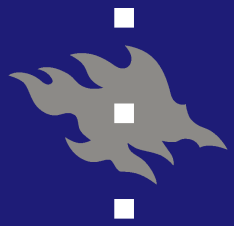


HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Extracting semantic frames with hand-written rules

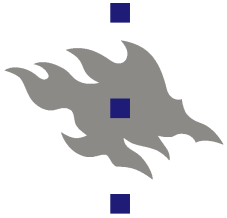
Krister Lindén, Sam Hardwick,
Miikka Silfverberg, Erik Axelson

{krister.linden, erik.axelson}@helsinki.fi,

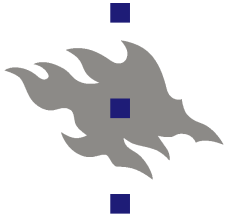
{sam.hardwick, miikka.silfverberg}@iki.fi

September 17, 2015

SFCM

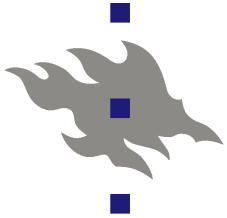


Frame semantics as a theory of meaning:



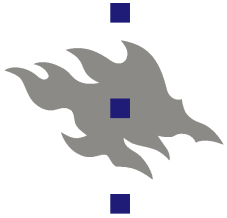
Frame semantics as a theory of meaning:

- Meanings contain information about the concepts they refer to (eg. the concept of *buying* implies two agentive participants, a buyer and a seller)



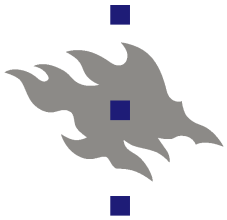
Frame semantics as a theory of meaning:

- Meanings contain information about the concepts they refer to (eg. the concept of *buying* implies two agentive participants, a buyer and a seller)
- The appearance of a concept-carrying word activates (or “evokes”) the frame of associated participants (the verb “buy” requires a direct object, the meaning “buy” requires an object of transaction)



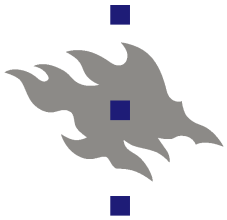
Frame semantics as a theory of meaning:

- Meanings contain information about the concepts they refer to (eg. the concept of *buying* implies two agentive participants, a buyer and a seller)
- The appearance of a concept-carrying word activates (or “evokes”) the frame of associated participants (the verb “buy” requires a direct object, the meaning “buy” requires an object of transaction)
- Frames can disambiguate meanings (the absence of compulsory elements or the presence of impossible elements carries information about the meaning)



A *semantic frame* is kind of like a pattern:

$$\left[_{\text{Size}}\ \left[_{\text{Entity}}\ \text{He}\right]\ \text{is}\ \left[_{\text{Degree}}\ \text{quite}\right]\ \left[_{\text{Lex-unit}}\ \text{tall}\right]\ \left[_{\text{Standard}}\ \text{for a jockey}\right]\right]$$

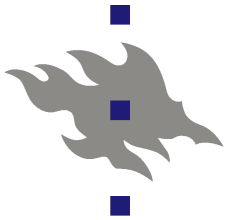


A *semantic frame* is kind of like a pattern:

$$\left[_{\text{Size}} \left[_{\text{Entity}} \text{He}\right] \text{is} \left[_{\text{Degree}} \text{quite}\right] \left[_{\text{Lex-unit}} \text{tall}\right] \left[_{\text{Standard}} \text{for a jockey}\right]\right]$$

We can write appropriate matching sub-expressions which are defined by syntactic expressions:

```
# Frame
define SizeFrame Entity Copula (Degree) LexUnit (Standard);
```



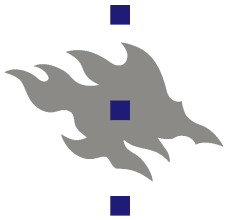
A *semantic frame* is kind of like a pattern:

$$[_{\text{Size}}\,[_{\text{Entity}}\,\text{He}]\,\text{is}\,[_{\text{Degree}}\,\text{quite}]\,[_{\text{Lex-unit}}\,\text{tall}]\,[_{\text{Standard}}\,\text{for a jockey}]]$$

We can write appropriate matching sub-expressions which are defined by syntactic expressions:

```
# Frame
define SizeFrame Entity Copula (Degree) LexUnit (Standard);


# Frame elements
define Entity Pronoun | NP EndTag(Entity);
define Degree AdverbialPhrase EndTag(Degree);
```



A *semantic frame* is kind of like a pattern:

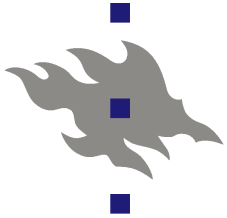
$$\left[_{\text{Size}} \left[_{\text{Entity}} \text{He}\right] \text{is} \left[_{\text{Degree}} \text{quite}\right] \left[_{\text{Lex-unit}} \text{tall}\right] \left[_{\text{Standard}} \text{for a jockey}\right]\right]$$

We can write appropriate matching sub-expressions which are defined by syntactic expressions:

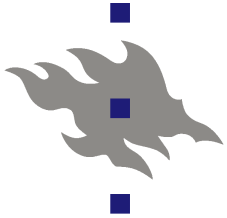
```
# Frame
define SizeFrame Entity Copula (Degree) LexUnit (Standard);


# Frame elements
define Entity Pronoun | NP EndTag(Entity);
define Degree AdverbialPhrase EndTag(Degree);


# Syntax and word classes
define NP Noun (CoordNP) (Of ([Det]) (AdjP) Ins(NP));
define Copula word ["<VBB>" | "<VBZ>"];
```

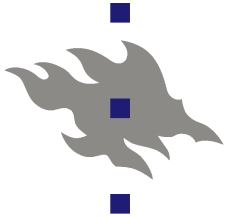


Are these just regexps?



Are these just regexps?

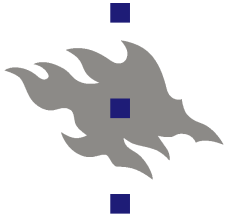
This is our open-source implementation of the `pmatch` concept introduced by Lauri Karttunen in SFCM 2011



Are these just regexps?

This is our open-source implementation of the `pmatch` concept introduced by Lauri Karttunen in SFCM 2011

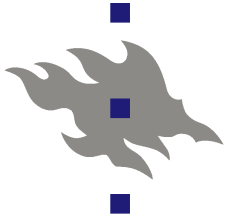
- RTNs, allowing context-free grammars and space efficiency



Are these just regexps?

This is our open-source implementation of the `pmatch` concept introduced by Lauri Karttunen in SFCM 2011

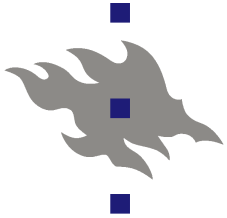
- RTNs, allowing context-free grammars and space efficiency
- With runtime contextual constraints



Are these just regexps?

This is our open-source implementation of the `pmatch` concept introduced by Lauri Karttunen in SFCM 2011

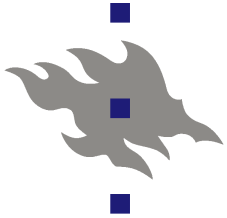
- RTNs, allowing context-free grammars and space efficiency
- With runtime contextual constraints
- Simultaneous parsing of rules with shared prefixes



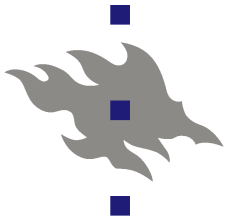
Are these just regexps?

This is our open-source implementation of the `pmatch` concept introduced by Lauri Karttunen in SFCM 2011

- RTNs, allowing context-free grammars and space efficiency
- With runtime contextual constraints
- Simultaneous parsing of rules with shared prefixes
- And weights!



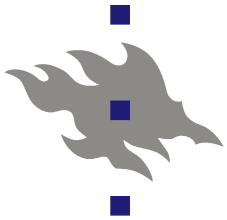
We used this approach in 2013-2014 to develop named entity recognition, in one case converting a pre-existing rule-based recogniser for Swedish and in another writing one for Finnish.



Motivation

Why rules?

State-of-the-art results using pure ML in semantic extraction aren’t that great

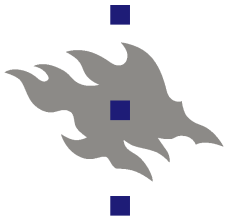


Motivation

Why rules?

State-of-the-art results using pure ML in semantic extraction aren’t that great

Existing methods don’t generally yield to fine-tuning by a user



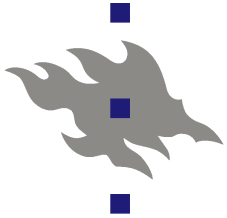
Motivation

Why rules?

State-of-the-art results using pure ML in semantic extraction aren't that great

Existing methods don't generally yield to fine-tuning by a user

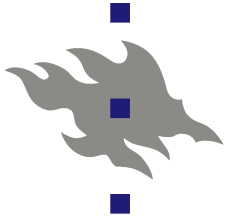
There isn't enough tagged text for most languages (eg. Finnish), or for most semantic frames in English



As a demonstration of the concept, we selected one FrameNet frame, *size*, and wrote an extractor for it. The rules were written in one day.

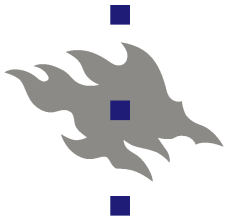
Raw text was first tokenized using a separate pattern-matcher and given morphological tags (in the case of English, POS tags) using a morphological transducer, and disambiguated with FinnPos.

```
# tokenize and perform morphological analysis
hfst-proc2 tokenizer |
# discriminative tagger, disambiguates morphology
finnpos-label model |
# tag the semantic frames
hfst-pmatch frame_tagger
```

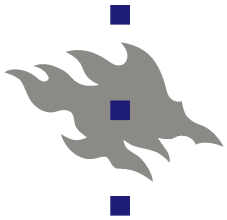


Frame extracting rules were on five levels:

- Literal words (eg. *and, a, the*)
- Word classes derived from the POS tagger (eg. *Copula, PersonalPronoun*) and the FrameNet lexical definition
- Context-free surface syntax (eg. *NounChunk, ForAPhrase*)
- Realisations of frame elements (eg. *Entity, Degree*)
- Orderings and optionality of elements (eg. *Size1, Size2*)



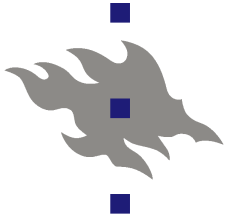
Proper evaluation was a challenge, because FrameNet's annotations are inconsistent and sparse



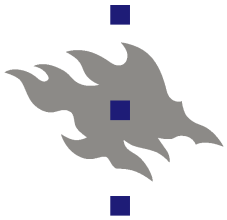
Proper evaluation was a challenge, because FrameNet's annotations are inconsistent and sparse

Decent proto-evaluation results: top-level coverage 89% and accuracy of 93% (of just the frame), all-element accuracy 70%.

About half of the errors were due to the extraction rules, half due to mistakes in eg. POS tagging.



Take-away: with a good morphological analyzer and disambiguator it's possible to write a decent custom semantic frame tagger in a short amount of time.



`+hfst-pmatch2fst+`(the compiler) and `hfst-pmatch` (the matching tool) are available on `hfst.sf.net`, though we're moving to `github.com/hfst` in the future