Natural Language Processing

Lecture 3
Introduction to NLP
Methods of Morphological Analysis

Languages:

- Natural languages: *English*, *Chinese*, *Russian* etc.;
- Formal languages: *programming languages* etc.;
- Artificial languages: *Esperanto*, *Elvish languages* etc.

Natural Language Processing

Natural-language processing (**NLP**) is a field of computer science, artificial intelligence concerned with the interactions between computers and human (natural) languages.

- ***1950*** - Turing test
- ***1954*** - Georgetown experiment (Machine Translation)
- ***1970s*** - conceptual ontologies
- ***1980s*** - ***1990s*** - statistical revolution
- ***Currently*** - Deep Learning algorithms

Computer Linguistics Tasks:

1. Information Retrieval: *Google*, *Yahoo!*;
2. Information Extraction: *RCO Fact Extractor*;
3. Machine Translation: *PROMT*, *Google Translate*;
4. Automatic Text Summarization: *TextAnalyst*, *Extractor*, *Text Miner*;
5. Corpus Linguistics: *RusCorpora*, *OpenCorpora*;
6. Expert Systems: *IBM Watson*, *Wolfram Alfa*;
7. Question Answering Systems: *IBM Watson*, *Siri*;
8. Electronic dictionaries, thesaurus, onthology creation;
9. Optical Character Recognition: *Fine Reader*;
10. Automatic Speech Recognition: *plug-in in Google Chrome*;
11. Text-To-Speech: *Google Translate*

Stages to build NLP system:

1. Analysis of graphemes (character level);
2. Morphological analysis (word level);
3. Fragmentational analysis (phrase level);
4. Syntax analysis (sentence level);
5. Semantic analysis (text level).

Discourse analysis - ?

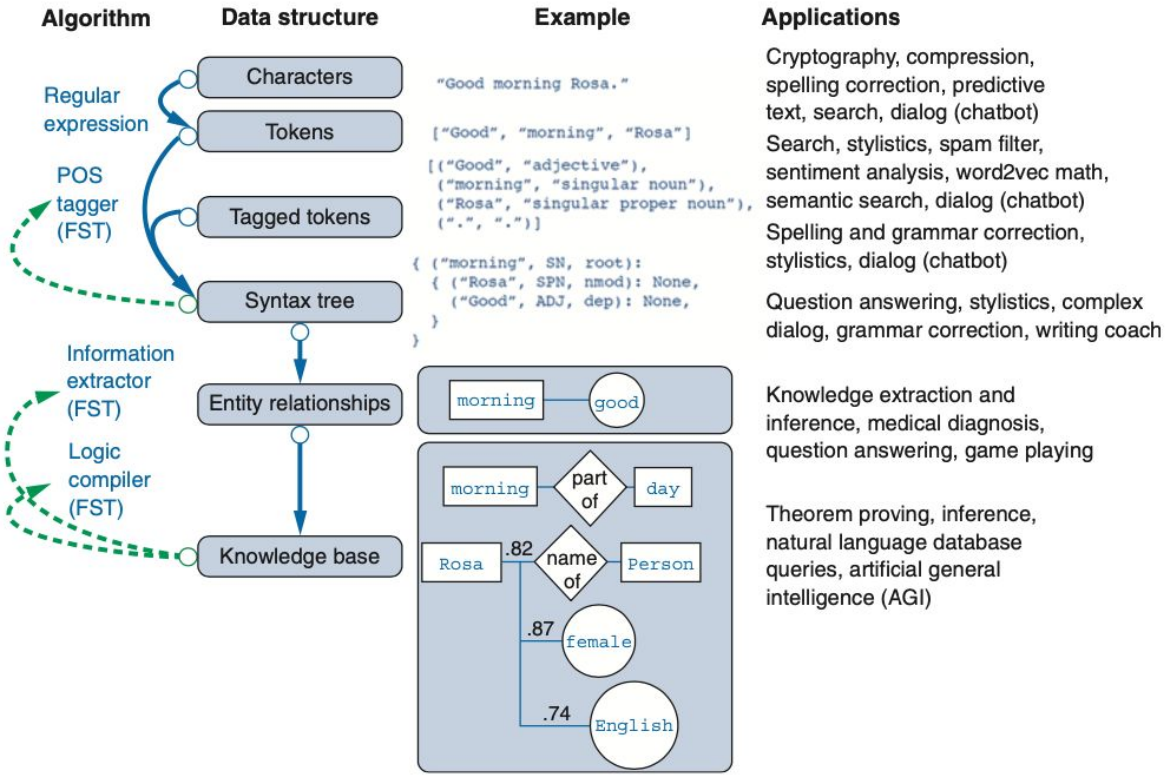


Figure 1.4 Example layers for an NLP pipeline

Analysis of graphemes: Tokenization

Tokenization is words, digits, punctuation marks, formula etc. extraction from the text.

Tokens are elements extracted from the text.

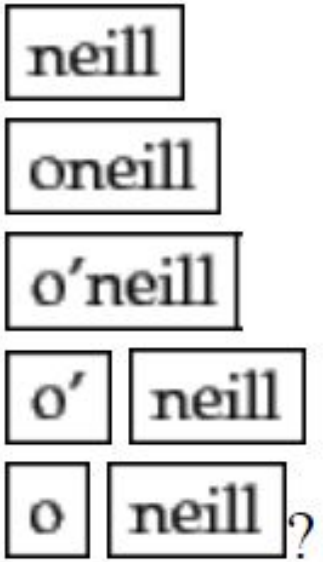
Input: “`Friends, Romans, Countrymen, lend me your ears`”

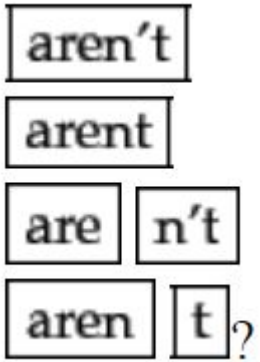
Output: `[“Friends”, “,”, “Romans”, “,” “Countrymen”, “,”, “lend”, “me”, “your”, “ears”]`

Ideas?

Tokenization: tricky cases

Mr. O'Neill thinks that the boys' stories about Chile's capital aren't amusing.





Tokenization: tricky cases

1. **Programming Languages**: C++, C#;
2. **Aircraft names**: B-52;
3. **Email addresses**: jblack@mail.yahoo.com;
4. **Web URLs**: http://stuff.big.com/new/specials.html;
5. **Numeric IP addresses**: 142.32.48.231;
6. **Package tracking numbers**: (1Z9999W99845399981)
7. and more...

Tokenization: hyphenation

Example 1: `co-education`

Example 2: `Hewlett-Packard`

Example 3: `the hold-him-back-and-drag-him-away maneuver`

Tokenization: other languages

- French: *l'ensemble*, *donne-moi* ‘give me’;
- German:
  - *Computerlinguistik* `computational linguistics';
  - *Lebensversicherungsgesellschaftsangestellter* `life insurance company employee'
- East Asian Languages (e.g., Chinese, Japanese, Korean, and Thai)

电脑坏了。

The computer is broken.

Tokenization: BPE

Byte pair encoding is a simple form of data compression in which the most common pair of consecutive bytes of data is replaced with a byte that does not occur within that data.

Tokenization: BPE

Suppose the data to be encoded is

```
aaabdaaabac
```

The byte pair "aa" occurs most often, so it will be replaced by a byte that is not used in the data, "Z". Now there is the following data and replacement table:

```
ZabdZabac
Z=aa
```

Then the process is repeated with byte pair "ab", replacing it with Y:

```
ZYdZYac
Y=ab
Z=aa
```

The only literal byte pair left occurs only once, and the encoding might stop here. Or the process could continue with recursive byte pair encoding, replacing "ZY" with "X":

```
XdXac
X=ZY
Y=ab
Z=aa
```

This data cannot be compressed further by byte pair encoding because there are no pairs of bytes that occur more than once.

To decompress the data, simply perform the replacements in the reverse order.

Analysis of graphemes: Segmentation

Segmentation is the retrieval of words boundaries in the text without spaces (e.g. Chinese or Japanese texts).

Example: *Itiseasytoreadtextwithoutspaces → It is easy to read text without spaces*

Sentence segmentation

S1: `Two high-ranking positions were filled Friday by Penn St. University President Graham Spanier.`

S2: `Two high-ranking positions were filled Friday at Penn St. University President Graham Spanier announced the appointments.`

Ideas?

Sentence segmentation

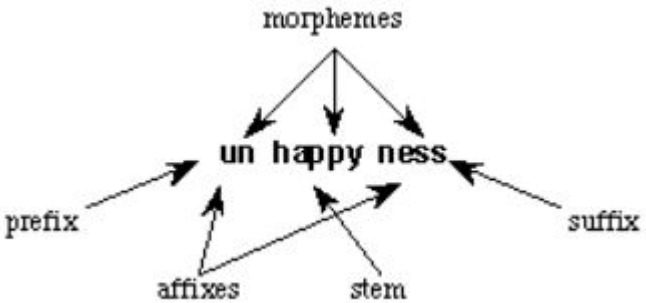
- **Case distinctions** — In languages and corpora where both uppercase and lowercase letters are consistently used, whether a word is capitalized provides information about sentence boundaries;
- **Part of speech**;
- **Word length** — the length of the words before and after a period;
- **Lexical endings** — using morphological analysis to recognize suffixes and thereby filter out words which were not likely to be abbreviations;
- **Prefixes and suffixes** — using both prefixes and suffixes of the words surrounding the punctuation mark as one contextual feature;
- **Internal punctuation** — using the presence of periods within a token as a feature;
- **Proper nouns** — using the presence of a proper noun to the right of a period as a feature.

Morphological Analysis

Morphology

Morphology is the study of the structure and formation of words.

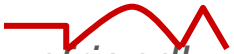
Its most important unit is the **morpheme**, which is defined as the "minimal unit of meaning".



Free morpheme can appear on its own
Bound morphemes have to be attached to a free morpheme

Morpheme is a minimal meaningful unit of a word.

Root is a morpheme with lexical meaning of a word.



Affix is a morpheme which modifies the lexical meaning of a word (e.g. prefix, suffix).

Allomorph is some complementary **morphs** (the phonetic realization of morpheme), which manifest a morpheme in its different morphological or phonological environments.

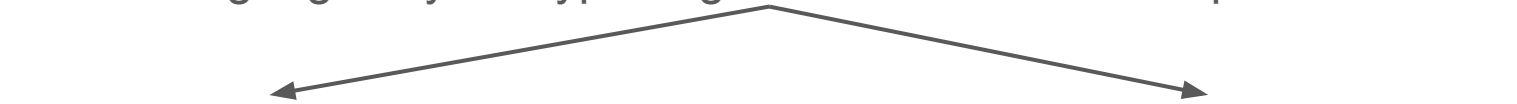
Lexemes: *illegal*, *impatient*, *irregular*, *inconsiderate*

Allomorphs: *il*, *im*, *ir*, *in* Morpheme: *in*

Paradigm is a list of all word forms.

Paradigm for verb *to be*: *am*, *is*, *are*, *was*, *were*, *will*

Languages by the type of grammatical features expression



Index of synthesis = M / W,
M - number of morphs in text;
W - number of words in text.

For analytical languages index < 2.0 (e.g. for English 1.68)

For synthetic languages index 2.0 - 3.0 (e.g. for Russian 2.33 - 2.45)

Languages by the type of morphological structure

- **Isolating** languages: isolated morphemes as a word (e.g. *Chinese*);
- **Agglutinative** languages: a lot of affixes in word, each affix has its own meaning (e.g. *Turkish*);
- **Inflectional** languages: affixes are homonymous (e.g. *Russian*)

Isolating languages (e.g. Mandarin Chinese)

Transliterated sentence: *gou bú ài chi qingcài*

may be literally translated as: *dog not like eat vegetable*

Depending on the context, it can mean any of the four following sentences:

- *the dog did not like to eat vegetables*
- *the dogs do not like to eat vegetables*
- *the dogs did not like to eat vegetables*
- *dogs do not like to eat vegetables*

Agglutinative languages (e.g. Turkish)

- *ler* = plural
- *i* = possessive (e.g. *his*, *her*, *its*)
- *den* = ablative (e.g. a grammatical "case" ending showing a source, e.g. *from a house*)
  - ev: house
  - evler: houses
  - evi: his/her house
  - evleri: his/her houses, their houses
  - evden: from the house
  - evlerden: from the houses
  - evinden: from his/her house
  - evlerinden: from his/her houses, from their houses

Inflecting languages (e.g. Latin)

amo = *I love*

Ending *o* is used to express the meanings:

- first person ("*I*" or "*we*"),
- singular,
- present tense,
- and also other meanings.

Stemming is the process of reducing inflected words to their word stem or root (the stem need not be identical to morphological root of the word): *'stems'*, *'stemmer'*, *'stemming'*, *'stemmed'* → *'stem'*.

Lemmatization is the process of getting the base form of the word: *'tables'* → *'table'*, *'written'* → *'write'*.

Tagging a wordform with its grammems: *'table'*: [Noun, sing]; *'book'*: [Noun, sing], [Verb, 1/2 person, sing/plur, Pr.Simple] / [Verb, 3 person, plur, Pr.Simple].

Paradigm derivation is the process of derivation all word forms from the base form.

Morphological analyzers

- **Dictionary-based**: using a table (a dictionary), which contains mapping from set of words on set of lemmas. For Russian Zaliznyak's dictionary is used. Downside: it is impossible to get information for word if the dictionary does not contain it.
- **Analytical**: using a set of rules for morphological transformations. Don't cope with all morphological tasks, but good for stemming, lemmatization and getting paradigm.

Lovins' algorithm (Lovins, 1968)

- 294 endings are defined;
- 29 conditions for removing one of the endings;
- 35 rules of wordform transformation after the ending removing

Example: '*nationally*' → '*nat*'. Two endings can be removed: '*ationally*' and '*ionally*'. But the first can't be removed because of the restriction: stem should be longer than 3 characters.

Downside: the algorithm requires linguists for rules and exceptions creating.

Porter’s algorithm (Porter, 1980)

Rule: <*condition*>, <*ending*> → <*new ending*>

Contains ~ 60 rules, each of them is applied to the input wordform.

Example:

(*m* > 0) *eed* → *ee* `agreed → agree`

Algorithm of Paice&Husk (Paice/Husk, 1990)

Table of rules for ending transformations (removing or replacement).

Rule contains:

- inverted ending;
- integrity mark “*” (optional);
- length of the removing ending (including 0);
- string with length > 1, which has to be added (optional);
- symbols ‘>’ (switching to the pointed entry) or ‘.’ (stopping).

Example: “*noi s4j>*”

Comparison

Original sentence	*Such an analysis can reveal features that are not easily visible from the variations in the individual genes.*
Lovins' algorithm	*Such an analysis can reve featur that ar not eas vis from th vari in th individu gen*
Porter's algorithm	*Such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene.*
Algorithm of Paice&Husk	*Such an analysis can rev feat that are not easy vis from the vary in the individ gen*

Metrics for algorithm performance

- Under-stemming index;
- Over-stemming index;
- Words number belonging to the same paradigm;
- Coefficient of the index compression;
- Modified Hamming distance

Under-stemming index - % analyzed wordforms, for which the same stem has to be produced, but different stems were produced.

Over-stemming index - % analyzed wordforms, for which different stems have to produced, but the same stem was produced.

Both metrics require annotated text corpora

Words number belonging to the same paradigm

$$\mathbf{MVC} = \mathbf{N} / \mathbf{S},$$

N - number of distinctive word usage before the algorithm performance,

S - number of distinctive stems produced by the algorithm

Coefficient of the index compression

Shows the difference between distinctive stems collection and distinctive wordforms collection:

$$\mathbf{ICF} = (\mathbf{N} - \mathbf{S}) / \mathbf{N},$$

N - number of distinctive word usage before the algorithm performance,

S - number of distinctive stems produced by the algorithm

Modified Hamming distance

Hamming distance is the number of positions where two strings (sequences) with equal length are different.

For strings with different lengths modified Hamming distance is used:

$$\mathbf{MHD = HD\ (1,\ P) + (Q - P)},$$

HD (1, P) - Hamming distance for the first **P** characters of the analyzed strings (**P** < **Q**).

Example:

Algorithm output: '*parties*' → '*party*'

MHD (*party*, *parties*) = 1 + 2 = 3, **HD** (1, P) = 1, **P** = 5

Part-of-Speech Tagging

Part-of-speech tagging is the process of assigning a part-of-speech marker to each word in an input text.

Tagging is a disambiguation task; words are ambiguous — have more than one possible part-of-speech — and the goal is to find the correct tag for the situation.

`book` can be a verb (`book that flight`) or a noun (`hand me that book`).

`that` can be a determiner (`Does that flight serve dinner`) or a complementizer (`I thought that your flight was earlier`).

The Penn Treebank Part-of-Speech Tagset

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coordinating conjunction	*and, but, or*	PDT	predeterminer	*all, both*	VBP	verb non-3sg present	*eat*
CD	cardinal number	*one, two*	POS	possessive ending	*'s*	VBZ	verb 3sg pres	*eats*
DT	determiner	*a, the*	PRP	personal pronoun	*I, you, he*	WDT	wh-determ.	*which, that*
EX	existential 'there'	*there*	PRP$	possess. pronoun	*your, one's*	WP	wh-pronoun	*what, who*
FW	foreign word	*mea culpa*	RB	adverb	*quickly*	WP$	wh-possess.	*whose*
IN	preposition/ subordin-conj	*of, in, by*	RBR	comparative adverb	*faster*	WRB	wh-adverb	*how, where*
JJ	adjective	*yellow*	RBS	superlatv. adverb	*fastest*	$	dollar sign	*$*
JJR	comparative adj	*bigger*	RP	particle	*up, off*	#	pound sign	*#*
JJS	superlative adj	*wildest*	SYM	symbol	+,%, &	"	left quote	' or "
LS	list item marker	*1, 2, One*	TO	"to"	*to*	"	right quote	' or "
MD	modal	*can, should*	UH	interjection	*ah, oops*	(	left paren	[, (, {, <
NN	sing or mass noun	*llama*	VB	verb base form	*eat*	)	right paren	],), }, >
NNS	noun, plural	*llamas*	VBD	verb past tense	*ate*	,	comma	,
NNP	proper noun, sing.	*IBM*	VBG	verb gerund	*eating*	.	sent-end punc	. ! ?
NNPS	proper noun, plu.	*Carolinas*	VBN	verb past part.	*eaten*	:	sent-mid punc	: ; ... – -

Figure 8.1 Penn Treebank part-of-speech tags (including punctuation).

Национальный Корпус Русского Языка

1. А. Б. Барский. Применение SPMD-технологии при построении сетевых баз данных с циркулирующей информацией // «Информационные технологии», 2004 [омонимия снята] Все примеры

Например, студент затребовал учебное пособие, с которым проработает несколько часов. [А. Б. Барский. Применение SPMD-технологии при построении сетевых баз данных с циркулирующей информацией // «Информационные

студ`ент	
Лемма	студент
Грамматика	S,m,anim,sg,nom
Семантика основная	t:hum r:concr
Доп. признаки	acomma, amark, genderred, numred

Сообщить об ошибке...

2. Информационные тех… …нологии», 2004 [омонимия снята] Все примеры

Кроме того, на … …цией по интересующим вопросам. [Информационные технологии в образовательной среде вуза // «Информационные технологии», 2004]

3. Классика на школьно… …имеры

Чичикова играл 19-летний студент Виталий Тулмач — высокий, изящный блондин с тонкими чертами лица (ничего общего с лысеющим и "добреющим" гоголевским персонажем), этакий лощёный сегодняшний мальчик, стоящий у магазина "Мода" или руководящий компьютерной группой. [Классика на школьной сцене // «Народное творчество», 2004] [омонимия снята] ←…→

4. Наши дети: Подростки (2004) [омонимия снята] Все примеры

Сеичас в 11-ом — «гуд студент», везде в пример ставят. [Наши дети: Подростки (2004)] [омонимия снята] ←…→

5. Спасительная эстафета игры // «Экран и сцена», 2004.05.06 [омонимия снята] Все примеры

Романтический студент, мечтающий о настоящей любви, в которой отказано ему в "земном" уделе, и скучный муж-каббалист равно скрывают его истинный облик. [Спасительная эстафета игры // «Экран и сцена», 2004.05.06] [омонимия снята] ←…→

POS-tags in Russian

Части речи

S — существительное (*яблоня, лошадь, корпус, вечность*)
A — прилагательное (*коричневый, таинственный, морской*)
NUM — числительное (*четыре, десять, много*)
ANUM — числительное-прилагательное (*один, седьмой, восьмидесятый*)
V — глагол (*пользоваться, обрабатывать*)
ADV — наречие (*сгоряча, очень*)
PRAEDIC — предикатив (*жаль, хорошо, пора*)
PARENTH — вводное слово (*кстати, по-моему*)
SPRO — местоимение-существительное (*она, что*)
APRO — местоимение-прилагательное (*который, твой*)
ADVPRO — местоименное наречие (*где, вот*)
PRAEDICPRO — местоимение-предикатив (*некого, нечего*)
PR — предлог (*под, напротив*)
CONJ — союз (*и, чтобы*)
PART — частица (*бы, же, пусть*)
INTJ — междометие (*увы, батюшки*)

POS-tags in RusCorpora (НКРЯ)

```
PARTS_OF_SPEECH = frozenset([
    'NOUN',   # имя существительное
    'ADJF',   # имя прилагательное (полное)
    'ADJS',   # имя прилагательное (краткое)
    'COMP',   # компаратив
    'VERB',   # глагол (личная форма)
    'INFN',   # глагол (инфинитив)
    'PRTF',   # причастие (полное)
    'PRTS',   # причастие (краткое)
    'GRND',   # деепричастие
    'NUMR',   # числительное
    'ADVB',   # наречие
    'NPRO',   # местоимение-существительное
    'PRED',   # предикатив
    'PREP',   # предлог
    'CONJ',   # союз
    'PRCL',   # частица
    'INTJ',   # междометие
])
```

POS-tags in OpenCorpora

The Hidden Markov Models

	NNP	MD	VB	JJ	NN	RB	DT
$<s>$	0.2767	0.0006	0.0031	0.0453	0.0449	0.0510	0.2026
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	0.0090	0.0025
MD	0.0008	0.0002	0.7968	0.0005	0.0008	0.1698	0.0041
VB	0.0322	0.0005	0.0050	0.0837	0.0615	0.0514	0.2231
JJ	0.0366	0.0004	0.0001	0.0733	0.4509	0.0036	0.0036
NN	0.0096	0.0176	0.0014	0.0086	0.1216	0.0177	0.0068
RB	0.0068	0.0102	0.1011	0.1012	0.0120	0.0728	0.0479
DT	0.1147	0.0021	0.0002	0.2157	0.4744	0.0102	0.0017

Figure 8.7 The A transition probabilities $P(t_i|t_{i-1})$ computed from the WSJ corpus without smoothing. Rows are labeled with the conditioning event; thus $P(VB|MD)$ is 0.7968.

The Hidden Markov Models

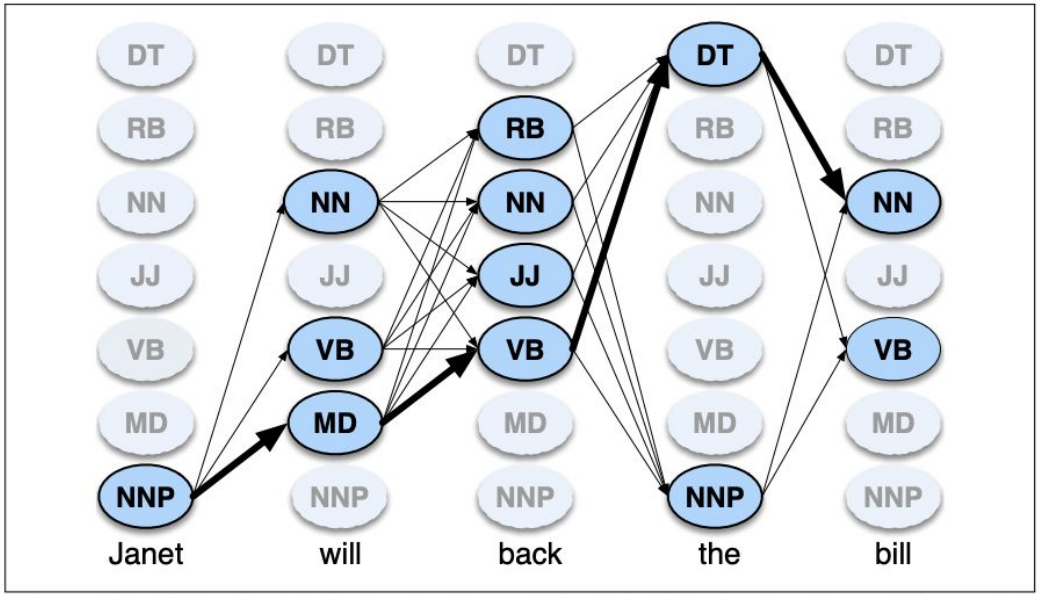


Figure 8.6 A sketch of the lattice for *Janet will back the bill*, showing the possible tags (q_i) for each word and highlighting the path corresponding to the correct tag sequence through the hidden states. States (parts of speech) which have a zero probability of generating a particular word according to the B matrix (such as the probability that a determiner DT will be realized as *Janet*) are greyed out.

Why does morphology matter?

- Information retrieval:
  - A query for ***phones*** should match both ***phone*** and ***phones***
- Language modeling:
  - If we have seen ***scrutinize***, we can predict ***scrutinized***
- Machine translation:
  - Swedish ***bilen*** corresponds to English ***the car***
- etc.

Morphological analyzers

- **Morphology software for English**: https://aclweb.org/aclwiki/Morphology_software_for_English
- **AOT**: http://aot.ru/technology.html
- **PyMorphy**: *documentation*: http://pymorphy.readthedocs.io/en/v0.5.6/index.html , *code*: https://github.com/kmike/pymorphy2
- **TreeTagger**: http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/
- **MyStem**: https://tech.yandex.ru/mystem/
- more: https://nlpub.ru/%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%B0

Thank you for your attention!