University of Toronto, Department of Computer Science
CSC 485H/2501H: Computational linguistics, Fall 2021

Assignment 3

Due date: 23:59 on Thursday, December 9, 2021.
Late assignments will not be accepted without a valid medical certificate or other documentation of an emergency.
For CSC485 students, this assignment is worth 33% of your final grade.
For CSC2501 students, this assignment is worth 25% of your final grade.

- **Read the whole assignment carefully.**
- Type the written parts of your submission in no less than 12pt font.
- What you turn in must be your own work. You may not work with anyone else on any of the problems in this assignment. If you need assistance, contact the instructor or TA for the assignment.
- Any clarifications to the problems will be posted on the Piazza forum for the class. You will be responsible for taking into account in your solutions any information that is posted there, or discussed in class, so you should check the page regularly between now and the due date.
- The starter code directory for this assignment is accessible on Teaching Labs machines at the path `/u/csc485h/fall/pub/a3/`. In this handout, code files we refer to are located in that directory.
- When implementing code, make sure to **read the docstrings** as some of them provide important instructions, implementation details, or hints.
- Fill in your name, student number, and UTORid on the relevant lines at the top of each file that you submit. (Do not add new lines; just replace the `NAME`, `NUMBER`, and `UTORid` placeholders.)

Overview: Symbolic Machine Translation

In this assignment, you will learn how to write phrase structure grammars for some different linguistic phenomena in two different languages: English and Chinese. You can use the two grammars to create an *interlingual machine translation system* by parsing in one, and generating in the other. Don't panic if you don't speak Chinese, and also don't cheer up yet if you can speak the language — it won't give you much of an advantage over other students. A facility with languages in general will help you, as will the ability to learn and understand the nuances between the grammars of two different languages.

In particular, you will start by working on agreement. Then, you will need to analyse the quantifier scoping difference between the two languages.

TRALE Instructions The TRALE system can be run with:

```
/h/u2/csc485h/fall/pub/trale/trale -fsug
```

(which you are welcome to alias). For this assignment, TRALE needs to start a graphical interface: Gralej. Therefore, if you don't have access to the labs and want to run TRALE remotely, you can either use:

- RDP over SSH[1],
- Remote Access Server NX[2],
- or connect to `teach.cs` using ssh with either the `-X` or `-Y` flag:
  `ssh -X myutorid@teach.cs.toronto.edu`.

1. Agreement: Determiners, Numbers and Classifiers [10 marks]

English expresses subject–verb agreement in person and number. English has two kinds of number: singular and plural. The subject of a clause must agree with its predicate: they should be both singular or both plural. However, the number of a direct object does not need to agree with anything.

(1) A professor steals a cookie.

(2) Two professors steal a cookie.

(3) * Two professors steals two cookies.

[1] https://www.teach.cs.toronto.edu/using_cdf/rdp.html

[2] https://www.teach.cs.toronto.edu/using_cdf/remote_access_server.html

(4) * A professor steal two cookies.

Chinese, on the other hand, does not exhibit subject–verb agreement. As shown in the examples below, most nouns do not inflect at all for plurality. Chinese does, however, have a classifier (CL) part of speech that English does not. Semantically, classifiers are similar to English collective nouns (a *bottle* of water, a *murder* of crows), but English collective nouns are only used when describing collectives. With very few exceptions, classifiers are mandatory in complex Chinese noun phrases. Different CLs agree with different classes of nouns that are sorted by mostly semantic criteria. For example, 教授 (jiaoshou) *professor* is a person and an occupation, so it should be classified by either 个 (ge) or 位 (wei) and cannot be classified by the animal CL 只 (zhi). However, the rules of determining a noun's class constitute a formal system that must be followed irrespective of semantic similarity judgements. For example, while cats and dogs are both pets and can both be classified by the animal CL 只 (zhi), 狗 (gou) *dog* can take another classifier, 条 (tiao), for "string-like" objects.

(5) 一 个 教授
yi ge jiaoshou
one *ge*-CL professor

(6) 两 个 教授
liang ge jiaoshou
two *ge*-CL professor

(7) 三 个 教授
san ge jiaoshou
three *ge*-CL professor

(8) * 三 教授
san jiaoshou
three professor

(9) * 三 只 教授
san zhi jiaoshou
three *zhi*-CL professor

(10) 一 只 猫
yi zhi mao
one *zhi*-CL cat

(11) 两 只 猫
liang zhi mao
two *zhi*-CL cat

(12) 三 只 猫
san zhi mao
three *zhi*-CL cat

(13) * 三 条 猫
san *tiao*-CL mao
three cat

(14) * 三 位 猫
san wei mao
three *wei*-CL cat

You should be familiar by now with the terminology in the English grammar starter code for this question. The Chinese grammar is fairly similar, but there is a new phrasal category called a classifier phrase (CLP), formed by a number and a classifier. The classifier phrase serves the same role as a determiner does in English.

The two grammars below don't appropriately constrain the NPs generated. You need to design your own rules and features to properly enforce agreement.

English Grammar:
Rules:
S → NP VP
VP → V NP
NP → Det N
NP → Num N

Lexicon:
a: det
one: Num
two: Num
three: Num
cat: N
cats: N
dog: N
dogs: N
professor: N
professors: N
see: V
sees: V
saw: V
chase: V
chases: V

Chinese Grammar:
Rules:
S → NP VP
VP → V NP
NP → CLP N
CLP → Num CL

Lexicon:
一 yi *one/a*: Num
两 liang *two*: Num
三 san *three*: Num
猫 mao *cat*: N
狗 gou *dog*: N
教授 jiaoshou *professor*: N
看见 kanjian *see*: V
追 zhui *chase*: V
个 ge: CL
位 wei: CL
只 zhi: CL
条 tiao: CL

Here is a list of all of the nouns in this question and their acceptable classifiers:

- 猫 mao *cat*: 只 zhi;
- 狗 gou *dog*: 只 zhi, 条 tiao;
- 教授 jiaoshou *professor*: 个 ge, 位 wei.

(a) (7 marks) Implement one grammar for each language pursuant to the specifications above. English: `q1_en.pl` and Chinese: `q1_zh.pl`.

Neither of your grammars need to handle embedded clauses, e.g., *a professor saw two cats chase a dog.* Similarly for Chinese, your grammar doesn't need to parse sentences like example (15):

(15) 一 个 教授 看见 两 只 猫 追 一 条 狗
yi ge jiaoshou kanjian liang zhi mao zhui yi tiao gou
A professor saw two cats chase a dog.

For the Chinese grammar, the lexical entries can be coded in either pinyin (the Romanized transcriptions of the Chinese characters) or in simplified Chinese characters.

(b) (2 marks) Use your grammars to parse and translate the following sentences. Save and submit all the translation results in the `.grale` format. The results of sentence (16) should be named `q1b_en.grale` and the results of sentence (17) should be named `q1b_zh.grale`.

(16) Two cats chase one dog

(17) 一 个 教授 追 两 条 狗
yi ge jiaoshou zhui liang tiao gou

Operational Instructions

- If you decide to use simplified Chinese characters, enter them in Unicode and use the `-u` flag when you run TRALE.
- Independently test your grammars in TRALE first, before trying to translate.
- Use the function `translate` to generate a semantic representation of your source sentence. If your sentence can be parsed, the function `translate` should open another `gralej` interface with all of the translation results.

  ```
  | ?- translate([two,cats,chase,one,dog]).
  ```

- To save the translation results, on the top left of the `Gralej` window (the window with the `INITIAL CATEGORY` entry and all of the translated sentences listed), click `File >> Save all >> TRALE format`.
- Don't forget to **close all of the windows** or kill both of the `Gralej` processes after you finish. Each `Gralej` process will take up one port in the server, and no one can use the server if we run out of ports.

(c) (1 mark) Compare your translator with Google Translate[3]. At its core, Google Translate is a neural machine translation (NMT) system. In a few sentences, describe the similarities and differences between Google Translate and your system. Your analysis should be submitted as the section 1(c) in `analysis.txt`.

2. Quantifier Scope [30 marks]

Quantifiers For this assignment, we will consider two quantifiers: the universal quantifier (*every*, 每 mei) and the existential quantifier (*a*, 一 yi). In English, both quantifiers behave as singular determiners.

(18) A professor stole every cookie.

(19) * A professor stole every cookies.

(20) * A professors stole every cookie.

In Chinese, both of these quantifiers behave more like numerical determiners. In addition, when a universal quantifier modifies an NP that occurs before the verb (such as with a universally quantified subject), the preverbal operator 都 (dou) is required. When a universally quantified NP occurs after the verb, the dou-operator must not appear with it.

[3] https://translate.google.ca/

(21) Every professor stole a cookie.

(22) A professor stole every cookie.

(23) 每 个 教授 都 偷了 一 块 饼干
mei ge jiaoshou *dou* toule yi kuai binggan
∀ *ge*-CL professor *dou* stole ∃ *kuai*-CL cookie

(24) *每 个 教授 偷了 一 块 饼干
mei ge jiaoshou toule yi kuai binggan
∀ *ge*-CL professor stole ∃ *kuai*-CL cookie

(25) 一 个 教授 偷了 每 块 饼干
yi ge jiaoshou toule mei kuai binggan
∃ *ge*-CL professor stole ∀ *kuai*-CL cookie

(26) *一 个 教授 都 偷了 每 块 饼干
yi ge jiaoshou *dou* toule mei kuai binggan
∃ *ge*-CL professor *dou* stole ∀ *kuai*-CL cookie

We shall simplify our analysis of NPs in this question to be a sequence of a quantifier, a classifier and a noun, and forget all about other determiners such as numbers.

Quantifier Scope Ambiguity In lecture, we talked about different kinds of ambiguity. Quantifier scope ambiguity was one of them. In many English sentences, no matter what the order of the quantifiers, there is a quantifier scope ambiguity. For example, there can be two readings of this sentence (27):

- ($\exists > \forall$) Every student read a book. The book's title is *The Old Man and the Sea*.
- ($\forall > \exists$) Every student read a book. Some students read *The Old Man and the Sea*.

($\exists > \forall$) means the existential quantifier outscopes the universal quantifier in a logical form representation of the sentence.

(27) Every student read a book
∀ student read ∃ book
Ambiguous: $\forall > \exists$ and $\exists > \forall$

(28) 每 个 学生 都 读过 一 本 书
mei ge xuesheng dou duguo yi ben shu
∀ *ge*-CL student *dou* read ∃ *ben*-CL book
Ambiguous: $\forall > \exists$ and $\exists > \forall$

(29) A student read every book
∃ student read ∀ book
Ambiguous: $\exists > \forall$ and $\forall > \exists$

(30) 一 个 学生 读过 每 本 书
yi ge xuesheng duguo mei ben shu
∃ *ge*-CL student read ∀ *ben*-CL book
Unambiguous: only $\exists > \forall$

The English sentences (27,29) have a scope ambiguity no matter what the order of the quantifiers. In Chinese, however, the sentence is only ambiguous if the universal quantifier came first (28).

Received a coded retreat message we have. — Master Yoda

Topicalization and Movement Topicalization is a linguistic phenomenon in which an NP appears at the beginning of a sentence in order to establish it as the topic of discussion in a sentence or to emphasize it in some other way. It plays an important role in the syntax of fixed-word-order languages because grammatical function is mainly determined by word order. Both Chinese and English exhibit topicalization. The entire object NP, for example, can be moved to the beginning of the sentence in either language. But in Chinese, object topicalization is more restricted when the subject is quantified: it can happen when the subject is universally quantified, but not when it is existentially quantified (33-36).

(31) A book, every student read.
∃ book ∀ student read
Ambiguous: $\forall > \exists$ and $\exists > \forall$

(32) Every book, a student read.
∀ book ∃ student read
Ambiguous: $\forall > \exists$ and $\exists > \forall$

(33) 一 本 书 每 个 学生 都 读过
yi ben shu mei ge xuesheng dou duguo
∃ *ben*-CL book ∀ *ge*-CL student *dou* read
Ambiguous: $\forall > \exists$ and $\exists > \forall$[4]

(34) 每 本 书 每 个 学生 都 读过
mei ben shu mei ge xuesheng dou duguo
∀ *ben*-CL book ∀ *ge*-CL student *dou* read

(35) * 一 本 书 一 个 学生 读过
yi ben shu yi ge xuesheng duguo
∃ *ben*-CL book ∃ *ge*-CL student read

(36) * 每 本 书 一 个 学生 都 读过
mei ben shu yi ge xuesheng dou duguo
∀ *ben*-CL book ∀ *ge*-CL student *dou* read

In English, subject–verb agreement is not affected by movement; the number and person of the subject should always agree with the predicate no matter where it occurs. Here, you can assume that Chinese also follows the subject–verb agreement in the same way that English does.

Figures 1 and 2 show the parse trees of sentences (31) and (33). Topicalization is generally analysed with gaps. An empty trace is left in the untopicalized position of the object NP, where

[4]This sentence may seem unambiguously $\exists > \forall$ to some native speakers. But consider this example: 一本书每个学生都读过。但两本书就不一定了。(*One book, every student has read, but two books, not necessarily.*) The $\forall > \exists$ reading is in fact available.

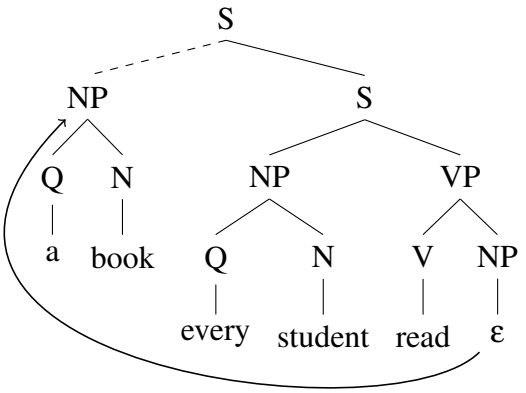


Figure 1: English topicalization parse tree: example (31).

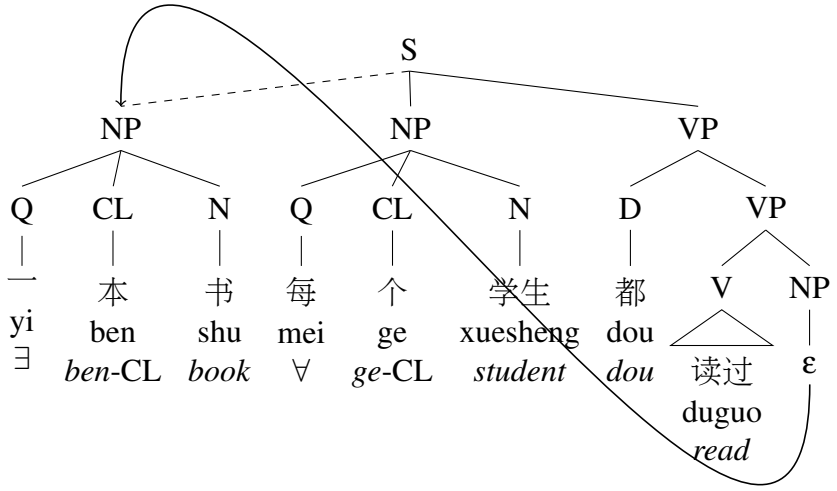


Figure 2: Chinese topicalization parse tree: example (33).

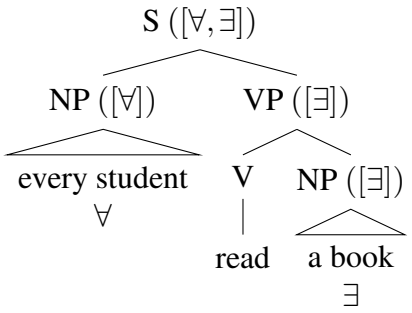
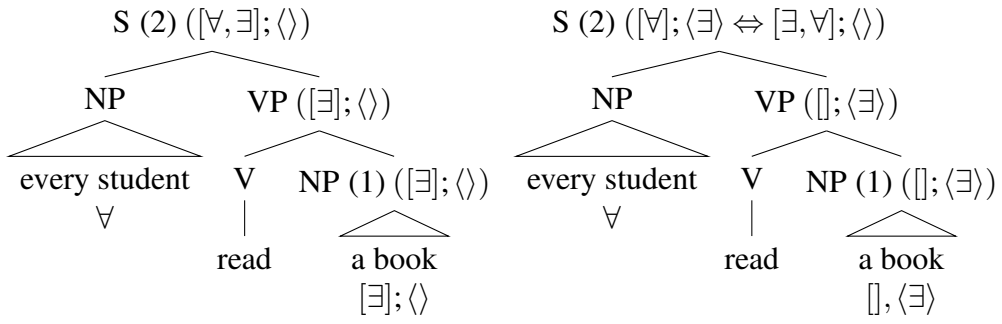


Figure 3: Quantifier scope tracking by maintain a list. The parse result of this sentence is $\forall > \exists$.



(a) $\forall > \exists$ (b) $\exists > \forall$

Figure 4: The basic idea of quantifier storage.

the gap is introduced. The gapped NP then percolates up the tree, and is finally unified with the topicalized NP at the left periphery of the sentence.[5]

Quantifier Storage But if quantifier scoping is a semantic effect, how do we represent it in syntax? When there is no ambiguity, keeping track of quantifier scope is pretty straightforward. As shown in figure 3, we can maintain a list-valued feature called a *quantifier stack* and record which quantifiers are seen as we ascend whilst building the parse tree. In practice, maintaining this stack is an instance of a more general process, called *beta reduction*, that is necessary to manage semantic expressions in the lambda calculus. We will cover this concept in greater detail in the tutorials.

To keep track of and resolve scope ambiguities, we can introduce another list: the *quantifier store* (represented by $\langle\rangle$). As shown in figure 4, having this option will allow us to generate parse trees for multiple readings. At (1), there is an option to store the quantifier in the quantifier store, and then we can retrieve it at the end (2).

[5] Although Chinese is an SVO (Subject-Verb-Object) language, there is a means of performing "double movement."

(1) 一个 学生 每 本 书 都 读过
yi ge xuesheng mei ben shu dou duguo
∃ *ge*-CL student ∀ *ben*-CL book *dou* read
A student every book read.

We will ignore these.

(a) (2 marks) Manually convert all readings of the sentences (29) and (30) to logical expressions. Put your logical forms in section 2(a) of `analysis.txt`. Use `exists` and `forall` for the quantifiers, and use => and the caret symbol ^ for implication and conjunction.

(b) (10 marks) Implement grammars for the syntax of quantifier scope ambiguity. You don't need to account for meanings, or for ambiguity in meanings (there should be no syntactic ambiguities). At this point, a correct grammar will produce exactly **one** parse for every grammatical sentence. Test your implementation before you move on to the next step.

(c) (10 marks) Augment your grammars to represent meaning and quantifier scope ambiguity. Marks for question 2(b) will be deducted if your work on this part causes errors in the syntactic predictions. Your grammar should generate more than one parse for each ambiguous sentence.

(d) (4 marks) Translate sentences (29) and (30), as you did in the first question.

Operational Instructions

- Use the function `translate` to generate semantic representations of your source sentences. If your sentences can be parsed, `translate` should open another `gralej` window and with all of the translation results.

  ```
  | ?- translate([a,student,read,every,book]).
  ```

- You will be prompted as follows to see the next parse.

  ```
  ANOTHER? y
  ...
  ANOTHER? y
  no
  ```

  Answer `y` to see the next parse until you reach the end. Each time TRALE will open a new `Gralej` window. You need to store all of your translation results by repeating the previous step. A `no` will be returned when you reach the end of your parses.

- Save your translations of sentence (29) as `q2d_29_1.grale`, `q2d_29_2.grale` ... and your translations of sentence (30) as `q2d_30_1.grale`, `q2d_30_2.grale` ...
- Submit a zip file `q2d.zip` containing all the translation results. You can use this command: `zip -r q2d.zip q2d_*.grale` to create the zip file.
- Again, don't forget to **close all the windows** and kill your `Gralej` processes after you finish.

(e) (4 marks) Again, compare your grammar-based translator with Google Translate. Report **at least one** instance of a difference between the translation given by your translator and Google Translate. Your analysis should be submitted as the section 2(e) in `analysis.txt`.

CSC 485H/2501H, Fall 2021: Assignment 3

Family name: _____

Given name: _____

Student #: _____

Date: _____

I declare that this assignment, both my paper and electronic submissions, is my own work, and is in accordance with the University of Toronto Code of Behaviour on Academic Matters and the Code of Student Conduct.

Signature: _____