MACH - report

1 The target

The target is to write a program that, given a raw string of text, does two things:

1. Identifies and resolves named entities occurring in the text, along with semantic relations occurring between them
2. Transforms the tags into RDFA tags and returns HTML - RDFA encoded text.

2 Milestones, or how to reach the target

The idea is to write a program as modular as possible. We all agree that the tough step in the target above is going to be 1. So I will focus on that one, in this short report. A rough pipeline might look like the following:

(syntax) parse the dependency tree of the text. This is useful insofar as it gives very valuable clues about the semantic role of the parts of the sentences.

(extraction) obtain from the dependency trees the words/sequences of words that are most likely to denote entities (nodes) or relations (edges).

(resolution) defer to sub-modules the duty of guessing/determining what does each extracted name refer to. This should include:

(geonames) identify names for localities

(people) identify the people behind their names

(relations) this will guess the (not-yet-guessed) edges of the graph. Should be done last, for a geoname being a geoname is in itself a clue of the semantic relations that it might be entertaining with other entities.

3 Progress and difficulties

The first try was to check the availability of off-the-shelf named entity resolution systems. To see what I have tried so far, you need to have installed Dupira (a Dutch parser), Freebase and CLAVIN-Rest's Python3 api.

Then:

```
from mach.utils.preprocess import Structure
import mach
struct = Structure('./mach/data/feed/article1777.txt')
mach.main.run(struct)
```

this will create a Structure (a wrapper class for the text and the tags), and tag it with the full available pipeline. At this moment, that means CLAVIN and Freebase.

I encountered several problems.

- the Dupira parser for Dutch sometimes just fails to parse sentences, such as "*Drie jaar krimp Italië blijft in een recessie.*" A new parser has to be tried.
- the CLAVIN geoparsing service relies on a Stanford NERF which is trained on English text. This means that it works really well (has won a couple of awards) but not on Dutch text. No proper off-the-shelf Dutch api seems to be available for this purpose: making one should not be too difficult.
- the Freebase api returns some interesting results, but is terribly imprecise. The point is that it will try to match *any* word to a named entity, so entering "*hier*" will return as top result 'Haier', in the category 'Consumentenelektronica'. Clearly not useful. However doing some selection yielded more promising results: depending on the syntax NOUN category identifier and on whether the word was capitalised, a threshold secured that only certain words be passed to Freebase.

At the present state of the system, most entities go unnoticed. "*Duitsland*" is not recognized, but "*Duitsers*" is tagged correctly:

```
'freebase_id=/en/germany',
'freebase_id=/m/0345h'
```

My guess is that the approach is promising, but many things need improvement. A few points:

- *Use more the syntactic information.* Tree parsing is expensive, but extremely powerful: use it more, or let it go!
- Make a custom powerful entity extractor, optimized for Dutch. *Tertium non datur*, in this case.

- Freebase has an huge ontology, and I think we really need one. Though this might not be the best one (in terms of precision).

[Pietro Pasotti, Amsterdam 2014]