed of symbols from a finite set of symbols.
- Finite-state automata define formal languages (without having to enumerate all the strings in the language)

Formal Languages

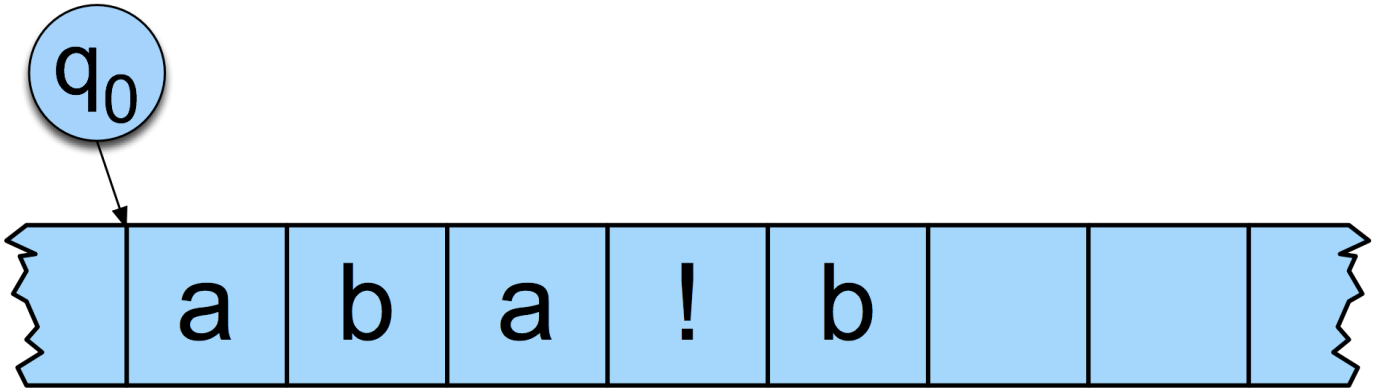
- FSAs can be viewed from two perspectives:
  - Generators to produce all and only the strings in the language
  - Acceptors that can tell you if a string is in the language

Recognition (识别)

- Recognition is the process of determining if a string should be accepted by a machine
- Or… it's the process of determining if a string is in the language we're defining with the machine
- Or… it's the process of determining if a regular expression matches a string
- Those all amount the same thing in the end

Recognition

- Traditionally, (Turing's idea) this process is depicted with a tape.



Recognition

- Simply a process of starting in the start state
- Examining the current input
- Consulting the table
- Going to a new state and updating the tape pointer.
- Until you run out of tape.

Key Points

- Deterministic (确定性) means that at each point in processing there is always one unique thing to do (no choices).
- Can be done with a simple table-driven interpreter as in Figure 2.12 (D-recognize)
- The algorithm is universal for all unambiguous (无歧义) regular languages.
  - To change the machine, you just change the table.

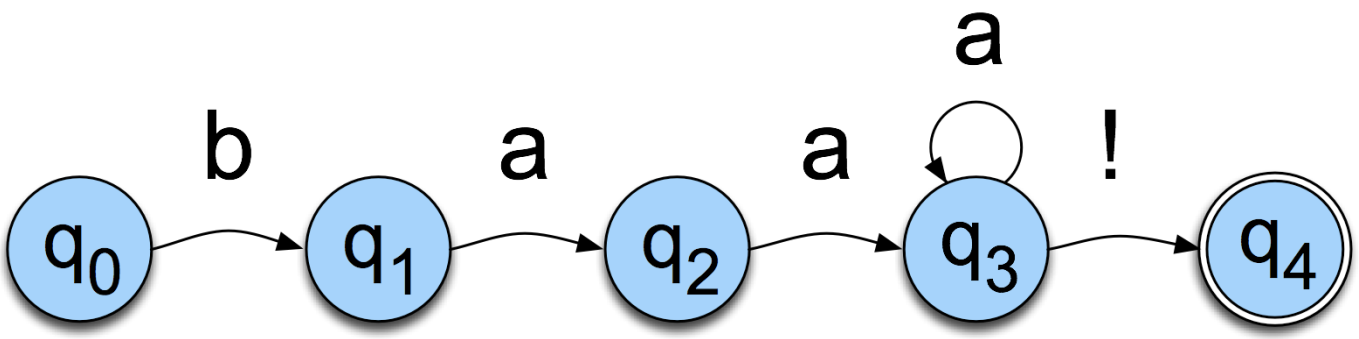
Key Points

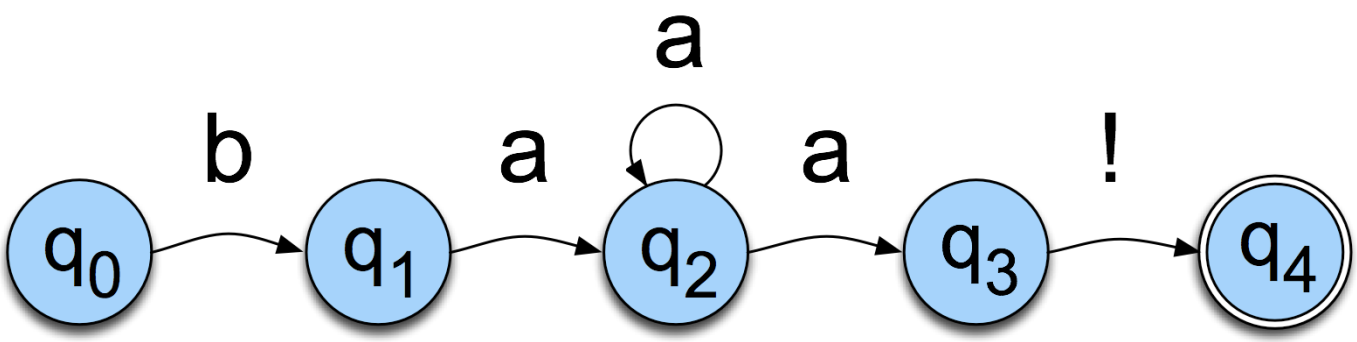
- Crudely therefore… matching strings with regular expressions (ala Perl, grep, etc.) is a matter of
  - translating the regular expression into a machine (a table) and
  - passing the table to an interpreter

Recognition as Search

- You can view this algorithm as a trivial kind of state-space search.
- States are pairings of tape positions and state numbers.
- Operators are compiled into the table
- Goal state is a pairing with the end of tape position and a final accept state

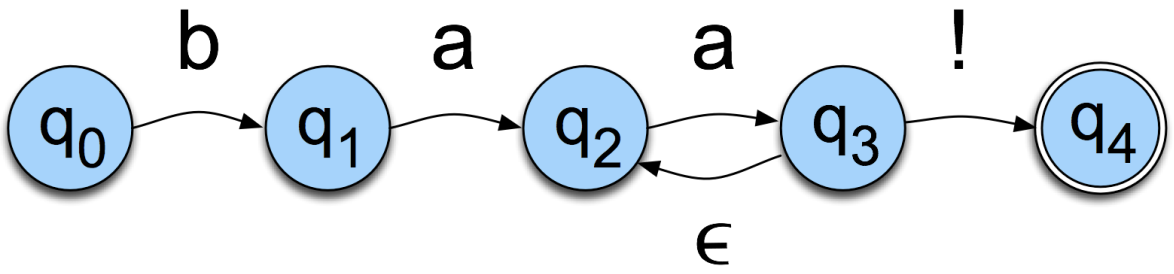
Non-Determinism





Non-Determinism cont.

- Yet another technique
  - Epsilon transitions
  - Key point: these transitions do not examine or advance the tape during recognition



Equivalence

- Non-deterministic machines can be converted to deterministic ones with a fairly simple construction
- That means that they have the same power; non-deterministic machines are not more powerful than deterministic ones in terms of the languages they can accept

ND Recognition

- Two basic approaches (used in all major implementations of Regular Expressions)
  1. Either take a ND machine and convert it to a D machine and then do recognition with that.
  2. Or explicitly manage the process of recognition as a state-space search (leaving the machine as is).

ND Recognition: Implementation

Program	(Original) Author	Version	Regex Engine
awk	Aho, Weinberger, Kernighan	*generic*	DFA
new *awk*	Brian Kernighan	*generic*	DFA
GNU *awk*	Arnold Robbins	*recent*	Mostly DFA, some NFA
MKS *awk*	Mortice Kern Systems		POSIX NFA
mawk	Mike Brennan	*all*	POSIX NFA
egrep	Alfred Aho	*generic*	DFA
MKS *egrep*	Mortice Kern Systems		POSIX NFA
GNU Emacs	Richard Stallman	*all*	Trad. NFA (POSIX NFA available)
Expect	Don Libes	*all*	Traditional NFA
expr	Dick Haight	*generic*	Traditional NFA
grep	Ken Thompson	*generic*	Traditional NFA
GNU *grep*	Mike Haertel	Version 2.0	Mostly DFA, but some NFA
GNU *find*	GNU		Traditional NFA
lex	Mike Lesk	*generic*	DFA
flex	Vern Paxson	*all*	DFA
lex	Mortice Kern Systems		POSIX NFA
more	Eric Schienbrood	*generic*	Traditional NFA
less	Mark Nudelman		Variable (usually Trad. NFA)
Perl	Larry Wall	*all*	Traditional NFA
Python	Guido van Rossum	*all*	Traditional NFA
sed	Lee McMahon	*generic*	Traditional NFA
Tcl	John Ousterhout	*all*	Traditional NFA
vi	Bill Joy	*generic*	Traditional NFA

ND Recognition: Search

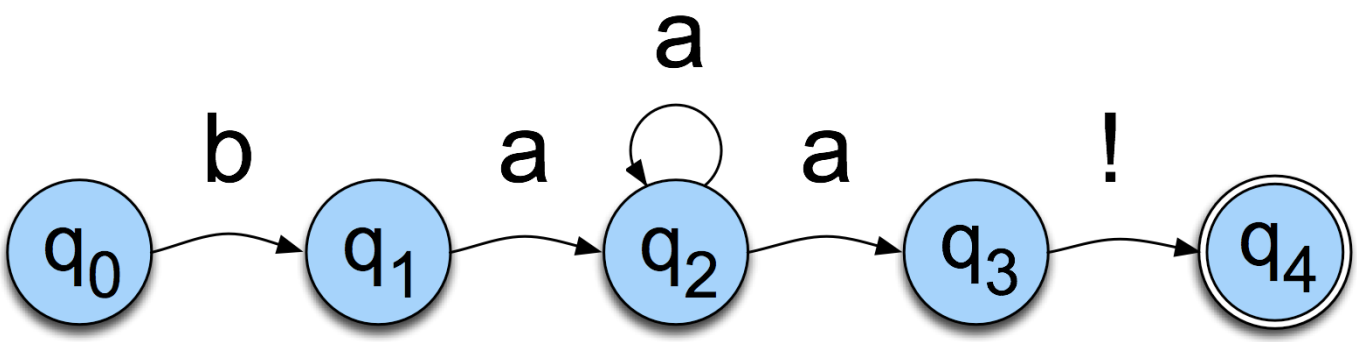
- In a ND FSA there exists at least one path through the machine for a string that is in the language defined by the machine.
- But not all paths directed through the machine for an accept string lead to an accept state.
- No paths through the machine lead to an accept state for a string not in the language.

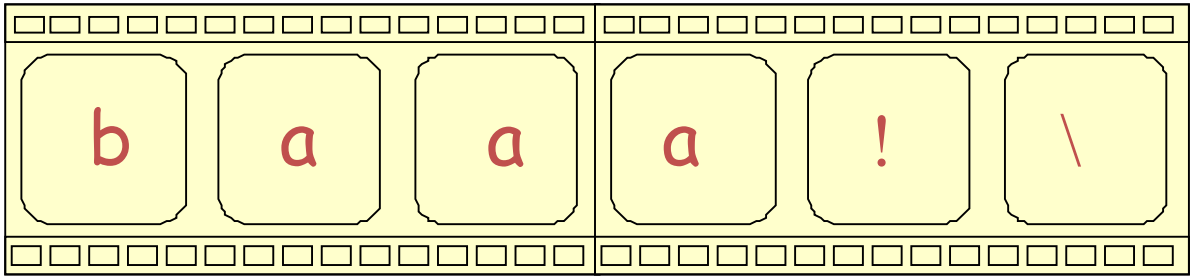
ND Recognition

- So success in a non-deterministic recognition occurs when a path is found through the machine that ends in an accept.

- Failure occurs when all of the possible paths lead to failure.

Example

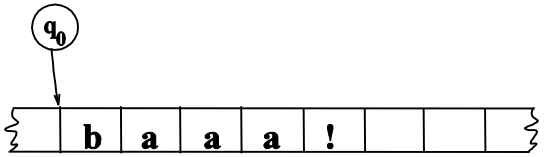




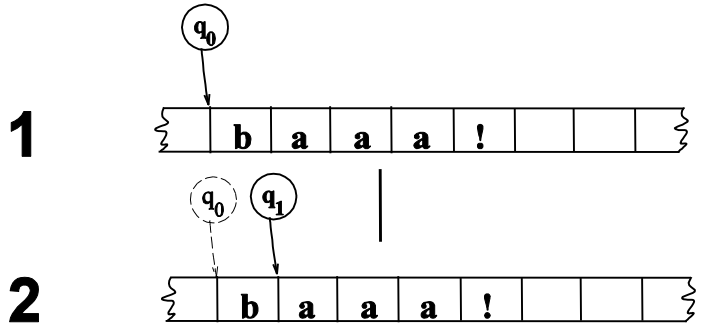
q_0 q_1 q_2 q_2 q_3 q_4

Example

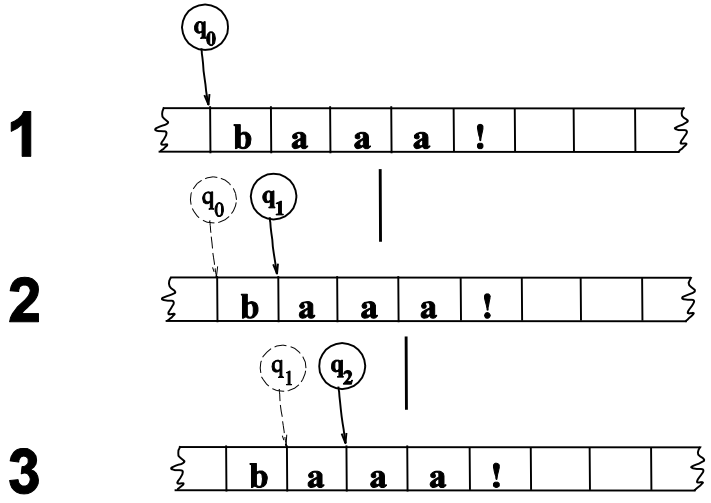
1



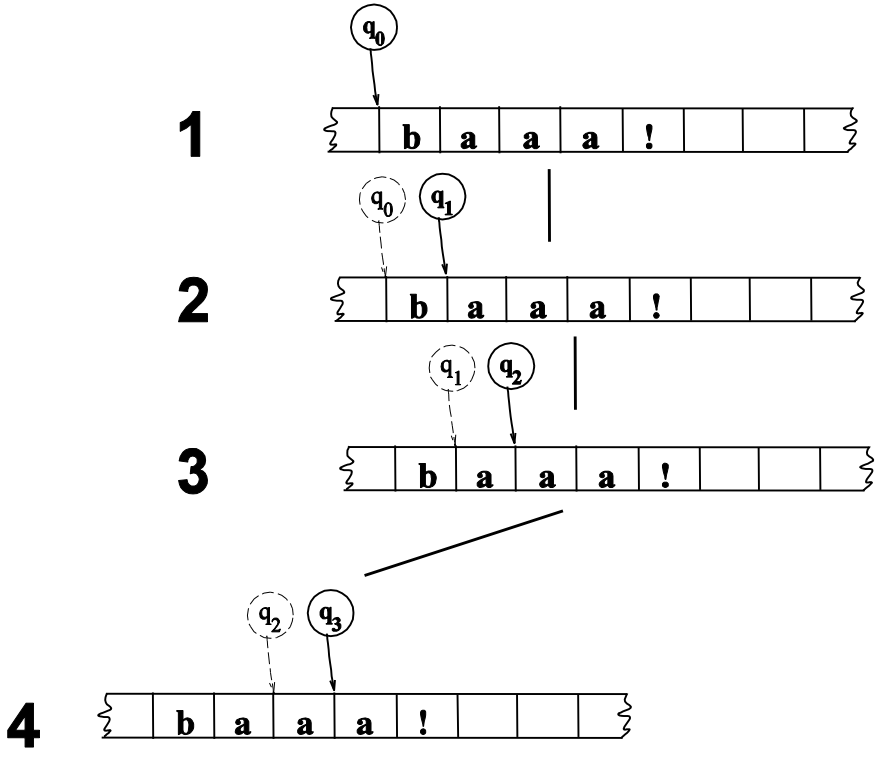
Example



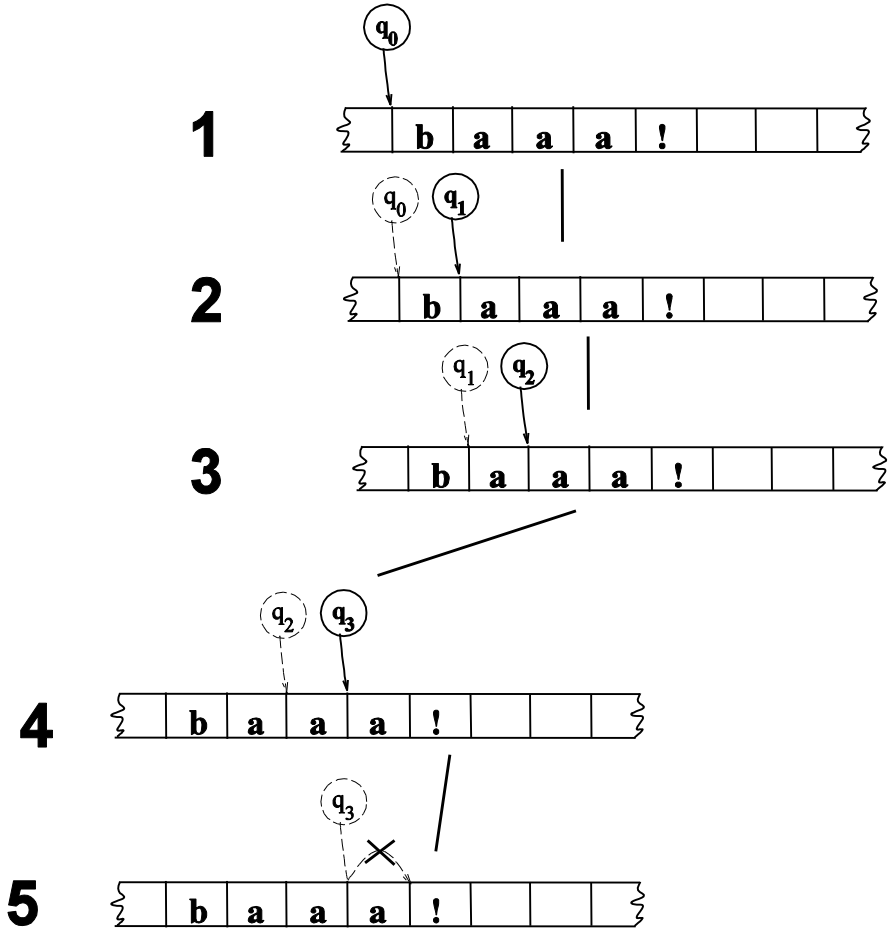
Example



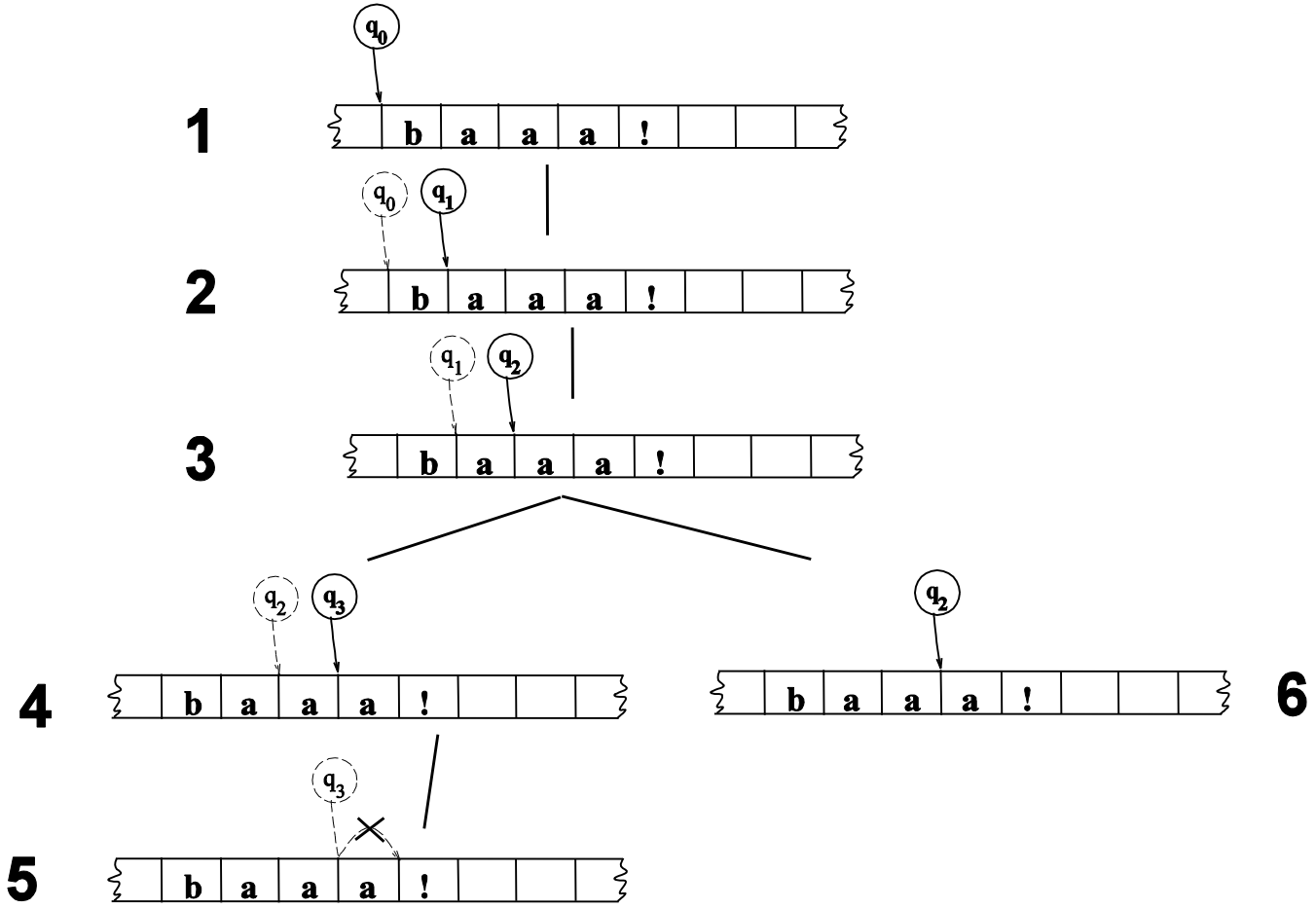
Example



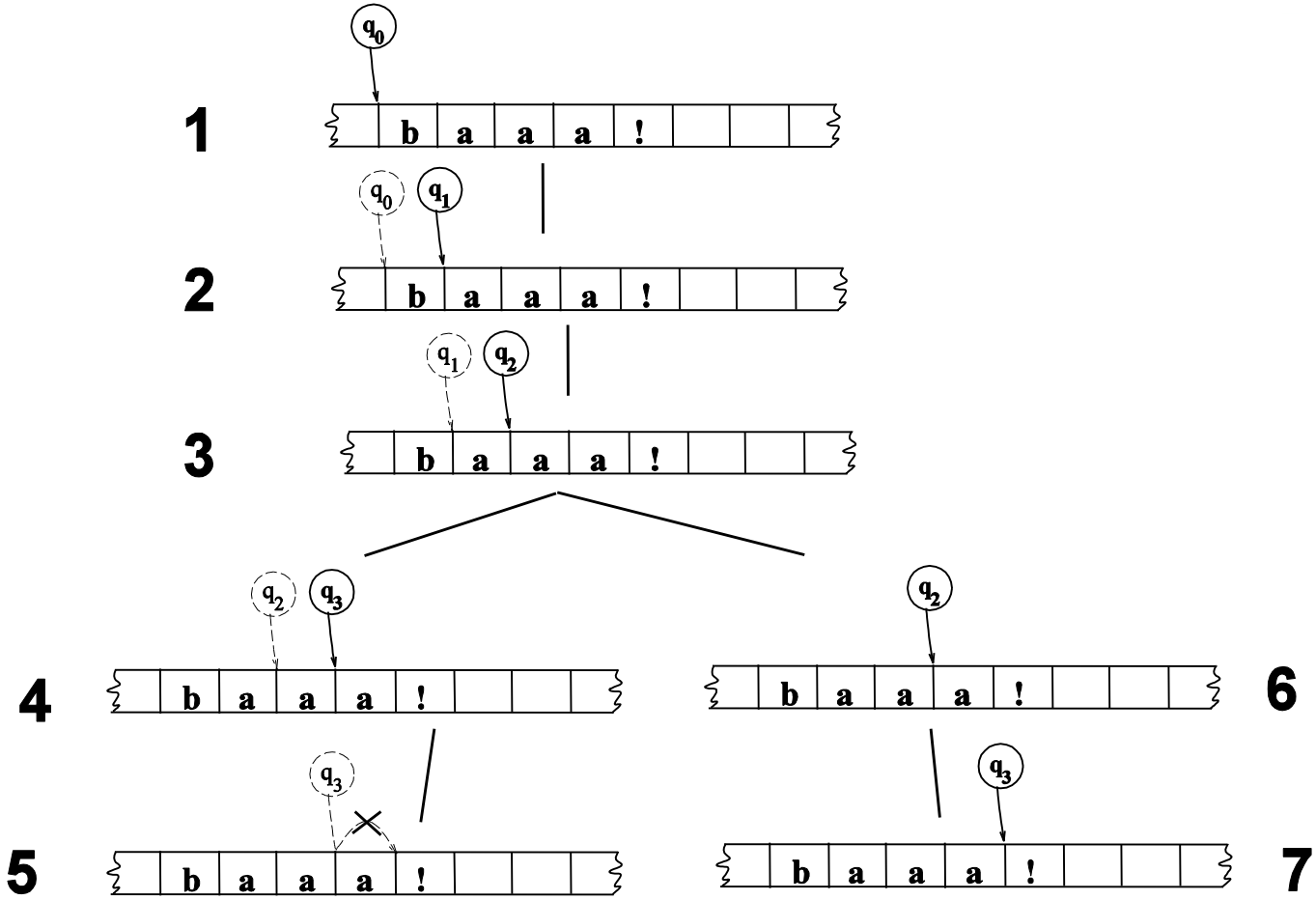
Example



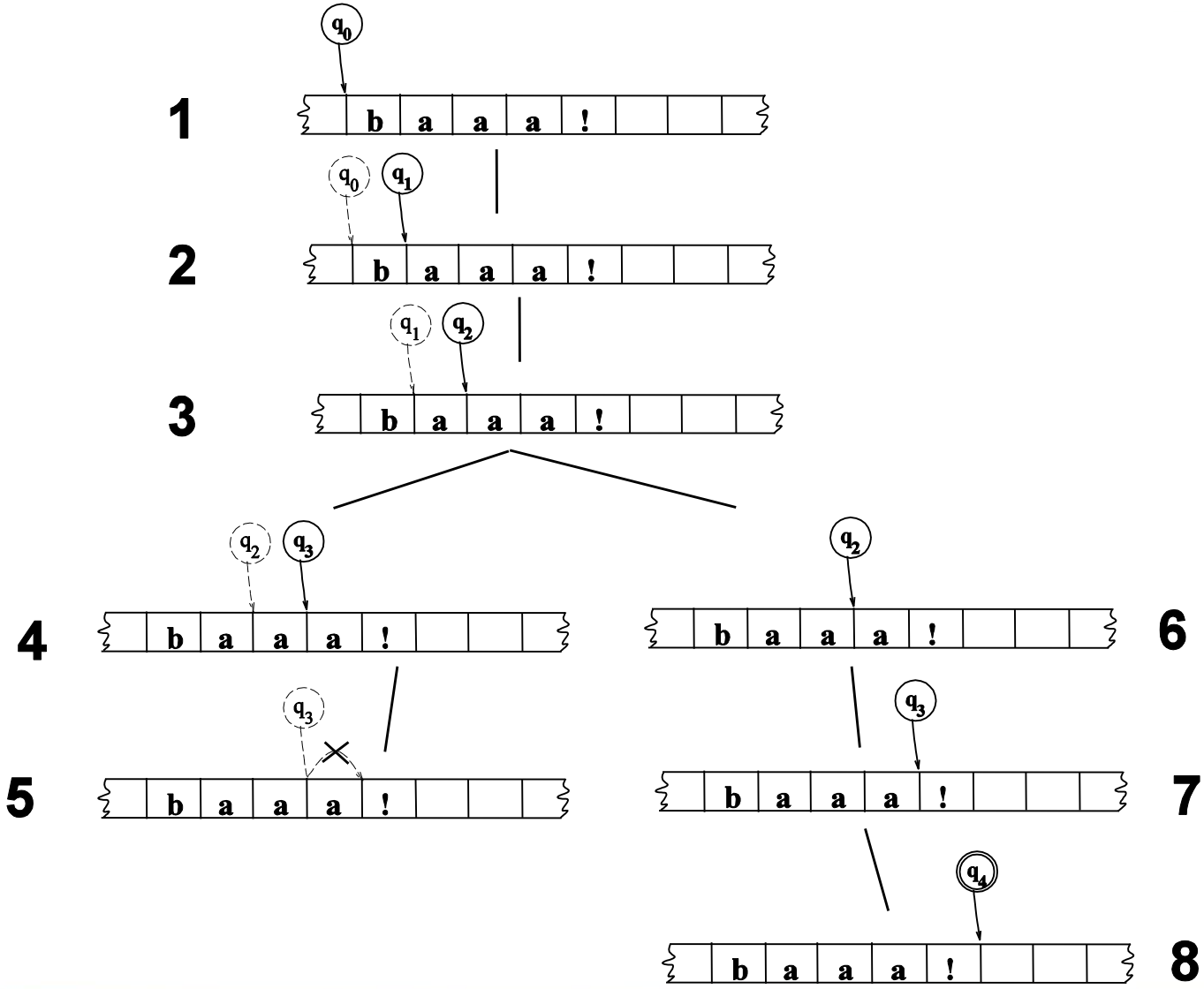
Example



Example



Example



Key Points

- States in the search space are pairings of tape positions and states in the machine.
- By keeping track of as yet unexplored states, a recognizer can systematically explore all the paths through the machine given an input.

- 自然语言处理概述与课程介绍
- 正则表达式 (Regular Expression)
- 有限状态自动机 (Finite State Automata)
- 分词 (Word Segmentation)

Text Normalization (文本规整)

- Every NLP task needs to do text normalization:
  1. Segmenting/tokenizing words in running text
  2. Normalizing word formats
  3. Segmenting sentences in running text

How many words?

- *I do uh main- mainly business data processing*
  - Fragments, filled pauses
- *Seuss's cat in the hat is different from other cats!*
  - **Lemma(词元)**: same stem (词干), part of speech (词性), rough word sense (词义)
    - cat and cats = same lemma
  - **Wordform(词形)**: the full inflected surface form
    - cat and cats = different wordforms

How many words?

they lay back on the San Francisco grass and looked at the stars and their

- **Type(词型)**: an element of the vocabulary.
- **Token(词例)**: an instance of that type in running text.
- How many?
  - 15 tokens (or 14)
  - 13 types (or 12)

How many words?

N = number of tokens

V = vocabulary = set of types

|*V*| is the size of the vocabulary

	Tokens = N	Types = |V|
Switchboard phone conversations	2.4 million	20 thousand
Shakespeare	884,000	31 thousand
Google N-grams	1 trillion	13 million

Church and Gale (1990): $|V| > O(N^{½})$

Simple Tokenization in UNIX

- (Inspired by Ken Church's UNIX for Poets.)

https://web.stanford.edu/class/cs124/kwc-unix-for-poets.pdf

- Given a text file, output the word tokens and their frequencies

```
tr -sc 'A-Za-z' '\n' < shakes.txt
     | sort
     | uniq -c
```

Change all non-alpha to newlines
Sort in alphabetical order
Merge and count each type

```
1945 A                    25 Aaron
  72 AARON                 6 Abate
  19 ABBESS                1 Abates
   5 ABBOT                 5 Abbess
 ... ...                   6 Abbey
                           3 Abbot
                         .... ...
```

The first step: tokenizing

```
tr -sc 'A-Za-z' '\n' < shakes.txt | head

THE
SONNETS
by
William
Shakespeare
From
fairest
creatures
We
...
```

The second step: sorting

```
tr -sc 'A-Za-z' '\n' < shakes.txt | sort | head

A
A
A
A
A
A
A
A
A
...
```

More counting

- Merging upper and lower case

```
tr 'A-Z' 'a-z' < shakes.txt | tr -sc 'A-Za-z' '\n' | sort | uniq -c
```

- Sorting the counts

```
tr 'A-Z' 'a-z' < shakes.txt | tr -sc 'A-Za-z' '\n' | sort | uniq -c | sort -n -r
```

```
  23243 the
  22225 i
  18618 and
  16339 to
  15687 of
  12780 a
  12163 you
  10839 my
  10005 in
   8954  d
```

What happened here?

Issues in Tokenization

- Finland's capital → Finland Finlands Finland's ?
- what're, I'm, isn't → What are, I am, is not
- Hewlett-Packard → Hewlett Packard ?
- state-of-the-art → state of the art ?
- Lowercase → lower-case lowercase lower case ?
- San Francisco → one token or two?
- m.p.h., PhD. → ??

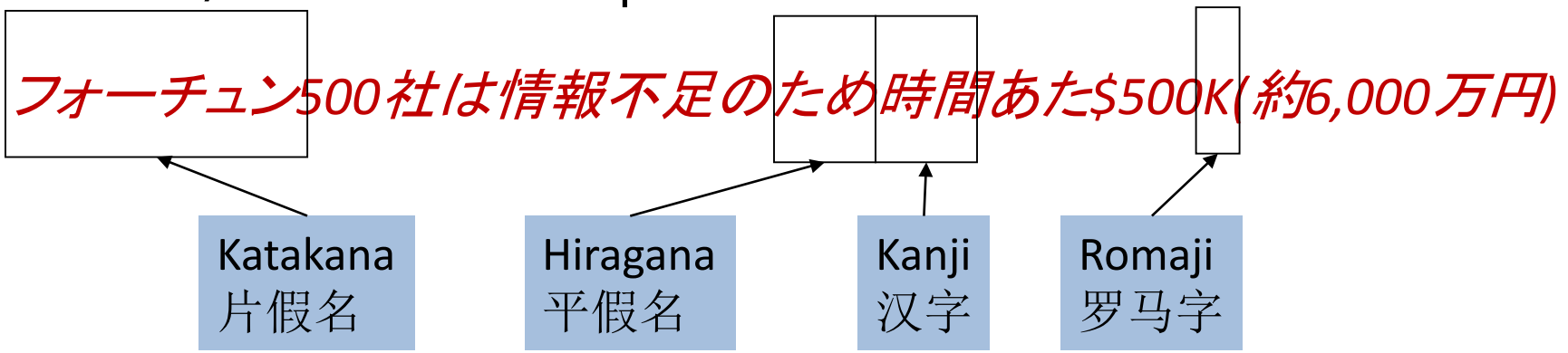
Tokenization: language issues

- French
  - *L'ensemble* → one token or two?
    - *L* ? *L'* ? *Le* ?
    - Want *l'ensemble* to match with *un ensemble*

- German noun compounds are not segmented
  - *Lebensversicherungsgesellschaftsangestellter*
  - 'life insurance company employee'
  - German information retrieval needs **compound (复合词) splitter**

Tokenization: language issues

- Chinese and Japanese no spaces between words:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
  - Sharapova now lives in US southeastern Florida
- Further complicated in Japanese, with multiple alphabets intermingled
  - Dates/amounts in multiple formats



Word Tokenization in Chinese

- Also called **Word Segmentation (分词)**
- Important for semantic representation
- An example

Word Tokenization in Chinese

- Chinese words are composed of characters
  - Characters are generally 1 syllable and 1 morpheme.
  - Average word is 2.4 characters long.

- Standard baseline segmentation algorithm:
  - Maximum Matching (also called Greedy)

Maximum Matching Word Segmentation Algorithm

- Given a wordlist of Chinese, and a string.

1) Start a pointer at the beginning of the string
2) Find the longest word in dictionary that matches the string starting at pointer
3) Move the pointer over the word in string
4) Go to 2

Max-match segmentation illustration

- Thecatinthehat — the cat in the hat
- Thetabledownthere — the table down there

  theta bled own there
- Doesn’ t generally work in English!

- But works astonishingly well in Chinese
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
- Modern probabilistic segmentation algorithms even better

Another option for text tokenization

Instead of

- white-space segmentation
- single-character segmentation

Use the data to tell us how to tokenize.

Subword tokenization (because tokens can be parts of words as well as whole words)

Subword tokenization

- Three common algorithms:
  - **Byte-Pair Encoding (BPE)** (Sennrich et al., 2016)
  - **Unigram language modeling tokenization** (Kudo, 2018)
  - **WordPiece** (Schuster and Nakajima, 2012)
- All have 2 parts:
  - A token **learner** that takes a raw training corpus and induces a vocabulary (a set of tokens).
  - A token **segmenter** that takes a raw test sentence and tokenizes it according to that vocabulary