Overview of Natural Language Processing
Part II & ACS L90

Lecture 2: Morphology

Weiwei Sun and Michael Schlichtkrull

Department of Computer Science and Technology
University of Cambridge

Michaelmas 2023/24

Some yinkish dripners blorked quastofically into the nindin with the pidibs

words have internal structures

...dripn+ER+S blork+ED quastofical+LY into the nindin with the pidib+S

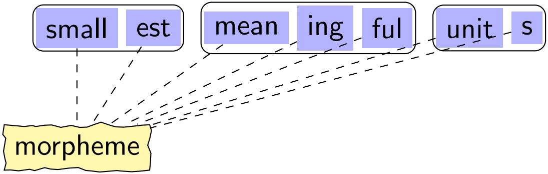
Lecture 2: Morphology

1. Morphology
2. Relevant NLP tasks
3. Finite state techniques
4. Byte-pair encoding

Morphology

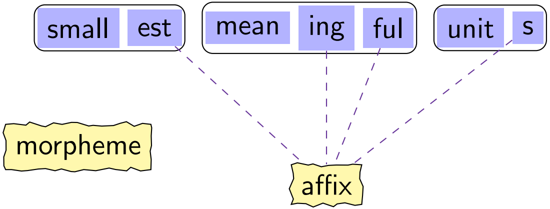
Morpheme

Morphemes are the *smallest meaningful units* of language. Words are composed of morpheme(s).



Morpheme

Morphemes are the *smallest meaningful units* of language. Words are composed of morpheme(s).

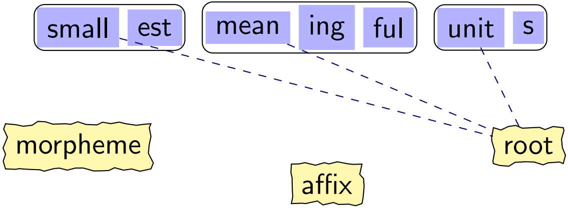


Affix: morpheme which only occurs in conjunction with other morphemes.

- suffix (unit*s*), prefix (*in*complete), infix, circumfix

Morpheme

Morphemes are the *smallest meaningful units* of language. Words are composed of morpheme(s).



Affix: morpheme which only occurs in conjunction with other morphemes.

- suffix (unit*s*), prefix (*in*complete), infix, circumfix

Root: nucleus of the word that affixes attach too.

Infix

Tagalog (Philippines)

infix: -um-

basa	b-um-asa	sulat	s-um-ulat
read	read.PAST	write	wrote

Infix

Tagalog (Philippines)

infix: -um-

basa	b-um-asa	sulat	s-um-ulat
read	read.PAST	write	wrote

Circumfix: occur on both sides

Dutch collectives

circumfix: ge-X-te

berg	ge-berg-te
mountain	mountains

*ge-berg
*berg-te

Source: J Hana & A Feldman. *ESSLLI 2013: Computational Morphology*.

http://ufal.mff.cuni.cz/~hana/teaching/2013-esslli/

Inflection and derivation

Inflection creates new forms of the same word

- e.g. *bring*, *brought*, *brings*, *bringing*
- generally fully productive (modulo irregular forms)
- tends to affect only its *syntactic function*

Derivation creates new words

- e.g. *logic*, *logical*, *illogical*, *illogicality*, *logician*, etc.
- generally semi-productive: e.g., *escapee*, *textee*, *?dropee*, *?snoree*, **cricketee* (* and ?)
- tends to be more irregular; the meaning is more idiosyncratic and less compositional.
- tends to affect the *meaning* of the word, and may change part-of-speech

Internal structure: ambiguity

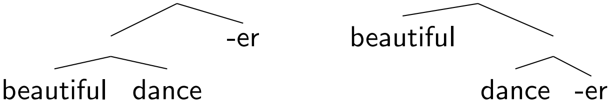
Structural ambiguity

unlockable

Capable of being unlocked. Not capable of being locked.

Can cross word boundaries

beautiful dancer



More about *beautiful dancer*: Larson (1998).

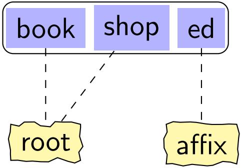
`semantics.uchicago.edu/kennedy/classes/f11/na/docs/larson08.pdf`

More about *unlockable*: `en.wiktionary.org/wiki/unlockable`

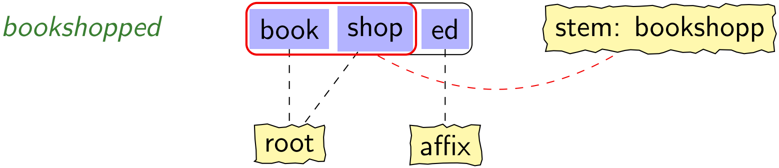
Stem: word without its inflectional affixes = roots + all derivational affixes.

Stem: word without its inflectional affixes = roots + all derivational affixes.

bookshopped

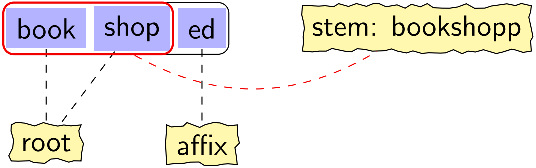


Stem: word without its inflectional affixes = roots + all derivational affixes.



Stem: word without its inflectional affixes = roots + all derivational affixes.

bookshopped



Lexeme: the set of all forms related by inflection (but not derivation).

{*bookshops*, *bookshopped*, *bookshopping*, ...}

Lemma: the *canonical/base/dictionary/citation* form of a lexeme chosen by convention.

bookshop (cf. the stem—*bookshopp*)

Phonaesthemе

slither, *slide*, *slip* etc have somewhat similar meanings; but *sl-* is not a morpheme.

Etymology: *slith*, *slid* and *slip* are historically related. See `www.etymonline.com/word/slide`

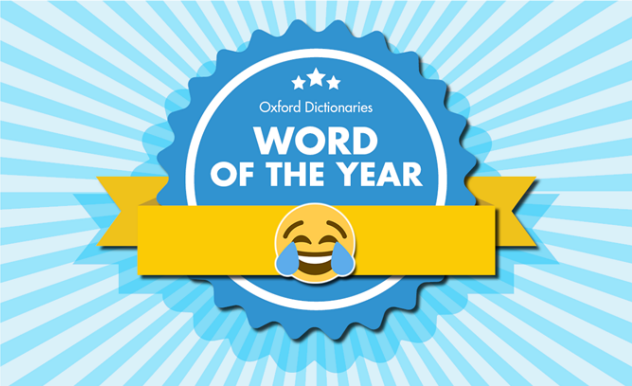
Phonaestheme

a pattern of sounds systematically paired with a certain meaning in a language

- *cl-*: related to a closing motion of a single object, such as *clam*, *clamp*, *clap*, *clasp*, *clench*, *cling*, *clip*, *clop*, *clutch*.
- *gl-*: related to light, as in *glance*, *glare*, *glass*, *gleam*, *glimmer*, *glint*, *glisten*, *glitter*, *gloaming*, *gloom*, *gloss*, *glow*.

Compound and multiword expression (1)

2015

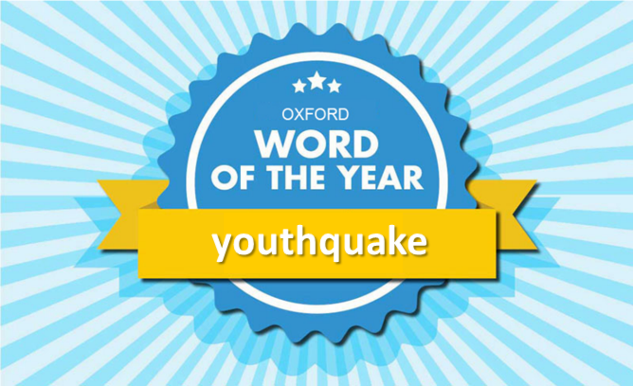


Compound and multiword expression (1)

2016

Compound and multiword expression (1)

2017



Compound and multiword expression (1)

2019

Compound and multiword expression (2)

Root: nucleus of the word that affixes attach too.

Compounds contain more than one root.

(1) a. youthquake
b. post-truth
c. railway
d. sunset

Multiword expression: combinations of two or more words that exhibit syntactic and semantic idiosyncratic behavior.

(2) a. climate emergency
b. computer science
c. random variable

Different types of multiword expressions

Fixed		**(Syntactically) flexible**
by and large		*put on the clothes* *put the clothes on*
Non-compositional	**Semi-compositional**	**Compositional**
kick the bucket	*spill the beans* (reveal the secret)	*strong tea*

Multiword expression and grammatical errors

(3) a. *At this moment* Carole was living with her husband but they didn't love each other any more.
   → At the moment

b. It is a *dream becames true* and was really unexpected for me!
   → dream come true

c. They go together in groups, then they prepare power point presentations and *at least* they present it in front of the other pupils and teachers.
   → finally

d. *By the other side*, I have never climbed a mountain but I always wanted to do it.
   → On the other hand

e. I tried to *take it on my stride* but I couldn't.
   → take it in my stride

f. However, I told my teacher that I am willing to *give a hand* next time.
   → lend a hand

from Shiva Taslimipoor

Code-mixed languages

Code-switching

a speaker alternates between two or more languages in the context of a single conversation or situation.

Cantonese-English (widely used in Hong Kong)

The English word "sure"/"cute" is mixed into an otherwise Cantonese sentence.

- 我唔sure
- cu唔cute啊

Text normalization

- Not using any punctuation at all
  Eh speak english mi malay not tt good (Eh, speak English! My Ma-lay is not that good.)
- Using spell-ing/punctuation for emphasis
  goooooood Sunday morning !!!!!! (Good Sunday morning!)
- Using phonetic spelling
  dat iz enuf (That is enough)
- Dropping vowel
  i hv cm to c my luv. (I have come to see my love.)
- Introducing local flavor
  yar lor where u go juz now (yes, where did you go just now?)
- Dropping verb
  I hv 2 go. Dinner w parents. (I have to go. Have dinner with parents.)

Examples are from Aw et al. (2005). `https://www.aclweb.org/anthology/P06-2005.pdf`

More: `noisy-text.github.io/norm-shared-task.html`

Relevant NLP Tasks

Form transformation

$$\text{natural language expressions} \underset{\text{production}}{\overset{\text{comprehension}}{\rightleftarrows}} \mathcal{R} \left\{ \begin{array}{l} \text{morphological structure} \\ \text{syntactic structure} \\ \text{semantic structure} \\ \text{discourse structure} \\ \text{application-related structure} \end{array} \right.$$

Computational tasks

		natural language expression	representation $\mathcal{R}$
LEMMATIZATION	→	word *saw*	lexeme {*see*, *saw*}
TAGGING	→	contextualized word *saw @ J saw M*	contextualized tag ⟨*see*, VERB.PAST⟩
SEGMENTATION	→	word *meaningful*	morphemes (subwords) *mean+ing+ful*
GENERATION	←	word *saw*	*abstract word* ⟨*see*, VERB.PAST⟩

Segmentation

antidisestablishmentarianism $\Rightarrow$ anti- dis- e- stabl -ish -ment -arian -ism

antidisestablishmentarianism

anti	dis	establish	ment	arian	ism

en.wikipedia.org/wiki/Antidisestablishmentarianism

www.etymonline.com/word/antidisestablishmentarianism

important for some application, e.g. bioinformatics

Word segmentation

Goal

- The written systems for some languages, e.g. Japanese and Chinese contain no word delimiters such as spaces.
- There is a need to develop algorithms that are able to automatically divide a string into its component words.

Example

解放大道路面积水问题

解放 / 大道 / 路面 / 积水 / 问题

解 / 放大/ 道路 / 面积 / 水 / 问题

Finite State Techniques

Language Is An Inherently Temporal Phenomenon

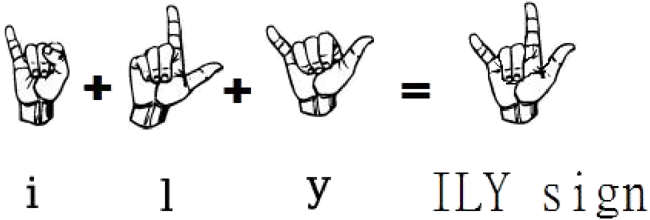
Orders matter!

- talk-ed $\neq$ *ed-talk
- re-write $\neq$ *write-re
- un-kind-ly $\neq$ *kind-un-ly

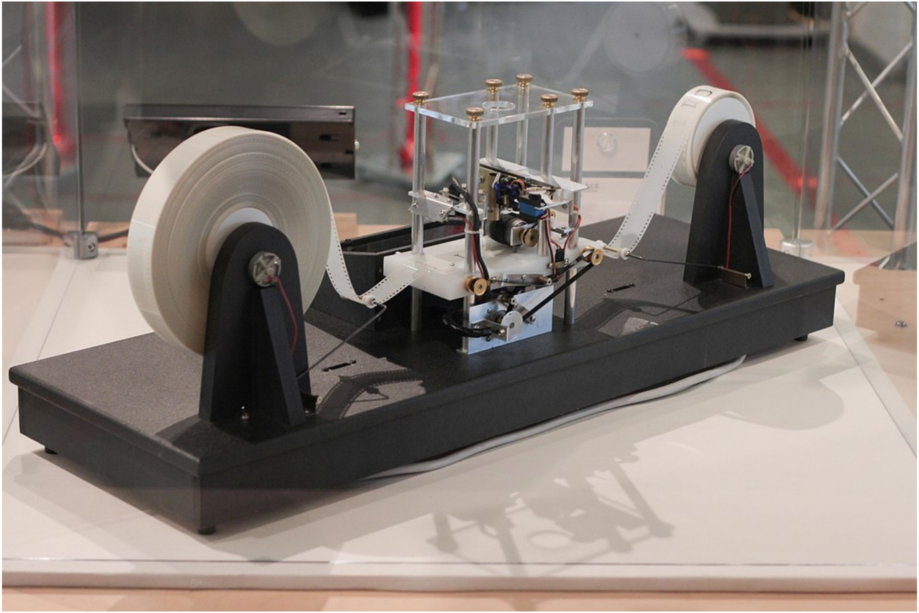
Language Is An Inherently Temporal Phenomenon

Orders matter!

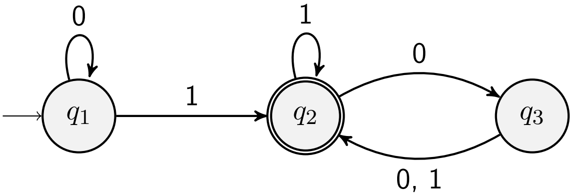
- talk-ed $\neq$ *ed-talk
- re-write $\neq$ *write-re
- un-kind-ly $\neq$ *kind-un-ly



Turing machine

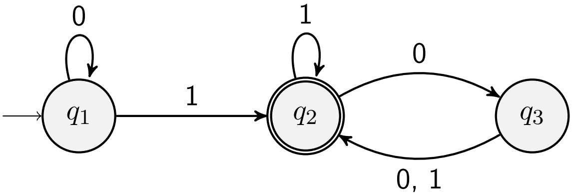


Finite-state automata



- Circles are states of the automaton.
- Arrows are called transitions.
- The automaton changes states by following transitions.
- The double circle indicates that this state is an accepting state. The automaton accepts the string if it ends in an accepting state.

Finite-state automata



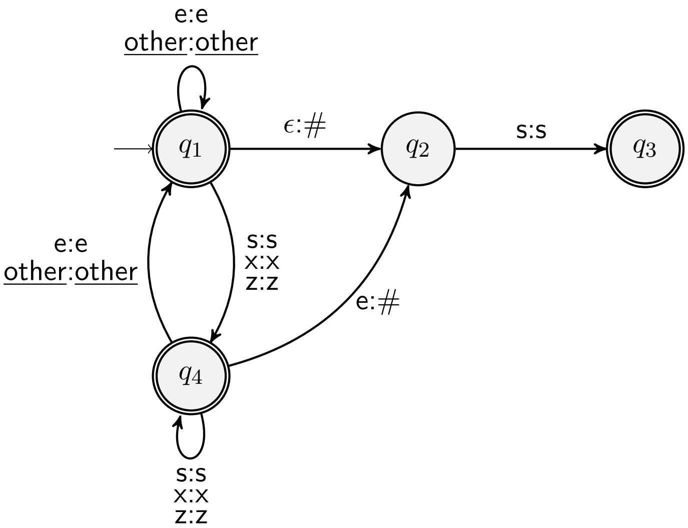
- Circles are states of the automaton.
- Arrows are called transitions.
- The automaton changes states by following transitions.
- The double circle indicates that this state is an accepting state. The automaton accepts the string if it ends in an accepting state.

- **Form Transformation**: agumenting transitions

  `input`$\rightarrow$`input:output`

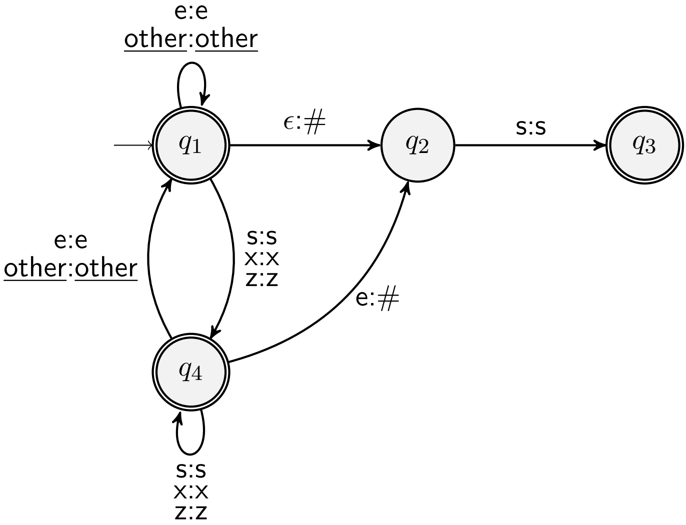
Finite state transducer

- *cakes* → *cake#s*
- *boxes* → *box#s*



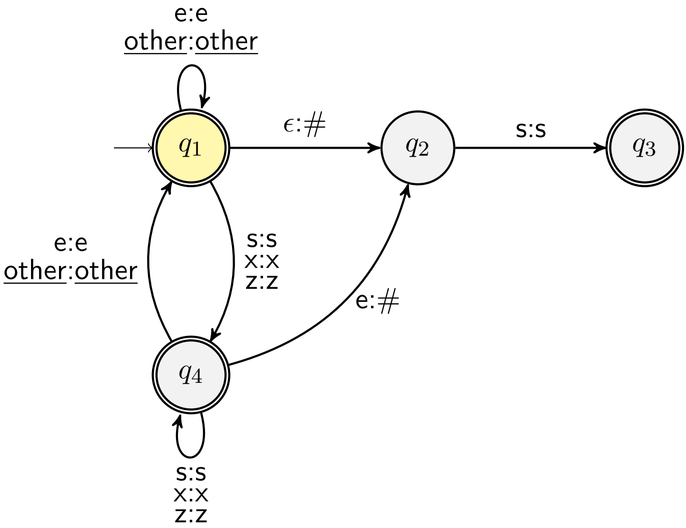
Analysing *boxes*

OUTPUT					
INPUT	b	o	x	e	s



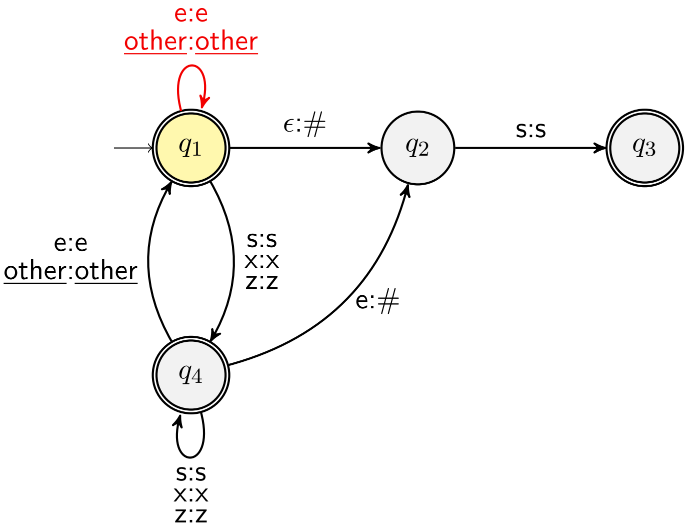
Analysing *boxes*

OUTPUT					
INPUT	b	o	x	e	s



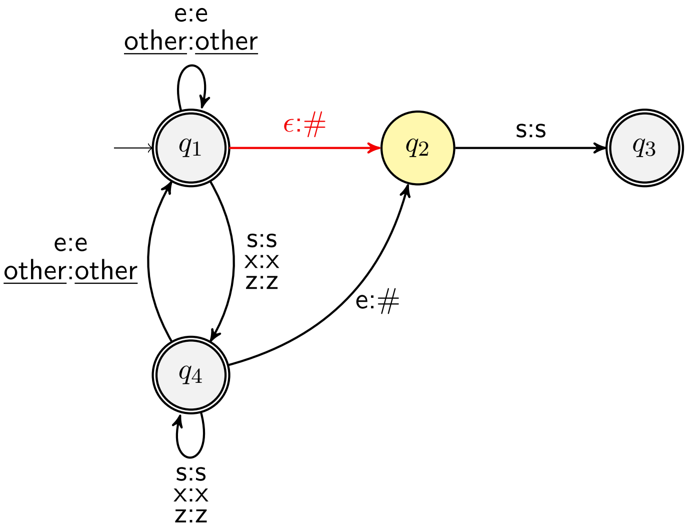
Analysing *boxes*

OUTPUT	b				
INPUT	b	o	x	e	s



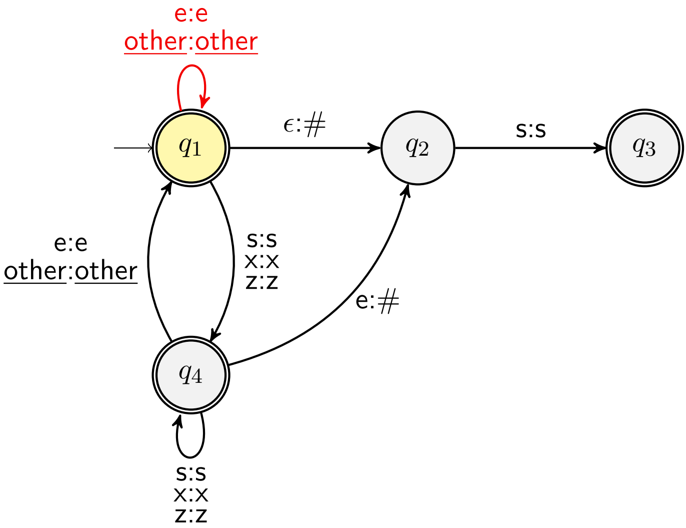
Analysing *boxes*

OUTPUT	b				
INPUT	b	o	x	e	s



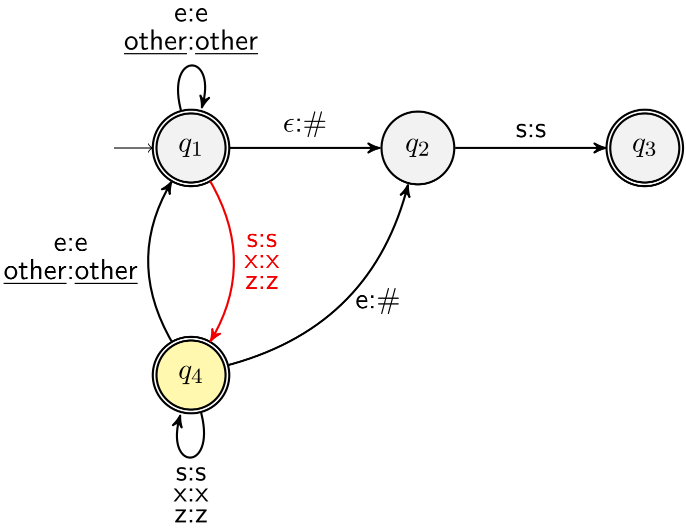
Analysing *boxes*

OUTPUT	b	o			
INPUT	b	o	x	e	s



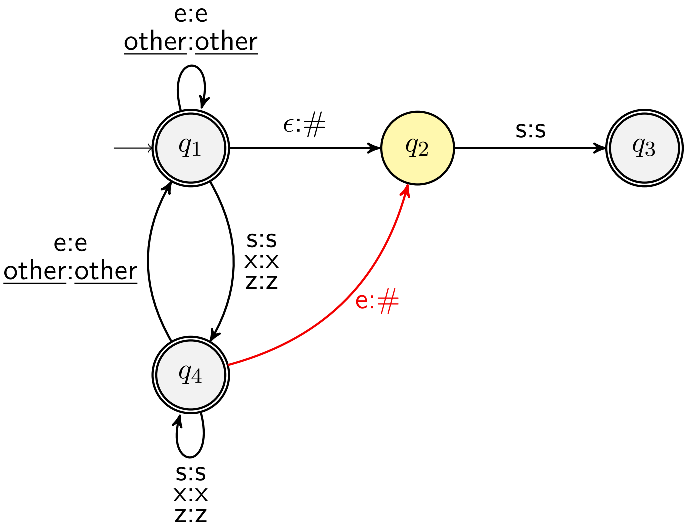
Analysing *boxes*

OUTPUT	b	o	x		
INPUT	b	o	x	e	s



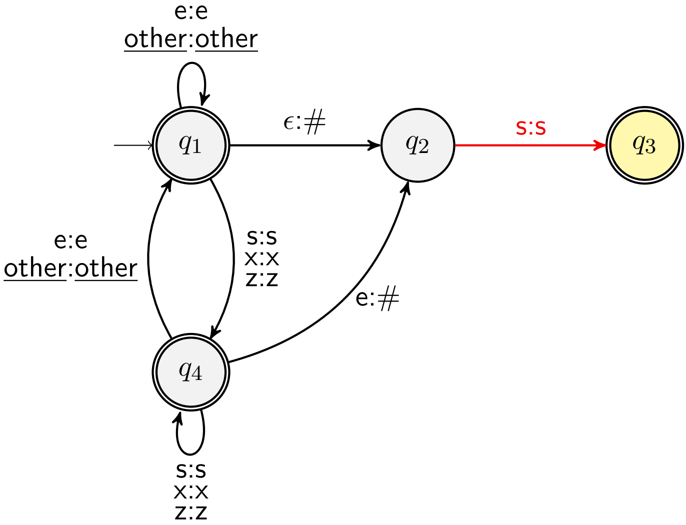
Analysing *boxes*

OUTPUT	b	o	x	#	
INPUT	b	o	x	e	s

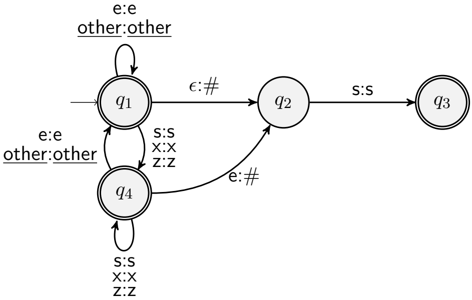


Analysing *boxes*

OUTPUT	b	o	x	#	s
INPUT	b	o	x	e	s

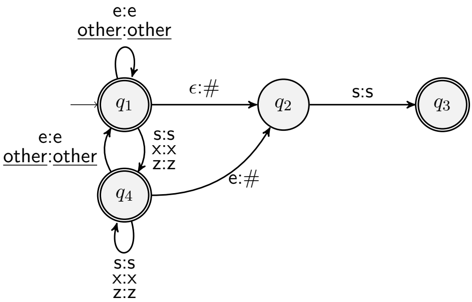


Finite-state machine



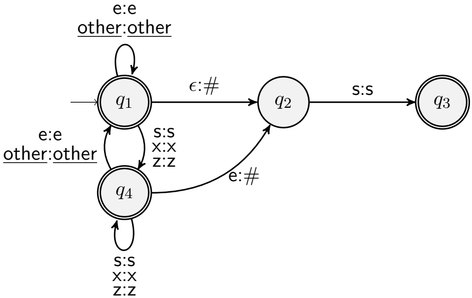
- A symbolic system that can recognize or transform forms.

Finite-state machine



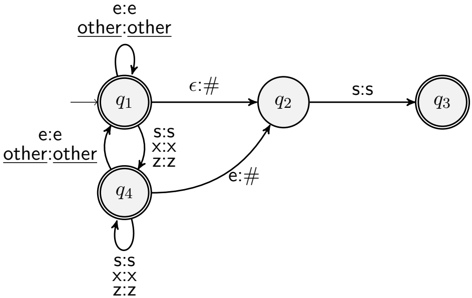
- A symbolic system that can recognize or transform forms.
- An automaton remembers only a finite amount of information.

Finite-state machine



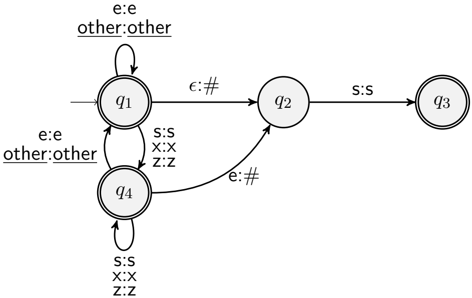
- A symbolic system that can recognize or transform forms.
- An automaton remembers only a finite amount of information.
- Information is represented by its states.

Finite-state machine



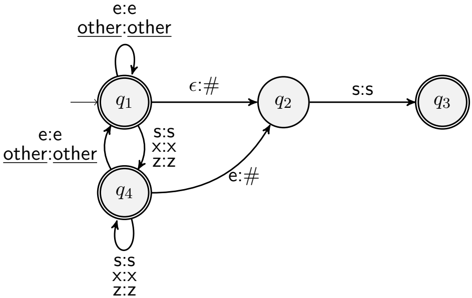
- A symbolic system that can recognize or transform forms.
- An automaton remembers only a finite amount of information.
- Information is represented by its states.
- State changes in response to inputs and may trigger outputs.

Finite-state machine



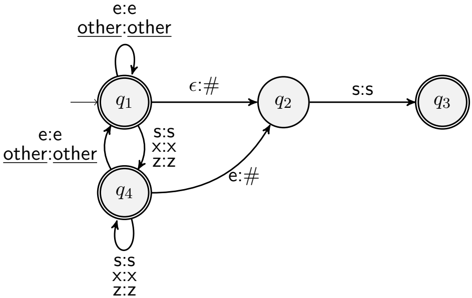
- A symbolic system that can recognize or transform forms.
- An automaton remembers only a finite amount of information.
- Information is represented by its states.
- State changes in response to inputs and may trigger outputs.
- Transition rules define how the state changes in response to inputs.

Finite-state machine



- A symbolic system that can recognize or transform forms.
- An automaton remembers only a finite amount of information.
- Information is represented by its states.
- State changes in response to inputs and may trigger outputs.
- Transition rules define how the state changes in response to inputs.
- Given a sequence of input symbols, a recognition process starts in the start state and follow the transitions in turn. Input is accepted if this process ends up in an accepting state.

Finite-state machine



- A symbolic system that can recognize or transform forms.
- An automaton remembers only a finite amount of information.
- Information is represented by its states.
- State changes in response to inputs and may trigger outputs.
- Transition rules define how the state changes in response to inputs.
- Given a sequence of input symbols, a recognition process starts in the start state and follow the transitions in turn. Input is accepted if this process ends up in an accepting state.
- Partial grammars for text preprocessing, tokenization, named entity recognition etc.

Cross-lingual variants

- English morphology is essentially concatenative
  cf. duplication in Chinese, e.g.

高兴	$\rightarrow$	高高兴兴
happy		高兴高兴

Cross-lingual variants

- English morphology is essentially concatenative
  cf. duplication in Chinese, e.g.

高兴	→	高高兴兴	
happy		高兴高兴	

- The phones making up a morpheme don't have to be contiguous, e.g. in Hebrew

Root	Pattern	PoS	Phonological Form	Gloss
ktb	**C**a**C**a**C**	v	katav	'wrote'
ktb	hi**CC**i**C**	v	hixtiv	'dictated'
ktb	mi**CC**a**C**	n	mixtav	'a letter'
ktb	**CC**a**C**	n	ktav	'writing, alphabet'

from E. Bender's tutorial (`faculty.washington.edu/ebender/papers/100things.pdf`)

Byte-Pair Encoding

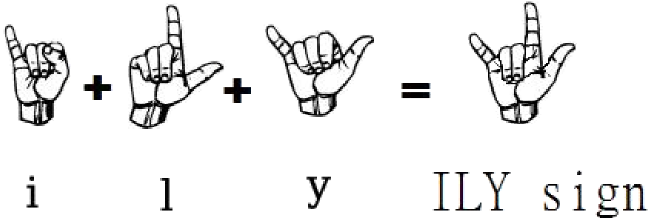
Form transformation

$$\text{natural language expressions} \; \underset{\text{production}}{\overset{\text{comprehension}}{\rightleftarrows}} \; \mathcal{R} \begin{cases} \text{morphological structure} \\ \text{syntactic structure} \\ \text{semantic structure} \\ \text{discourse structure} \\ \text{application-related structure} \end{cases}$$

Are there some magical algorithms that are able to automatically induce useful representations from data?

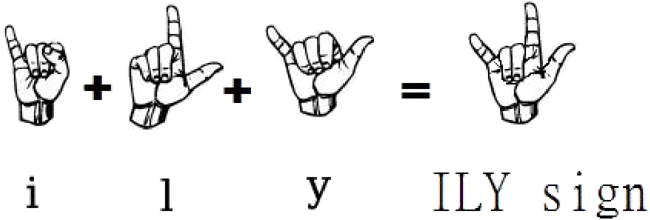
Subword tokenisation

Isn't it just one symbol?



Subword tokenisation

Isn't it just one symbol?



Phonaesthemе: It is difficult to hard-code the knowledge

- *cl-*: related to a closing motion of a single object, such as *clam*, *clamp*, *clap*, *clasp*, *clench*, *cling*, *clip*, *clop*, *clutch*.

Byte-Pair Encoding (BPE)

BPE was initially developed as an algorithm to compress texts, and then used by OpenAI for tokenization when pretraining the GPT model.

- Start from a small base vocabulary, e.g. 256 ASCII code.
- Add new tokens to the vocabulary until the desired vocabulary size is reached by learning merges,
  which are rules to merge two elements of the existing vocabulary together into a new one.
- At each step, the BPE algorithm search for the most frequent pair, namely two consecutive tokens, of existing tokens.

from `https://huggingface.co/learn/nlp-course/chapter6/5?fw=pt`

Example

```
("hug", 10), ("pug", 5), ("pun", 12), ("bun", 4), ("hugs",
5)
```

on whiteboard

Readings

Required

- Ann's lecture notes.
  `https://www.cl.cam.ac.uk/teaching/1920/NLP/materials.html`
- E. Bender. 100 Things You Always Wanted to Know about Linguistics But Were Afraid to Ask. NAACL-HLT 2012 tutorial. `faculty.washington.edu/ebender/papers/100things.pdf` Please read Numbers #7–#27.

Optional

* J. Hana & A. Feldman. Computational Morphology. ESSLLI 2013 course. `ufal.mff.cuni.cz/~hana/teaching/2013-esslli/`
* M. Mohri. Finite-State Transducers in Language and Speech Processing. CL 1997 paper. `www.aclweb.org/anthology/J97-2003/`