# Brian — PS 14 — 2025-03-28 — Solution

*EIWL3* Sections 35 and 36

## Exercises from *EIWL3* Section 35

In[1]:= `(* 35.1 *) Interpreter["Location"]["Eiffel Tower"]`

Out[1]= `GeoPosition[{48.8583, 2.29444}]`

In[2]:= `(* 35.2 *) Interpreter["University"]["U of T"]`

Out[2]= `University of Toronto`

In[3]:= `(* 35.3 *) Interpreter["Chemical"][{"C2H4", "C2H6", "C3H8"}]`

Out[3]= `{ ethylene , ethane , propane }`

In[4]:= `(* 35.4 *) Interpreter["Date"]["20140108"]`

Out[4]= `Wed 8 Jan 2014`

In[5]:= 
```
(* 35.5 *) (* I was able to get this far: *)
Interpreter["University"][
 Table[StringJoin["U of ", letter], {letter, Capitalize[Alphabet[]]}]]
(* See below *)
```

Out[5]= { Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] , University of Birjand ,

University of California-Berkeley , Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] ,

The University of Edinburgh , Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] ,

University of Georgia , University of Houston , University of Illinois at Urbana-Champaign ,

Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] ,

Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] , University of Lethbridge ,

University of Michigan-Ann Arbor , Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] ,

Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] ,

University of Phoenix-Online Campus ,

Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] ,

University of Regina , University of Saskatchewan , University of Toronto ,

Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] ,

Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] ,

Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] ,

Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] ,

Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] ,

Failure[ ⚠ Message: No university interpretation found. Try again.
Tag: InterpretationFailure ] }

In[6]:= `(* Then I had to look up how to proceed. The trick is to wrap *)`
`(* the results in Cases[ ... , _Entity]. This eliminates all *)`
`(* the failures. *)`
`Cases[Interpreter["University"][`
`  Table[StringJoin["U of ", letter], {letter, Capitalize[Alphabet[]]}]], _Entity]`

Out[6]= { University of Birjand , University of California-Berkeley , The University of Edinburgh ,

University of Georgia , University of Houston , University of Illinois at Urbana-Champaign ,

University of Lethbridge , University of Michigan-Ann Arbor , University of Phoenix-Online Campus ,

University of Regina , University of Saskatchewan , University of Toronto }

In[7]:= `(* 35.6 *)` `CommonName /@ Cases[Interpreter["Movie"][`

`CommonName /@` all US states with District of Columbia ADMINISTRATIVE DIVISIONS [ *capital city* ]], _Entity]

Out[7]= {Phoenix, Honolulu, Topeka, Annapolis, Lincoln, Santa Fe, Expedition: Bismarck,
 Columbus, Providence, Nashville, Olympia, Madison, Cheyenne}

In[8]:= `(* 35.7 *) Cases[`
`Interpreter["Movie"][StringJoin /@ Permutations[{"a", "i", "l", "m"}]], _Entity]`

Out[8]= { Liam , Mai , Apples , Mai }

In[9]:= `(* 35.8 *) TextCases[WikipediaData["gunpowder"], "Country"]`

Out[9]= {China, China, China, Chinese, Chinese, Chinese, Chinese, China, China,
 Japan, Mongols, Syrian, Chinese, Chinese, Chinese, Chinese, China, Syria,
 Egypt, Persia, Chinese, Persian, Chinese, Mongols, Mongols, Chinese,
 Turkish, Chinese, Turkish, Chinese, Turkish, Greek, Mongols, Mongols,
 Mongols, Mongols, China, Mongols, Japan, Italy, Germany, Italy, France,
 France, France, France, Ireland, German, Great Britain, India, India,
 India, Mongols, Mongol, India, Mongol, Mongol, India, India, Portuguese,
 Bengal, French, Portuguese, French, India, Chinese, Mongol, Chinese, Mongol,
 Portuguese, Portuguese, Portuguese, Spaniards, German, China, China, China,
 China, Chinese, China, China, Peru, Chile, Peru, French, British, Germany,
 French, U, S, United Kingdom, United States, China, United States, German,
 United Kingdom, US, Great Britain, American, Great Britain, United States,
 Indian, Chilean, Great Britain, Germany, United States, Australia, UK}

In[10]:= `(* 35.9 *) TextCases["She sells seashells by the sea shore.", "Noun"]`
Out[10]=
{seashells, sea, shore}

In[11]:= `StringTake[WikipediaData["computers"], 1000]`

Out[11]=

A computer is a machine that can be programmed to automatically carry out
sequences of arithmetic or logical operations (computation). Modern digital
electronic computers can perform generic sets of operations known as programs.
These programs enable computers to perform a wide range of tasks. The term
computer system may refer to a nominally complete computer that includes
the hardware, operating system, software, and peripheral equipment needed
and used for full operation; or to a group of computers that are linked
and function together, such as a computer network or computer cluster.
A broad range of industrial and consumer products use computers
as control systems, including simple special-purpose devices like
microwave ovens and remote controls, and factory devices like industrial
robots. Computers are at the core of general-purpose devices such as
personal computers and mobile devices such as smartphones. Computers
power the Internet, which links billions of computers and users.

In[12]:= `(* 35.10 *) Length[TextCases[StringTake[WikipediaData["computers"], 1000], #]] & /@`
`{"Noun", "Verb", "Adjective"}`

Out[12]=

{54, 23, 20}

In[13]:= `(* 35.11 *) TextStructure[TextSentences[WikipediaData["computers"]]〚1〛]`

Out[13]=

| A | computer | is | a | machine | that | can | be | programmed | to | automa |
|---|---|---|---|---|---|---|---|---|---|---|
| Determiner | Noun | Verb | Determiner | Noun | Wh-Determiner | Verb | Verb | Verb | Preposition | A |
| Noun Phrase | | | Noun Phrase | | Wh-Noun Phrase | | | | | Adve |

In[14]:= `(* 35.12 *)Keys[Take[Reverse[Sort[`
`    Counts[TextCases[ExampleData[{"Text", "AliceInWonderland"}], "Noun"]]]], 10]]`

Out[14]=
{Rabbit, door, voice, time, Mouse, way, moment, thing, head, garden}

In[15]:= `(* 35.13 *)`
`TextStructure[TextSentences[WikipediaData["language"]]〚1〛, "ConstituentGraphs"]`

Out[15]=
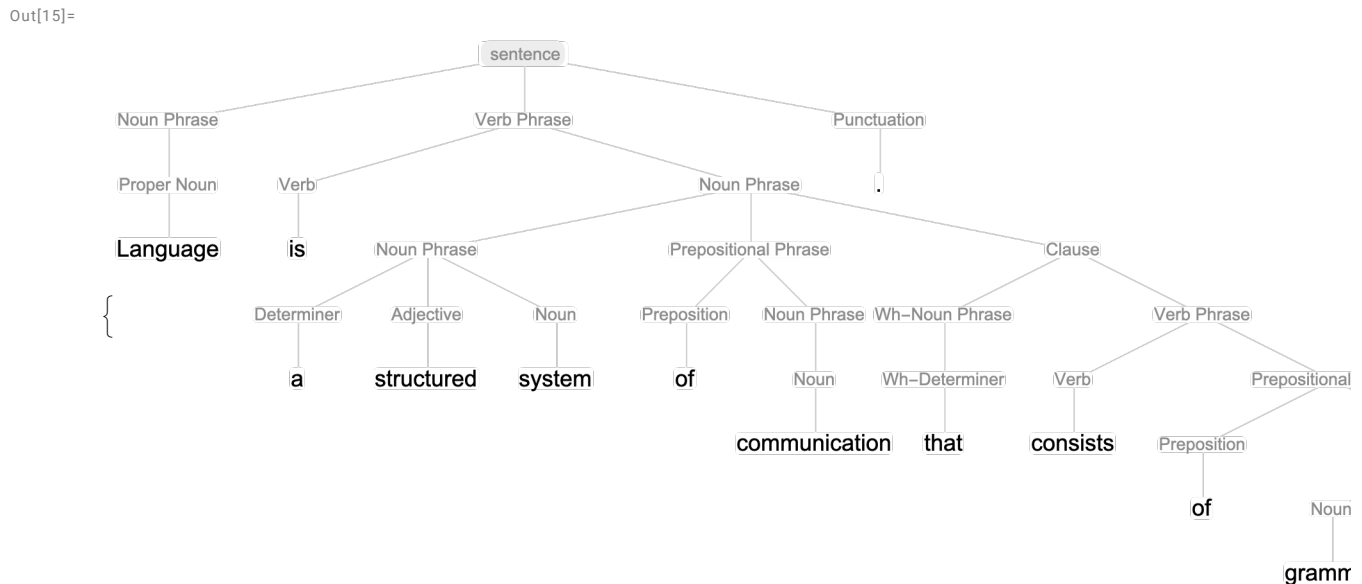


In[16]:= `(* 35.14 *) Length[Flatten[TextCases[WordList[], #]]] & /@`
`  {"Noun", "Verb", "Adjective", "Adverb"}`

Out[16]=
{22 728, 5894, 7146, 2824}

In[25]:= `(* 35.15 *) Flatten[WordTranslation[#, "French"] & /@ IntegerName /@ Range[2, 10]]`

Out[25]=
{deux, trois, quatre, cinq, six, sept, huit, neuf, dix}

## Exercises from *EIWL3* Section 36

In[18]:= `(* 36.1 *) CloudPublish[Delayed[Style[RandomInteger[1000], 100]]]`

Out[18]=
CloudObject[https://www.wolframcloud.com/obj/03213f78-7362-48cc-b78b-6bb8326ec9d7]

In[19]:= `(* 36.2 *)CloudPublish[FormFunction[{"number" → "Number"}, #number`$^2$` &]]`

Out[19]=
CloudObject[https://www.wolframcloud.com/obj/d386114c-bc12-4cfd-bc8a-cf7ccfbc8d44]

In[20]:= `(* 36.3 *)CloudPublish[FormFunction[{"x" → "Number", "y" → "Number"}, #x #y &]]`

Out[20]=
CloudObject[https://www.wolframcloud.com/obj/09523e1a-d0d4-470a-af78-5119d93ba7ac]

In[21]:= `(* 36.4 *) CloudPublish[`
`  FormFunction[{"topic" → "String"}, WordCloud[TextWords[WikipediaData[#topic]]] &]]`

Out[21]=
CloudObject[https://www.wolframcloud.com/obj/2dc41ee4-991f-4c00-90ae-5326f3ebcfb7]

In[22]:= `(* 36.5 *) (* I don't know what Wolfram meant by "repeatedly. *)CloudPublish[`
`  FormFunction[{"string" → String}, Style[StringReverse[#string], 100] &]]`

Out[22]=
CloudObject[https://www.wolframcloud.com/obj/9c0e72ee-07ea-451e-b9de-cf8be96a1226]

In[23]:= `(* 36.6 *)CloudPublish[FormFunction[{"n" → Integer},`
`   Graphics[Style[RegularPolygon[#n], RandomColor[]]] &]]`

Out[23]=
CloudObject[https://www.wolframcloud.com/obj/308e2e50-a0fc-4377-91b2-b158036c80d5]

In[24]:= `(* 36.7 *)CloudPublish[FormFunction[{"location" → Location, "count" → Integer},`
`   GeoListPlot[GeoNearest["Volcano", #location, #count]] &]]`

Out[24]=
CloudObject[https://www.wolframcloud.com/obj/210df29b-cf03-4d9c-9003-b343707ace1f]