LangDevCon_2018_{

↦ **Machine Learning**\n

↦ ↦ **meets**\n

↦ ↦ **Code**\n

↦ **Formatting**\n

}

Sebastian

itemis

Holger

Formatting

Beautiful Code!

Valid, but not Readable!

```
grammar org.eclipse.xtext.example.domainmodel.Domainmodel with
org.eclipse.xtext.xbase.Xbase generate domainmodel "http://www.xtext.org/
example/Domainmodel" DomainModel: importSection=XImportSection?
elements+=AbstractElement*; AbstractElement: PackageDeclaration|
Entity;PackageDeclaration:'package'
name=QualifiedName'{'elements+=AbstractElement*'}';Entity:'entity'name=ValidID('
extends'superType=JvmParameterizedTypeReference)?'{'features+=Feature*'}';Featur
e:Property|
Operation;Property:name=ValidID':'type=JvmTypeReference;Operation:'op'name=Valid
ID'('(params+=FullJvmFormalParameter(','params+=FullJvmFormalParameter)*)?')'(':
' type=JvmTypeReference)?body=XBlockExpression;
```

Same Code, but Readable!

```
grammar org.eclipse.xtext.example.domainmodel.Domainmodel with org.eclipse.xtext.xbase.Xbase

generate domainmodel "http://www.xtext.org/example/Domainmodel"

DomainModel:
    importSection=XImportSection? elements+=AbstractElement*;

AbstractElement:
      PackageDeclaration
    | Entity;

PackageDeclaration:
    'package' name=QualifiedName '{'
        elements+=AbstractElement*
    '}';

Entity:
    'entity' name=ValidID ('extends' superType=JvmParameterizedTypeReference)? '{' features+=Feature* '}';

Feature:
      Property
    | Operation;

Property:
    name=ValidID ':' type=JvmTypeReference;

Operation:
    'op' name=ValidID '(' (params+=FullJvmFormalParameter (',' params+=FullJvmFormalParameter)*)? ')'
    (':' type=JvmTypeReference)? body=XBlockExpression;
```

Same Code, but Readable!

```
grammar org.eclipse.xtext.example.domainmodel.Domainmodel with org.eclipse.xtext.xbase.Xbase

generate domainmodel "http://www.xtext.org/example/Domainmodel"

DomainModel:
    importSection=XImportSection? elements+=AbstractElement*;

AbstractElement:
      PackageDeclaration
    | Entity;

PackageDeclaration:
    'package' name=QualifiedName '{'
        elements+=AbstractElement*
    '}';

Entity:
    'entity' name=ValidID ('extends' superType=JvmParameterizedTypeReference)? '{' features+=Feature* '}';

Feature:
      Property
    | Operation;

Property:
    name=ValidID ':' type=JvmTypeReference;

Operation:
    'op' name=ValidID '(' (params+=FullJvmFormalParameter (',' params+=FullJvmFormalParameter)*)? ')'
    (':' type=JvmTypeReference)? body=XBlockExpression;
```

Spaces

Same Code, but Readable!

```
grammar org.eclipse.xtext.example.domainmodel.Domainmodel with org.eclipse.xtext.xbase.Xbase

generate domainmodel "http://www.xtext.org/example/Domainmodel"

DomainModel:
    importSection=XImportSection? elements+=AbstractElement*;

AbstractElement:
      PackageDeclaration
    | Entity;

PackageDeclaration:
    'package' name=QualifiedName '{'
        elements+=AbstractElement*
    '}';

Entity:
    'entity' name=ValidID ('extends' superType=JvmParameterizedTypeReference)? '{' features+=Feature* '}';

Feature:
      Property
    | Operation;

Property:
    name=ValidID ':' type=JvmTypeReference;

Operation:
    'op' name=ValidID '(' (params+=FullJvmFormalParameter (',' params+=FullJvmFormalParameter)*)? ')'
    (':' type=JvmTypeReference)? body=XBlockExpression;
```

Linebreak

Same Code, but Readable!

```
grammar org.eclipse.xtext.example.domainmodel.Domainmodel with org.eclipse.xtext.xbase.Xbase

generate domainmodel "http://www.xtext.org/example/Domainmodel"

DomainModel:
    importSection=XImportSection? elements+=AbstractElement*;

AbstractElement:
      PackageDeclaration
    | Entity;

PackageDeclaration:
    'package' name=QualifiedName '{'
        elements+=AbstractElement*
    '}';

Entity:
    'entity' name=ValidID ('extends' superType=JvmParameterizedTypeReference)? '{' features+=Feature* '}';

Feature:
      Property
    | Operation;

Property:
    name=ValidID ':' type=JvmTypeReference;

Operation:
    'op' name=ValidID '(' (params+=FullJvmFormalParameter (',' params+=FullJvmFormalParameter)*)? ')'
    (':' type=JvmTypeReference)? body=XBlockExpression;
```

Indentation

Same Code, but Readable!

```
grammar org.eclipse.xtext.example.domainmodel.Domainmodel with org.eclipse.xtext.xbase.Xbase

generate domainmodel "http://www.xtext.org/example/Domainmodel"

DomainModel:
    importSection=XImportSection? elements+=AbstractElement*;

AbstractElement:
      PackageDeclaration
    | Entity;

PackageDeclaration:
    'package' name=QualifiedName '{'
        elements+=AbstractElement*
    '}';

Entity:
    'entity' name=ValidID ('extends' superType=JvmParameterizedTypeReference)? '{' features+=Feature* '}';

Feature:
      Property
    | Operation;

Property:
    name=ValidID ':' type=JvmTypeReference;

Operation:
    'op' name=ValidID '(' (params+=FullJvmFormalParameter (',' params+=FullJvmFormalParameter)*)? ')'
    (':' type=JvmTypeReference)? body=XBlockExpression;
```

Alignment

But there are already formatters out there!

But there are already formatters out there!

But there are already formatters out there!

Academic	Standard	Team
$38*	**$98***	**$2,700***
Jindent - Java Code Formatter	Jindent - Java Code Formatter	Jindent - Java Code Formatter
Non-commercial Single User License	Commercial Single User License	Commercial Site License
1 User	1 User	50 Users
✓ Money Back Guarantee ✓ Free Minor Updates ✓ Technical Support ✓ Console Tool, Commander, Customizer ✓ All Plugins	✓ Money Back Guarantee ✓ Free Minor Updates ✓ Technical Support ✓ Console Tool, Commander, Customizer ✓ All Plugins	✓ Money Back Guarantee ✓ Free Minor Updates ✓ Technical Support ✓ Console Tool, Commander, Customizer ✓ All Plugins
PayPal Buy Now · share*it! Buy Now	PayPal Buy Now · share*it! Buy Now	PayPal Buy Now · share*it! Buy Now

*Please note that German customers and customers from the European Union without VAT-ID pay 19% VAT.

What’s the Point?

Formatting is a Matter of *Taste*!

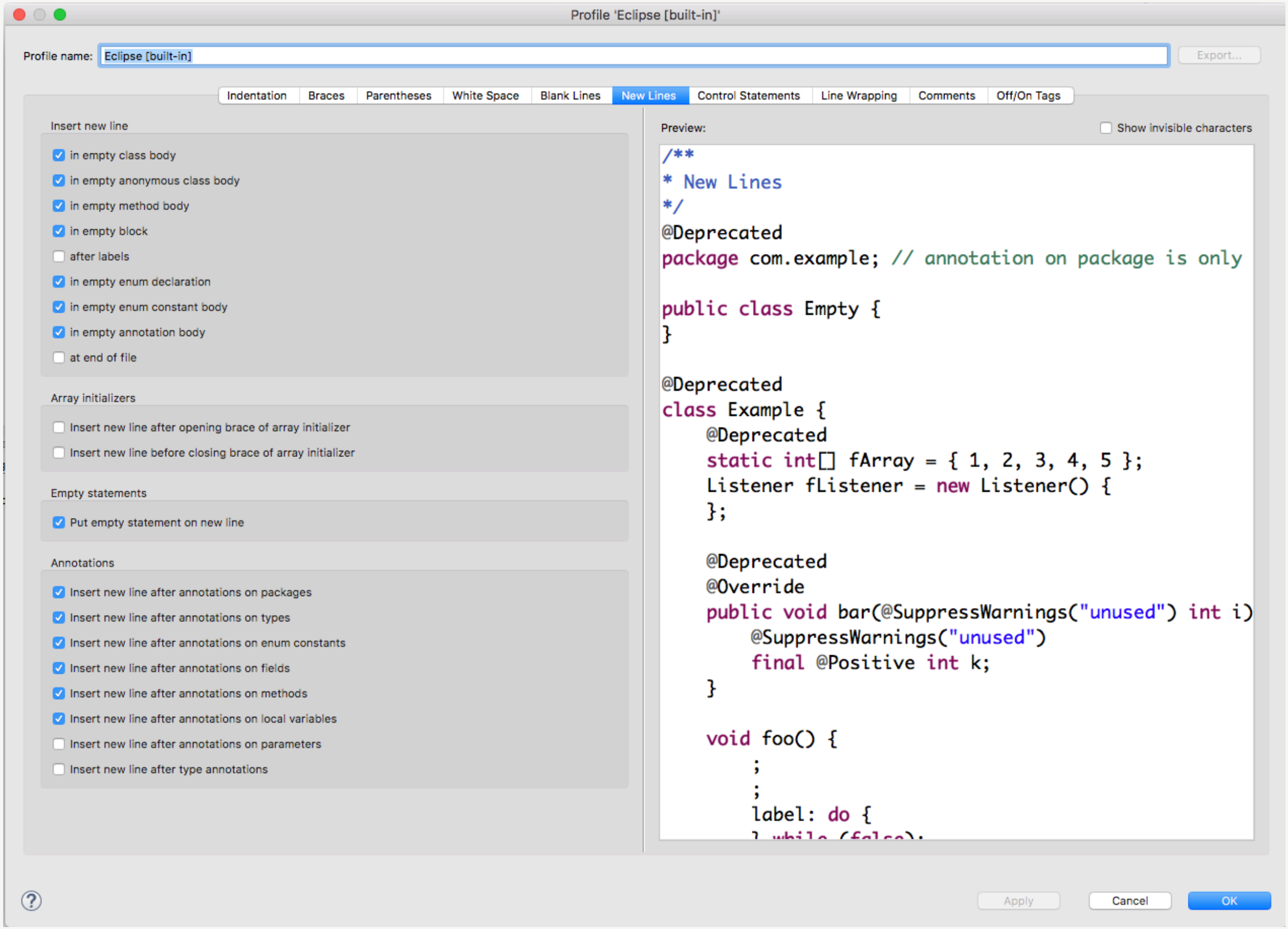
Natural Formatting

```
public NotificationChain myMethod(EObject otherEnd, int featureID) {
    switch (featureID) {
      ….
    }
    return super.eInverseRemove(otherEnd, featureID);
}
```

EMF

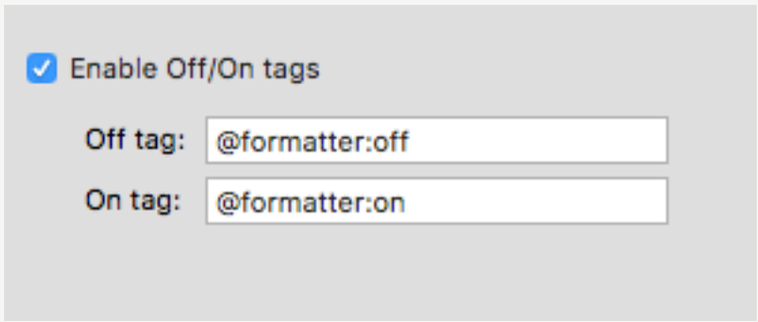
```
public NotificationChain myMethod(EObject otherEnd, int featureID)
{
    switch (featureID)
    {
      ….
    }
    return super.eInverseRemove(otherEnd, featureID);
}
```

Eclipse JDT


Profile 'Eclipse [built-in]'
Profile name: Eclipse [built-in]
Export...
Indentation
Braces
Parentheses
White Space
Blank Lines
New Lines
Control Statements
Line Wrapping
Comments
Off/On Tags
Insert new line
in empty class body
in empty anonymous class body
in empty method body
in empty block
after labels
in empty enum declaration
in empty enum constant body
in empty annotation body
at end of file
Array initializers
Insert new line after opening brace of array initializer
Insert new line before closing brace of array initializer
Empty statements
Put empty statement on new line
Annotations
Insert new line after annotations on packages
Insert new line after annotations on types
Insert new line after annotations on enum constants
Insert new line after annotations on fields
Insert new line after annotations on methods
Insert new line after annotations on local variables
Insert new line after annotations on parameters
Insert new line after type annotations
Preview:
Show invisible characters
/**
* New Lines
*/
@Deprecated
package com.example; // annotation on package is only
public class Empty {
}
@Deprecated
class Example {
@Deprecated
static int[] fArray = { 1, 2, 3, 4, 5 };
Listener fListener = new Listener() {
};
@Deprecated
@Override
public void bar(@SuppressWarnings("unused") int i)
@SuppressWarnings("unused")
final @Positive int k;
}
void foo() {
;
;
label: do {
Apply
Cancel
OK

What, if there is *no Option*?

Manual Formatting FTW




```
/**
 * @formatter:off
 */
public void foo() {};
/**
 * @formatter:on
 */
```

What, if there is *no Formatter*?

Domain Specific Languages

The Hardest Program I've Ever Written

Bob Nystrom
Author of dartfmt

http://journal.stuffwithstuff.com/2015/09/08/the-hardest-program-ive-ever-written/

Rich API to Define Formatters

```
def dispatch void format(Entity entity, extension IFormattableDocument document) {
    val open = entity.regionFor.keyword("{")
    val close = entity.regionFor.keyword("}")
    entity.regionFor.feature(ABSTRACT_ELEMENT__NAME).surround[oneSpace]
    entity.superType.surround[oneSpace]
    open.append[newLine]
    interior(open, close)[indent]
    format(entity.getSuperType(), document);
    for (Feature feature : entity.features) {
        feature.format
        feature.append[setNewLines(1, 1, 2)]
    }
}
```

Options are *not* coming *for free*!

What if **formatting** *would work* **by** *using an* **example***?*

Towards a **Universal Code Formatter** through Machine Learning

06/2016

Terence Parr, Jurgen Vinju

https://arxiv.org/abs/1606.08866

CodeBuff

Mimic developer during formatting

(Invalid) Simplification of ML / AI

$$\begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ .. \\ n_k \end{pmatrix} \mapsto \begin{pmatrix} m_1 \\ .. \end{pmatrix}$$

CodeBuff

Xtext Grammar Language

```
grammar org.eclipse.xtext.example.domainmodel.Domainmodel with org.eclipse.xtext.xbase.Xbase

generate domainmodel "http://www.xtext.org/example/Domainmodel"

DomainModel:
    importSection=XImportSection? elements+=AbstractElement*;

AbstractElement:
      PackageDeclaration
    | Entity;

PackageDeclaration:
    'package' name=QualifiedName '{'
        elements+=AbstractElement*
    '}';

Entity:
    'entity' name=ValidID ('extends' superType=JvmParameterizedTypeReference)? '{' features+=Feature* '}';

Feature:
      Property
    | Operation;

Property:
    name=ValidID ':' type=JvmTypeReference;

Operation:
    'op' name=ValidID '(' (params+=FullJvmFormalParameter (',' params+=FullJvmFormalParameter)*)? ')'
    (':' type=JvmTypeReference)? body=XBlockExpression;
```

```
PackageDeclaration:
  'package' name=QualifiedName '{'
        elements+=AbstractElement*
'}';
```

split(Document)↦Seq<*Token*>

```
PackageDeclaration:
    'package' name=QualifiedName '{'
        elements+=AbstractElement*
    '}';
```

format(Token)↦*TextChange*

```
PackageDeclaration:
  'package' name=QualifiedName '{'
        elements+=AbstractElement*
  '}';
```

format(Token)↦*TextChange*

```
PackageDeclaration:
  'package' name=QualifiedName '{'\n
  .... ....elements+=AbstractElement*
  '}';
```

$$format(\text{Token}) \mapsto TextChange$$

$$prop(Token) \mapsto TokenProp$$

$$match(TokenProp) \mapsto WS$$

$$apply(WS) \mapsto NewText$$

WS:=(1. sp
2. nl
3. indent(*Token*)
4. align(*Token*)
5. nothing)

$$WS := (sp, nl)$$

$$WS := (8, 1)$$

```
PackageDeclaration:
  'package' name=QualifiedName '{'\n
  .... ....elements+=AbstractElement*
  '}';
```

$$prop(Token) \mapsto TokenProp$$

$$match(TokenProp) \mapsto (sp, nl)$$

$$apply(sp, nl) \mapsto NewText$$

$$prop(Token) \mapsto TokenProp$$

$$TokenProp := Vector<..>$$

TokenProp:=*Vector*<..>

1. Token Type (String, Identifier, …)
2. Prev Token, Next Token
3. Paired Token ({ + }, : + ;, [+])
4. Parent Node Type
5. List Index and List Size

$$\mathit{TokenProp} := V\langle 21 \rangle$$

$$prop(Token) \mapsto V\langle 21\rangle$$

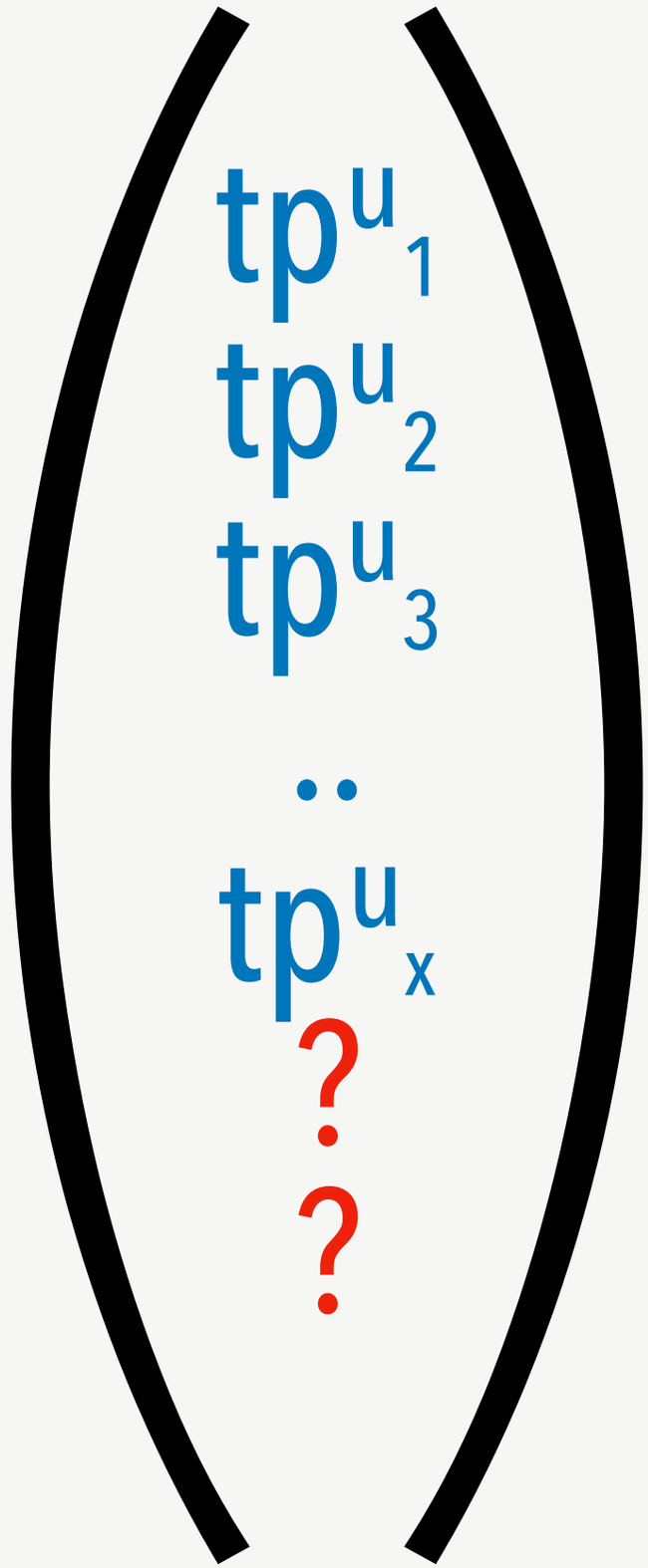
$$match\ (V\langle 21\rangle) \mapsto (sp, nl)$$

$$apply(sp, nl) \mapsto NewText$$

1. Train Corpus Documents
    a. Compute *prop*(Token)↦*tp*
    b. Store (*tp*, *sp*, *nl*)

$$
\begin{matrix}
tp^{t_1}{}_1 & tp^{t_2}{}_1 & \cdots & tp^{t_c}{}_1 \\
tp^{t_1}{}_2 & tp^{t_2}{}_2 & & tp^{t_c}{}_2 \\
tp^{t_1}{}_3 & tp^{t_2}{}_3 & & tp^{t_c}{}_3 \\
\vdots & \vdots & & \vdots \\
tp^{t_1}{}_x & tp^{t_2}{}_x & & tp^{t_c}{}_x \\
sp^{t_1}{}_1 & sp^{t_2}{}_1 & & sp^{t_c}{}_1 \\
nl^{t_1}{}_2 & nl^{t_2}{}_2 & & nl^{t_c}{}_2
\end{matrix}
$$

1. Train Corpus Documents
    a. Compute *prop*(Token)↦*tp*
    b. Store (*tp*, *sp*, *nl*)
2. Format Unknown Document


tp^{u}_{1}
tp^{u}_{2}
tp^{u}_{3}
..
tp^{u}_{x}
?
?

1. Train Corpus Documents
    a. Compute *prop*(Token)↦*tp*
    b. Store (*tp*, *sp*, *nl*)
2. Format Unknown Document
    a. Compute *prop*(Token)↦tp^u
    b. Find *match*(tp^u)↦ (sp^u, nl^u)
    c. Apply WS to Document

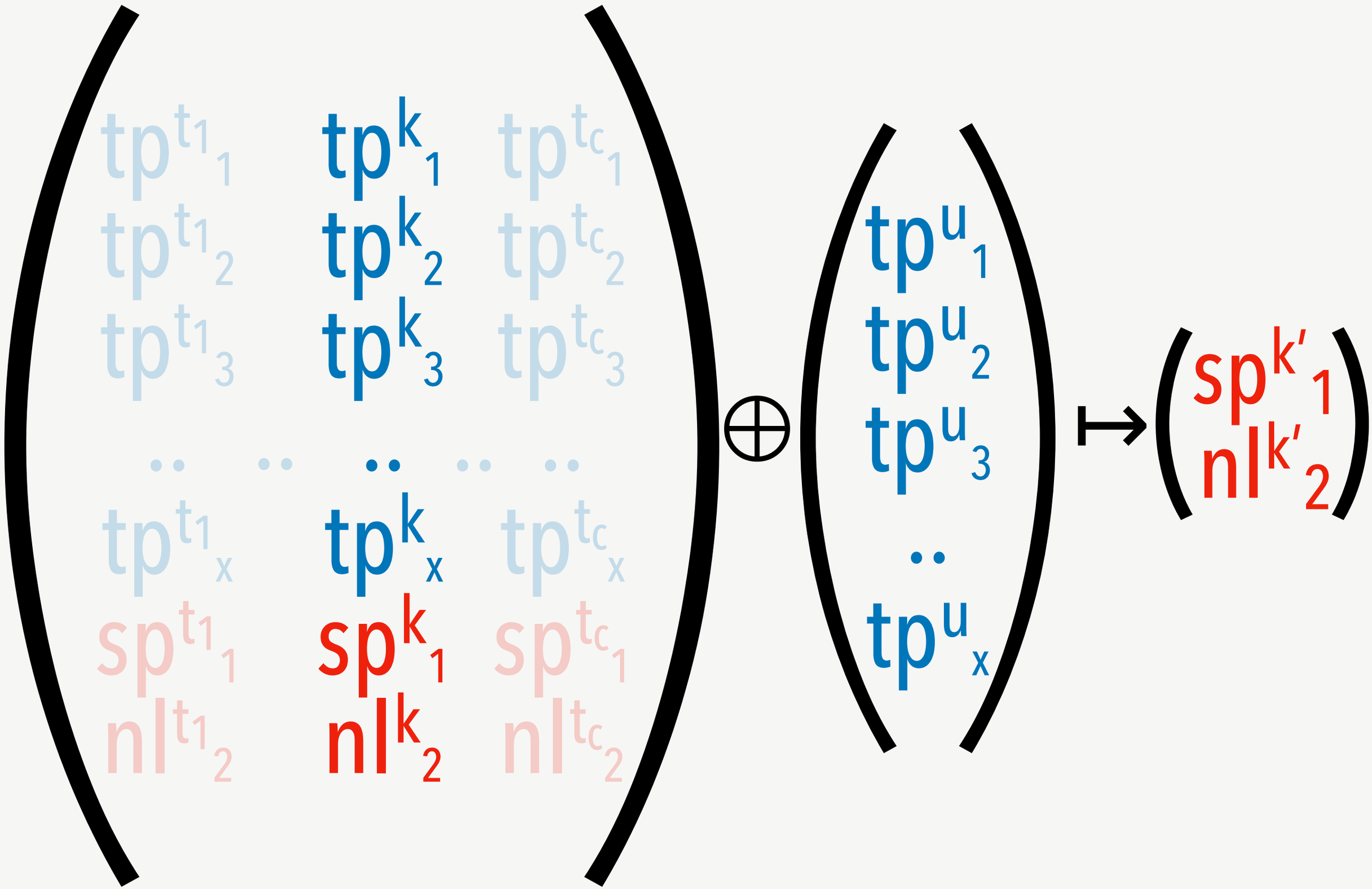
$$
\begin{pmatrix}
tp^{t_1}_1 & tp^{t_2}_1 & \cdots & tp^{t_c}_1 \\
tp^{t_1}_2 & tp^{t_2}_2 & \cdots & tp^{t_c}_2 \\
tp^{t_1}_3 & tp^{t_2}_3 & \cdots & tp^{t_c}_3 \\
\cdots & \cdots & \cdots & \cdots \\
tp^{t_1}_x & tp^{t_2}_x & \cdots & tp^{t_c}_x \\
sp^{t_1}_1 & sp^{t_2}_1 & \cdots & sp^{t_c}_1 \\
nl^{t_1}_2 & nl^{t_2}_2 & \cdots & nl^{t_c}_2
\end{pmatrix}
\oplus
\begin{pmatrix}
tp^{u}_1 \\
tp^{u}_2 \\
tp^{u}_3 \\
\cdots \\
tp^{u}_x
\end{pmatrix}
\mapsto
\begin{pmatrix}
sp^{u}_1 \\
nl^{u}_2
\end{pmatrix}
$$

$$match(tp^u) \mapsto (sp^u, nl^u)$$

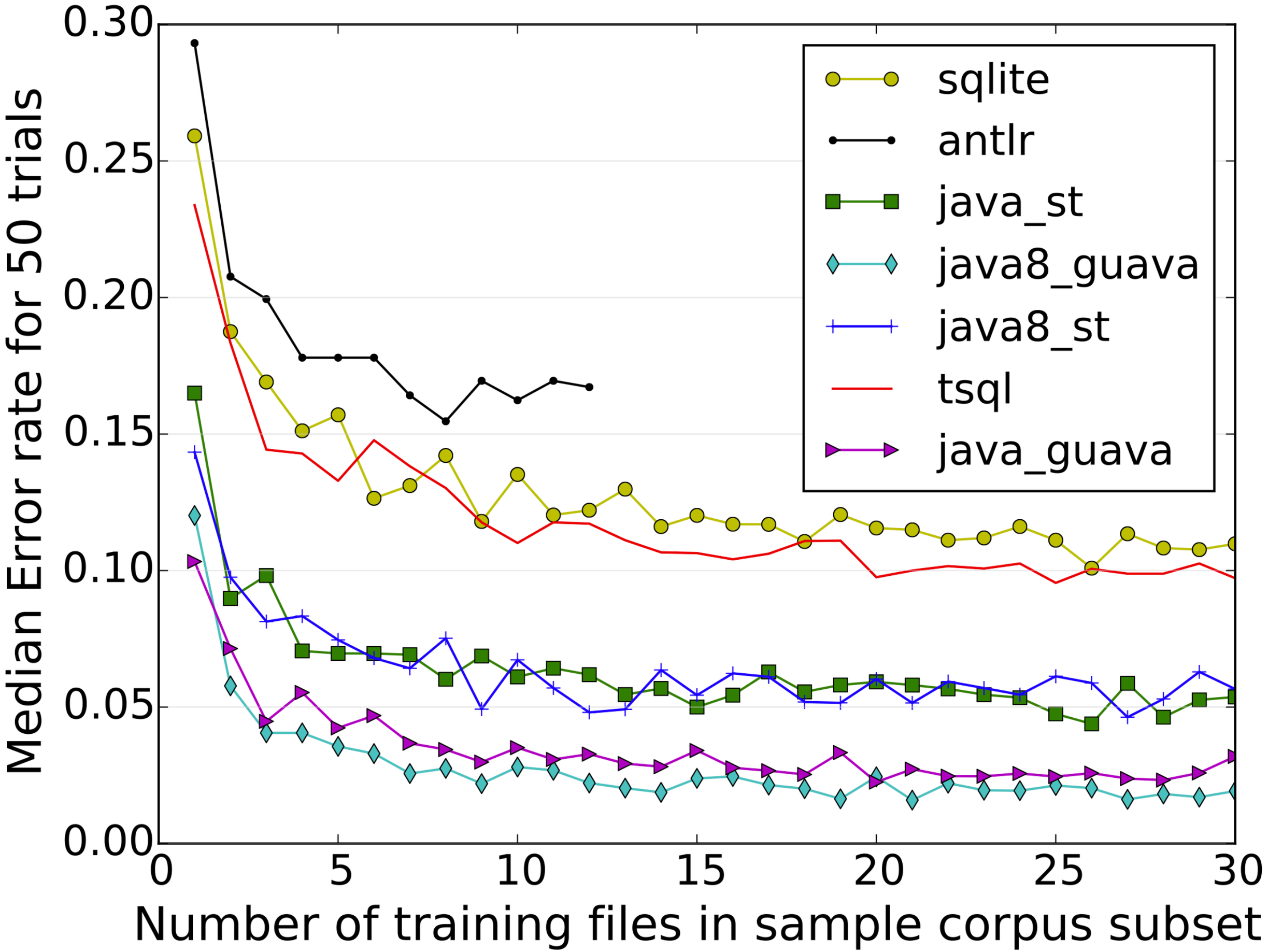
K Nearest Neighbour

KNN

1. Select k most similar $V\langle 21 \rangle$ as v_s
2. Select best $(sp^u, nl^u)_b$ from v_s by weigh function b
3. $k = 11$


$tp^{t_1}_1$ $tp^{t_1}_2$ $tp^{t_1}_3$ tp^{k}_1 tp^{k}_2 tp^{k}_3 $tp^{t_c}_1$ $tp^{t_c}_2$ $tp^{t_c}_3$
$tp^{t_1}_x$ tp^{k}_x $tp^{t_c}_x$
$sp^{t_1}_1$ sp^{k}_1 $sp^{t_c}_1$
$nl^{t_1}_2$ nl^{k}_2 $nl^{t_c}_2$
$\oplus$
tp^{u}_1 tp^{u}_2 tp^{u}_3 tp^{u}_x
$\mapsto$
$sp^{k'}_1$ $nl^{k'}_2$

Demo


Median Error rate for 50 trials
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
25
30
Number of training files in sample corpus subset
sqlite
antlr
java_st
java8_guava
java8_st
tsql
java_guava

Misclassification Error Rate

0.25
0.20
0.15
0.10
0.05
0.00

java_st n=59
java8_st n=59
java_guava n=511
java8_guava n=511
antlr n=12
sqlite n=36
tsql n=36
sqlite_noisy n=36
tsql_noisy n=36

Grammar and corpus size

Done!