

Práctico 2: Implementación de sistema basado en reglas

- Idea: armar un "toy system" para ver cómo funcionan y entender la tarea del lingüista/desarrollador
- Lingüistas usan mucho Prolog (PROgramming in LOGic) dado que permite crear reglas, hechos y relaciones de forma relativamente fácil
 - En PLN gramáticas libres de contexto, análisis morfológico, etc
- Hacer el sistema de 0 es mucho trabajo → reusemos código!

Qué hay que hacer?

- Este práctico se basa en el material de un curso de TA por D. Arnold, donde se provee el código Prolog de un sistema por transferencia e interlingua
- Para este práctico hay que:
 1. Modificar el código para que el sistema traduzca En-Sp o En-XX (XX no puede ser Japonés o Griego)
 2. Documentar el proceso (problemas, soluciones, opiniones, etc)
 3. Extender el sistema para que trate nuevos fenómenos lingüísticos (por ej phrasal verbs) o para que acepte mas vocabulario o para que sea multilingüe (En-XX-YY, XX o YY puede ser Japonés o Griego → podría traducir de Griego a Español? Y de Español a Griego?)
 4. Mandarme por mail sus resultados (código + pequeño informe) o alguna dirección desde donde descargarlo→ Todo esto en la medida de lo posible!
- La próxima clase discutimos cómo les fue

Recursos a utilizar

- Material del curso disponible en
 - <http://courses.essex.ac.uk/LG/LG619/mt/>
- Código disponible en
 - <http://courses.essex.ac.uk/LG/LG619/mt/code/>
 - Modificar archivos transfer.pl y il.pl
 - Puede que el navegador no reconozca el archivo → abrir la version .txt
- Prolog como herramienta para PLN
 - <http://www.bowdoin.edu/~allen/nlp/nlp2.html>
 - <http://www.ling.helsinki.fi/kit/2004k/ct1272/Bangor/clbook.html>
 - <http://www.informatics.sussex.ac.uk/research/groups/nlp/gazdar/nlp-in-prolog/index.html>