Tagging and Statistically Translating Latin Sentences

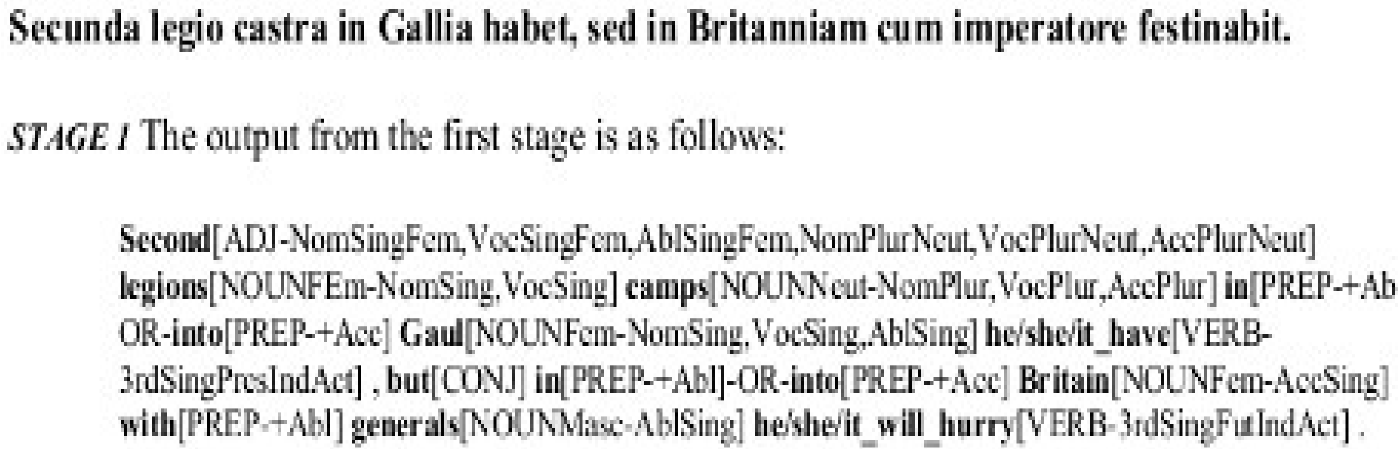
Andrew Runge
Computer Systems Lab 2009-2010

Abstract

In developing language translation software, an increasingly common method is to tag words based on their role in the sentence in order to determine where they should be in the sentence, and then put them in that slot to create a basic, sometimes awkward translation.This project works to create the intitial work for developing a fully automated translation. The program tags and translates the words in the sentence, allowing them to then be organized into the proper order via statistical translation methods.

Background

The biggest focus for language translation is to maintain the original meaning of the sentence when it is translated. As such, it is crucial not only to properly translate the words, but to maintain a sensible word order in order to preserve the original meaning. Machine learning methods, such as word tagging, allow the program to rule out possibilities for what the possible functions of a given word are. One such example used by McMahon and Smith was a method for determining the role of words in a sentence based on their context and similarities that they shared with other words. Another experiment by Bowden discussed a method of tagging the words for every possible set of characteristics they could have, and then systematically narrowing down the possibilities until you can more easily order the words based on their characteristics in Latin. This is a similar method to the one that I am employing. After identifying the words' roles in the sentence, then it is important to put them in the correct order via statistical analysis. Chen et al. demonstrate the effectiveness of statistical generation of sentence structure with their project using n-grams to create possible sentences. Another method of statistical analysis was demonstrated in Alam et al. In which they generated sentences based on the probability that two parts of speech could go together. Their program suffered largely due to the tagging and original translation methods that they used, but I intend to recreate the system they used for sorting the words into the correct order. Finally, to test the veracity of the program's translation, sometimes humans are integrated into the translation process in order to correct any mistaken translations, such as what was done in Barrachina et al. with Computer Assisted Translation.



Secunda legio castra in Gallia habet, sed in Britanniam cum imperatore festinabit.

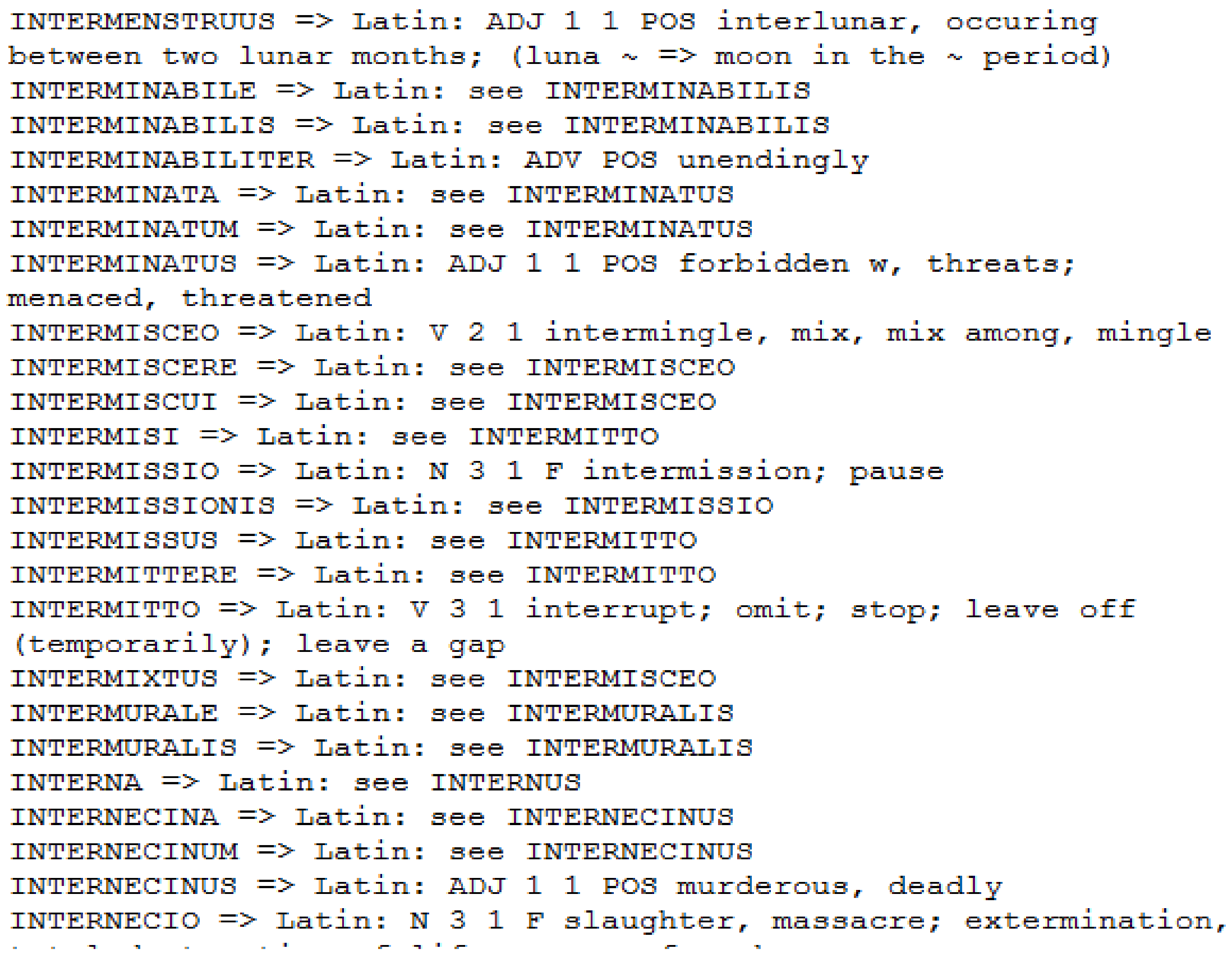
STAGE 1 The output from the first stage is as follows:

Second[ADJ-NomSingFem,VocSingFem,AblSingFem,NomPlurNeut,VocPlurNeut,AccPlurNeut] **legions**[NOUNFEm-NomSing,VocSing] **camps**[NOUNNeut-NomPlur,VocPlur,AccPlur] **in**[PREP-+Abl]-OR-**into**[PREP-+Acc] **Gaul**[NOUNFem-NomSing,VocSing,AblSing] **he/she/it_have**[VERB-3rdSingPresIndAct] , **but**[CONJ] **in**[PREP-+Abl]-OR-**into**[PREP-+Acc] **Britain**[NOUNFem-AccSing] **with**[PREP-+Abl] **generals**[NOUNMasc-AblSing] **he/she/it_will_hurry**[VERB-3rdSingFutIndAct] .

Figure 1: Sample of the method employed by Bowden to tag all words based on all their possible sets of characteristics.

Tagging

With Latin, it is very easy to identify various parts of speech. However, figuring out exactly what the translations are for those parts of speech can still be tricky. The purpose of tagging the words is to identify all the possibilities for how the word could be translated. This way, when it comes down to figuring out the exact meaning, there are a limited number of possibilities for what each word could be.

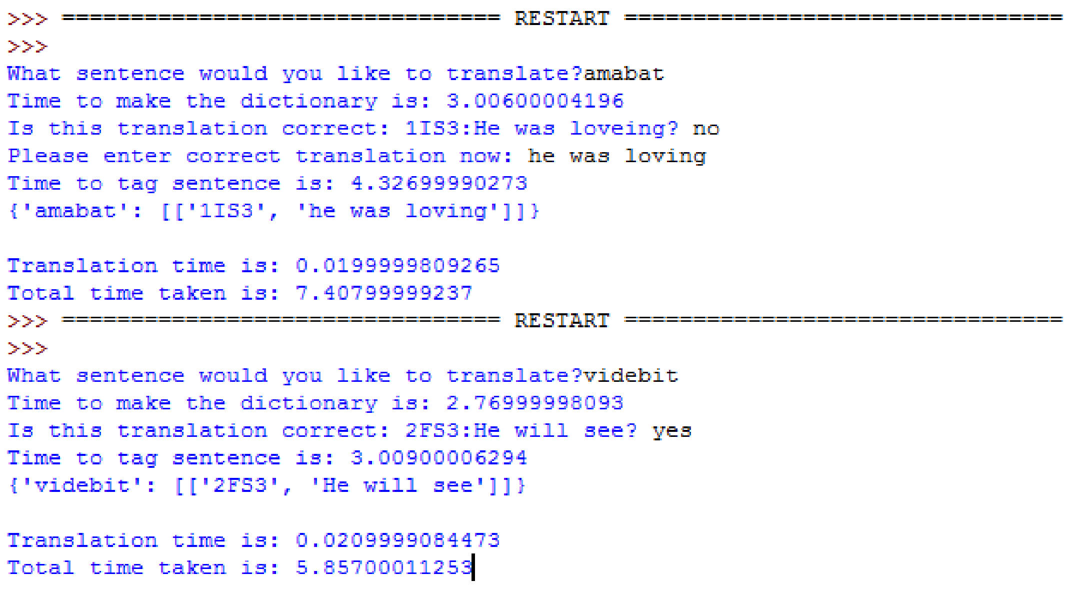


```
INTERMENSTRUUS => Latin: ADJ 1 1 POS interlunar, occuring
between two lunar months; (luna ~ => moon in the ~ period)
INTERMINABILE => Latin: see INTERMINABILIS
INTERMINABILIS => Latin: see INTERMINABILIS
INTERMINABILITER => Latin: ADV POS unendingly
INTERMINATA => Latin: see INTERMINATUS
INTERMINATUM => Latin: see INTERMINATUS
INTERMINATUS => Latin: ADJ 1 1 POS forbidden w, threats;
menaced, threatened
INTERMISCEO => Latin: V 2 1 intermingle, mix, mix among, mingle
INTERMISCERE => Latin: see INTERMISCEO
INTERMISCUI => Latin: see INTERMISCEO
INTERMISI => Latin: see INTERMITTO
INTERMISSIO => Latin: N 3 1 F intermission; pause
INTERMISSIONIS => Latin: see INTERMISSIO
INTERMISSUS => Latin: see INTERMITTO
INTERMITTERE => Latin: see INTERMITTO
INTERMITTO => Latin: V 3 1 interrupt; omit; stop; leave off
(temporarily); leave a gap
INTERMIXTUS => Latin: see INTERMISCEO
INTERMURALE => Latin: see INTERMURALIS
INTERMURALIS => Latin: see INTERMURALIS
INTERNA => Latin: see INTERNUS
INTERNECINA => Latin: see INTERNECINUS
INTERNECINUM => Latin: see INTERNECINUS
INTERNECINUS => Latin: ADJ 1 1 POS murderous, deadly
INTERNECIO => Latin: N 3 1 F slaughter, massacre; extermination,
```

Figure 2: Screenshot of the primary dictionary..

Translation

In translation, the program takes each of the possible tags and generates the translation for that set of characteristics. Once each translation has been created, the program asks for input from the user to determine whether the translation for the given word is correct or not. If it is, it will be recorded into a translation dictionary, so it can be easily accessed later. If it is not, it will replace the translation with the new user input one. This way, if the program does make some sort of mistake, then the user can correct it so as to maximize the accuracy of the output translation.



```
>>> ================================ RESTART ================================
>>>
What sentence would you like to translate?amabat
Time to make the dictionary is: 3.00600004196
Is this translation correct: 1IS3:He was loveing? no
Please enter correct translation now: he was loving
Time to tag sentence is: 4.32699990273
{'amabat': [['1IS3', 'he was loving']]}

Translation time is: 0.0199999809265
Total time taken is: 7.40799999237
>>> ================================ RESTART ================================
>>>
What sentence would you like to translate?videbit
Time to make the dictionary is: 2.76999998093
Is this translation correct: 2FS3:He will see? yes
Time to tag sentence is: 3.00900006294
{'videbit': [['2FS3', 'He will see']]}

Translation time is: 0.0209999084473
Total time taken is: 5.85700011253
```

Figure 3: Sample of code as it tags various forms of nouns and translates these forms and corrects errors.

Additional Work

The program is able to tag and translate single words. The next steps that should be taken for the improvement of this program would be to eliminate those tags which are grammatically impossible, such as a plural subject with a singular verb, as well as applying the statistical analysis on the sentence in order to generate multiple hypotheses for how the sentence could be translated.