# Instructions for NAACL HLT 2013 Proceedings[*]

**Author 1**
XYZ Company
111 Anywhere Street
Mytown, NY 10000, USA
author1@xyz.org

**Author 2**
ABC University
900 Main Street
Ourcity, PQ, Canada A1A 1T2
author2@abc.ca

## 1 Introduction

Our project is to create a machine translator for the synthetic language Lojban. Lojban is a language derived from the language Loglan, which was developed by linguists starting in the 1950s. The goal of Lojban overall is to enhance communication with both other humans, as well as with computers. The language accomplishes this in many ways. First of all, every attempt is made in to remove all ambiguity from sentences, a notorious problem in many natural languages. For example, the English sentence They are hunting dogs. This sentence is syntactically ambiguous as it is unclear whether there are people hunting dogs or the dogs being referred to hunt. Second of all, the language has no exceptions to any rules, unlike any natural languages. While in English we have rules like I before E, except after C and these rules have weird exceptions. These two features make it both very hard and very easy to think of a formulaic way to go between Lojban and natural languages. On one hand, the rules of the language will always hold, and so once the rules have been coded, no exceptions will have to be accounted for. But on the other hand in Lojban the meaning of a sentence is unambiguous, which makes translation into a language with ambiguity difficult. Our goal with this project is not to provide one hundred percent accurate descriptions of all Lojban sentences into English, but to be able to take simpler Lojban sentences and translate into non-ambiguous English as closely as possible.

[*]

## 2 Lojban Structure

The core of the Lojban sentence is the bridi, which is simply a phrase that describes a relationship between two things. The simplest form of a bridi is one that consists of one selbri, a word that describes the relationship between things, and one or more gismu, which is simply a word which describes an object. For example, in english the equivalent of a simple bridi is Joe is the father of Jill. in Lojban is the father of would be considered the selbri, and joe, and jill would be gismu. The actual Lojban phrase for this is Joe patfu Jill It is important to note that in Lojban, because the selbri only describes the relationship of the gismu, the positioning of the selbri in or around the gismu is unimportant, the only thing that matters is the ordering of the gismu.

The unimportance of the positioning of the selbri inside the bridi seems like a negligible concern when programming for bridi that only contain a selbri and gismu, and indeed it is. Bridi do not have to contain only a selbri and gismu however; the positions that the gismu fill are called sumti, and are only required to describe some object, which can be done with another selbri. For example Joe is the father of that girl who talks to Jim we used the same selbri, and the first argument remains Joe but we replaced the second argument with that girl who talks to jim. The Lojban phrase for this is Joe patfu nixli tavla Jim, in this case it is unclear whether Joe is talking to Jim, or if Joes daughter is talking to Jim. In this lojban sentence, due to the fact that the left most selbri uses the closest sumti to it (unless otherwise indicated). This phrase actually means that Joe is talking to Jim, with Joe being the father of the girl. This example also shows how difficult it can be to avoid ambiguity when translating from Lojban to English.

These sentences can be modified to mean anything, using cmavo, which are nothing more than logical operators. for example, the example given on page 95 of The Complete Lojban Language is about co, which inverts tanru. Tanru are a grouping of a modifier and an object to be modified.

ta blanu zdani $\rightarrow$ that is-a-blue type-of-house $\rightarrow$ that is a blue house

ta zdani co blanu $\rightarrow$
that is-a-house of-type-blue $\rightarrow$ that is a blue house

This obviously makes is complicated to gather a fluid english translation if two Lojban phrases can mean the same thing, like the example above. Thankfully even if we cannot quite get to the acceptable English translation, the psuedo-English translation still makes sense to native speakers of English, and it is reflective of exactly what the Lojban phrase is. This was a rather simple example, but many cmavo are related to grouping tanru, and modifying the relational structure of selbri. They even make parsing more difficult because cmavo are found near selbri and between tanru where one would hope to find the related word, but instead what is found is an operator that modifies meaning without having any visible presence in the English translation.

Cmavo are among a long list of things that our group is not fully committed to implementing. While they are extremely important to the meaning of lojban sentences, and surely many ideas cannot be expressed without them, they pose a major parsing issue and there are other more important concepts of the language that need to be addressed, for example, selbri, gismu, and tanru, which can be used to express a massive amount of ideas without the complexity of cmavo added.

## 3 Resources

The lojban community has already created a translator similar to the goal of this project which can be found at (http://www.lojban.org/jboski/index.php). It takes the lojban sentence and parses it into its logical form, and the output given is that logical lojban directly translated into english. Observe when the input mi tavla do (i talk to you) is given, the response from the translator is [1(2[tavla1 (talk-er(s)) :] mi I, me)2 [is, does] 3tavla talk-ing3 (4[tavla2 (talked to thing(s)) :] do you)4]1 , This is an interesting translation for multiple reasons, first of all it leaves the lojban words inside of the translation, with their definitions attached to them. It also places parenthesis and other grouping symbols so that any meaning carried with groupings in the lojban is not lost in the english, and helps new users of lojban to understand how the language works.

While this translator is great for showing the inner workings of lojban, and keeping the logical grouping together, it does not function as a plain Lojban to English translator. To interpret these translations one must observe all of the possible translations of each word, for example, I/me, and the translator does not always include all of the correct articles, an equivalent sentence in english could be I am talking to you, but the am article is not included. Another issue with the translation is how cluttered it becomes with all of the grouping symbols, indeed when the translation first comes back it is unreadable without first deciphering where all of the grouping markers are and how they change the meaning. Overall, unless one desires to understand the details of how lojban conveys meaning, this translation is quite a burden. Nevertheless, this translator was very useful to us in testing our programs functionality.

## 4 Our project

Our teams goal is to take a similar design where perfect english is not returned, but the translation is still fully intelligible to a native English speaker. As opposed to the translator of lojban.org, however, we want to provide the user with output that does not have to be analyzed before they can gather any sort of meaning. The end goal of this project is to provide the user with output that can be immediately understood without the need to any deep analytical thought.

One of the obstacles for this project is the fact that none of the members of the team are considered experts on lojban, or even to be proficient in it. Therefore we will be relying on the book The Complete

Lojban Language by John Woldemar Cowan. This book is published by the Logical Language Group, and is considered the Lojban bible. Following the logical progression of the book, which teaches the reader the language of Lojban in a methodical way that starts with simple statements then fully fleshes out each of the complexities of the language with great detail we plan to implement general rules first, then add in more subtle and obscure rules.

For our implementation we are using the pyparsing package to parse and lexically analyze the lojban sentences. Pyparsing allows a programmer to define rules about how a sentence is parsed, and into what entities the sentence is parsed. For example, to parse a simple bridi sentence with three arguments we used four different rules to account for the four placements for the selbri. (Recall that a bridi with three arguments consists of three sumti and one selbri).

When we tried to parse one such bridi sentence (mi vecnu ta ti) we got a list with four elements– one for each word. When we used a sentence with a nested bridi (mi vecnu ta mi vecnu ta ti) we got a list with four elements, one of which was nested list. The nested list was the nested bridi (mi vecnu ta ti).

To use pyparsing, we had to create several rules or constraints to correctly parse the sentence. To do this, we used an extended Backus-Naur representation. Backus-Naur representations, originally invented by computer scientists John Backus and Peter Naur to represent non-ambiguous programming languages, are a useful construct in representing Lojban.

To get the vocabulary we parsed the text files from the Lojban website. This required a lot of work, because they were not very well formatted, had lots of extra info, and there was no consistent syntax they used throughout. For example, some lines had synonyms, some did not. We parsed these files into arrays of objects for later use.

## References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Association for Computing Machinery. 1983. *Computing Reviews*, 24(11):503–512.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.