Extracting Semantic Frames using hfst-pmatch

Sam Hardwick
University of Helsinki

Miikka Silfverberg
University of Helsinki

Krister Lindén
University of Helsinki

{sam.hardwick, miikka.silfverberg}@iki.fi, krister.linden@helsinki.fi

Abstract

We use `hfst-pmatch` (Lindén et al., 2013), a pattern-matching tool mimicking and extending Xerox `fst` (Karttunen, 2011), for demonstrating how to develop a semantic frame extractor. We select a FrameNet (Baker et al., 1998) frame and write shallowly syntactic pattern-matching rules based on part-of-speech information and morphology from either a morphological automaton or tagged text.

1 Introduction

`pmatch` is a pattern-matching operation for text based on regular expressions. It uses a context-free grammar on regular expression terminals, which allows for recursive, self-referencing rules which would not be possible in a fully regular formalism. The matched patterns may be efficiently tagged, extracted and modified by the rules.

Large-scale named-entity recognisers (NERs) have been developed in `pmatch` for Swedish and Finnish. Here we demonstrate a bottom-up approach to using it to identify the frame "Size" in FrameNet.

2 A Semantic Frame

A semantic frame (Fillmore, 1976) is a description of a *type* of event, relation or entity and related participants. For example, in FrameNet, a database of semantic frames, the description of an `Entity` in terms of physical space occupied by it is an instance of the semantic frame `Size`. The frame is evoked by a lexical unit (LU), also known as a frame evoking element (FEE), which is a word (in this case an adjective) such as "big" or "tiny", descriptive of the size of the `Entity`. Apart from `Entity`, which is a core or compulsory element, the frame may identify a `Degree` to which the `Entity` deviates from the norm ("a **really** big dog") and a `Standard` to which it is compared ("tall **for a jockey**").

Lexical Unit (LU)	Adjective describing magnitude (large, tiny, ...)
Entity (E)	That which is being described (house, debt, ...)
Degree (D), optional	Intensity or extent of description (really, quite, ...)
Standard (S), optional	A point of comparison (for a jockey, ...)

Table 1: The semantic frame *Size*.

For example:

$$\Big[_{\text{Size}} \Big[_{\text{E}} \text{He}\Big] \text{ is } \Big[_{\text{D}} \text{quite}\Big] \Big[_{\text{LU}} \text{tall}\Big] \Big[_{\text{S}} \text{for a jockey}\Big]\Big]$$

Figure 1: A tagged example of *Size*

3 A Rule

A `pmatch` ruleset consists of a number of named regular expressions and functions, exactly one of which is the top-level rule which is named `TOP` or is introduced with the directive `regex`. For example:

```
define my_colours {green} | {red};
define TOP my_colours EndTag(colour);
```

Listing 1: Introducing `pmatch` syntax

The effect of the directive `EndTag()` is to tag whatever is matched by its rule (here shown with an unintentional effect):

```
The light went <colour>green</colour>
and the mechanism was
trigge<colour>red</colour>.
```

To avoid tagging the “red” at the end of “triggered”, we need to add a word boundary to the rule. This could be accomplished by defining eg. W to be whitespace (Whitespace), punctuation (Punct) or the limit of input (#). W may then be interpolated in rules when we want to capture whitespace inside the pattern, or checked for with run-time context checking just to make sure there is a word boundary at the edge of our rule (LC() and RC() check left and right contexts respectively).

A simple and common syntactic realisation of the Size frame is a single noun phrase containing one of the LUs, such as “the big brown dog that ran away”. Here we’d like to identify “big” as LU, “brown dog” as Entity and the combination as Size. Our first rule for identifying this type of construction might be

```
define LU {small} | {large} |
  {big} EndTag(LU);
define Size1 LU (Adjective)
  [Noun].t(Entity);
define TOP Size1 EndTag(Size);
```

Listing 2: A simplified first rule

This ruleset has been simplified for brevity – it has only a few of the permitted LUs, and word boundary issues have not been addressed.

The [].t() syntax in the definition of Size1 is a tag delimiter that controls the area tagged as Entity. The extra Adjective is optional, which is conveyed by the surrounding parentheses.

We can verify that our rules extract instances of our desired pattern by compiling them with hfst-pmatch2fst and running the compiled result with hfst-pmatch --extract-tags. In the following we have inputted the text of the King James Bible from Project Gutenberg (gutenberg.org) and added some extra characters on both sides for a concordance-like effect:

```
...
there lay a <Size><LU>small</LU>
round <Entity>thing</Entity></Size>
...
there was a <Size><LU>great</LU>
<Entity>cry</Entity></Size> in Egypt
...
saw that <Size><LU>great</LU>
<Entity>work</Entity></Size> which
...
```

pmatch may be operated in various modes. In locate mode the position and length of each match is given, and only the outermost tag is supplied. match mode (which is the default) tags and outputs running text, and extract mode does the same but omits parts of the input that aren’t matched by a rule. Matches may also be extracted via an API call, for example in order to achieve the above-seen concordance effect.

A natural next step is to add optional non-core elements, such as an adverb preceding the LU being tagged as Degree and a noun phrase beginning with “for a” following it as Standard.

```
define Size1 [Adverb].t(Degree)
  LU (Adjective) [Noun].t(Entity)
  [{for a} NP].t(Standard);
```

Listing 3: Extending the rule with optional elements

Here are some examples this rule finds in the British National Corpus (Consortium, 2007).

```
...
presence of an <Size>
  <Degree>arbitrarily</Degree>
  <LU>small</LU> <Entity>
  amount</Entity></Size> of dust
...
one <Size><LU>small</LU>
  <Entity>step</Entity>
  <Standard>for a man</Standard>
  </Size>
...
```

We can see that in “small amount of dust” we might want to tag not just the immediate noun as Entity but the entire noun phrase (which could be implemented up to a context-free definition of a noun phrase), and in “one small step for a man” a common indirect use of the Standard construction.

The FrameNet corpus itself is a good source for finding more cases.

As well as correct matches, such as “small round thing" in the biblical example, we have metaphorical meanings of `Size`, such as "great cry". This may or may not be desired – perhaps we wish to do further processing to identify the target domains of such metaphors, or perhaps we wish to be able to annotate physical size and physical size only.

3.1 Incorporating Semantic Information

Size is a very metaphorical concept, and syntactic rules as above will produce a large amount of matches that relate to such uses, eg. "a great cry" or "a big deal". If we wish to refine our rules to detect such uses, there are a few avenues for refinement.

First of all, some LUs are much more metaphorical than others. During the rule-writing process, a training set taken from a corpus (ideally a tagged corpus, but in this case taken from from a collection of appearances of the LU) is subjectively perused for more or less metaphorical cases.

A "great man" is almost certainly a metaphorical use, whereas a "large man" is almost certainly concrete. Accuracy may be improved by requiring "great" to be used together with a common concrete complement, like "great crowd". Improvements are rejected or accepted on the basis of performance on the training set.

There are also semantic classifications of words, such as WordNet (Miller, 1995). We may compile the set of hyponyms of *physical entity* and require them to appear as the nouns in our rules.

```
define phys_entity
  @txt"phys_entity.txt";
! a list of singular baseforms
! can be expanded to include
! eg. plurals by suitably composing
! it with a dictionary automaton
define phys_entities
  phys_entity .o. noun_baseform_expander;
```

Listing 4: Reading an external linguistic resource

3.2 Incorporating Part-of-speech Information

We have hitherto used named rules for matching word classes, like `Noun`, without specifying how they are written. Even our collection of LUs might need some closer attention – for example "little" could be an adverb.

Considering that in writing our rules we are effectively doing shallow syntactic parsing, even a very simple way to identify parts of speech may suffice: a morphological dictionary. For example, a finite-state transducer representing English morphology may be used to define the class of common nouns as in listing 5.

```
! The file we want to read
define English @bin"english.hfst";
! We compose it with a noun filter
! and extract the input side
define Noun English .o.
 [?+ "<NN1>" | "<NN2>"].u;
! (NN1 is singular, NN2 plural)
```

Listing 5: Using a dictionary to write POS rules

If we have the use of a part-of-speech tagger, we may write our rules to act on its output, as in table 6.

```
define Noun LC(W) Wordchar+
 ["<NN1>"|"<NN2>"] RC(W);
```

Listing 6: Using tags in pre-tagged text

4 Increasing Coverage

Having considered for each rule where `Degree` and `Standard` may occur, coverage may be evaluated by also finding those cases where a LU is used as an adjective but hasn't been tagged, eg.

```
define TOP Size1 | Size2 | ...
 [LU].t(NonmatchingLU);
```

The valid match is always the longest possible one, so `NonmatchingLU` will be the tag only if no subsuming `SizeN` rule applies.

For example in

```
the moving human body is
<NonmatchingLU>large</NonmatchingLU>
obtrusive and highly visible
```

We see another realisation of the frame: the `Entity` being followed by a copula, and the `LU` appearing to the right. We could write the rule `Size2` to capture this, adding positions for non-core elements either by linguistic reasoning or by searching the corpus.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics -*

Volume 1, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

The BNC Consortium. 2007. *The British National Corpus*. Oxford University Computing Services, version 3 (BNC XML) edition.

Charles J. Fillmore. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, 280(1):20–32.

Lauri Karttunen. 2011. Beyond morphology: Pattern matching with FST. In Cerstin Mahlow and Michael Piotrowski, editors, *Systems and Frameworks for Computational Morphology*, volume 100 of *Communications in Computer and Information Science*, pages 1–13, Berlin Heidelberg. Springer.

Krister Lindén, Erik Axelson, Senka Drobac, Sam Hardwick, Juha Kuokkala, Jyrki Niemi, Tommi Pirinen, and Miikka Silfverberg. 2013. HFST – a system for creating NLP tools. In Cerstin Mahlow and Michael Piotrowski, editors, *Systems and Frameworks for Computational Morphology*, volume 380 of *Communications in Computer and Information Science*, pages 53–71, Berlin Heidelberg. Springer.

George A. Miller. 1995. Wordnet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.