

ICPC 2017  
EARLY RESEARCH ACHIEVEMENTS

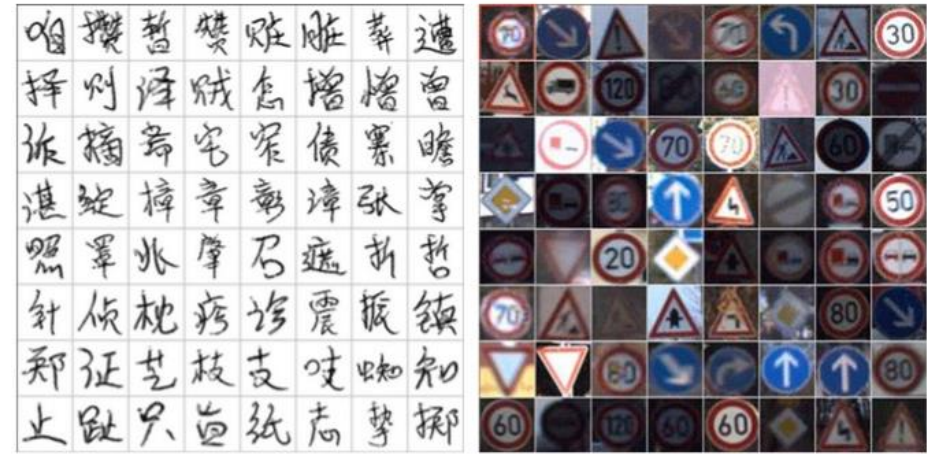
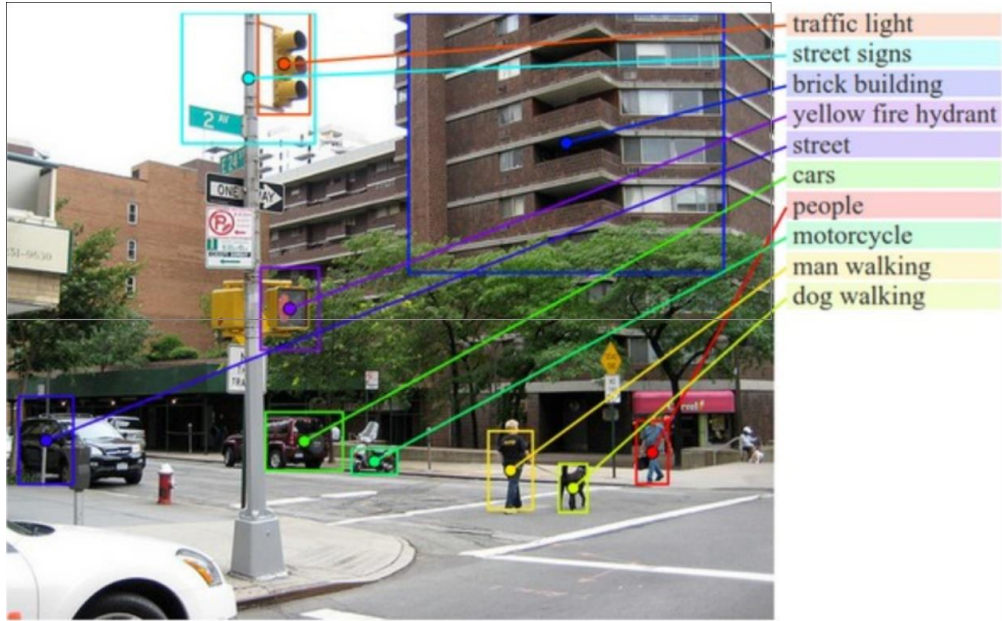
# REPLICATING PARSER BEHAVIOR USING NEURAL MACHINE TRANSLATION

Carol V. Alexandru, Sebastiano Panichella, Harald C. Gall  
{alexandru,panichella,gall}@ifi.uzh.ch  
23. May 2017



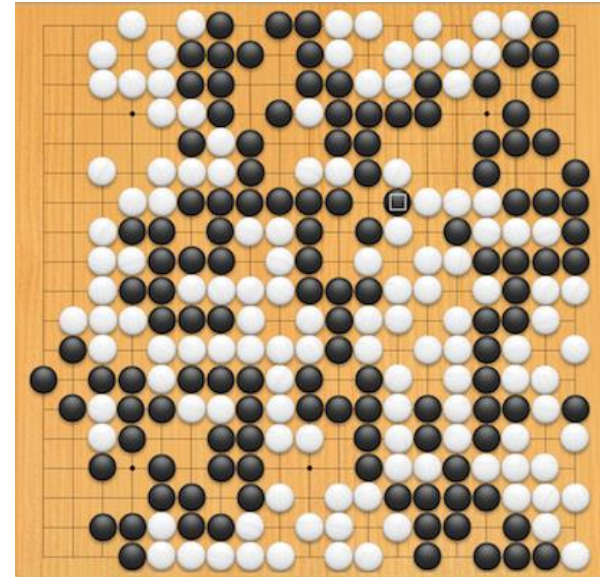
University of  
Zurich <sup>UZH</sup>

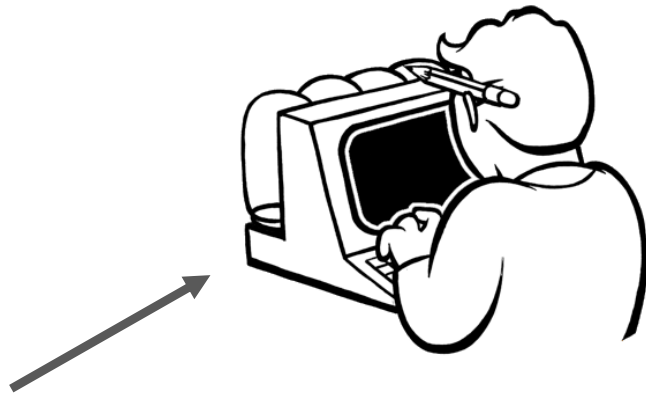
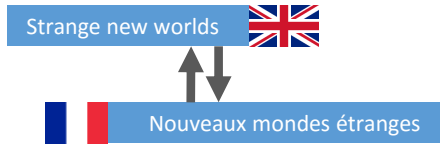
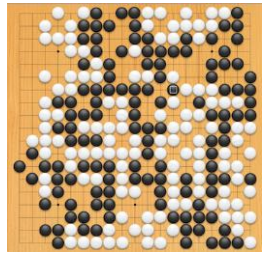
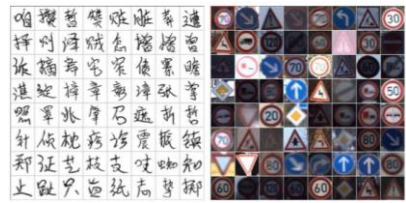


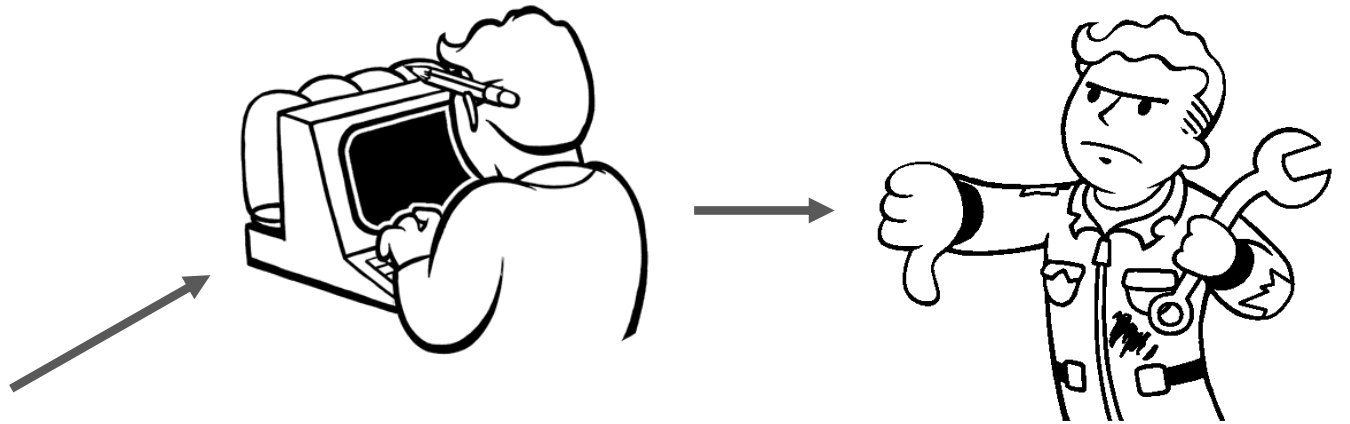
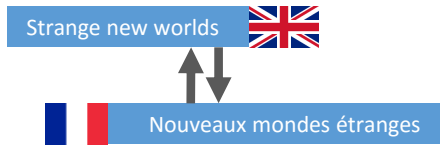
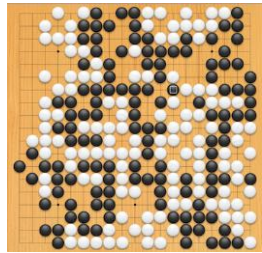
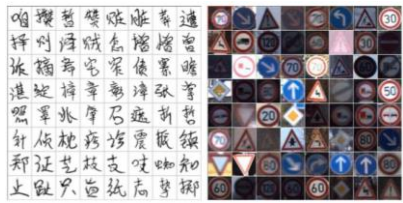


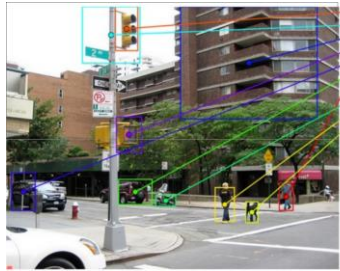
Strange new worlds 

 Nouveaux mondes étranges

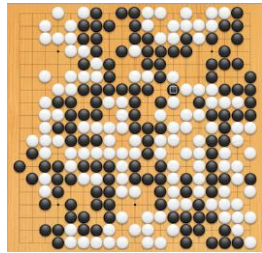
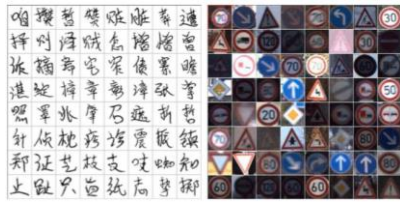








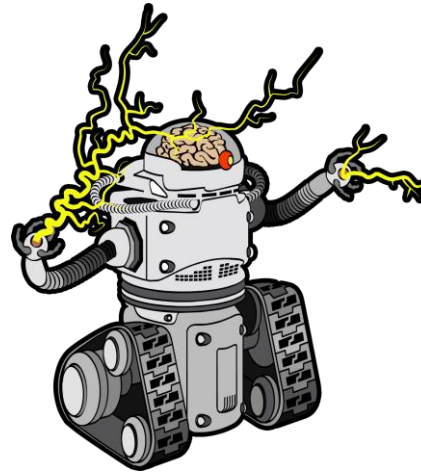
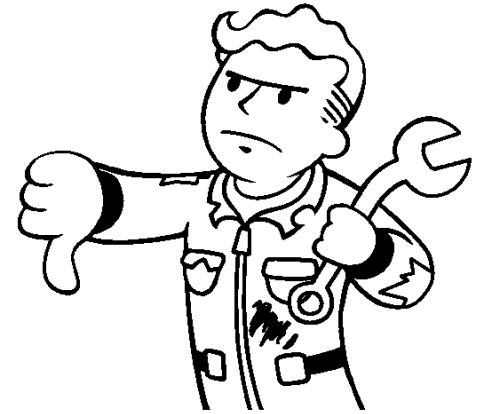
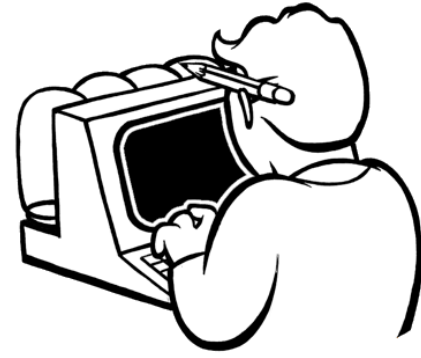
- traffic light
- street signs
- brick building
- yellow fire hydrant
- street
- cars
- people
- motorcycle
- man walking
- dog walking



Strange new worlds 



Nouveaux mondes étranges



```
public int sum(int[] numbers) {  
    int s = 0;  
    for (int n : numbers) {  
        s = s - n;  
    }  
    return s;  
}
```

```
public int sum(int[] numbers) {  
    int s = 0;  
    for (int n : numbers) {  
        s = s - n;  
    }  
    return s;  
}
```

```
public int sum(int[] numbers) {  
    int s = 0;  
    for (int n : numbers) {  
        s = s - n;  
    }  
    return s;  
}
```

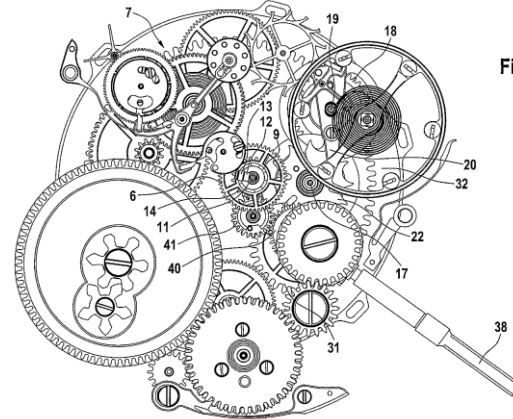


Fig. 3

Difficult to  
codify, hard to  
detect and fix



```
public int sum(int[] numbers) {  
    int s = 0;  
    for (int n : numbers) {  
        s = s - n;  
    }  
    return s;  
}
```

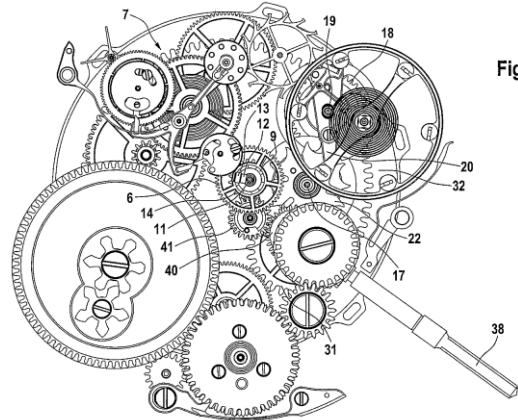
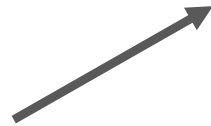
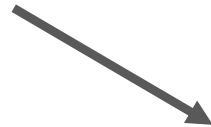


Fig. 3

Difficult to  
codify, hard to  
detect and fix



Human  
blackbox brain  
can easily spot  
this

Where to begin?

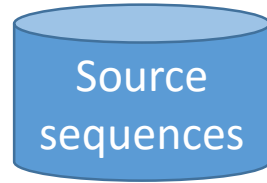
```
print("Hello world")
```

Where to begin?

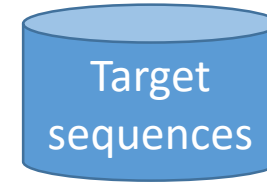
```
print("Hello world")
```

Can we teach a machine to "read" code?

# Neural Machine Translation

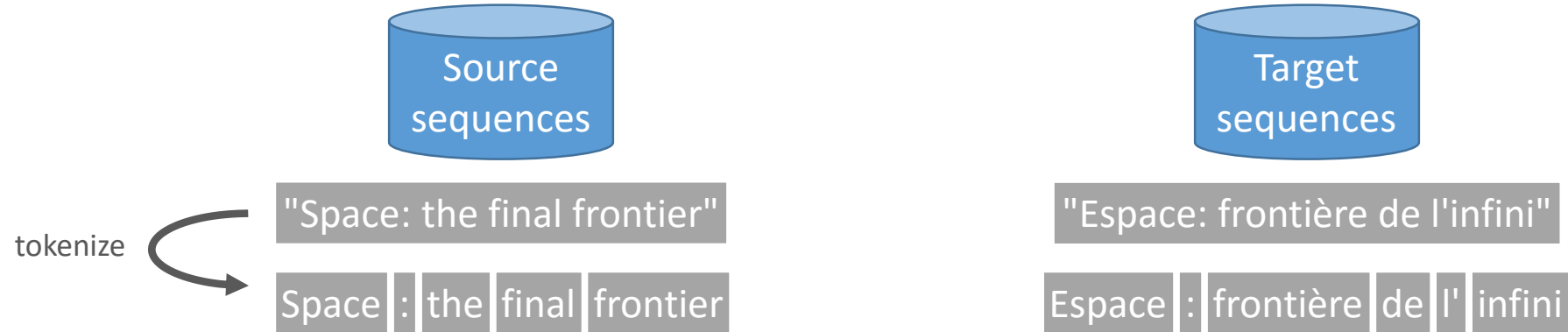


"Space: the final frontier"

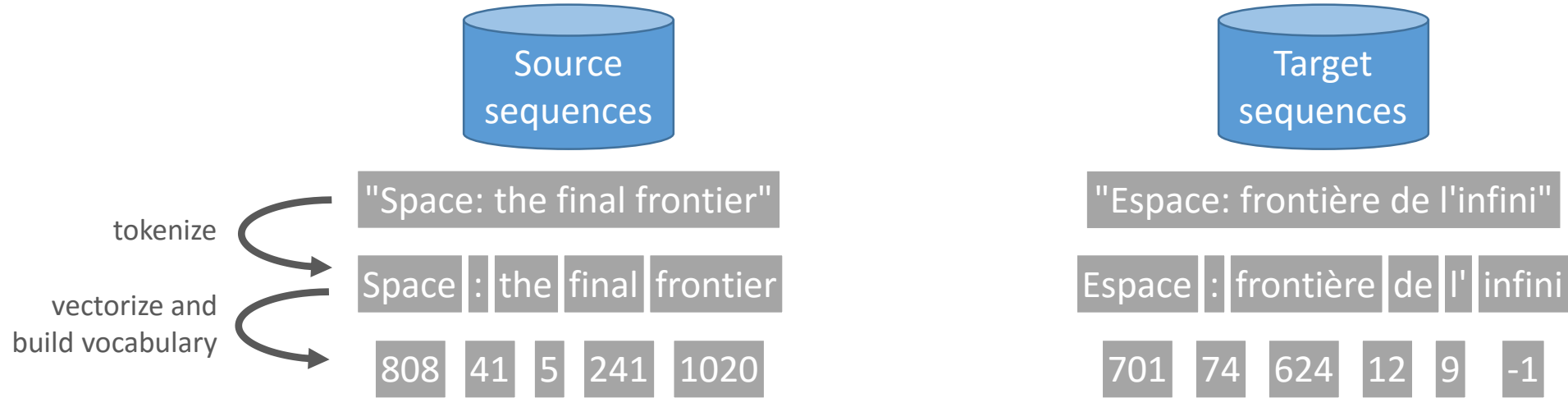


"Espace: frontière de l'infini"

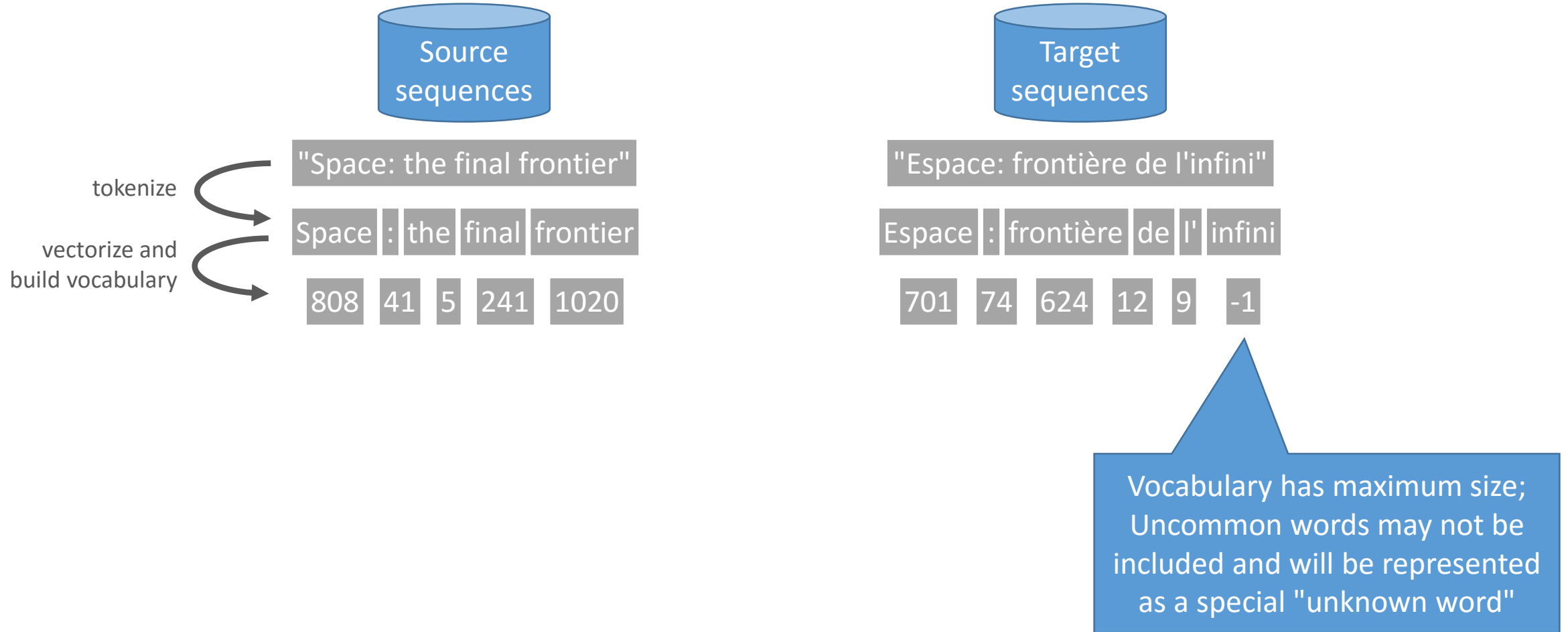
# Neural Machine Translation



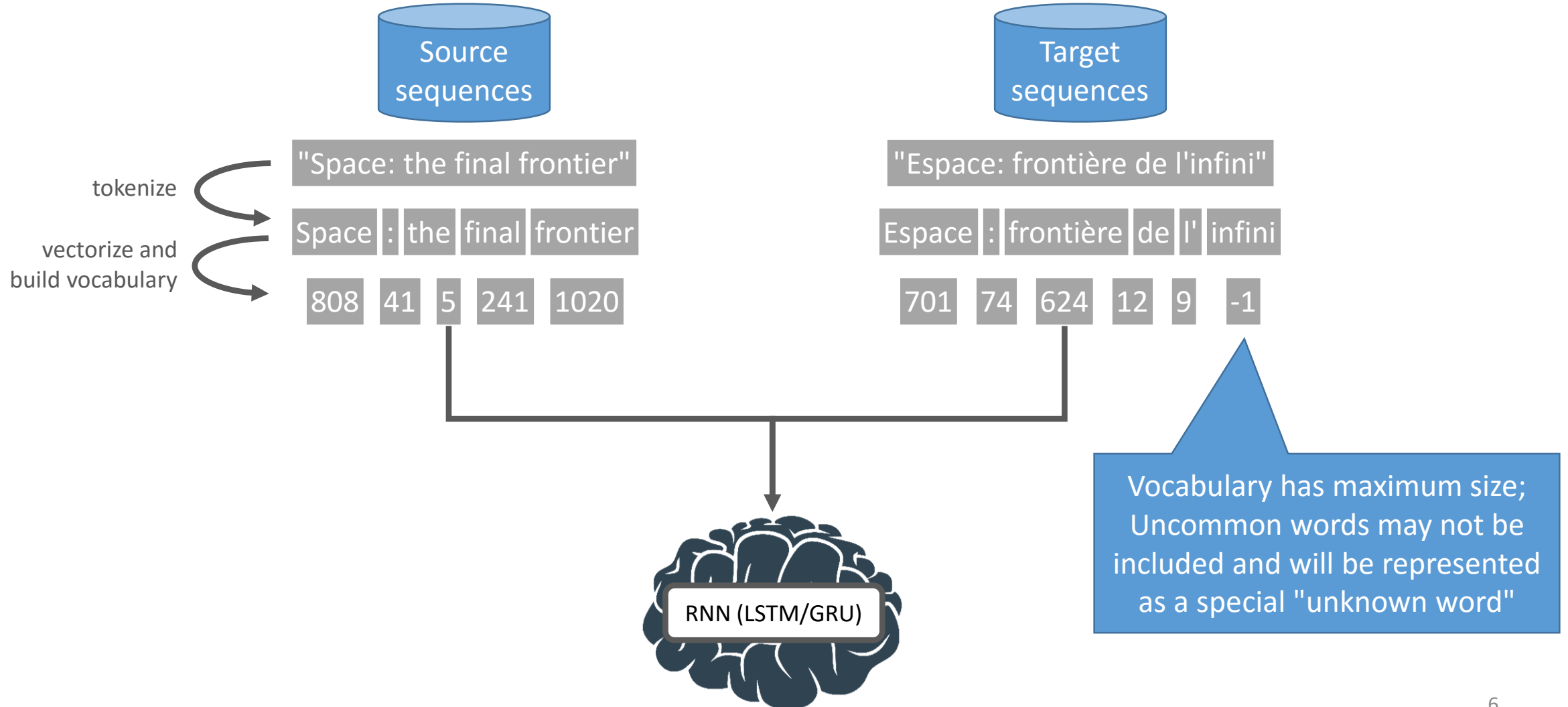
# Neural Machine Translation



# Neural Machine Translation



# Neural Machine Translation

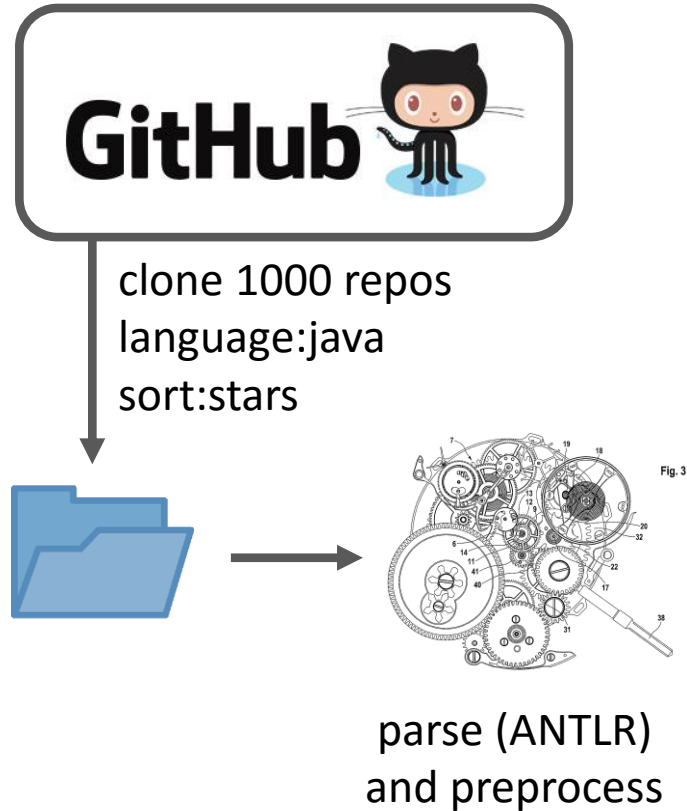




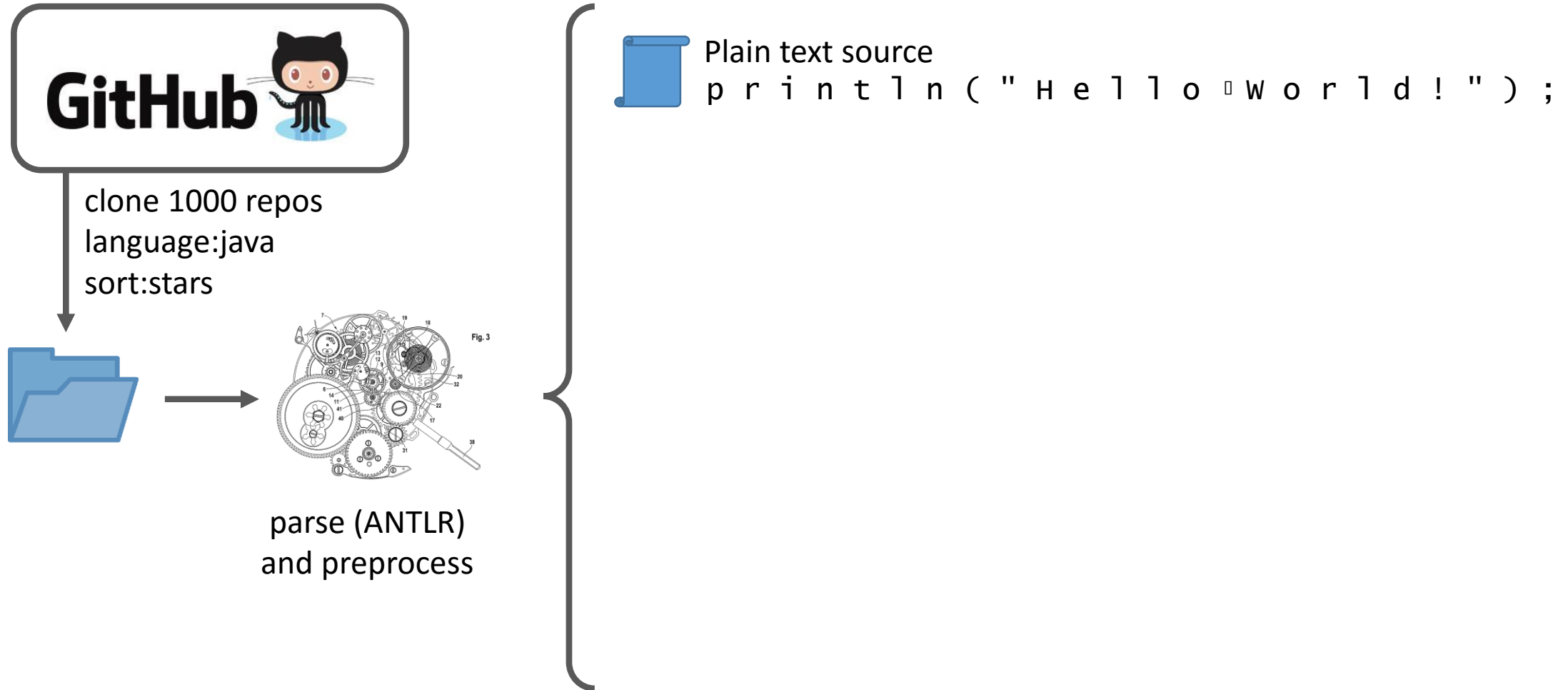
# Data Gathering and Preparation



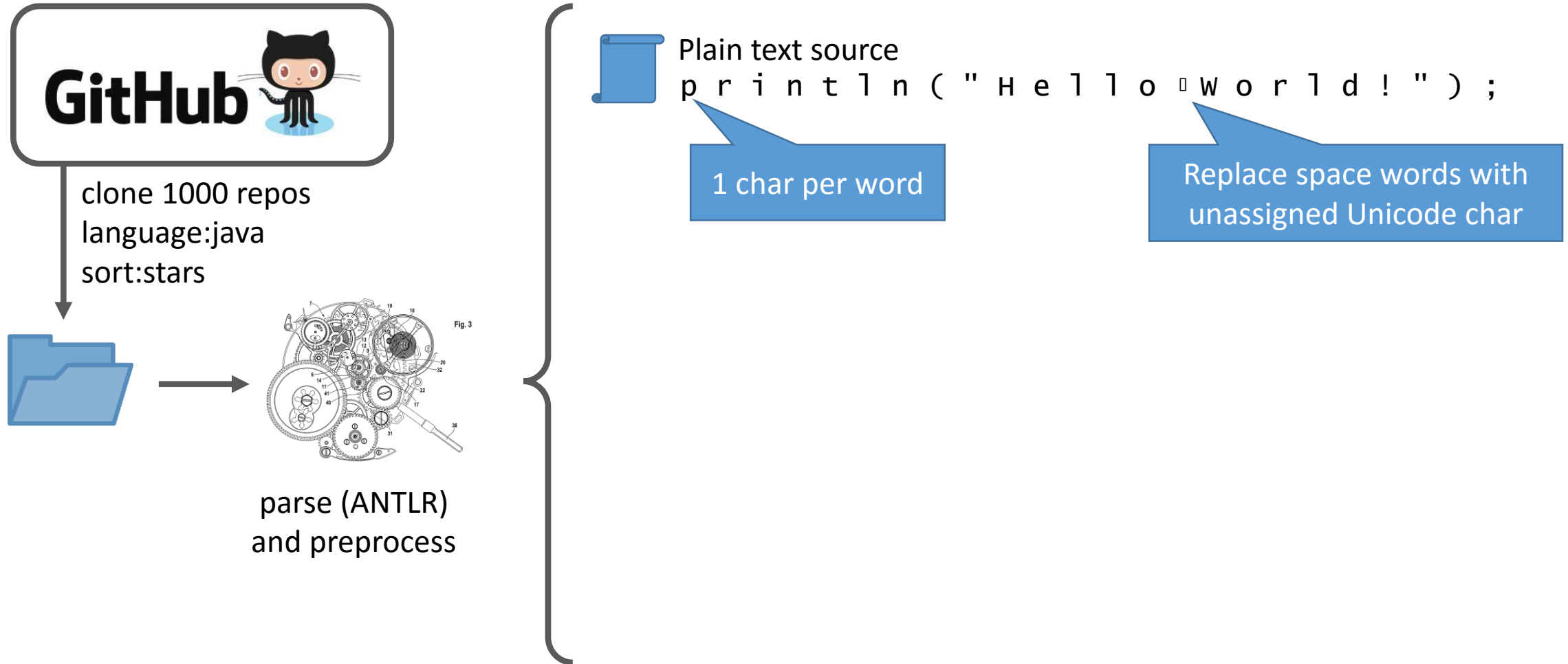
# Data Gathering and Preparation



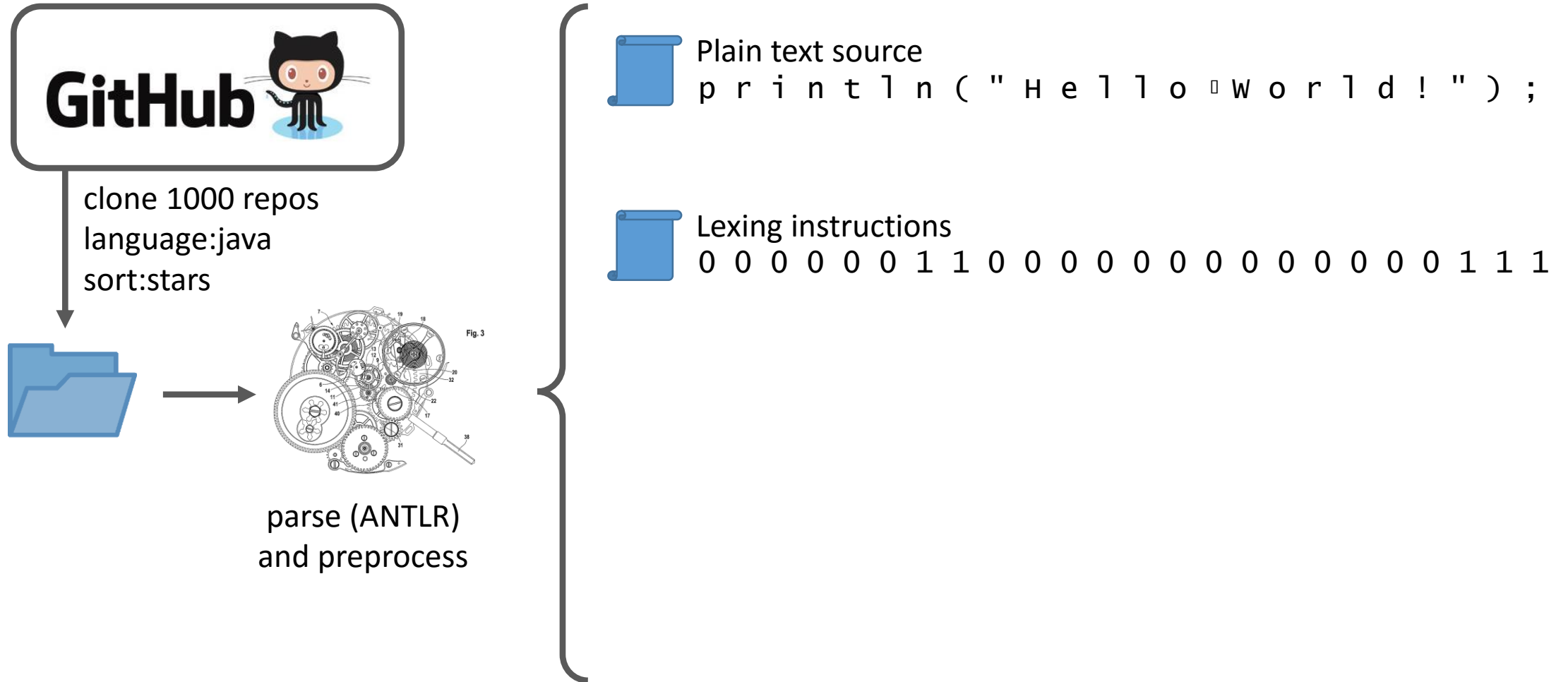
# Data Gathering and Preparation



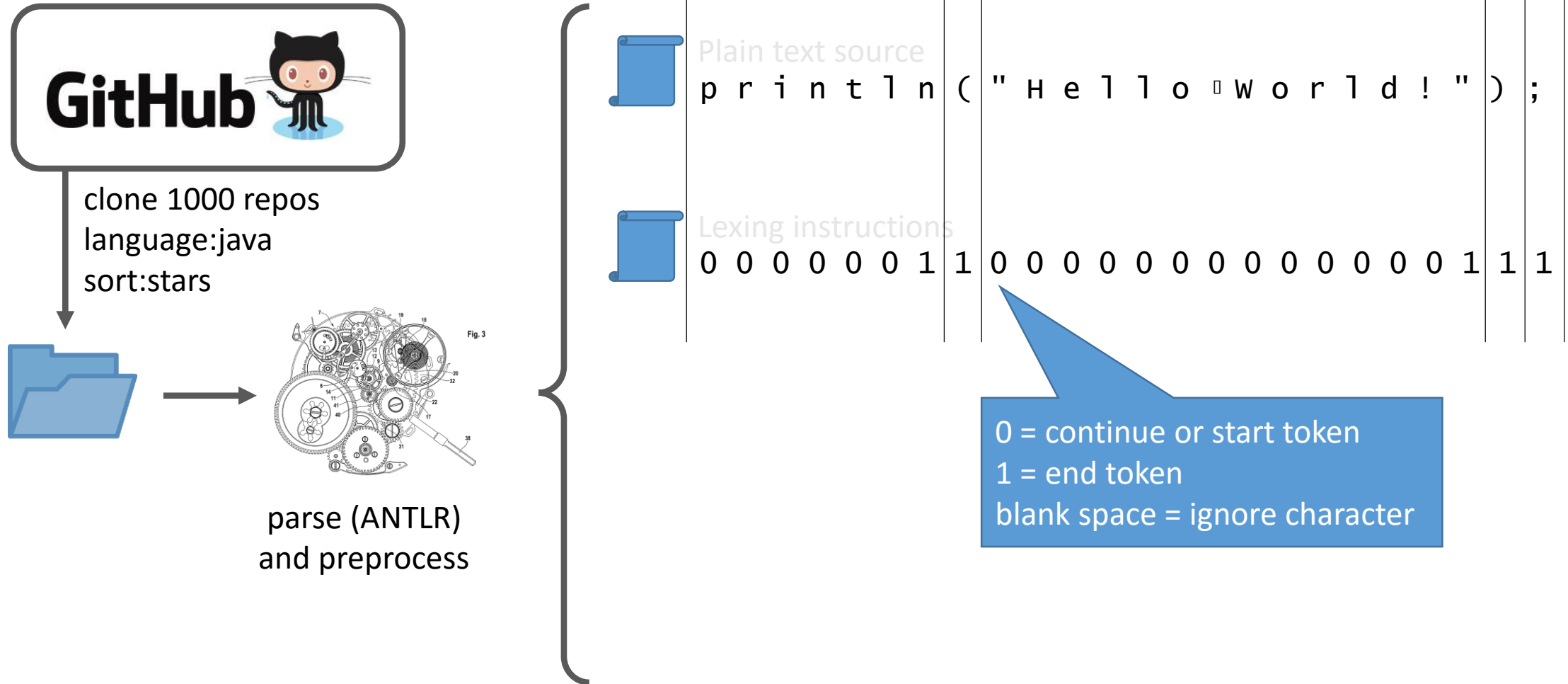
# Data Gathering and Preparation



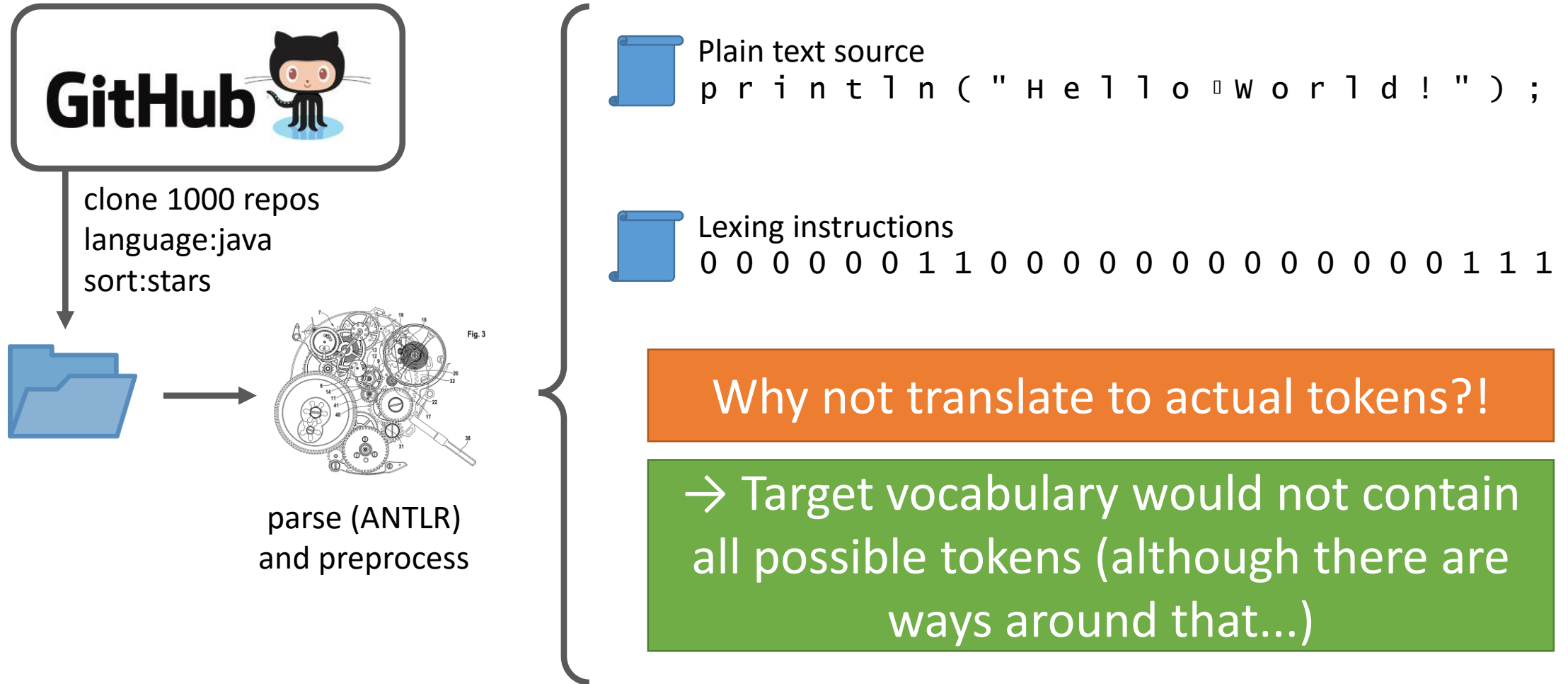
# Data Gathering and Preparation



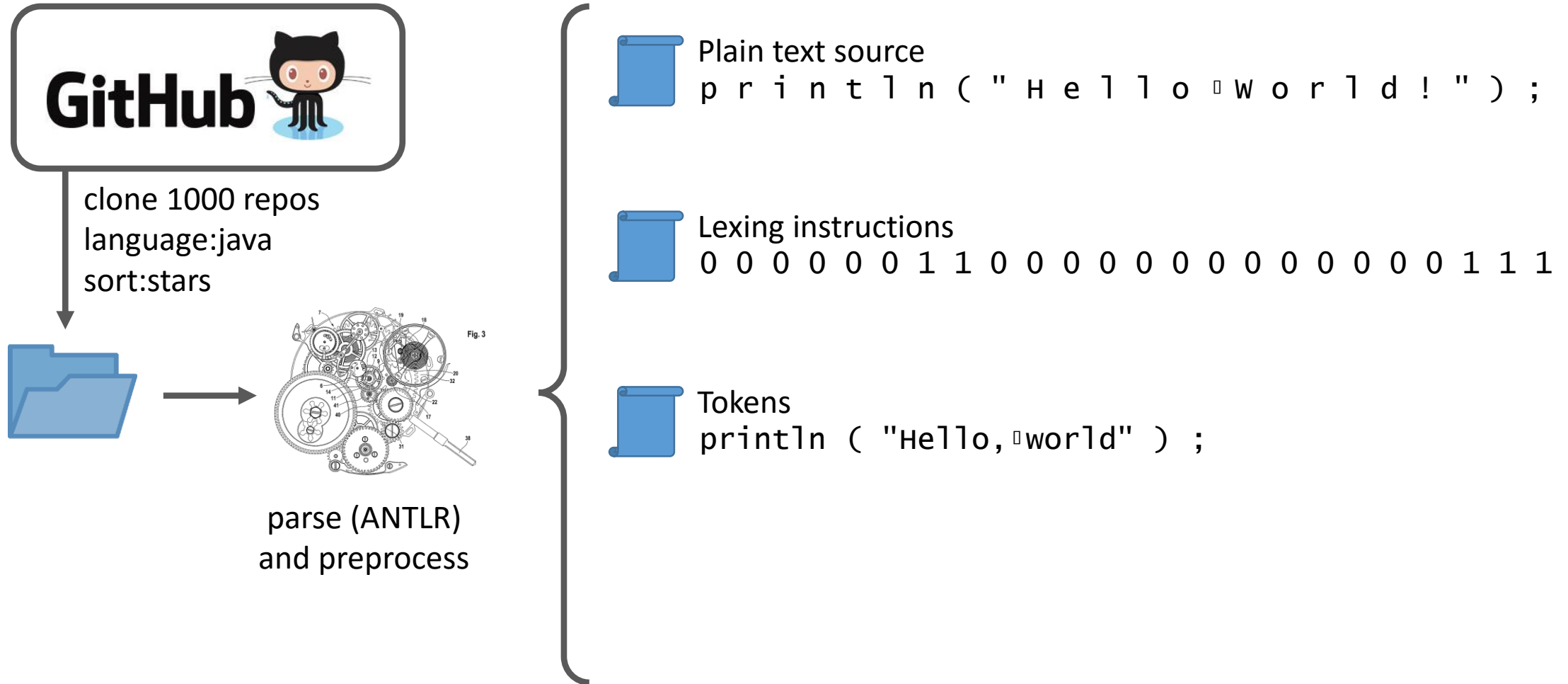
# Data Gathering and Preparation



# Data Gathering and Preparation

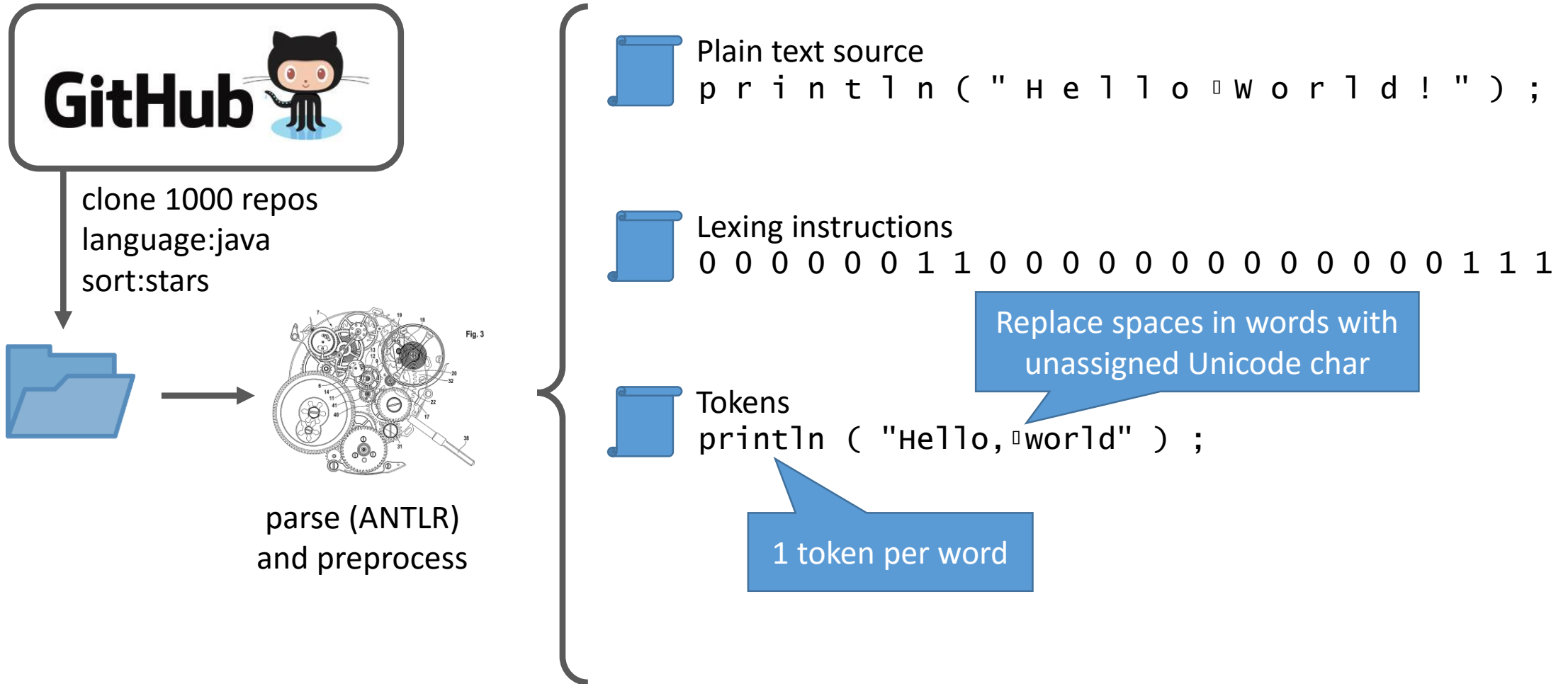


# Data Gathering and Preparation

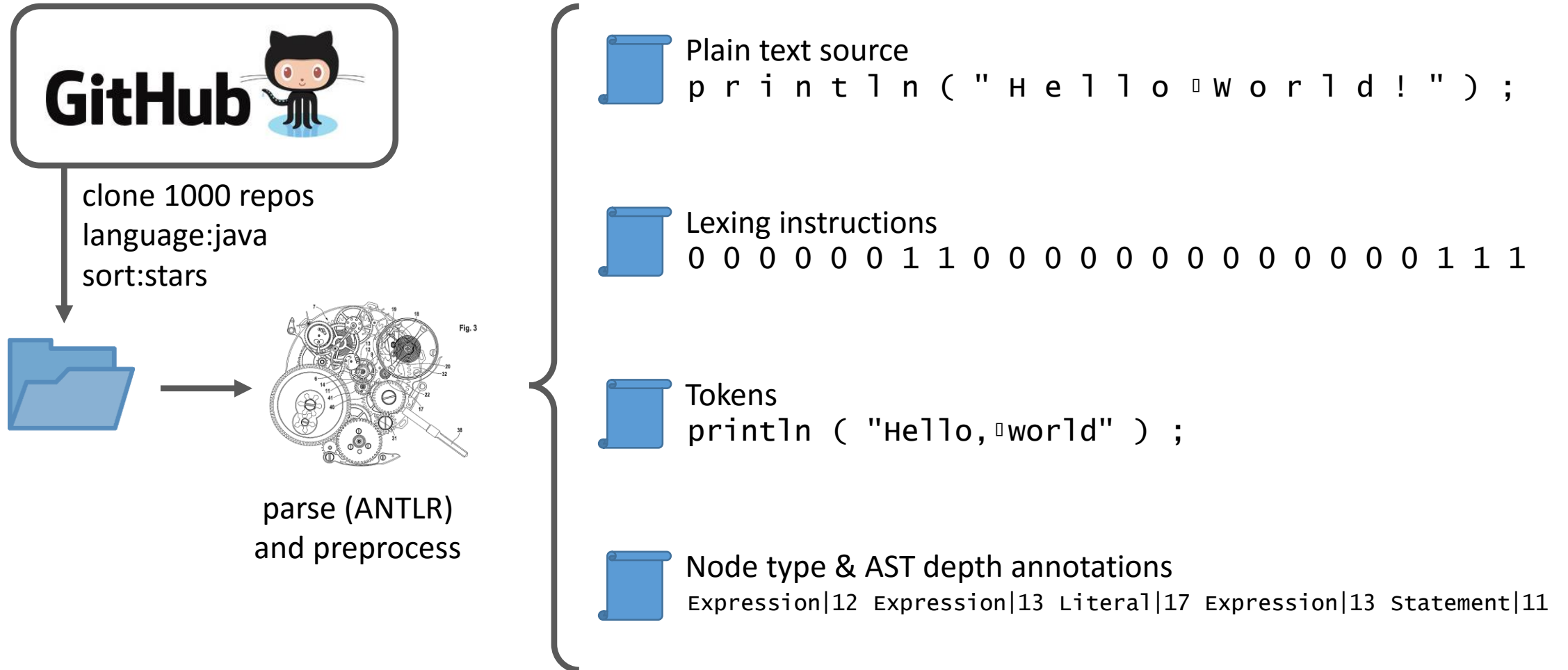




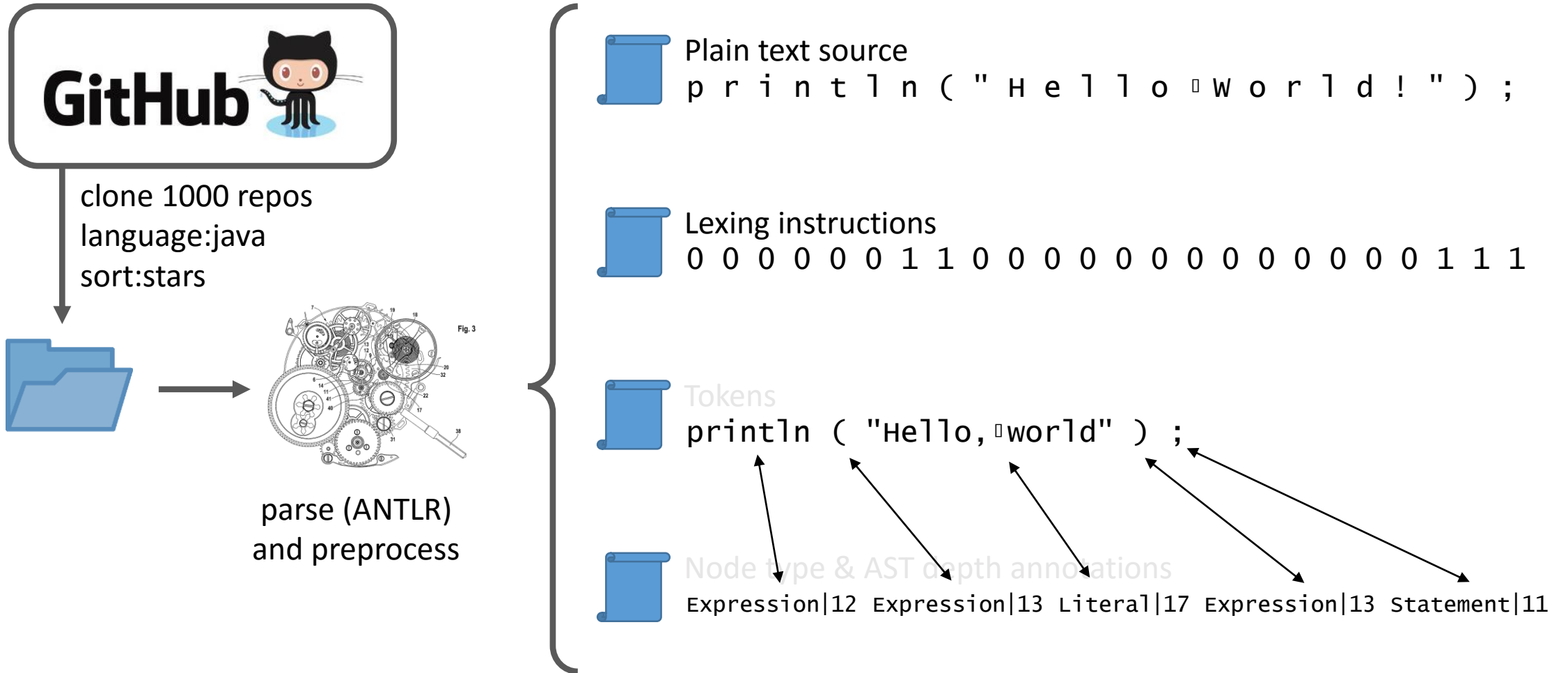
# Data Gathering and Preparation



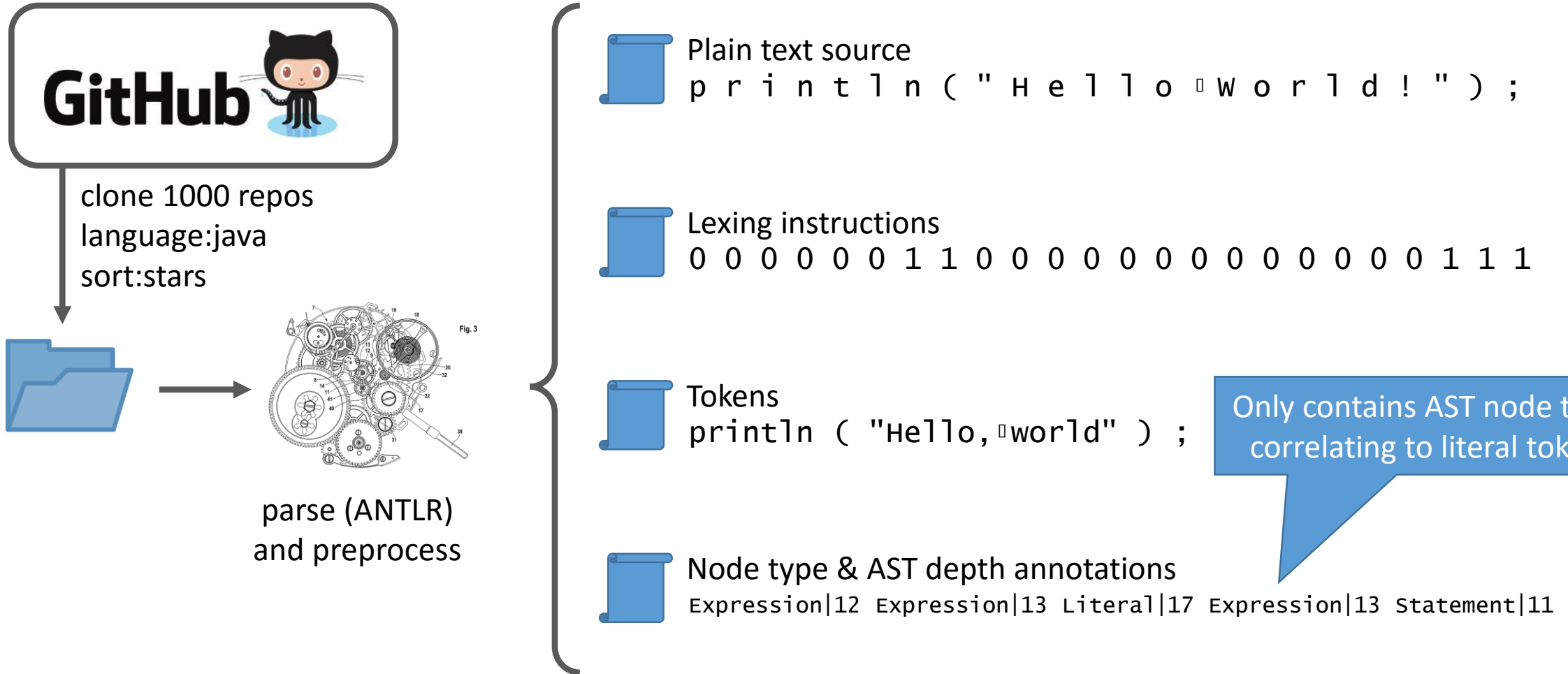
# Data Gathering and Preparation



# Data Gathering and Preparation



# Data Gathering and Preparation



# Data Gathering and Preparation



Plain text source

```
println("Hello World!");
```

Creation of 2x2 datasets for two translations steps:  
plaintext → tokens      tokens → annotations

Data creation tool is **open source** - define your own extractions and translations and apply them easily to 1000s of repos:

<https://bitbucket.org/sealuzh/parsenn>

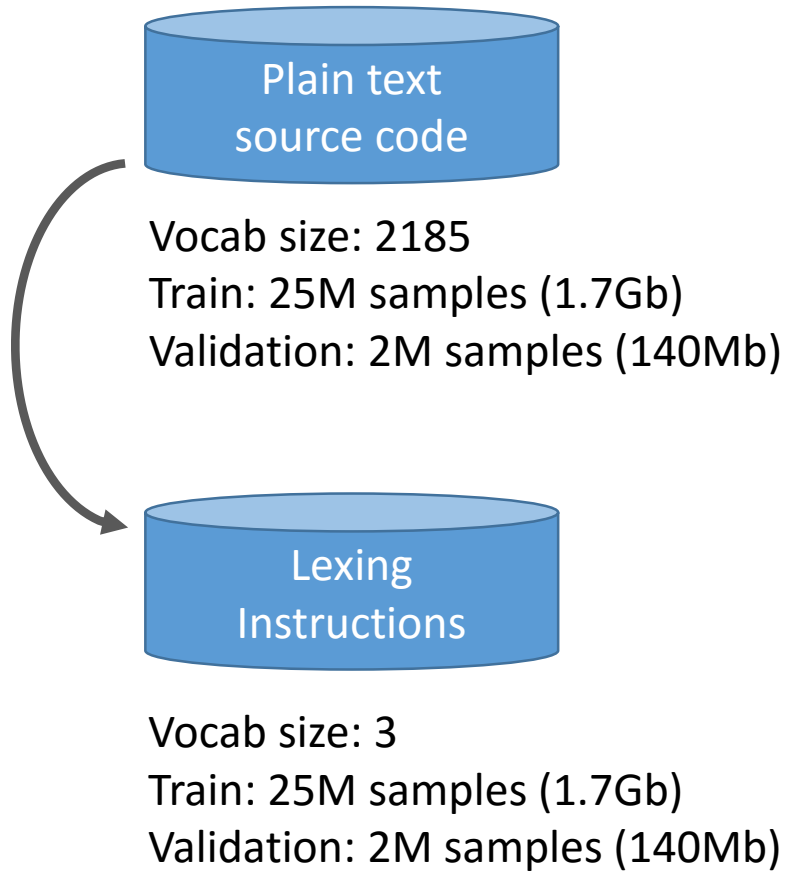
parse (ANTLR)  
and preprocess



Node type & AST depth annotations

```
Expression|12 Expression|13 Literal|17 Expression|13 Statement|11
```

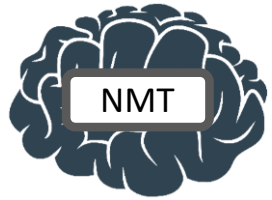
# Results: Tokenization



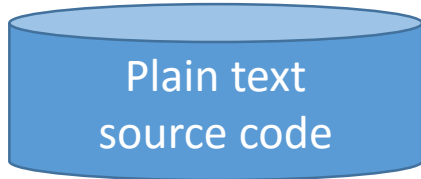
```
import android.graphics.Bitmap;  
import com.facebook.common.references.ResourceReferenceCleaner;  
public class SimpleBitmapReleaser implements IResourceReferenceCleaner {  
    private static SimpleBitmapReleaser sInstance;  
    public static SimpleBitmapReleaser getInstance() {  
        if (sInstance == null) {  
            sInstance = new SimpleBitmapReleaser();  
        }  
    }  
}
```

```
0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1  
0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0  
0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0  
0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1  
0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1  
0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1 1 1  
0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1  
1
```

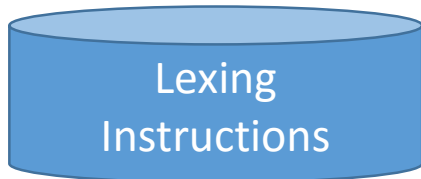
# Results: Tokenization



Bi-RNN  
7 epochs  
7 days  
**Perplexity: 1.11**



Vocab size: 2185  
Train: 25M samples (1.7Gb)  
Validation: 2M samples (140Mb)



Vocab size: 3  
Train: 25M samples (1.7Gb)  
Validation: 2M samples (140Mb)

```
import android.graphics.Bitmap;  
import com.facebook.common.references.ResourceReferenceCleaner;  
public class SimpleBitmapReleaser implements IResourceReferenceCleaner {  
    private static SimpleBitmapReleaser instance;  
    public static SimpleBitmapReleaser getInstance() {  
        if (instance == null) {  
            instance = new SimpleBitmapReleaser();  
        }  
    }  
}
```

```
0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1  
0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0  
0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0  
0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1  
0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1  
0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1 1 1  
0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1  
1
```

# Results: Tokenization

What is **perplexity**?

In the context of NMT:

Perplexity describes how "confused" a probability model is on a given test data set. A perfect model has perplexity 1.

Lower Perplexity is better

Meaning of perplexity value depends on **target vocab size**



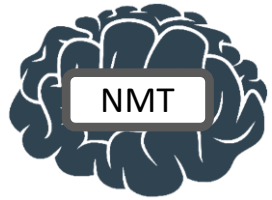
Bi-RNN  
7 epoch  
7 days  
Perple

Resource  
elements | R  
instance |  
instance |

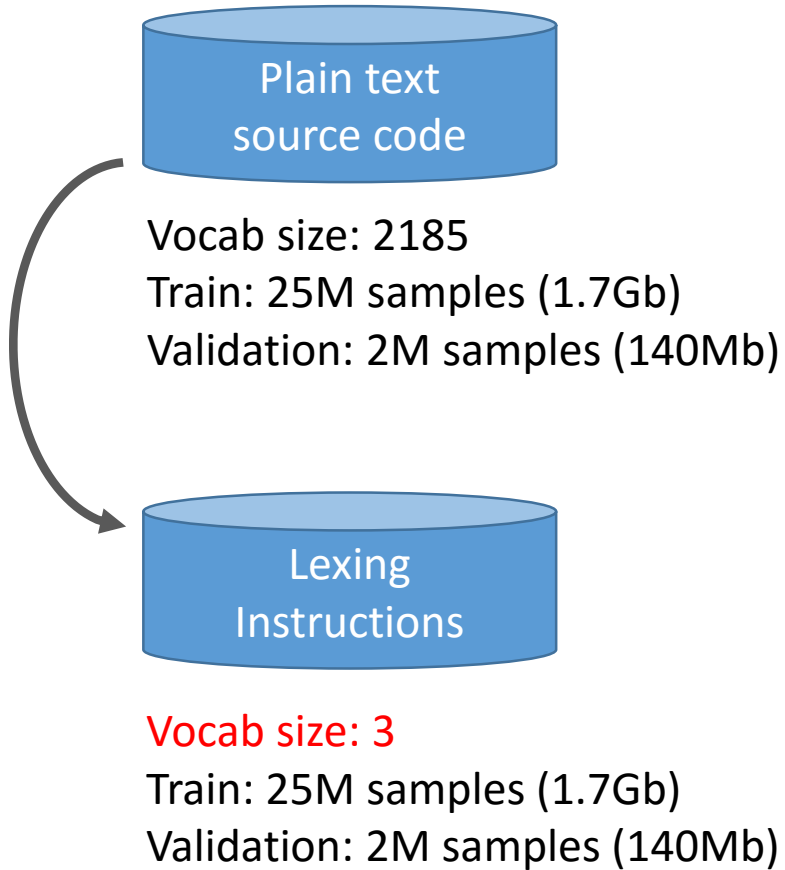
0 0 0 0 0 0  
0 0 0 0 0 1 1 0  
0 0 0 0 0 0 1 1  
0 0 0 0 0 0 0 1



# Results: Tokenization



Bi-RNN  
7 epochs  
7 days  
**Perplexity: 1.11**

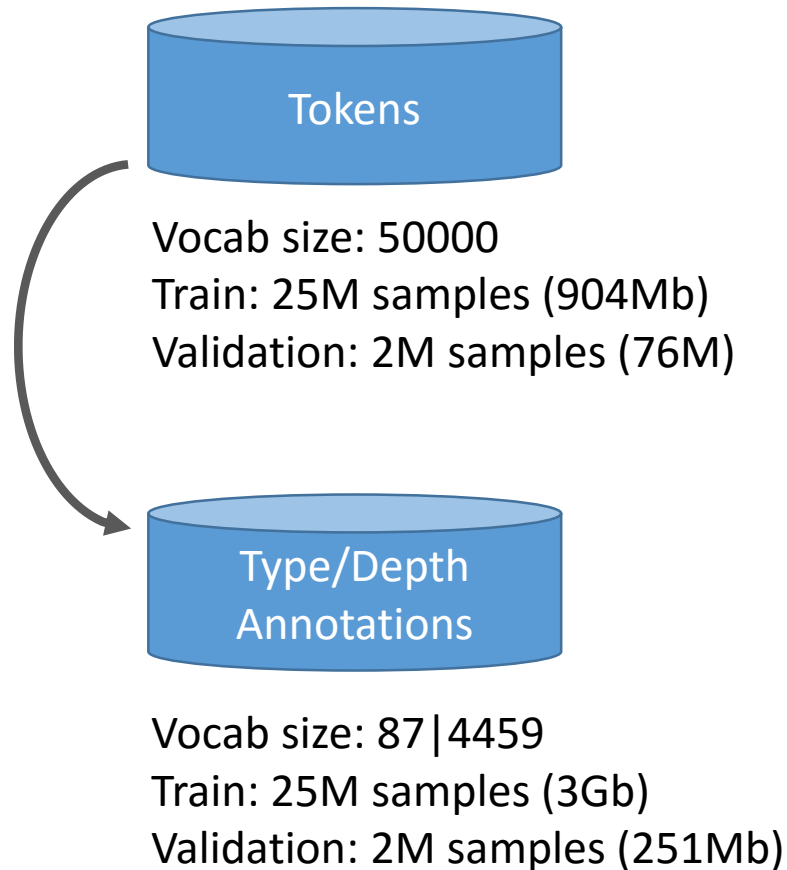


```
import android.graphics.Bitmap;  
import com.facebook.common.references.ResourceReferenceCleaner;  
public class SimpleBitmapReleaser implements IResourceReferenceCleaner {  
    private static SimpleBitmapReleaser instance;  
    public static SimpleBitmapReleaser getInstance() {  
        if (instance == null) {  
            instance = new SimpleBitmapReleaser();  
        }  
    }  
}
```

```
0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1  
0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0  
0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0  
0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1  
0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1  
0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1 1 1  
0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1  
1
```



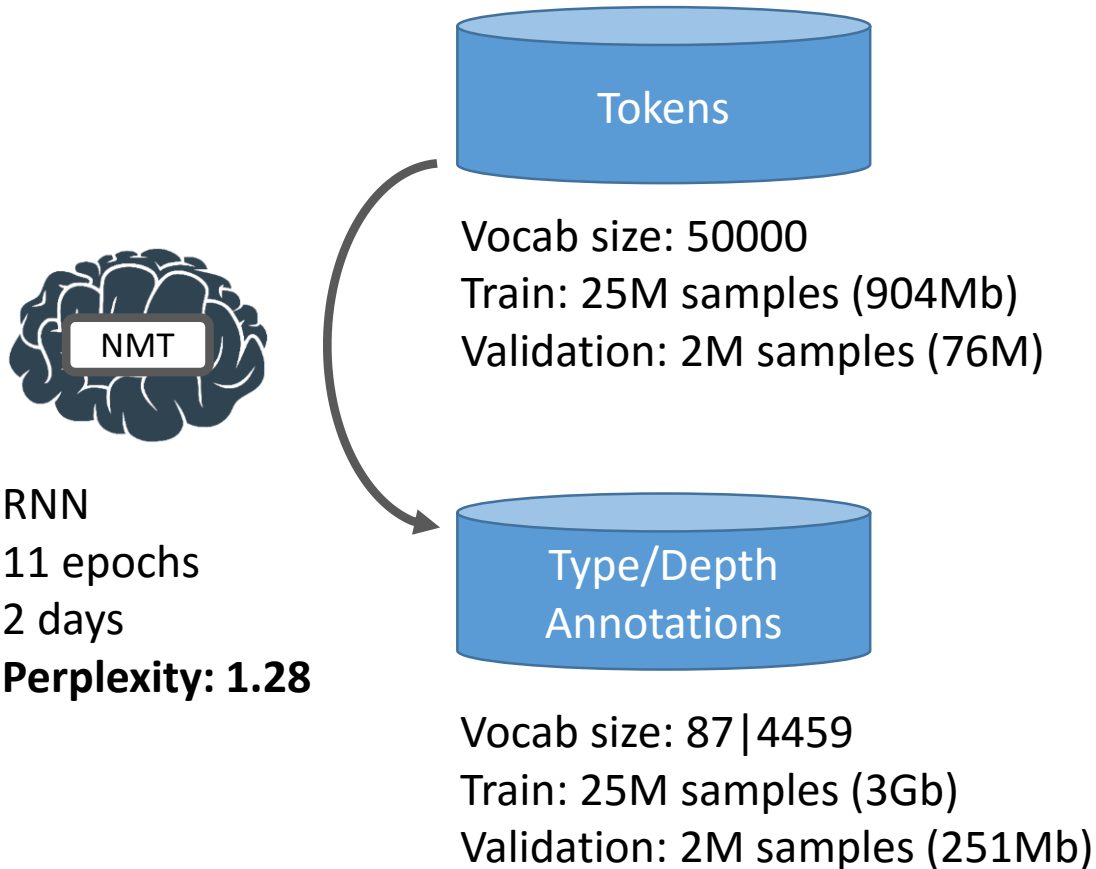
# Results: Token Annotation



```
import android . graphics . Bitmap ;  
import com . facebook . common . references . ResourceReleaser ;  
public class SimpleBitmapReleaser implements ResourceReleaser < Bitmap > {  
    private static SimpleBitmapReleaser sInstance ;  
    public static SimpleBitmapReleaser getInstance ( ) {  
        if ( sInstance == null ) {  
            sInstance = new SimpleBitmapReleaser ( ) ;  
        }  
    }  
}
```

```
ImportDeclaration|2 QualifiedName|3 QualifiedName|3 QualifiedName|3 QualifiedName|3 Quali  
ImportDeclaration|2 QualifiedName|3 QualifiedName|3 QualifiedName|3 QualifiedName|3 Quali  
ClassOrInterfaceModifier|3 ClassDeclaration|3 ClassDeclaration|3 ClassDeclaration|3 Class  
ClassOrInterfaceModifier|7 ClassOrInterfaceModifier|7 ClassOrInterfaceType|9 VariableDecl  
ClassOrInterfaceModifier|7 ClassOrInterfaceModifier|7 ClassOrInterfaceType|9 MethodDecla  
IfStatement|12 ParExpression|13 Primary|16 Expression|14 Literal|17 ParExpression|13  
Block|14
```

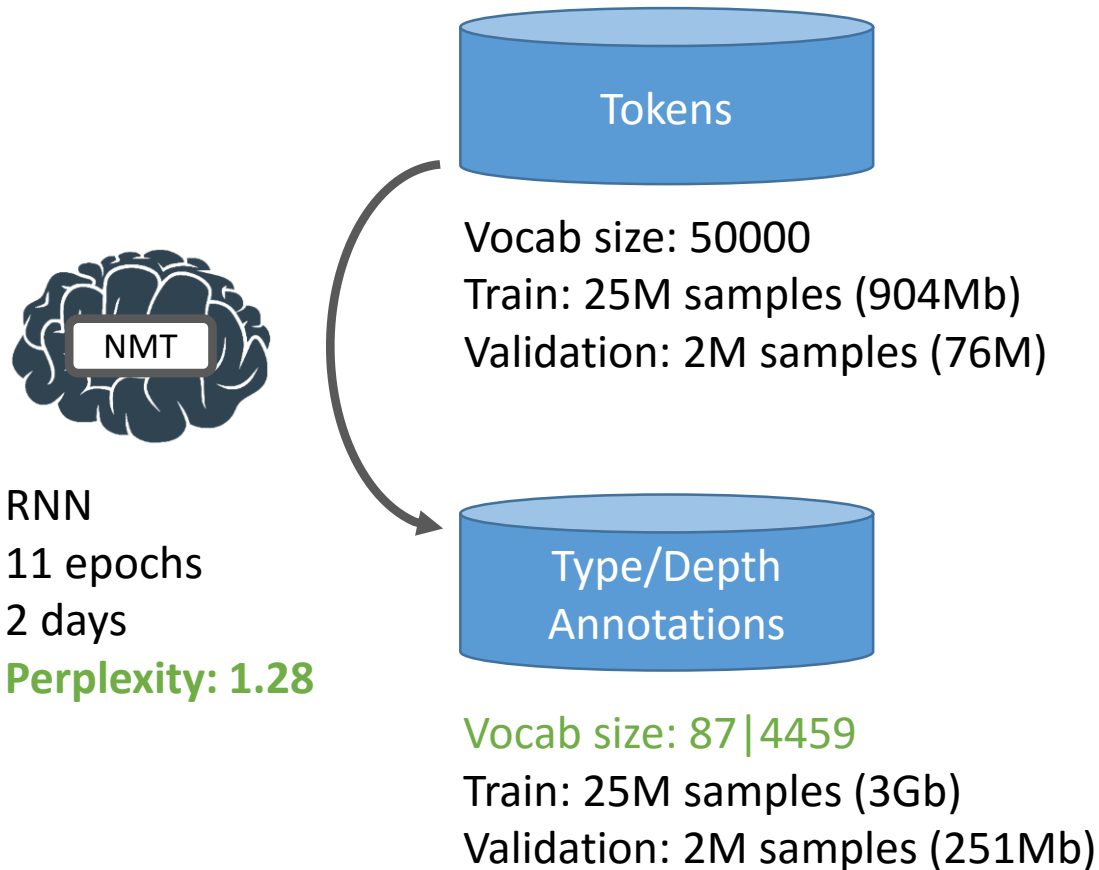
# Results: Token Annotation



```
import android . graphics . Bitmap ;  
import com . facebook . common . references . ResourceReleaser ;  
public class SimpleBitmapReleaser implements ResourceReleaser < Bitmap > {  
    private static SimpleBitmapReleaser sInstance ;  
    public static SimpleBitmapReleaser getInstance ( ) {  
        if ( sInstance == null ) {  
            sInstance = new SimpleBitmapReleaser ( ) ;  
        }  
    }  
}
```

```
ImportDeclaration|2 QualifiedName|3 QualifiedName|3 QualifiedName|3 QualifiedName|3 Quali  
ImportDeclaration|2 QualifiedName|3 QualifiedName|3 QualifiedName|3 QualifiedName|3 Quali  
ClassOrInterfaceModifier|3 ClassDeclaration|3 ClassDeclaration|3 ClassDeclaration|3 Class  
ClassOrInterfaceModifier|7 ClassOrInterfaceModifier|7 ClassOrInterfaceType|9 VariableDecl  
ClassOrInterfaceModifier|7 ClassOrInterfaceModifier|7 ClassOrInterfaceType|9 MethodDeclara  
IfStatement|12 ParExpression|13 Primary|16 Expression|14 Literal|17 ParExpression|13  
Block|14
```

# Results: Token Annotation



```
import android . graphics . Bitmap ;  
import com . facebook . common . references . ResourceReleaser ;  
public class SimpleBitmapReleaser implements ResourceReleaser < Bitmap > {  
    private static SimpleBitmapReleaser sInstance ;  
    public static SimpleBitmapReleaser getInstance ( ) {  
        if ( sInstance == null ) {  
            sInstance = new SimpleBitmapReleaser ( ) ;  
        }  
    }  
}
```

```
ImportDeclaration|2 QualifiedName|3 QualifiedName|3 QualifiedName|3 QualifiedName|3 Quali  
ImportDeclaration|2 QualifiedName|3 QualifiedName|3 QualifiedName|3 QualifiedName|3 Quali  
ClassOrInterfaceModifier|3 ClassDeclaration|3 ClassDeclaration|3 ClassDeclaration|3 Class  
ClassOrInterfaceModifier|7 ClassOrInterfaceModifier|7 ClassOrInterfaceType|9 VariableDecl  
ClassOrInterfaceModifier|7 ClassOrInterfaceModifier|7 ClassOrInterfaceType|9 MethodDeclara  
IfStatement|12 ParExpression|13 Primary|16 Expression|14 Literal|17 ParExpression|13  
Block|14
```

# Results: Token Annotation

A successful example:

```
List<Throwable> errors = TestHelper.trackPluginErrors();
```

```
000110000000011 000001 1 0000000001100000000000000000011111
```

```
[ClassOrInterfaceType|14] [TypeArguments|15] [ClassOrInterfaceType|18] [TypeArguments|15]  
[VariableDeclaratorId|15] [VariableDeclarator|14]  
[Primary|19] [Expression|17] [Expression|17] [Expression|16] [Expression|16]  
[LocalVariableDeclarationStatement|11]
```

Vocab size: 87|4459

Train: 25M samples (3Gb)

Validation: 2M samples (251Mb)

IfStatement|12 ParExpression|13 Primary|16 Expression|14 Literal|17 ParExpression|13  
Block|14

# Take-home messages:

- NN can learn to "read" code (tokens / syntactic elements)
  - What else could we teach? Type resolution? Calls & attribute access? Inheritance?
  - Could we follow the "human path" of learning to program to teach an AI?

# Take-home messages:

- NN can learn to "read" code (tokens / syntactic elements)
  - What else could we teach? Type resolution? Calls & attribute access? Inheritance?
  - Could we follow the "human path" of learning to program to teach an AI?
- *"If only we had good data"*
  - Bug reports, commit messages etc. are *still* unstructured. This needs to change if we want to leverage deep learning in SE and PC.



ICPC 2017  
EARLY RESEARCH ACHIEVEMENTS

THANKS FOR LISTENING

Data creation tool: [t.uzh.ch/Hb](http://t.uzh.ch/Hb)

Paper: [t.uzh.ch/Hc](http://t.uzh.ch/Hc)



University of  
Zurich <sup>UZH</sup>



Carol V. Alexandru, Sebastiano Panichella, Harald C. Gall  
{alexandru,panichella,gall}@ifi.uzh.ch

23. May 2017