

More (Advanced) JAPE

Module 1

June 2015

© 2015 The University of Sheffield

This material is licenced under the Creative Commons

Attribution-NonCommercial-ShareAlike Licence

(<http://creativecommons.org/licenses/by-nc-sa/3.0/>)

Outline

1 Debugging JAPE Grammars

2 Using Java in JAPE

- Common idioms

Outline

1 Debugging JAPE Grammars

2 Using Java in JAPE

- Common idioms

Debugging JAPE Grammars

- Read the error messages, they are helpful!
 - line numbers etc. refer to the original JAPE files
 - description usually highlights the exact problem

```
file:/home/gate/plugins/ANNIE/resources/NE/name.jape:  
Encountered " <kleeneOp> "? "" at line 1580, column 10.  
Was expecting one of:  
    "\" " ...  
    <ident> ...  
    "|" ...  
    "{" ...  
    "(" ...  
    ")" ...
```

Debugging JAPE Grammars

When trying to understand how annotations were created by a grammar try the new **enableDebugging** option:

- **addedByPR:** the name of the JAPE PR running the grammar that produced the annotation
- **addedByPhase:** the name of the phase (usually the filename) in which the annotation was created
- **addedByRule:** the name of the rule responsible for creating the annotation

Outline

1 Debugging JAPE Grammars

2 Using Java in JAPE

■ Common idioms

Beyond Simple Actions

It's often useful to do more complex operations on the RHS than simply adding annotations, e.g.

- Set a new feature on one of the matched annotations
- Delete annotations from the input
- More complex feature value mappings, e.g. concatenate several LHS features to make one RHS one.
- Collect statistics, e.g. count the number of matched annotations and store the count as a document feature.

JAPE has no special syntax for these operations, but allows blocks of arbitrary Java code on the RHS.

- Don't worry if you are not a (Java) developer
- The rest of this section will show you a number of 'recipes' which you can edit slightly for specific tasks
- These ideas can be cut-and-pasted together to perform more complex actions
- If you do want to understand these examples in more detail then the GATE API will be covered in the developer track on Friday



- The examples covered in this session cover common scenarios
 - accessing annotations and features
 - removing annotations
 - accessing document features
 - using the text under an annotation
- There are lots more examples on the GATE wiki
 - `https://gate.ac.uk/wiki/jape-repository/`

JAPE With Java RHS Template

```
1 Imports: { import static gate.Uutils.*; }
2
3 Phase: Example
4 Input: Token // and any other input annotation types
5 Options: control = appelt
6
7 Rule: Example1
8 (
9   // Normal JAPE LHS goes here
10 ):label
11 -->
12 {
13   //Java code goes in here
14 }
```

Java Block Variables

The variables available to Java RHS blocks are:

- doc** The document currently being processed.
- inputAS** The `AnnotationSet` specified by the `inputASName` runtime parameter to the JAPE transducer PR. Read or delete annotations from here.
- outputAS** The `AnnotationSet` specified by the `outputASName` runtime parameter to the JAPE transducer PR. Create new annotations in here.
- ontology** The ontology (if any) provided as a runtime parameter to the JAPE transducer PR.
- bindings** The bindings map. . .

Bindings

- `bindings` is a `Map` from `string` to `AnnotationSet`
- Keys are labels from the LHS.
- Values are the annotations matched by the label.

```
1 (
2   {Token.string == "University"}
3   {Token.string == "of"}
4   ({Lookup.minorType == city}):uniTown
5 ):orgName
```

- `bindings.get("uniTown")` contains one annotation (the `Lookup`)
- `bindings.get("orgName")` contains three annotations (two `Tokens` plus the `Lookup`)

A Simple Example

This is a simple example of a Java RHS that prints the type and features of each annotation it matches. Give it a try!

```
1 Rule: ListEntities
2 ({Person}|{Organization}|{Location}):ent
3 -->
4 {
5     // get the annotations that matched
6     AnnotationSet ents = bindings.get("ent");
7
8     for(Annotation e : ents) {
9         // display the type and features of each
10        System.out.println("Type: " + e.getType());
11        System.out.println("Features: " + e.getFeatures());
12    }
13 }
```

- **Application:** hands-on/jape/example1.xgapp
- **JAPE file:** hands-on/jape/resources/simple.jape

Named Java Blocks

```
1 -->
2 :uniTown{
3     uniTownAnnots.iterator().next().getFeatures()
4         .put("hasUniversity", Boolean.TRUE);
5 }
```

- You can label a Java block with a label from the LHS
- The block will only be called if there is at least one annotation bound to the label
- Within the Java block there is a variable `labelAnnots` referring to the `AnnotationSet` bound to the label
 - i.e. `AnnotationSet xyAnnots = bindings.get("xy")`
- you can have any number of `:bind.Type = {}` assignment expressions and blocks of Java code, separated by commas.

Common Idioms for Java RHS

Setting a new feature on one of the matched annotations

```
1 Rule: LcString
2 ({Token}):tok
3 -->
4 :tok {
5     for(Annotation a : tokAnnots) {
6         // get the FeatureMap for the annotation
7         FeatureMap fm = a.getFeatures();
8         // get the "string" feature
9         String str = (String)fm.get("string");
10        // convert it to lower case and store
11        fm.put("lcString", str.toLowerCase());
12    }
13 }
```

Exercise 2: Modifying Existing Annotations

- Load `hands-on/jape/exercise2.xgapp`
- As before, this is ANNIE plus an extra transducer, this time loading
`hands-on/jape/resources/general-pos.jape`.
- Modify the Java RHS block to add a `generalCategory` feature to the matched `Token` annotation holding the first two characters of the POS tag (the `category` feature).
 - `String.substring(startIndex, endIndex)`
- Remember to reinitialize the “Exercise 2 Transducer” after editing the JAPE file.
- Test it by running the “Exercise 2” application.

Common Idioms for Java RHS

Removing matched annotations from the input

```
1 Rule: Location
2 ({Lookup.majorType = "location"}):loc
3 -->
4 :loc.Location = { kind = :loc.Lookup.minorType,
5     rule = "Location"},
6 :loc {
7     inputAS.removeAll(locAnnots);
8 }
```

This can be useful to stop later phases matching the same annotations again.

Common Idioms for Java RHS

Accessing the string covered by a match

```
1 Rule: Location
2 ({Lookup.majorType = "location"}):loc
3 -->
4 :loc {
5     String str = stringFor(doc,locAnnots);
6 }
```

Example: Contained Annotations

To get annotations contained within the span of the match

```
1 Rule: NPtokens
2 ({NounPhrase}):np
3 -->
4 :np {
5     List<String> posTags = new ArrayList<String>();
6     for(Annotation tok :
7         getContained(inputAS, npAnnots, "Token")) {
8         posTags.add(
9             (String)tok.getFeatures().get("category"));
10    }
11    FeatureMap fm =
12        npAnnots.iterator().next().getFeatures();
13    fm.put("posTags", posTags);
14    fm.put("numTokens", (long)posTags.size());
15 }
```

Exercise 3: Working with Contained Annotations

- Load `hands-on/jape/exercise3.xgapp`
- As before, this is ANNIE plus an extra transducer, this time loading `hands-on/jape/resources/exercise3-main.jape`.
- This is a multiphase grammar containing the `general-pos.jape` from exercise 2 plus `num-nouns.jape`.
- Modify the Java RHS block in `num-nouns.jape` to count the number of nouns in the matched `Sentence` and add this count as a feature on the sentence annotation.
- Remember to reinitialize the “Exercise 3 Transducer” after editing the JAPE file.
- Test it by running the “Exercise 3” application.

Passing state between rules

To pass state between rules, use document features:

```
1 Rule: Section
2 ({SectionHeading}):sect
3 -->
4 :sect {
5     doc.getFeatures().put("currentSection",
6         stringFor(doc, sectAnnots));
7 }
8
9 Rule: Entity
10 ({Entity}):ent
11 -->
12 :ent {
13     entAnnots.iterator().next().getFeatures()
14         .put("inSection",
15             doc.getFeatures().get("currentSection"));
16 }
```

Returning from RHS blocks

- You can **return** from a Java RHS block, which prevents any later blocks or assignments for that rule from running, e.g.

```
1 -->
2 :uniTown{
3   String townString = stringFor(doc, uniTownAnnots);
4   // don't add an annotation if this town has been seen before. If we
5   // return, the UniversityTown annotation will not be created.
6   if (!( (Set) doc.getFeatures().get("knownTowns") )
7       .add(townString) ) return;
8 },
9 :uniTown.UniversityTown = {}
```

Annotation Sets and Ordering

- An AnnotationSet is a set, so it is not ordered

```
10 Rule: SimpleNPRule1
11 (
12   ({Token.generalCategory=="DT"})?
13   ({Token.generalCategory=="JJ"}) [0, 4]
14   ({Token.generalCategory=="NN"})+
15 ) : nnp
16 -->
17 : nnp {
18   System.out.println("_____");
19   System.out.println(stringFor(doc, nnpAnnots));
20   System.out.println("The individual tokens:");
21
22   for(Annotation tok : nnpAnnots) {
23     System.out.println(stringFor(doc, tok));
24   }
25 }
```

- The grammar for this example is in `hands-on/jape/resources/match-nps.jape`. To run the example yourself, load `exercise2.xgapp` in GATE Developer, load an extra JAPE Transducer PR, and give it as a parameter this grammar file. Finally, add the resulting new PR at the end of the Exercise 2 application and re-run it.

Annotation Sets and Ordering (Continued)

- Here is a sample output, if you execute this rule on our test document

```
waste management businesses  
Now printing the matched individual tokens:  
businesses  
waste  
management
```

- Instead use `inDocumentOrder(AnnotationSet as)` which returns a list containing the annotations in the given annotation set, in document order

Exceptions

- Any `JapeException` or `RuntimeException` thrown by a Java RHS block will cause the JAPE Transducer PR to fail with an `ExecutionException`
- For non-fatal errors in a RHS block you can throw a `gate.jape.NonFatalJapeException`
- This will print debugging information (phase name, rule name, file and line number) but will not abort the transducer execution.
 - However it will interrupt this rule, i.e. if there is more than one block or assignment on the RHS, the ones after the `throw` will not run.