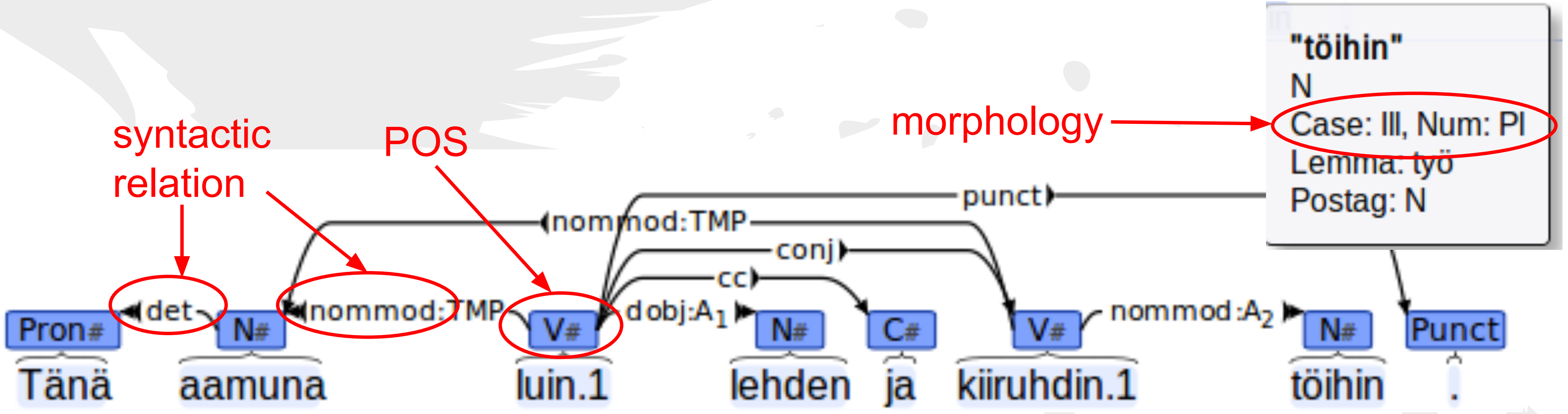
Seeding the Finnish Internet ParseBank with CommonCrawl: An Experience Report

Filip Ginter & Juhani Luotolahti
Department of Information Technology
University of Turku, Finland
Turku NLP group
bionlp.utu.fi   www.evexdb.org

The Finnish Internet ParseBank

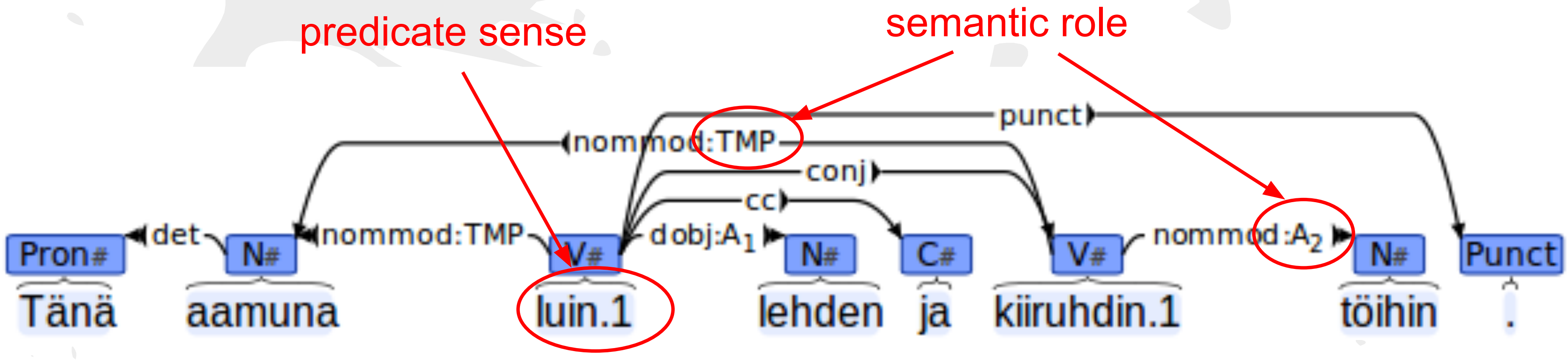
- **DATA:** Get *everything* we can find on the Internet in Finnish
- Morphosyntactic parse + semantic role labeling
- Clustering into (hopefully) linguisticially interesting sub-co rpora
- **WHY:**
  - Why not! :D
  - Resource for linguistic research
  - Resource for statistical NLP

Syntactic analysis of Finnish



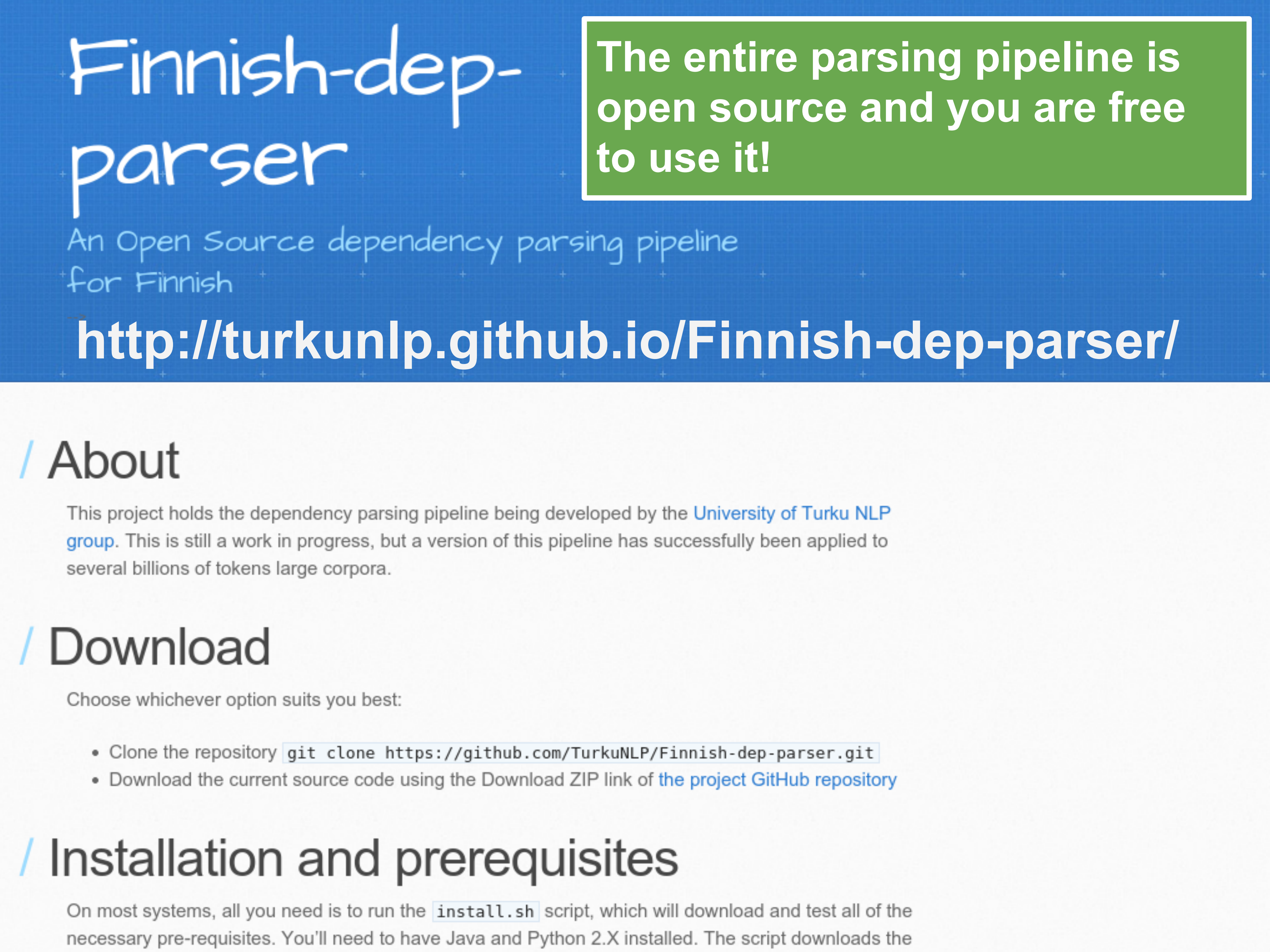
Semantic role labeling

Who **did** ***what*** **to** ***whom*****,** ***when*****, and** ***why*****?**
(semantic role labeling)



The entire parsing pipeline is open source and you are free to use it!

http://turkunlp.github.io/Finnish-dep-parser/



Finnish-dep-parser

An Open Source dependency parsing pipeline for Finnish

/ About

This project holds the dependency parsing pipeline being developed by the University of Turku NLP group. This is still a work in progress, but a version of this pipeline has successfully been applied to several billions of tokens large corpora.

/ Download

Choose whichever option suits you best:

- Clone the repository `git clone https://github.com/TurkuNLP/Finnish-dep-parser.git`
- Download the current source code using the Download ZIP link of the project GitHub repository

/ Installation and prerequisites

On most systems, all you need is to run the `install.sh` script, which will download and test all of the necessary pre-requisites. You'll need to have Java and Python 2.X installed. The script downloads the

CommonCrawl enters the picture

- [Initial project idea dates back to 2012]
- Didn’t want to restrict the crawl to the .fi domain only - and didn’t want to run our own crawl
- 10 minutes on Google brings us to the CommonCrawl site
- The obvious way to proceed: run language recognition on the CommonCrawl data, grab only Finnish, process

Getting the data: Starting point

- We have very good computational resources available to us, and we don't pay for them
- EC2 compute time costs, and full parsing is a computationally intensive operation
  - 8000 CPU hours on the last run
- ~~Data download from EC2 costs~~
  - ~~Prohibitive to download the whole CC~~
- ~~Good compromise: run language recognition on EC2 and only download the Finnish part~~

Not true - as we learned. There are no egress fees for CC data.

Getting the data: Starting point

- Lots of processing power at the Centre for Scientific Computing (CSC) in Finland
  - Traditional cluster machine with a batch job system
- Solid in-group experience with running complex NLP on large datasets at CSC (www.evexdb.org)
- Zero experience with Amazon EC2, zero experience with Hadoop (not at CSC)

Getting the data (cont.)

- The process reads plain text Common Crawl data from Amazon Public Datasets
  - Stored as key-value pairs in hadoop sequence files
- Language is checked on the first 400 bytes of each plain text document
- Pages detected as Finnish are uploaded in regular intervals to CSC for parsing

Amazon EC2

- Amazon Elastic Compute Cloud Instance
  - Can access Amazon public datasets
  - Fully functional box and easy to setup
  - m1.medium instance used for text gathering
    - 1 CPU/2 CU 3.75 GB memory $0.070/hour
    - Outbound data transfer also charged
      - $0.12/GB

Getting the data (cont.)

- Running on a single amazon EC2 instance
- A single Python process
  - Uses boto library to read the S3 filesystem
  - Hadoop library to read hadoop sequence files
  - Chromium compact language detection for language detection
    - FAST! The language detection is the most intensive part of the pipeline
- The process took about a month to run
- [We could have gotten the data much faster with a real map/reduce job]

RTFM

- Later we noticed EC offered a High-CPU Medium instance for almost the same price as the medium instance we used
  - D’oh!
- C3 Large instance
  - 2 CPU / 5 CU1.7 GB memory $0.130/hour
- Used later when grabbing the HTML sources

Deduplication and Filtering

- The plain text contains a lot of lists, menus, product catalogues, and the like
- Since we are interested in sentence-structured text suitable for parsing, the data must be filtered

- Also: Web data contains a lot of duplicate material
- For our purposes it must be deduplicated

Filtering the Text

- Filtering is done on a line by line basis
- Based on features such as:
  - token count
  - tokens recognized as Finnish by a morphological analyser
  - special character count
  - numerical character count
  - whether line starts with an uppercase token
- Potential lines concatenated in blocks
- ...after that the parsing pipeline splits the text into sentences, etc...

Deduplicating

- Done on document level
- After sentence splitting, every sentence is hashed
- If a document contains more than 90 % sentences seen previously, it is discarded

What we got in the end

Clean text good for parsing:

- 1.5 billion tokens
- 116 million sentences
- ~4 million urls

- Roughly 65 % of the originally gathered data was discarded

Next

- [bear in mind: we did the CC processing 2 years ago]
- 1.5B tokens was <u>way</u> less than we expected / hoped
- We started an in-house crawl with all .fi domains and all CC finnish pages as seeds
- That crawl still runs: the first batch of 3.2B tokens parsed few weeks ago
- We will run the CC job again in 2015

Syntactic structure search

- We have hundreds of millions of parse trees - now what?
- A real query:

  “Can you find me all verbs that have an object and a subject. The subject must be a noun in the partitive case and it must not have a numerical modifier, unless that modifier is in the partitive case as well. Oh, and the verb must not be the head of a complement clause. Thanks!”

Syntactic structure search

- Complex query with negations
- As of last Friday:
  - 40M trees searched, 500M tokens, single PC with 128GB of memory (60GB free when running the numbers below)
  - Data in OS cache: **20 seconds**, 17K hits
  - Data not in OS cache (cold run): ~10min
- We hope this will evolve into a general complex syntax query system on top of treebanks and big parsebanks
  - Everything designed to support parse graphs (not trees)

Partitivegate

- Disaster hits!
- When browsing through the results, we discover we have a perfect list of parser errors :(
- Why?
    - Subject/object distinction not trivial in Finnish
    - Swapping subject<->object a common parsing mistake if both in partitive
- What now?
    - Hope distributional semantics methods will help us re-rank the results and find the most likely correct

Partitivegate

- After re-ranking with a method based on vector space embeddings (word2vec), we found one real example yesterday night
- "...ihmisiä siemaili samppanjaa..."
- Not exactly a happy end, but getting there :)

Released by the project so far

http://bionlp.utu.fi/finnish-internet-parsebank.html

- 5-grams
- Google-style syntactic n-grams
- Vector Space Embeddings (word2vec)
- NoSketchEngine with a sample of the data

  http://bio3-ett.utu.fi/nse/

  username: guest

  password: voikukka

Recap - CommonCrawl

- Fast start, lots of data quickly
- At least in the 2011 version, coverage of Finnish not that great - ended up running own crawl
- Language detection on EC and subsequent processing locally a good, cheap option
  - Maybe it's in the newest CC metadata?
- Good seed for own language-specific crawl

Recap (cont.)

- Technically, we possibly could have gotten much better mileage if we learned how to use EC2 properly
  - Spot instances, etc...
  - Then again, using days of work to save two hundred bucks is not super-efficient either
  - We don’t want to use EC2
- Parsing quite intense CPU-wise, last round about 8K CPU-hours

Recap (cont.)

- Finding rare, linguistically interesting phenomena is not going to become trivial just because we have lots of data
  - Hardly all get masked by a common parser error the way the Finnish partitive subjects do, though
- Complex search in syntactically parsed corpora needs specialized tools
  - Those we know do not scale
  - Now we have our own and will make it available