Comp349 Spoken Language Dialogue Systems
Week 10: Language Modelling

Robert Dale
Robert.Dale@mq.edu.au
Context: The Architecture of an SLDS

- Speech Recognition
- Language Understanding
- Dialog Management
- Database
- Language Generation
- Speech Synthesis
Today's Focus: From Acoustic Signal to Word Sequence

y uh r s t ae n d ih ng oa n m ay f uh t

you’re standing on my foot

Phoneme Sequence via an Acoustic Model

Word Sequence via the Lexicon and a Language Model
The Lexicon

- Defines the mapping between orthographic words and phoneme sequences
- The recognizer:
  - hypothesises phoneme sequences on the basis of the acoustic model
  - uses the lexicon to map the hypothesized phoneme sequence to a hypothesized word
- The lexicon is effectively a grammar for building words out of phonemes: it tells us what the valid phoneme sequences are
# A Sample Lexicon

<table>
<thead>
<tr>
<th>Word</th>
<th>Sound</th>
<th>Word</th>
<th>Sound</th>
<th>Word</th>
<th>Sound</th>
</tr>
</thead>
<tbody>
<tr>
<td>able</td>
<td>e b * l</td>
<td>butter</td>
<td>b ^ ! * r</td>
<td>echo</td>
<td>E k o</td>
</tr>
<tr>
<td>adam</td>
<td>a ! * m</td>
<td>cat</td>
<td>k a t</td>
<td>even</td>
<td>i v * n</td>
</tr>
<tr>
<td>agent</td>
<td>e dZ * n t</td>
<td>caught</td>
<td>k O t</td>
<td>father</td>
<td>f A D * r</td>
</tr>
<tr>
<td>alive</td>
<td>* l aj v</td>
<td>child</td>
<td>tS aj l d</td>
<td>focus</td>
<td>f o k * s</td>
</tr>
<tr>
<td>aphid</td>
<td>e f * d</td>
<td>coat</td>
<td>k o t</td>
<td>golf</td>
<td>g A l f</td>
</tr>
<tr>
<td>apple</td>
<td>a p * l</td>
<td>compass</td>
<td>k ^ m p * s</td>
<td>grammar</td>
<td>g r a m * r</td>
</tr>
<tr>
<td>asked</td>
<td>a s k t</td>
<td>confuse</td>
<td>k * n f j u z</td>
<td>hotel</td>
<td>h o t E l</td>
</tr>
<tr>
<td>ball</td>
<td>b O l</td>
<td>cow</td>
<td>k aw</td>
<td>india</td>
<td>I n d i *</td>
</tr>
<tr>
<td>bird</td>
<td>b * r d</td>
<td>cry</td>
<td>k r aj</td>
<td>iraq</td>
<td>* r A k</td>
</tr>
<tr>
<td>book</td>
<td>b U k</td>
<td>dogs</td>
<td>d A g z</td>
<td>iraq</td>
<td>I r A k</td>
</tr>
<tr>
<td>bought</td>
<td>b O t</td>
<td>dogs</td>
<td>d O g z</td>
<td>iraq</td>
<td>aj r A k</td>
</tr>
</tbody>
</table>
The Role of the Lexicon

Phoneme Sequence via an Acoustic Model

you’re standing on my foot

The lexicon determines the possible valid chunkings
Lexicon Size Depends on the Application

• A desktop dictation package requires a large lexicon:
  – Dragon NaturallySpeaking 8 has a vocabulary of 300,000 words; special versions add to this — DNS 8 Legal comes with a dictionary containing nearly 30,000 legal terms
• Current telephony-based apps require much smaller lexica:
  – Taxi booking: maybe 100-200 general purpose words, 650 suburb names, for a total of around 1000-1500 words
Language Models and the Lexicon

• The lexicon provides a model of possible words (i.e., phoneme sequences) that might be heard
• It doesn't by itself say anything about the likelihood (i.e. probability) of those words
• For this we need a language model
Today's Agenda

1. Language Models
2. Grammars and Syntax
3. Writing Grammars
Today's Agenda

1. Language Models
2. Grammars and Syntax
3. Writing Grammars
Basic Observation

- In any given context, some words are more likely than others.
- If we can capture the probabilities in some way, we can help the speech recognizer decide what it is hearing.
- We encode these probabilities in a language model.
Two Kinds of Language Model

• Grammar-based:
  – Used in small vocabulary speaker-independent contexts (like telephony-based dialog systems) where many speakers are likely

• N-gram-based:
  – Used in large vocabulary speaker-dependent contexts (like dictation) where the system can be tuned to characteristics of a single speaker
A Bad Language Model

[Slide borrowed from Joshua Goodman]
A Bad Language Model

by Jim Unger

[Slide borrowed from Joshua Goodman]
A Bad Language Model

[Slide borrowed from Joshua Goodman]
What’s a Language Model?

• A Language Model is a probability distribution over word sequences:
  – $P(\text{"And nothing but the truth"}) \approx 0.001$
  – $P(\text{"And nuts sing on the roof"}) \approx 0$
Observation

• Not all words are equally likely:
  – It has been estimated that 8000 morphemes is sufficient to handle 95% of texts
  – Typically, the 15 most frequent words account for 25% of tokens
  – The 100 most frequent words account for 60% of tokens
<table>
<thead>
<tr>
<th>Rank</th>
<th>Spoken English</th>
<th>French</th>
<th>Written English</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>the</td>
<td>de</td>
<td>the</td>
<td>der</td>
</tr>
<tr>
<td>2</td>
<td>and</td>
<td>of</td>
<td>of</td>
<td>die</td>
</tr>
<tr>
<td>3</td>
<td>to</td>
<td>la</td>
<td>to</td>
<td>und</td>
</tr>
<tr>
<td>4</td>
<td>to</td>
<td>et</td>
<td>in</td>
<td>in</td>
</tr>
<tr>
<td>5</td>
<td>of</td>
<td>des</td>
<td>of</td>
<td>des</td>
</tr>
<tr>
<td>6</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>den</td>
</tr>
<tr>
<td>7</td>
<td>that</td>
<td>est</td>
<td>that</td>
<td>zu</td>
</tr>
<tr>
<td>8</td>
<td>was</td>
<td>was</td>
<td>was</td>
<td>das</td>
</tr>
<tr>
<td>9</td>
<td>is</td>
<td>un</td>
<td>is</td>
<td>von</td>
</tr>
<tr>
<td>10</td>
<td>it</td>
<td>du</td>
<td>it</td>
<td>fur</td>
</tr>
</tbody>
</table>
Unigrams:
The Simplest Kind of Language Model

• Consider a word's probability to be related to its frequency in the language: the higher the frequency, the more likely the word

• Note: different contexts lead to different probabilities
  – "How many apples should I buy?"
  – "How many passengers are travelling?"

• Consider the grammars you wrote for this week's tutorial
Key Topics

- The Lexicon
- Grammar-based Language Models
- N-gram-based Language Models
Grammars as Language Models

- A grammar is one kind of language model
- A grammar defines the possible sequences of words that can be recognised at some point in the dialog
Ways of Helping the Recogniser

• Specify a grammar for each dialog state
• Make some parts of a grammar more likely than others by adding probabilities
• Dynamically refine the grammar: e.g. populate airport city grammars on the basis of the country under discussion
A Grammar with Probabilities

NAME
[
  2WS_SUB~0.001312147
  3AW_SUB~0.002891474
  3M_AUSTRALIA_SUB~0.002072124
  6IX_SUB~0.001169652
  6PR_SUB~0.001169652
  92.9PMFM_SUB~0.001656512
  94.5FM_SUB~0.001852443
  A_B_C_SPARE_PARTS_SUB~0.001211213
  A_N_A_HOTEL_SUB~0.005397023
  AAMI_SUB~0.035600162
  AAPT_SUB~0.006471677
  ABC_RADIO_SUB~0.01284242
  ABC_TAXIS_SUB~0.003117092
  ABC_TELEVISION_SUB~0.004607359
  ... ]
Grammar Compilation

Colour →
[i like]  (blue | aquamarine | yellow | green | red | scarlet)
Grammar Compilation #1

\[ I \in \{\text{like, acquamarine, blue, yellow, scarlet, red, green}\} \]
• Probabilities can be added to the arcs in the grammar to reflect likelihood of different paths being taken.
Key Topics

- The Lexicon
- Grammar-based Language Models
- N-gram-based Language Models
Large Vocabulary Speech Recognition

- Suppose you want to allow the user to say just about anything
- One approach:
  Utterance $\rightarrow$ Word+
Large Vocabulary Speech Recognition

At any point, recognize any word in the lexicon.
Adding Unigram Probability

Each transition conditioned by inverse frequency of word

Word1: 0.002%

Word2: ..

...

Word43171

Word43172: 0.0007%
The Predictive Power of N-Grams

- We can do better than unigram probabilities
- Example input:
  - Yesterday I went to the …
- Bigrams:
  - Next word is something that can or is likely to follow 'the'
- Trigrams:
  - Next word is something that can or is likely to follow 'to the'
N-Gram Language Modelling

• Basic idea:
  – the probability of occurrence of a symbol is conditioned upon the prior occurrence of N-1 other symbols
  – typically constructed from statistics obtained from a large corpus of text using the co-occurrences of words in the corpus to determine word sequence probabilities

• N is typically 2 or 3
Word N-Gram Models

• Bigram Model:
  - gives the probability of all possible next words on the basis of the current word only
  - \( p(w_2 \mid w_1) \)

• Trigram model:
  - based on two words of context
  - \( p(w_3 \mid w_1w_2) \)
Why use N-gram language modelling?

- Allows large vocabulary applications
- A context free grammar of reasonable complexity can never foresee all the different utterance patterns that callers may use in spontaneous speech input
Grammar Based Models vs N-Grams

- **Grammars**
  - typically constructed by hand
  - difficult to maintain as they get larger
  - may be overly restrictive
- **N-gram models**
  - computed automatically
  - greater **perplexity**: more scope for misrecognition
Take-home Messages

• A lexicon maps phoneme sequences to orthographic words
• We need to guide the search for words: a language model predicts the next word in the sequence
• Language models can be grammar-based or n-gram based
Today's Agenda

1. Language Models
2. Grammars and Syntax
3. Writing Grammars
Key Topics

• What is a Grammar?
• The Elements of Syntax
What is a Grammar?

• A grammar is a set of rules which can
  – generate a construct from a list of terminals;
  – recognize that a construct obeys the grammar's rules

• A construct that obeys the grammar’s rules is said to be well-formed; everything else is ill-formed
What is a Grammar?

• A grammar formally defines the set of allowable phrases that can be recognized by the speech engine
• A grammar defines the syntax of a language
• It may return a semantic structure corresponding to the meaning of the recognized phrase
Phrase Structure Grammars

- A grammar is defined by means of a set of rewrite rules
- Example:
  
  \[
  \begin{align*}
  A & \rightarrow b \ C \\
  C & \rightarrow d \ e
  \end{align*}
  \]
- Symbols in the grammar are either terminal symbols or non-terminal symbols
- Terminal symbols cannot be rewritten any further
A Parse Tree

A
  b
  C
    e
    d
    b
Why Bother with Syntax?

- The autonomy of syntax:
  - Colourless green ideas sleep furiously
  - Furiously ideas colourless sleep green
- Syntax provides a level of analysis that is domain-independent
- Syntax structures data for subsequent semantic processing
A Simple Grammar for Natural Language

- $S \rightarrow NP \; VP$
- $NP \rightarrow Det \; N$
- $VP \rightarrow V \; NP$
- $Det \rightarrow \text{the}$
- $Det \rightarrow \text{a}$
- $N \rightarrow \text{cat}$
- $N \rightarrow \text{biscuit}$
- $V \rightarrow \text{ate}$
A Simple Grammar for Natural Language

• This grammar covers (or generates) the following strings:
  – The cat ate the biscuit.
  – The cat ate a biscuit.
  – A cat ate the biscuit.
  – A cat ate a biscuit.
  – The biscuit ate the biscuit.
  – The biscuit ate a biscuit.
  – A biscuit ate the biscuit.
  – A biscuit ate a biscuit.
  – The cat ate the cat.
  – The cat ate a cat.
  – A cat ate the cat.
  – A cat ate a cat.
  – The biscuit ate the cat.
  – The biscuit ate a cat.
  – A biscuit ate the cat.
  – A biscuit ate a cat.
A Parse Tree

S
   /\   /
  NP  VP
 /\   /\  \
Det N  V Det NP
the cat ate a biscuit
Grammars and Lexica

- Generally we separate out information about the structures in the language and information about the lexical elements in the language.
- Grammar contains non-terminals, some of which are pre-terminals.
- The pre-terminals correspond to the parts of speech in the lexicon.
A Simple Grammar and Lexicon

- The Grammar
  \[ S \rightarrow NP \ VP \]
  \[ NP \rightarrow \text{Det} \ N \]
  \[ VP \rightarrow V \ NP \]

- The Lexicon
  the, a: Det
  cat, biscuit: N
  ate: V
Today's Agenda

- What is a Grammar?
- The Elements of Syntax
Words

• Every word has a **Syntactic Category** or **Part of Speech**
• Two broad types: **Open Class** words vs **Closed Class** words
• Open classes:
  – Nouns, Verbs, Adjectives, Adverbs
• Closed classes:
  – Determiners, Conjunctions, Prepositions
• A semantic distinction: **Content words** vs **Function words**
• Parts of speech are important because they constrain how sentences can be put together
The cat sat on the mat.
The Open Classes

• Nouns
  – projector, money, infidelity, amazement, antidisestablishmentarianism …

• Verbs
  – run, fly, walk, procrastinate, believe …

• Adjectives
  – crazy, green, hungry, unbelievable, amazed, smart …

• Adverbs
  – slowly, hungrily, unbelievably …
The Closed Classes

• Determiners
  — a, the, this, that, these, those …

• Conjunctions
  — and, but, therefore, because …

• Prepositions
  — in, on, under, between, to, from …
Nouns and Noun Phrases

• Generally, nouns correspond to **things**
• A noun serves as the **head** of a **noun phrase**
• Typically, a noun phrase picks out (refers to) some object in the world
  – **The table** is in the corner.
  – **The table in the corner** is mine.
  – **The table you’re sitting at now** used to be mine.
  – It’s **the only table my mother could find in the antique store**.
Noun Phrases

• Simplest noun phrases are pronouns
  – I, you, me, he, she, it, we, they …

• Proper names are noun phrases
  – Melbourne
  – President Clinton
  – The Sydney Morning Herald
Noun Phrases

• Most noun phrases have a common noun as head
• Common nouns correspond to general concepts
  – table
  – cloud
  – timetable
  – amazement
  – disbelief
Common Nouns

• Nouns can be count or mass
• **Count nouns** describe countable objects
  – table, stapler, belief, freeway …
• **Mass nouns** describe quantities of substance
  – flour, intimidation, meat, water …
• Many nouns can be used in both count and mass senses:
  – I bought *some paper*/I bought a paper.
  – John ate *the fish*/John made the pie with *fish*. 
The Components of a Noun Phrase

- The head is preceded by Specifiers and Qualifiers
- The head is followed by Complements
Specifiers

• Specifiers consist of Ordinals, Cardinals and Determiners

• Ordinals
  – first, second, ..., fifty-sixth, ...

• Cardinals
  – two, five hundred, a billion and six, ...

• Determiners ...
Determiners

- Articles
  - a, an, the …
- Demonstratives
  - this, these, those …
- Possessives
  - his, her, my, John’s, the richest man in the world’s …
- Quantifiers
  - some, every, most, no, both …
Determiners

- Determiners are **definite** or **indefinite**
- **Definite determiners**
  - the, this, that, those, these
- **Indefinite determiners**
  - a, an, some
Qualifiers

• These are adjectives or nouns used to **modify** the head noun

• Adjectives:
  – Words that attribute qualities to objects
  – the **angry** man, the **unbelievable** exercise, the **hard-to-see** insect

• Nouns as modifiers:
  – Mass or count nouns
  – the **video** recorder, the **data** projector, the **class** timetable
Noun Phrase Complements

• Complements follow the head

• Two major types: Prepositional Phrases and Relative Clauses

• Prepositional Phrases
  – the first man on the moon, the fastest road between Melbourne and Sydney, the dog in the window …

• Relative Clauses
  – the first man who got to the moon, the fastest road which you can find between Melbourne and Sydney, the dog that is in the window
Noun Phrases Summarised

- Proper Names
- Pronouns
- Complex NPs: Specifiers + Qualifiers + Head + Complements
The Open Classes

• Nouns
  – projector, money, infidelity, amazement, antidisestablishmentarianism …

• Verbs
  – run, fly, walk, procrastinate, believe …

• Adjectives
  – crazy, green, hungry, unbelievable, amazed, smart …

• Adverbs
  – slowly, hungrily, unbelievably …
• Every sentence contains a main verb

• Sentences are of different types
  – Statements, also referred to as assertions or declaratives
  – Questions, also referred to as interrogatives
  – Commands, also referred to as imperatives

• Questions can be of two types
  – Yes/No questions
  – Wh-questions
Verbs

- **Auxiliary Verbs** typically take a verb phrase as complement
  - be, do, have
- **Modal Verbs** also typically take a verb phrase as complement
  - will, can, could
- **Main Verbs**
  - justify, mediate, disenfranchise, congratulate, pester
Verb Tenses

- Present Progressive: He is walking
- Past Progressive: He was walking
- Future Progressive: He will be walking
- Present Perfect Progressive: He has been walking
- Future Perfect Progressive: He will have been walking
- Past Perfect Progressive: He had been walking
Verb Subcategorisation

- Different verbs subcategorise for different complements
  - Intransitive verbs take no object
    - I laughed, He will have been running
  - Transitive verbs take an object
    - I tasted a cake, He ran a machine
  - Ditransitive verbs take two objects
    - I gave John a book
    - I sent Mary a letter
Verb Subcategorisation

• Many verbs typically allow clausal complements:
  – I know that Mary stole the ribbon
  – I agreed to buy the shares
• Many verbs require Prepositional Phrase complements of specific types:
  – Mary blamed the theft on the children.
  – Zoe conferred with the committee about the solution.
The Open Classes

- **Nouns**
  - projector, money, infidelity, amazement, antidisestablishmentarianism ...

- **Verbs**
  - run, fly, walk, procrastinate, believe ...

- **Adjectives**
  - crazy, green, hungry, unbelievable, amazed, smart ...

- **Adverbs**
  - slowly, hungrily, unbelievably ...
Adjectives

- Adjective phrases often consist of single adjectives
  - the green tree
  - the difficult problem
  - the ecstatic confederation
  - John’s favourite Alanis Morissette album

- Two kinds of adjective phrases
  - prenominal
  - verb complement
Adjective Phrases

- Prenominal Adjective Phrases are limited in variety:
  - the very long summer
  - the most desperate student
  - the too-stupid-to-see-he-was-being-robbed consumer
More complex forms typically appear as complements of verbs.

- The committee was pleased with the price of the NRMA shares.
- The storekeeper was angry at the imposition of the GST.
- Bill seemed happy that his dog was finally becoming vegetarian.
The Open Classes

- Nouns
  - projector, money, infidelity, amazement, antidisestablishmentarianism ...

- Verbs
  - run, fly, walk, procrastinate, believe ...

- Adjectives
  - crazy, green, hungry, unbelievable, amazed, smart ...

- Adverbs
  - slowly, hungrily, unbelievably ...
Adverbs

- The category ‘adverb’ is something of a catch-all for things not covered by the other categories
- Adverbs can modify verbs or adjectives:
  - He ate the cake **slowly**.
  - She **cautiously** opened the gate.
  - His **unbelievably** rich uncle visited.
  - Samantha’s **very** worn-out sofa did the job.
Today's Agenda

1. Language Models
2. Grammars and Syntax
3. Writing Grammars
Key Topics

- Types of Grammars
- Semantic Grammars
- How To Write a Grammar
Types of Grammars

- **Syntactic grammars**
  - Domain-independent
  - Capture generalizations about language as a whole
  - Goal is broad coverage

- **Semantic grammars**
  - Domain-dependent
  - Capture generalizations within a restricted domain
  - Limited coverage is okay
Syntactic vs Semantic Grammars

- Syntactic grammars deliver a structure from which a semantic representation can be derived
- Semantic grammars map more directly to a semantics
Today's Agenda

- Types of Grammars
- Semantic Grammars
- How To Write a Grammar
Semantic Grammars

- Domain-dependent
- More restricted coverage than ‘broad coverage’ grammars
- Don’t make distinctions that are unimportant for the domain at hand
Sample Data for an Auto Attendant System

- Q: Who would you like to speak to?
  - Jim in Finance
  - Jim on Extension 3323
  - The CEO
  - Can you put me through to ...
  - Can I speak to ...
  - Can I have ...
Bottom-Up Generalisation

- Jim in Finance $\rightarrow$ NAME in DEPARTMENT
- Jim on Extension 3323 $\rightarrow$ NAME on EXTENSION
- The CEO $\rightarrow$ ROLE
- Can you put me through to ... $\rightarrow$ REQUESTPHRASE
- Can I speak to ... $\rightarrow$ REQUESTPHRASE
- Can I have ... $\rightarrow$ REQUESTPHRASE
- I’m looking for ... $\rightarrow$ REQUESTPHRASE
Collating the Grammar Rules

- RESPONSE \rightarrow [REQUESTPHRASE] PERSON
- PERSON \rightarrow NAME
- PERSON \rightarrow NAME in DEPARTMENT
- PERSON \rightarrow NAME on EXTENSION
- PERSON \rightarrow ROLE
- NAME \rightarrow FIRSTNAME [SECONDNAME]
Today's Agenda

- Types of Grammars
- Semantic Grammars
- How To Write a Grammar
Steps in Writing a Grammar

1. Assemble some representative data
2. Start with the simpler cases
3. Add grammar rules that capture generalizations
4. Add more complex cases
5. Repeat from Step 3
A GetDate Dialog State

System: On what date do you want to travel?
User: The fifteenth of January.
Some Representative Data

- Today.
- Tomorrow.
- May ten.
- On the sixteenth of January.
A Simple Grammar: First Cut

DATE → today | tomorrow | DATE-NP | DATE-PP
DATE-NP → May ten
DATE-PP → on the sixteenth of January
A Simple Grammar:  
Generalise DATE-NP

\[
\text{DATE} \rightarrow \text{today} \mid \text{tomorrow} \mid \text{DATE-NP} \mid \text{DATE-PP} \\
\text{DATE-NP} \rightarrow \text{MONTH} \ \text{DAYNUMBER} \\
\text{MONTH} \rightarrow \text{January} \mid \text{February} \mid \text{March} \mid \ldots \\
\text{DAYNUMBER} \rightarrow \text{one} \mid \text{two} \mid \ldots \mid \text{thirty-one} \\
\text{DATE-PP} \rightarrow \text{on the sixteenth of January}
\]
A Simple Grammar: Generalise DATE-PP

\[
\begin{align*}
\text{DATE} & \rightarrow \text{today} \mid \text{tomorrow} \mid \text{DATE-NP} \mid \text{DATE-PP} \\
\text{DATE-NP} & \rightarrow \text{MONTH DAYNUMBER} \\
\text{MONTH} & \rightarrow \text{January} \mid \text{February} \mid \text{March} \mid \ldots \\
\text{DAYNUMBER} & \rightarrow \text{one} \mid \text{two} \mid \ldots \mid \text{thirty-one} \\
\text{DATE-PP} & \rightarrow \text{on the DAYNUMBER of MONTH}
\end{align*}
\]
A Simple Grammar: Recognize Ordinal vs Cardinal Distinction

DATE → today | tomorrow | DATE-NP | DATE-PP
DATE-NP → MONTH CARDINALDAYNUMBER
MONTH → January | February | March | ...
CARDINALDAYNUMBER → one | two | ...

DATE-PP → on the ORDINALDAYNUMBER of MONTH
ORDINALDAYNUMBER → first | second | ...

DATE-PP → thirty-one
A Simple Grammar: Generalise DATE-NP

DATE → today | tomorrow | DATE-NP | DATE-PP
DATE-NP → MONTH CARDINALDAYNUMBER
DATE-NP → the ORDINALDAYNUMBER of MONTH
MONTH → January | February | March | …
CARDINALDAYNUMBER → one | two | … | thirty-one
DATE-PP → on DATE-NP
ORDINALDAYNUMBER → first | second | … | thirty-first
A Simple Grammar: Tidy-up

DATE → today | tomorrow | DATE-NP | DATE-PP
DATE-NP → (MONTH CARDINALDAYNUMBER) | (the ORDINALDAYNUMBER of MONTH)
DATE-PP → on DATE-NP
MONTH → January | February | March | …
CARDINALDAYNUMBER → one | two | … | thirty-one
ORDINALDAYNUMBER → first | second | … | thirty-first
A Simple Grammar: Recognize Other DATE-NP Possibilities

DATE-NP → (MONTH CARDINALDAYNUMBER) | (MONTH ORDINALDAYNUMBER) | (the ORDINALDAYNUMBER of MONTH) | (ORDINALDAYNUMBER of MONTH) | (ORDINALDAYNUMBER MONTH) | (CARDINALDAYNUMBER MONTH)
DATE-NP → (MONTH CARDINALDAYNUMBER) | (MONTH ORDINALDAYNUMBER) | ([the] ORDINALDAYNUMBER [of] MONTH) | (CARDINALDAYNUMBER MONTH)
A Simple Grammar: An Alternative Factorisation

DATE-NP → CARDINALDATE-NP | ORDINALDATE-NP
CARDINALDATE-NP → (MONTH CARDINALDAYNUMBER) | (CARDINALDAYNUMBER MONTH)
ORDINALDATE-NP → (MONTH ORDINALDAYNUMBER) | ([the] ORDINALDAYNUMBER [of] MONTH)
A Simple Grammar: Another Factorisation

\[
\begin{align*}
\text{DATE-NP} & \rightarrow \text{MONTH-DAY-NP} | \text{DAY-MONTH-NP} \\
\text{MONTH-DAY-NP} & \rightarrow (\text{MONTH \ CARDINALDAYNUMBER}) | (\text{MONTH \ ORDINALDAYNUMBER}) \\
\text{DAY-MONTH-NP} & \rightarrow (\text{CARDINALDAYNUMBER \ MONTH}) | ([\text{the}] \ ORDINALDAYNUMBER \ [of] \ \text{MONTH})
\end{align*}
\]
A Simple Grammar:
Yet Another Factorisation

\[
\begin{align*}
\text{DATE-NP} & \rightarrow \text{MONTH-DAY-NP} \mid \text{DAY-MONTH-NP} \\
\text{MONTH-DAY-NP} & \rightarrow \\
& \hspace{1em} \text{MONTH (CARDINALDAYNUMBER} \mid \text{ORDINALDAYNUMBER}) \\
\text{DAY-MONTH-NP} & \rightarrow \\
& \hspace{1em} (\text{CARDINALDAYNUMBER} \mid ([\text{the}] \text{ORDINALDAYNUMBER} \text{[of]})) \\
& \hspace{1em} \text{MONTH}
\end{align*}
\]
Collect More Data

System: On what date do you want to travel?
User: In five days' time.
User: The day after tomorrow.
User: Next Friday.
A Simple Grammar: Recognize Relative vs Absolute Distinction

```
DATE → RELATIVE-DATE | ABSOLUTE-DATE
RELATIVE-DATE → today | tomorrow | RELATIVE-NP | RELATIVE-PP
ABSOLUTE-DATE → DATE-NP | DATE-PP
RELATIVE-NP → the day after tomorrow
RELATIVE-PP → (next DAY-NAME) | (in NUMBER days' time)
```
More Data

- Need to cater for various degrees of specification:
  - On Friday.
  - On the nineteenth.
  - On Friday the nineteenth.
  - On the nineteenth of October.
  - On the nineteenth of October two thousand and seven.