Named Entity Recognition and Morphosyntactic Tagging with Conditional Random Fields

Jakub Waszczuk
Institute of Computer Science, Polish Academy of Sciences

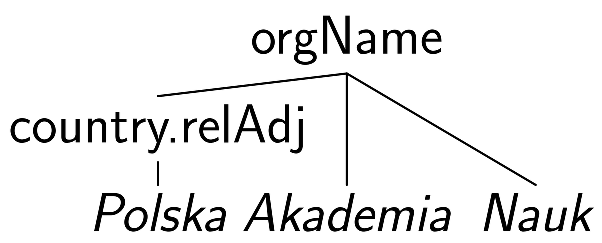
NERF

Introduction

- Both tools are implemented in a **Haskell** programming language, which combines advantages of high-level programming and type safety with excellent performance of generated programs (GHC compiler).
- Highly **modular design** – individual components are implemented as separate Cabal packages (Cabal is a package managment system for Haskell). Packages will be released via a **Hackage**, the public repository for Haskell libraries.
- All libraries developed under the hood of the NERF system or CRF tagger are/will be available under the **BSD** license.

Linear model

- **Extended IOB** encoding method serves to represent tree-like NE structures with label sequences.



Polska	B_country.relAdj#B_orgName
Akademia	I_orgName
Nauk	I_orgName

- **First-order linear conditional random field** (CRF) is used to model label sequences. Each label is treated as an atomic entity. The linear CRF is implemented as a stand-alone library and distributed as a Cabal package.
- A separate **library for observation extraction** is beeing developed. It can be used together with a user-defined observation schema as a source of input information for the CRF modeling toolkit.

Approximate NE searching

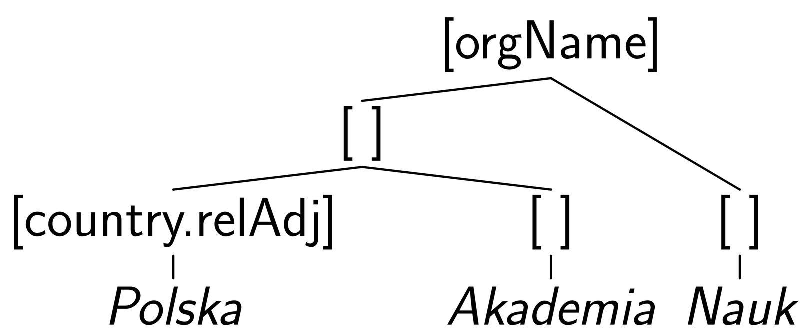
- **Position- and character-dependent cost function**, which can be specified by a library user.

```
data Cost a = Cost
    { insert :: Pos -> a -> Weight
    , delete :: Pos -> a -> Weight
    , subst  :: Pos -> a -> a -> Weight }
```

- General purpose **approximate dictionary searching library** parameterized over character type.
- **Depth-first search** on a **Trie** – all entries with edit distance lower than the threshold are returned.
- **Shortest-path search** on a **Directed Acyclic Word Graph** (DAWG) with explicit node identifiers. Only the nearest (with respect to the edit distance) dictionary entry is returned.

Tree model (in preparation)

NEs are represented as a **forest of independent binary trees**, where tree nodes keep information about NE types. The actual structure of NEs has to be binarized.



A **CRF-PCFG** method is used to model NE trees. The method is modified to incorporate additional **Boolean cut-off function** δ which can potentialy reduce the size of the search space.

$$T_i^j(x) = \begin{cases} \{Leaf(x,i)\} & \text{if } i = j \\ \{Node(x, t_l, t_r) : (x', y, z) \in R, \\ \quad x = x', k \in \{i, .., j-1\}, \\ \quad \delta(i,k,y),\ \delta(k+1,j,z), \\ \quad t_l \in T_i^k(y),\ t_r \in T_{k+1}^j(z)\} & \text{if } i < j \end{cases}$$

- The cut-off function is equally important for parameter estimation as it is for NER.
- By means of the cut-off function heuristics like **greedy search** can be represented.
- **External knowledge** (e.g. dictionary of NEs) also can be exploited via the cut-off function. For example, the dictionary can serve as an indicator of where NEs are allowed to appear.
- Preliminary version of the tree model has been implemented. Correctness of algorithms has been tested using the **Quickcheck** library.

Constrained CRF Tagger

Constrained CRFs

To utilize **morphosyntactic analysis results** we modify the basic definition of the linear CRF model:

$$p_\theta(\mathbf{y}|\mathbf{x},\mathbf{r}) = \begin{cases} Z_\theta(\mathbf{x},\mathbf{r})^{-1} \prod_{i=1}^n \phi_\theta(x_i, y_i, y_{i-1}) & \text{if } \mathbf{y} \in \prod_i r_i \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{x}$ is an input sentence, $\mathbf{y}$ is a sequence of output labels (e.g. morphosyntactic tags), ϕ is a potential defined with respect to a particular sentence position, Z is a normalization factor, θ is a set of model parameters and, finally, $\mathbf{r}$ is a **sequence of restrictions (potential morphosyntactic interpretations) for individual words**. This change alone (which has to be taken into consideration throughout the entire implementation, though) results in a **significant speed-up** of the CRF model training and morphosyntactic disambiguation.

Guessing

Marginal probabilities determined with respect to the first-order constrained CRF model are used to **guess potential morphosyntactic interpretations** of unknown words.

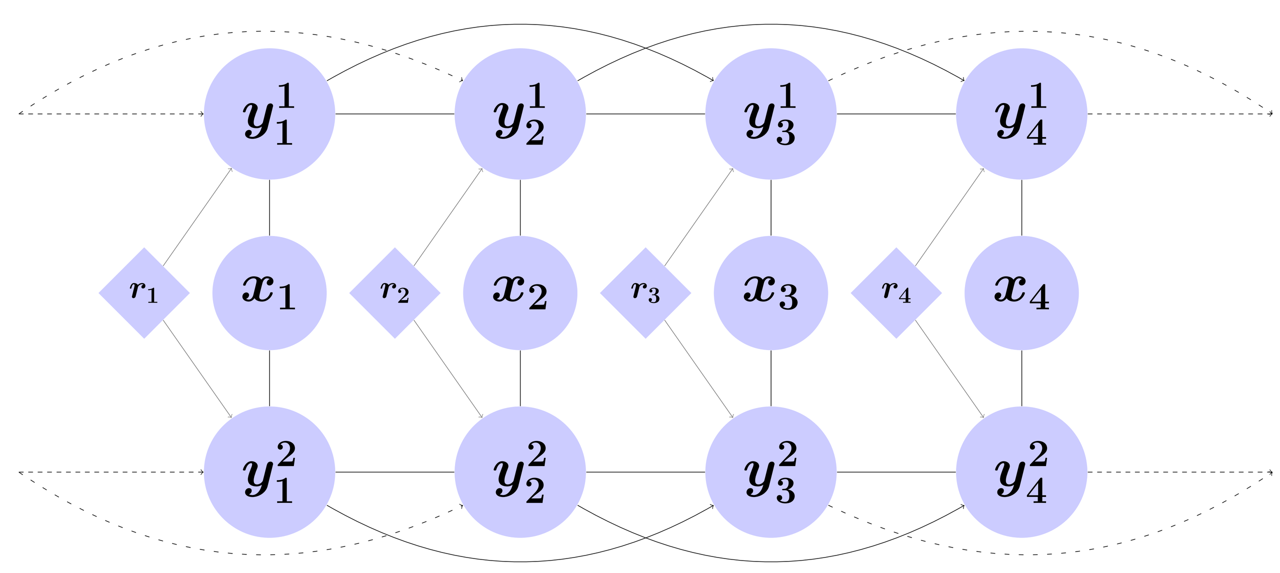
Observation schema:

- Prefixes and suffixes of lengths 1 and 2,
- A Boolean value indicating if the word is known,
- Packed shape of the word and information whether the word is positioned at the beginning of the sentence combined into one observation.

Szef	{ subst:sg:nom:m1 }
administracji	{ subst:sg:gen:f , subst:sg:dat:f , subst:sg:loc:f , subst:pl:gen:f }
Wołodymyr	$U \to$ { subst:sg:nom:m2, subst:sg:nom:n , subst:pl:nom:m1, ... }
Łatwyn	$U \to$ { subst:sg:nom:m2, subst:sg:gen:m1 , subst:sg:nom:m1, ... }

Disambiguation

Second-order, constrained and **layered** CRF is used for disambiguation. Morphosyntactic tags are divided between separate layers (example with two layers, y_i^1 and y_i^2, is shown below) according to a **user-defined configuration**. Labels in individual layers are treated as atomic entities.



Observation schema consists of lowered orthographic words at positions $i-1$, i and $i+1$ for each position i associated with a known word. For unknown words additional set of observation types is included:

- Lowered prefixes of length 1, 2 and 3 of the current word,
- Lowered suffixes of length 1, 2 and 3 of the current word,
- Packed shape of the word and information, whether the word is positioned at the beginning of the sentence, combined into a one observation.

Evaluation and comparison

Evaluation of the tagging system (guessing + disambiguation) has been performed on the **one-million, balanced National Corpus of Polish subcorpus** (NCP). It involved obligatory resegmentation (sentence splitting and tokenization) and reanalysis of the evaluation part. All tools have been evaluated on the **same extract of the NCP corpus**, and with respect to **exactly the same corpus partitioning**.

Tagger	Acc_{lower}	Acc_{upper}	Acc^K_{lower}	Acc^U_{lower}
Pantera	88.99%	89.28%	91.27%	14.74%
WMBT	89.71%	90.04%	91.20%	41.45%
WCRFT	90.34%	90.67%	91.89%	40.13%
Constrained	**91.12%**	**91.44%**	**92.10%**	**59.19%**

Table: Average accuracy measures obtained by individual taggers during the 10-fold cross validation on the NCP corpus.