

LIN 177 Final Project

Matthew Gans

The linguistic phenomenon modeled in this final project is the French number system. This model generates the numbers 1-1000. The French number system is a mixed-base system. The system is base 10 up until the number 69. From 80 to 99, the system changes to base 20. The numbers 70-79 are quite different in their creation than the previous two sets of numbers. The numbers 1-19 do not seem to follow a pattern in their creation and, as such, are lexically conditioned. The numbers 20-69 follow a similar style to English, in that there is a word for every multiple of 10 and numbers between multiples of 10 are created by the word for that multiple of 10 followed by the word for the number in the ones place. For example, the number 32 is created by combining the word “trente” (meaning 30) and the word “deux” (meaning 2). Thus, 32 is “trente-deux”. The only numbers that do not exactly follow this pattern between 20-69 are 21, 31, etc. The number 21 is created by combining the word “vingt” (meaning 20), the word “et” (meaning and), and the word “un” (meaning 1). Thus, 21 is “vingt-et-un”. The numbers 70-79 are created by taking the word “soixante” and adding the words for 10-19. So, the word for 70 is “soixante-dix”, which literally means “sixty-ten”. As stated prior, the numbers 80-99 switches to base 20. The word for 80 is “quatre-vingts” which literally means “four-twenty”. All numbers between 81-99 are created using “quatre-vingt” and then adding the words for 1-19 to the end. We have now reached 100, which has no relation to any previous number. The word for 100 is “cent”. Any number between 101-199 is created by taking “cent” and adding the word for the number comprising the tens and ones place. For example, 125 is “cent-vingt-cinq”. For numbers 200+ (up to 999), we add the word in the hundreds place to the front. Instead of 125, imagine we had 225. We know that 125 is “cent-vingt-cinq”. We also know that 2 is “deux”. So, 225 is “deux-cent-vingt-cinq”. The last number generated is 1000, which similarly to 100, has no relation to any previous number. The word for 1000 is “mille”.

I chose this topic to model because I have always been interested in mathematics. As a Computer Science major, I have taken plenty of math courses over my college career. Modeling a number system allows me to combine my interest in math with the linguistic nature of the class.

This topic, and the model of it, relates to the class material as it uses the IPA characters/sounds to lexically define the words that build up the number system. It also utilizes a form of affixation, in particular prefixation and suffixation. When creating new numbers, words are appended to the front or back of another word. Creating new numbers is recursive in nature. Using a previous example, the number 225 can be generated in the following manner: 1) begin with “vingt” (20), 2) append “cinq” to the end to get “vingt-cinq” (25), 3) append “cent” to the front to get “cent-vingt-cinq” (125), 4) append “deux” to the front to get “deux-cent-vingt-cinq” (225). A similar program is

found in the textbook, which models the Mandarin number system, but only generates numbers between 1-100.

The program is made up of numerous predicates to generate each different class of numbers. I included a predicate that calls each of the other predicates, as well as the numbers that were lexically conditioned. By querying “french(Number).”, all numbers between 1-1000 will be generated. Alternatively, all other predicates can be queried by using “french(Number, _).”, but this will not generate the lexically conditioned numbers.

The program generates all numbers between 1-1000, but I was not able to find a way to generate all numbers in order. This is due to the way some sets of numbers are generated by a separate predicate. Another problem (which does not affect the output, but is more a readability problem) is that I use plain letters in the categorical information to differentiate between sets of the lexically conditioned numbers, e.g. x. I would definitely like to change this, but I am unsure of a set of words that differentiates them in the way I want.

I believe that the model is principled. Each predicate generates a set of numbers and only that set. All numbers between 1-1000 are generated by some predicate in the model, or are lexically conditioned.