>>> Teaching computers to understand politicians
>>> *or: Semantic Analyses of Swedish Parliamentary Data*

Name: Stian Rødven Eide[†]
Date: May 23, 2019

Språk-
BANKEN

[†]stian.rodven.eide@svenska.gu.se

>>> Swedish parliamentary data

* The Swedish parliamentary data is available from
  * Riksdagen, the Swedish parliament
    HTML encapsuled in XML/JSON/SQL or plain text
  * Språkbanken, University of Gothenburg
    Linguistically annotated XML
* Some things are not in Språkbanken
  * Ledamöter (data on MPs)
  * Voteringar (voting data)
  * Anföranden (speeches - with extensive metadata)

>>> Språkbanken's XML

```
<corpus id="rd-prot">
  <dokument (document metadata)>
    <paragraph>
      <sentence>
        <ne (named entity, time -
             when identified in the text)>
          <w (linguistic annotation)>
```

>>> My work on parliamentary data

* Semantic analyses, including
    * Named entity recognition/resolution (NER)
    * Argumentation mining (AM)
    * Sentiment analysis (SA)
* Also in a diachronic perspective, tracking changes over time

>>> My mostly used sources

* Anföranden (speeches)
    * Contains the same text as the protocols, but with extensive metadata
* Ledamöter (MPs)
    * Provides additional metadata for NER
* Voteringar (votes)
    * Basic feature for SA, useful for AM
* Neither of these are in Språkbanken, but can easily be annotated through the Sparv API (annotation pipeline)

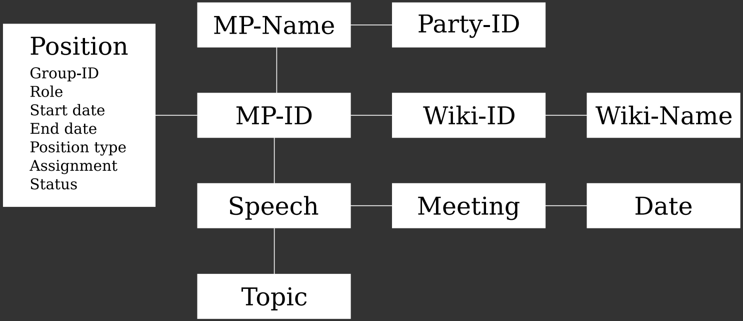
>>> A typical speech document

Property	Description
dok_hangar_id	Internal document ID
dok_id	Meeting + speech no.
dok_titel	Protocol title
dok_rm	Parliamentary year
dok_nummer	Number of meeting
dok_datum	Date of speech
avsnittsrubrik	Topic title
underrubrik	Topic subtitle
kammaraktivitet	Type of debate
anforande_id	Unique speech ID
anforande_nummer	Speech number in debate
talare	Speaker name
parti	Speaker party
anforandetext	Full speech text
intressent_id	Speaker's ID
rel_dok_id	Document being debated
replik	Speech type
systemdatum	Date of publishing

>>> Python and Prolog

* I work with Python and Prolog, using JSON from Riksdagen
* JSON can trivially be converted to a Python dictionary
* Prolog for linking and querying data
    * A speech may be replying to a person without using a name
    * With linked data, this can more easily be resolved
    * Also, additional metadata may provide useful features for machine learning

>>> A semantic graph of MPs and debates



Position
Group-ID
Role
Start date
End date
Position type
Assignment
Status
MP-Name
Party-ID
MP-ID
Wiki-ID
Wiki-Name
Speech
Meeting
Date
Topic