Using the Alpino chart generator

Table of Contents

Introduction

Recent versions of Alpino include a chart generator that uses the Alpino grammar. The generator is included in the latest Alpino snapshots that are available from:

http://www.let.rug.nl/~vannoord/alp/Alpino/binary/

This document provides some (short) instructions on using the generator.

Basic usage

You can generate from a parsed sentence with the - (minus) Hdrug command. Provide the analysis you want to generate from as an argument. For example:

```
1 |: * Dit is een zin .
[...]
Found 6 solution(s)
2 |: - 1
[...]
dit is een zin
G#undefined|1|dit is een zin|p(19.392927511841602,[])
een zin is dit
G#undefined|2|een zin is dit|p(24.38472796607936,[])
dit is 'n zin
G#undefined|3|dit is 'n zin|p(34.237826717179765,[])
'n zin is dit
G#undefined|4|'n zin is dit|p(35.92385918538579,[])
cputime total 1230 msec
Found 4 solution(s)
```

If a generation suite is specified (using the *generation_suite* flag), realizations can also be generated from abstract dependency trees (ADTs) from the selected generation suite with the *gen* command:

```
3 |: gen 55
[...]
ik heb de trein gemist
K#55|1|1.000|rougeN|1.000|rougeL|1.000|rougeS|1.000|rougeSu
G#55|1|ik heb de trein gemist|p(40.40712236100052,[])
de trein heb ik gemist
K#60|2|1.000|rougeN|1.000|rougeL|1.000|rougeS|1.000|rougeSu
G#60|2|de trein heb ik gemist|p(43.60989380865559,[])
cputime total 1090 msec
```

```
Found 2 solution(s)
```

Flags

Fluency

<glossentry> *fluency_model*
<glossdef>
Path to the n-gram language model module.
</glossdef>
</glossentry>
<glossentry> *fluency_model_maxent*
<glossdef>
Path to the maxent language model module.
</glossdef>
</glossentry>
<glossentry> *fluency_feature_weights*
<glossdef>
Weights for maxent fluency features.
</glossdef>
</glossentry>
<glossentry> *ngram_model_words*
<glossdef>
Path to the word automaton for the n-gram language model.
</glossdef>
</glossentry>
<glossentry> *ngram_model_{unigrams,bigrams,trigrams}*
<glossdef>
Path to the unigrams/bigrams/trigrams tuple automaton for the n-gram language model.
</glossdef>
</glossentry>
<glossentry> *ngram_model_tags*
<glossdef>
Path to the tags automaton for the maxent language model.
</glossdef>
</glossentry>
<glossentry> *ngram_model_tag_{unigrams,bigrams,trigrams}*
<glossdef>
Path to the tag unigrams/bigrams/trigrams tuple automaton for the maxent language model.
</glossdef>
</glossentry>
<glossentry> *use_fluency_model*
<glossdef>
Enable use of a fluency model, *ngram* will use the n-gram language model, *maxent* the maximum entropy model.
</glossdef>
</glossentry>

Other

<glossentry> *introduce_punct*
<glossdef>
Introduce punctuation. There are three possible settings for this flag. *off* introduces no punctuation, *minimal* introduces as little punctuation as possible using iterative deepening, and *on* introduces punctuation where it is allowed.
</glossdef>
</glossentry>
<glossentry> *number_analyses*

<glossdef>
The number of realizations to return. If a fluency model is used, the N most probable realizations are returned.
</glossdef>
</glossentry>

End hooks

train_fluency

The *train_fluency* end hook is used to extract and print features used for training the maximum entropy model. For instance:

```
$ Alpino end_hook=train_fluency
[...]
1 |: gen 1
G#1#1#1.0#21.06905461962647@ngram_lm|1@r1(n_adj_n)|1@r1(np_det_n)|[...]
```