

LDyH-PR4

Generado por Doxygen 1.12.0

1 Práctica 4	1
1.0.1 Código de la práctica 4 de la asignatura Laboratorio de Desarrollo y Herramientas	1
1.1 Notes:	1
1.2 Links:	1
1.2.1 The OpenNLP Manual	1
1.2.2 Where to find POS Tagger training data in TreeBank II format	1
1.2.3 A source for Chunker training data	1
1.2.4 A good source for general English text to use for training tasks	1
2 Índice de espacios de nombres	3
2.1 Lista de paquetes	3
3 Índice de clases	5
3.1 Lista de clases	5
4 Índice de archivos	7
4.1 Lista de archivos	7
5 Documentación de espacios de nombres	9
5.1 Paquete org.fogbeam.example.opennlp	9
5.2 Paquete org.fogbeam.example.opennlp.training	9
6 Documentación de clases	11
6.1 Referencia de la clase org.fogbeam.example.opennlp.ChunkerMain	11
6.1.1 Documentación de funciones miembro	11
6.1.1.1 main()	11
6.2 Referencia de la clase org.fogbeam.example.opennlp.training.ChunkerTrainer	12
6.2.1 Documentación de funciones miembro	12
6.2.1.1 main()	12
6.3 Referencia de la clase org.fogbeam.example.opennlp.DocumentClassifierMain	12
6.3.1 Documentación de funciones miembro	12
6.3.1.1 main()	12
6.4 Referencia de la clase org.fogbeam.example.opennlp.training.DocumentClassifierTrainer	13
6.4.1 Documentación de funciones miembro	13
6.4.1.1 main()	13
6.5 Referencia de la clase org.fogbeam.example.opennlp.NameFinderMain	14
6.5.1 Documentación de funciones miembro	14
6.5.1.1 main()	14
6.6 Referencia de la clase org.fogbeam.example.opennlp.training.NameFinderTrainer	14
6.6.1 Documentación de funciones miembro	15
6.6.1.1 main()	15
6.7 Referencia de la clase org.fogbeam.example.opennlp.ParserMain	16
6.7.1 Documentación de funciones miembro	16
6.7.1.1 main()	16

6.8 Referencia de la clase <code>org.fogbeam.example.opennlp.training.ParserTrainer</code>	17
6.8.1 Documentación de funciones miembro	17
6.8.1.1 <code>main()</code>	17
6.9 Referencia de la clase <code>org.fogbeam.example.opennlp.PartOfSpeechTaggerMain</code>	17
6.9.1 Documentación de funciones miembro	17
6.9.1.1 <code>main()</code>	17
6.10 Referencia de la clase <code>org.fogbeam.example.opennlp.training.PartOfSpeechTaggerTrainer</code>	18
6.10.1 Documentación de funciones miembro	18
6.10.1.1 <code>main()</code>	18
6.11 Referencia de la clase <code>org.fogbeam.example.opennlp.SentenceDetectionMain</code>	18
6.11.1 Documentación de funciones miembro	18
6.11.1.1 <code>main()</code>	18
6.12 Referencia de la clase <code>org.fogbeam.example.opennlp.training.SentenceDetectionTrainer</code>	19
6.12.1 Documentación de funciones miembro	19
6.12.1.1 <code>main()</code>	19
6.13 Referencia de la clase <code>org.fogbeam.example.opennlp.TokenizerMain</code>	20
6.13.1 Documentación de funciones miembro	20
6.13.1.1 <code>main()</code>	20
6.14 Referencia de la clase <code>org.fogbeam.example.opennlp.training.TokenizerTrainer</code>	20
6.14.1 Documentación de funciones miembro	20
6.14.1.1 <code>main()</code>	20
7 Documentación de archivos	23
7.1 Referencia del archivo <code>README.md</code>	23
7.2 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/ChunkerMain.java</code>	23
7.3 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/DocumentClassifierMain.java</code>	23
7.3.1 Descripción detallada	23
7.4 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/NameFinderMain.java</code>	24
7.4.1 Descripción detallada	24
7.5 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/ParserMain.java</code>	24
7.5.1 Descripción detallada	24
7.6 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/PartOfSpeechTaggerMain.java</code>	25
7.6.1 Descripción detallada	25
7.7 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/SentenceDetectionMain.java</code>	25
7.7.1 Descripción detallada	25
7.8 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/TokenizerMain.java</code>	26
7.8.1 Descripción detallada	26
7.9 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/training/ChunkerTrainer.java</code>	26
7.9.1 Descripción detallada	26
7.10 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/training/DocumentClassifier↵ Trainer.java</code>	27
7.10.1 Descripción detallada	27
7.11 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/training/NameFinderTrainer.java</code>	27

7.11.1 Descripción detallada	27
7.12 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/training/ParserTrainer.java</code> . . .	28
7.13 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/training/PartOfSpeechTagger← Trainer.java</code>	28
7.13.1 Descripción detallada	28
7.14 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/training/SentenceDetection← Trainer.java</code>	29
7.14.1 Descripción detallada	29
7.15 Referencia del archivo <code>src/main/java/org/fogbeam/example/opennlp/training/TokenizerTrainer.java</code> .	29
7.15.1 Descripción detallada	29
Índice alfabético	31

Capítulo 1

Práctica 4

1.0.1. Código de la práctica 4 de la asignatura Laboratorio de Desarrollo y Herramientas.

Code to accompany Phil's June 2015 Tri-JUG presentation on OpenNLP.

1.1. Notes:

TBD...

1.2. Links:

1.2.1. The OpenNLP Manual

- <http://opennlp.apache.org/documentation/1.5.3/manual/opennlp.html>

1.2.2. Where to find POS Tagger training data in TreeBank II format

- <http://stackoverflow.com/questions/8949517/is-there-any-treebank-for-free>

1.2.3. A source for Chunker training data

- <http://www.cnts.ua.ac.be/conll2000/chunking/>

1.2.4. A good source for general English text to use for training tasks

<http://www.gutenberg.org>

Capítulo 2

Índice de espacios de nombres

2.1. Lista de paquetes

Estos son los paquetes con breves descripciones (si están disponibles):

org.fogbeam.example.opennlp	9
org.fogbeam.example.opennlp.training	9

Capítulo 3

Índice de clases

3.1. Lista de clases

Lista de clases, estructuras, uniones e interfaces con breves descripciones:

org.fogbeam.example.opennlp.ChunkerMain	11
org.fogbeam.example.opennlp.training.ChunkerTrainer	12
org.fogbeam.example.opennlp.DocumentClassifierMain	12
org.fogbeam.example.opennlp.training.DocumentClassifierTrainer	13
org.fogbeam.example.opennlp.NameFinderMain	14
org.fogbeam.example.opennlp.training.NameFinderTrainer	14
org.fogbeam.example.opennlp.ParserMain	16
org.fogbeam.example.opennlp.training.ParserTrainer	17
org.fogbeam.example.opennlp.PartOfSpeechTaggerMain	17
org.fogbeam.example.opennlp.training.PartOfSpeechTaggerTrainer	18
org.fogbeam.example.opennlp.SentenceDetectionMain	18
org.fogbeam.example.opennlp.training.SentenceDetectionTrainer	19
org.fogbeam.example.opennlp.TokenizerMain	20
org.fogbeam.example.opennlp.training.TokenizerTrainer	20

Capítulo 4

Índice de archivos

4.1. Lista de archivos

Lista de todos los archivos con breves descripciones:

src/main/java/org/fogbeam/example/opennlp/ ChunkerMain.java	
Programa principal para realizar el análisis de fragmentos gramaticales (chunks) en una oración	23
src/main/java/org/fogbeam/example/opennlp/ DocumentClassifierMain.java	
Programa principal para clasificar texto en categorías usando un modelo preentrenado de OpenNLP	23
src/main/java/org/fogbeam/example/opennlp/ NameFinderMain.java	
Programa principal para la detección de nombres propios en texto utilizando OpenNLP	24
src/main/java/org/fogbeam/example/opennlp/ ParserMain.java	
Programa principal para el análisis sintáctico de oraciones utilizando OpenNLP	24
src/main/java/org/fogbeam/example/opennlp/ PartOfSpeechTaggerMain.java	
Programa principal para el etiquetado gramatical (POS tagging) de oraciones utilizando OpenNLP	25
src/main/java/org/fogbeam/example/opennlp/ SentenceDetectionMain.java	
Programa principal para la detección de oraciones utilizando OpenNLP	25
src/main/java/org/fogbeam/example/opennlp/ TokenizerMain.java	
Programa principal para la tokenización de múltiples archivos de texto usando OpenNLP	26
src/main/java/org/fogbeam/example/opennlp/training/ ChunkerTrainer.java	
Clase para entrenar un modelo de fragmentación gramatical (chunker) utilizando OpenNLP	26
src/main/java/org/fogbeam/example/opennlp/training/ DocumentClassifierTrainer.java	
Clase para entrenar un modelo de clasificación de documentos utilizando OpenNLP	27
src/main/java/org/fogbeam/example/opennlp/training/ NameFinderTrainer.java	
Clase para entrenar un modelo de reconocimiento de nombres propios utilizando OpenNLP	27
src/main/java/org/fogbeam/example/opennlp/training/ ParserTrainer.java	
Clase para entrenar un modelo de análisis sintáctico de oraciones utilizando OpenNLP	28
src/main/java/org/fogbeam/example/opennlp/training/ PartOfSpeechTaggerTrainer.java	
Clase para entrenar un modelo de etiquetado gramatical (Part-of-Speech) usando OpenNLP	28
src/main/java/org/fogbeam/example/opennlp/training/ SentenceDetectionTrainer.java	
Clase para entrenar un modelo de detección de oraciones usando OpenNLP	29
src/main/java/org/fogbeam/example/opennlp/training/ TokenizerTrainer.java	
Clase para entrenar un modelo de tokenización usando OpenNLP	29

Capítulo 5

Documentación de espacios de nombres

5.1. Paquete org.fogbeam.example.opennlp

Paquetes

- package [training](#)

Clases

- class [ChunkerMain](#)
- class [DocumentClassifierMain](#)
- class [NameFinderMain](#)
- class [ParserMain](#)
- class [PartOfSpeechTaggerMain](#)
- class [SentenceDetectionMain](#)
- class [TokenizerMain](#)

5.2. Paquete org.fogbeam.example.opennlp.training

Clases

- class [ChunkerTrainer](#)
- class [DocumentClassifierTrainer](#)
- class [NameFinderTrainer](#)
- class [ParserTrainer](#)
- class [PartOfSpeechTaggerTrainer](#)
- class [SentenceDetectionTrainer](#)
- class [TokenizerTrainer](#)

Capítulo 6

Documentación de clases

6.1. Referencia de la clase org.fogbeam.example.opennlp.ChunkerMain

Métodos públicos estáticos

- static void [main](#) (String[] args) throws Exception
Método principal del programa.

6.1.1. Documentación de funciones miembro

6.1.1.1. main()

```
static void org.fogbeam.example.opennlp.ChunkerMain.main (  
    String[] args) throws Exception [static]
```

Método principal del programa.

Este método carga un modelo de fragmentación, procesa una oración tokenizada y etiquetada gramaticalmente (POS tags) y genera etiquetas de fragmentos (chunk tags) para cada token.

Parámetros

<i>args</i>	Argumentos de línea de comandos (no utilizados en este programa).
-------------	---

Excepciones

<i>Exception</i>	En caso de que ocurra un error inesperado.
------------------	--

< Flujo de entrada para cargar el modelo de fragmentación.

< Modelo de fragmentación cargado desde el archivo.

< Etiquetas de fragmentos generadas para los tokens.

< Probabilidades asociadas a las etiquetas.

Los fragmentos generados contienen etiquetas que representan el tipo de estructura gramatical. Ejemplo:

- B-NP: Inicio de una frase nominal.
- I-NP: Continuación de una frase nominal.
- I-VP: Continuación de una frase verbal.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/org/fogbeam/example/opennlp/ChunkerMain.java`

6.2. Referencia de la clase

org.fogbeam.example.opennlp.training.ChunkerTrainer

Métodos públicos estáticos

- static void [main](#) (String[] args) throws Exception
Metodo principal para entrenar un modelo de fragmentación.

6.2.1. Documentación de funciones miembro

6.2.1.1. main()

```
static void org.fogbeam.example.opennlp.training.ChunkerTrainer.main (
    String[] args) throws Exception [static]
```

Metodo principal para entrenar un modelo de fragmentación.

Este metodo lee datos de entrenamiento en formato CoNLL2000, entrena un modelo de fragmentación utilizando OpenNLP y guarda el modelo generado en un archivo para su posterior uso.

Parámetros

<i>args</i>	Argumentos de línea de comandos (no utilizados).
-------------	--

Excepciones

<i>Exception</i>	En caso de errores durante la ejecución.
------------------	--

< Codificación de los datos de entrenamiento.

< Modelo de fragmentación generado.

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/org/fogbeam/example/opennlp/training/ChunkerTrainer.java](#)

6.3. Referencia de la clase

org.fogbeam.example.opennlp.DocumentClassifierMain

Métodos públicos estáticos

- static void [main](#) (String[] args) throws Exception
Metodo principal del programa.

6.3.1. Documentación de funciones miembro

6.3.1.1. main()

```
static void org.fogbeam.example.opennlp.DocumentClassifierMain.main (
    String[] args) throws Exception [static]
```

Metodo principal del programa.

Este metodo carga un modelo de categorización de documentos, clasifica un texto de entrada y determina la categoría más adecuada.

Parámetros

<i>args</i>	Argumentos de línea de comandos (no utilizados en este programa).
-------------	---

Excepciones

<i>Exception</i>	En caso de errores durante la ejecución.
------------------	--

< Flujo de entrada para cargar el modelo de categorización.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/org/fogbeam/example/opennlp/[DocumentClassifierMain.java](#)

6.4. Referencia de la clase org.fogbeam.example.opennlp.training.DocumentClassifierTrainer

Métodos públicos estáticos

- static void [main](#) (String[] args)
Método principal para entrenar un modelo de clasificación de documentos.

6.4.1. Documentación de funciones miembro**6.4.1.1. main()**

```
static void org.fogbeam.example.opennlp.training.DocumentClassifierTrainer.main (
    String[] args) [static]
```

Método principal para entrenar un modelo de clasificación de documentos.

Este método procesa un archivo de datos de entrenamiento, entrena un modelo de categorización de documentos y guarda el modelo entrenado en un archivo para su posterior uso.

Parámetros

<i>args</i>	Argumentos de línea de comandos (no utilizados).
-------------	--

Excepciones

<i>Exception</i>	En caso de errores durante la ejecución.
------------------	--

< Modelo de clasificación de documentos generado.

< Flujo de entrada para leer los datos de entrenamiento.

< Flujo de salida para guardar el modelo entrenado.

< Ruta del archivo donde se guardará el modelo.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/org/fogbeam/example/opennlp/training/[DocumentClassifierTrainer.java](#)

6.5. Referencia de la clase

org.fogbeam.example.opennlp.NameFinderMain

Métodos públicos estáticos

- static void [main](#) (String[] args) throws Exception
Metodo principal del programa.

6.5.1. Documentación de funciones miembro

6.5.1.1. main()

```
static void org.fogbeam.example.opennlp.NameFinderMain.main (
    String[] args) throws Exception [static]
```

Metodo principal del programa.

Este metodo carga un modelo de detección de entidades nombradas (NER, por sus siglas en inglés), identifica nombres propios en un conjunto de tokens y muestra las entidades detectadas.

Parámetros

<i>args</i>	Argumentos de línea de comandos (no utilizados en este programa).
-------------	---

Excepciones

<i>Exception</i>	En caso de errores durante la ejecución.
------------------	--

< Flujo de entrada para cargar el modelo NER.

Itera sobre las entidades detectadas (nombres propios). Cada objeto Span contiene:

- Índice de inicio y fin de la entidad en el array de tokens.
- Tipo de entidad (por ejemplo, "person").

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/org/fogbeam/example/opennlp/NameFinderMain.java](#)

6.6. Referencia de la clase

org.fogbeam.example.opennlp.training.NameFinderTrainer

Métodos públicos estáticos

- static void [main](#) (String[] args) throws Exception
Metodo principal para entrenar un modelo de detección de nombres propios.

6.6.1. Documentación de funciones miembro

6.6.1.1. `main()`

```
static void org.fogbeam.example.opennlp.training.NameFinderTrainer.main (  
    String[] args) throws Exception [static]
```

Método principal para entrenar un modelo de detección de nombres propios.

Este método utiliza datos de entrenamiento en formato de texto para crear un modelo capaz de identificar entidades nombradas (nombres de personas) en un texto tokenizado.

Parámetros

<i>args</i>	Argumentos de línea de comandos (no utilizados).
-------------	--

Excepciones

<i>Exception</i>	En caso de errores durante el entrenamiento o escritura del modelo.
------------------	---

< Conjunto de caracteres utilizado para leer los datos.

< Modelo de detección de nombres propio generado.

< Flujo de salida para guardar el modelo entrenado.

< Ruta del archivo donde se guardará el modelo.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/org/fogbeam/example/opennlp/training/NameFinderTrainer.java`

6.7. Referencia de la clase `org.fogbeam.example.opennlp.ParserMain`

Métodos públicos estáticos

- `static void main (String[] args) throws Exception`
Método principal del programa.

6.7.1. Documentación de funciones miembro

6.7.1.1. `main()`

```
static void org.fogbeam.example.opennlp.ParserMain.main (
    String[] args) throws Exception [static]
```

Método principal del programa.

Este método carga un modelo de análisis sintáctico, analiza una oración de entrada y genera un árbol sintáctico que representa su estructura gramatical. También muestra el árbol sintáctico en forma de texto y como un árbol codificado.

Parámetros

<i>args</i>	Argumentos de línea de comandos (no utilizados en este programa).
-------------	---

Excepciones

<i>Exception</i>	En caso de errores durante la ejecución.
------------------	--

< Flujo de entrada para cargar el modelo de análisis sintáctico.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/org/fogbeam/example/opennlp/ParserMain.java`

6.8. Referencia de la clase org.fogbeam.example.opennlp.training.ParserTrainer

Métodos públicos estáticos

- static void [main](#) (String[] args)

6.8.1. Documentación de funciones miembro

6.8.1.1. main()

```
static void org.fogbeam.example.opennlp.training.ParserTrainer.main (
    String[] args) [static]
```

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/org/fogbeam/example/opennlp/training/[ParserTrainer.java](#)

6.9. Referencia de la clase org.fogbeam.example.opennlp.PartOfSpeechTaggerMain

Métodos públicos estáticos

- static void [main](#) (String[] args)
Método principal del programa.

6.9.1. Documentación de funciones miembro

6.9.1.1. main()

```
static void org.fogbeam.example.opennlp.PartOfSpeechTaggerMain.main (
    String[] args) [static]
```

Método principal del programa.

Este método carga un modelo de etiquetado gramatical, procesa una oración tokenizada y genera etiquetas gramaticales (POS) para cada token, junto con las probabilidades asociadas.

Parámetros

<i>args</i>	Argumentos de línea de comandos (no utilizados en este programa).
-------------	---

< Flujo de entrada para cargar el modelo de etiquetado gramatical.

Registra los resultados para cada token. Muestra:

- El token original.
- Su etiqueta gramatical (POS tag).
- La probabilidad asociada a esa etiqueta.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/org/fogbeam/example/opennlp/[PartOfSpeechTaggerMain.java](#)

6.10. Referencia de la clase

org.fogbeam.example.opennlp.training.PartOfSpeechTaggerTrainer

Métodos públicos estáticos

- static void [main](#) (String[] args)

Metodo principal para entrenar un modelo de etiquetado gramatical (POS).

6.10.1. Documentación de funciones miembro

6.10.1.1. main()

```
static void org.fogbeam.example.opennlp.training.PartOfSpeechTaggerTrainer.main (
    String[] args) [static]
```

Metodo principal para entrenar un modelo de etiquetado gramatical (POS).

Este metodo procesa un archivo de datos de entrenamiento en formato texto, entrena un modelo de etiquetado gramatical y lo guarda en un archivo para su posterior uso.

Parámetros

<i>args</i>	Argumentos de línea de comandos (no utilizados).
-------------	--

< Modelo de etiquetado gramatical generado.

< Flujo de entrada para leer los datos de entrenamiento.

< Flujo de salida para guardar el modelo entrenado.

< Ruta del archivo donde se guardará el modelo entrenado.

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/org/fogbeam/example/opennlp/training/PartOfSpeechTaggerTrainer.java](#)

6.11. Referencia de la clase

org.fogbeam.example.opennlp.SentenceDetectionMain

Métodos públicos estáticos

- static void [main](#) (String[] args) throws Exception

Metodo principal del programa.

6.11.1. Documentación de funciones miembro

6.11.1.1. main()

```
static void org.fogbeam.example.opennlp.SentenceDetectionMain.main (
    String[] args) throws Exception [static]
```

Metodo principal del programa.

Este metodo carga un modelo de detección de oraciones, procesa un texto de entrada desde un archivo, detecta oraciones individuales en el texto y las muestra en la salida.

Parámetros

<code>args</code>	Argumentos de línea de comandos (no utilizados en este programa).
-------------------	---

Excepciones

<code>Exception</code>	En caso de errores durante la ejecución.
------------------------	--

< Flujo de entrada para cargar el modelo de detección de oraciones.

< Flujo de entrada para leer los datos de demostración.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/org/fogbeam/example/opennlp/SentenceDetectionMain.java`

6.12. Referencia de la clase `org.fogbeam.example.opennlp.training.SentenceDetectionTrainer`

Métodos públicos estáticos

- static void `main` (String[] args) throws Exception
Método principal para entrenar un modelo de detección de oraciones.

6.12.1. Documentación de funciones miembro

6.12.1.1. `main()`

```
static void org.fogbeam.example.opennlp.training.SentenceDetectionTrainer.main (  
    String[] args) throws Exception [static]
```

Método principal para entrenar un modelo de detección de oraciones.

Este método procesa un archivo de datos de entrenamiento, entrena un modelo de detección de oraciones y guarda el modelo resultante en un archivo para uso futuro.

Parámetros

<code>args</code>	Argumentos de línea de comandos (no utilizados).
-------------------	--

Excepciones

<code>Exception</code>	Si ocurre un error durante el entrenamiento o la serialización.
------------------------	---

< Codificación utilizada para leer los datos de entrenamiento.

< Modelo de detección de oraciones generado.

< Flujo de salida para guardar el modelo entrenado.

< Archivo donde se almacenará el modelo.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/org/fogbeam/example/opennlp/training/SentenceDetectionTrainer.java`

6.13. Referencia de la clase

org.fogbeam.example.opennlp.TokenizerMain

Métodos públicos estáticos

- static void [main](#) (String[] args) throws Exception
Método principal del programa.

6.13.1. Documentación de funciones miembro

6.13.1.1. main()

```
static void org.fogbeam.example.opennlp.TokenizerMain.main (
    String[] args) throws Exception [static]
```

Método principal del programa.

Este método recibe una lista de archivos de texto como entrada, procesa el contenido de cada archivo utilizando un modelo de tokenización, y escribe los tokens generados en un archivo de salida único.

Parámetros

<i>args</i>	Argumentos de línea de comandos: <ul style="list-style-type: none"> • Nombres de archivos de entrada. • Último argumento debe ser el nombre del archivo de salida.
-------------	--

Excepciones

<i>Exception</i>	En caso de errores durante la ejecución.
------------------	--

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/org/fogbeam/example/opennlp/[TokenizerMain.java](#)

6.14. Referencia de la clase

org.fogbeam.example.opennlp.training.TokenizerTrainer

Métodos públicos estáticos

- static void [main](#) (String[] args) throws Exception
Método principal para entrenar un modelo de tokenización.

6.14.1. Documentación de funciones miembro

6.14.1.1. main()

```
static void org.fogbeam.example.opennlp.training.TokenizerTrainer.main (
    String[] args) throws Exception [static]
```

Método principal para entrenar un modelo de tokenización.

Este método carga datos de entrenamiento, entrena un modelo de tokenización y lo guarda en un archivo para su uso posterior.

Parámetros

<i>args</i>	Argumentos de línea de comandos (no utilizados).
-------------	--

Excepciones

<i>Exception</i>	Si ocurre un error durante la lectura, el entrenamiento o la escritura.
------------------	---

< Codificación usada para leer los datos de entrenamiento.

< Modelo de tokenización generado.

< Flujo de salida para guardar el modelo entrenado.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/org/fogbeam/example/opennlp/training/TokenizerTrainer.java`

Capítulo 7

Documentación de archivos

7.1. Referencia del archivo README.md

7.2. Referencia del archivo

src/main/java/org/fogbeam/example/opennlp/ChunkerMain.java

Programa principal para realizar el análisis de fragmentos gramaticales (chunks) en una oración.

```
import java.io.FileInputStream;
```

Gráfico de dependencias incluidas en ChunkerMain.java:

7.3. Referencia del archivo

src/main/java/org/fogbeam/example/opennlp/DocumentClassifierMain.java

Programa principal para clasificar texto en categorías usando un modelo preentrenado de OpenNLP.

```
import java.io.FileInputStream;
```

Gráfico de dependencias incluidas en DocumentClassifierMain.java:

Clases

- class [org.fogbeam.example.opennlp.DocumentClassifierMain](#)

Paquetes

- package [org.fogbeam.example.opennlp](#)

7.3.1. Descripción detallada

Programa principal para clasificar texto en categorías usando un modelo preentrenado de OpenNLP.

Este programa carga un modelo de categorización de documentos, clasifica un texto de ejemplo en una categoría específica y muestra el resultado.

7.4. Referencia del archivo

src/main/java/org/fogbeam/example/opennlp/NameFinderMain.java

Programa principal para la detección de nombres propios en texto utilizando OpenNLP.

```
import java.io.FileInputStream;
```

Gráfico de dependencias incluidas en NameFinderMain.java:

Clases

- class [org.fogbeam.example.opennlp.NameFinderMain](#)

Paquetes

- package [org.fogbeam.example.opennlp](#)

7.4.1. Descripción detallada

Programa principal para la detección de nombres propios en texto utilizando OpenNLP.

Este programa carga un modelo de detección de entidades nombradas, procesa un conjunto de tokens para identificar nombres propios y muestra los resultados.

7.5. Referencia del archivo

src/main/java/org/fogbeam/example/opennlp/ParserMain.java

Programa principal para el análisis sintáctico de oraciones utilizando OpenNLP.

```
import java.io.FileInputStream;
```

Gráfico de dependencias incluidas en ParserMain.java:

Clases

- class [org.fogbeam.example.opennlp.ParserMain](#)

Paquetes

- package [org.fogbeam.example.opennlp](#)

7.5.1. Descripción detallada

Programa principal para el análisis sintáctico de oraciones utilizando OpenNLP.

Este programa carga un modelo de análisis sintáctico, procesa una oración de entrada y genera un árbol sintáctico que describe su estructura gramatical.

7.6. Referencia del archivo `src/main/java/org/fogbeam/example/opennlp/PartOfSpeechTaggerMain.java`↵

Programa principal para el etiquetado gramatical (POS tagging) de oraciones utilizando OpenNLP.

```
import java.io.FileInputStream;
```

Gráfico de dependencias incluidas en `PartOfSpeechTaggerMain.java`:

Clases

- class [org.fogbeam.example.opennlp.PartOfSpeechTaggerMain](#)

Paquetes

- package [org.fogbeam.example.opennlp](#)

7.6.1. Descripción detallada

Programa principal para el etiquetado gramatical (POS tagging) de oraciones utilizando OpenNLP.

Este programa carga un modelo de etiquetado gramatical preentrenado, procesa una oración de ejemplo para asignar etiquetas gramaticales (part-of-speech, POS) a cada palabra y muestra los resultados.

7.7. Referencia del archivo `src/main/java/org/fogbeam/example/opennlp/SentenceDetectionMain.java`↵

Programa principal para la detección de oraciones utilizando OpenNLP.

```
import java.io.FileInputStream;
```

Gráfico de dependencias incluidas en `SentenceDetectionMain.java`:

Clases

- class [org.fogbeam.example.opennlp.SentenceDetectionMain](#)

Paquetes

- package [org.fogbeam.example.opennlp](#)

7.7.1. Descripción detallada

Programa principal para la detección de oraciones utilizando OpenNLP.

Este programa carga un modelo preentrenado de detección de oraciones, procesa un texto de entrada y divide el contenido en oraciones individuales.

7.8. Referencia del archivo

src/main/java/org/fogbeam/example/opennlp/TokenizerMain.java

Programa principal para la tokenización de múltiples archivos de texto usando OpenNLP.

```
import java.util.ArrayList;
```

Gráfico de dependencias incluidas en TokenizerMain.java:

Clases

- class [org.fogbeam.example.opennlp.TokenizerMain](#)

Paquetes

- package [org.fogbeam.example.opennlp](#)

7.8.1. Descripción detallada

Programa principal para la tokenización de múltiples archivos de texto usando OpenNLP.

Este programa permite procesar múltiples archivos de texto proporcionados como argumentos, aplica un modelo de tokenización para separar el texto en tokens, y guarda el resultado en un archivo de salida único.

7.9. Referencia del archivo

src/main/java/org/fogbeam/example/opennlp/training/Chunker↵ Trainer.java

Clase para entrenar un modelo de fragmentación gramatical (chunker) utilizando OpenNLP.

```
import java.nio.charset.Charset;
```

Gráfico de dependencias incluidas en ChunkerTrainer.java:

Clases

- class [org.fogbeam.example.opennlp.training.ChunkerTrainer](#)

Paquetes

- package [org.fogbeam.example.opennlp.training](#)

7.9.1. Descripción detallada

Clase para entrenar un modelo de fragmentación gramatical (chunker) utilizando OpenNLP.

Este programa utiliza datos de entrenamiento en formato CoNLL2000 para generar un modelo de fragmentación, que puede ser usado posteriormente para identificar estructuras gramaticales.

7.10. Referencia del archivo

src/main/java/org/fogbeam/example/opennlp/training/DocumentClassifierTrainer.java ↩

Clase para entrenar un modelo de clasificación de documentos utilizando OpenNLP.

```
import java.io.BufferedOutputStream;
```

Gráfico de dependencias incluidas en DocumentClassifierTrainer.java:

Clases

- class [org.fogbeam.example.opennlp.training.DocumentClassifierTrainer](#)

Paquetes

- package [org.fogbeam.example.opennlp.training](#)

7.10.1. Descripción detallada

Clase para entrenar un modelo de clasificación de documentos utilizando OpenNLP.

Esta clase utiliza datos de entrenamiento en formato de texto para generar un modelo de clasificación de documentos basado en categorías predefinidas.

7.11. Referencia del archivo

src/main/java/org/fogbeam/example/opennlp/training/NameFinderTrainer.java ↩

Clase para entrenar un modelo de reconocimiento de nombres propios utilizando OpenNLP.

```
import java.io.BufferedOutputStream;
```

Gráfico de dependencias incluidas en NameFinderTrainer.java:

Clases

- class [org.fogbeam.example.opennlp.training.NameFinderTrainer](#)

Paquetes

- package [org.fogbeam.example.opennlp.training](#)

7.11.1. Descripción detallada

Clase para entrenar un modelo de reconocimiento de nombres propios utilizando OpenNLP.

Este programa entrena un modelo NER (Named Entity Recognition) basado en un conjunto de datos de entrenamiento, específicamente para la detección de nombres de personas.

7.12. Referencia del archivo

src/main/java/org/fogbeam/example/opennlp/training/ParserTrainer.java

```
import java.util.logging.Logger;
```

Gráfico de dependencias incluidas en ParserTrainer.java:

Clases

- class [org.fogbeam.example.opennlp.training.ParserTrainer](#)

Paquetes

- package [org.fogbeam.example.opennlp.training](#)

7.13. Referencia del archivo

src/main/java/org/fogbeam/example/opennlp/training/PartOfSpeechTaggerTrainer.java

Clase para entrenar un modelo de etiquetado gramatical (Part-of-Speech) usando OpenNLP.

```
import java.io.BufferedOutputStream;
```

Gráfico de dependencias incluidas en PartOfSpeechTaggerTrainer.java:

Clases

- class [org.fogbeam.example.opennlp.training.PartOfSpeechTaggerTrainer](#)

Paquetes

- package [org.fogbeam.example.opennlp.training](#)

7.13.1. Descripción detallada

Clase para entrenar un modelo de etiquetado gramatical (Part-of-Speech) usando OpenNLP.

Este programa entrena un modelo de etiquetado gramatical (POS) a partir de un conjunto de datos de entrenamiento. Los modelos generados permiten etiquetar palabras de un texto con su categoría gramatical (por ejemplo, sustantivo, verbo, adjetivo).

7.14. Referencia del archivo

src/main/java/org/fogbeam/example/opennlp/training/SentenceDetectionTrainer.java

Clase para entrenar un modelo de detección de oraciones usando OpenNLP.

```
import java.io.BufferedOutputStream;
```

Gráfico de dependencias incluidas en SentenceDetectionTrainer.java:

Clases

- class [org.fogbeam.example.opennlp.training.SentenceDetectionTrainer](#)

Paquetes

- package [org.fogbeam.example.opennlp.training](#)

7.14.1. Descripción detallada

Clase para entrenar un modelo de detección de oraciones usando OpenNLP.

Este programa entrena un modelo de detección de oraciones a partir de un conjunto de datos de entrenamiento. Los modelos generados permiten dividir textos en oraciones individuales.

7.15. Referencia del archivo

src/main/java/org/fogbeam/example/opennlp/training/TokenizerTrainer.java

Clase para entrenar un modelo de tokenización usando OpenNLP.

```
import java.io.BufferedOutputStream;
```

Gráfico de dependencias incluidas en TokenizerTrainer.java:

Clases

- class [org.fogbeam.example.opennlp.training.TokenizerTrainer](#)

Paquetes

- package [org.fogbeam.example.opennlp.training](#)

7.15.1. Descripción detallada

Clase para entrenar un modelo de tokenización usando OpenNLP.

Esta clase utiliza un archivo de datos de entrenamiento para generar un modelo que puede dividir textos en tokens, tales como palabras o puntuaciones.

Índice alfabético

main

org.fogbeam.example.opennlp.ChunkerMain, [11](#)

org.fogbeam.example.opennlp.DocumentClassifierMain, [main](#), [13](#)
[12](#)

org.fogbeam.example.opennlp.NameFinderMain, [14](#)
[14](#)

org.fogbeam.example.opennlp.ParserMain, [16](#)

org.fogbeam.example.opennlp.PartOfSpeechTaggerMain, [17](#)
[17](#)

org.fogbeam.example.opennlp.SentenceDetectionMain, [18](#)
[18](#)

org.fogbeam.example.opennlp.TokenizerMain, [20](#)

org.fogbeam.example.opennlp.training.ChunkerTrainer, [12](#)
[12](#)

org.fogbeam.example.opennlp.training.DocumentClassifierTrainer, [13](#)
[13](#)

org.fogbeam.example.opennlp.training.NameFinderTrainer, [15](#)
[15](#)

org.fogbeam.example.opennlp.training.ParserTrainer, [17](#)
[17](#)

org.fogbeam.example.opennlp.training.PartOfSpeechTaggerTrainer, [18](#)
[18](#)

org.fogbeam.example.opennlp.training.SentenceDetectionTrainer, [19](#)
[19](#)

org.fogbeam.example.opennlp.training.TokenizerTrainer, [20](#)
[20](#)

org.fogbeam.example.opennlp, [9](#)

org.fogbeam.example.opennlp.ChunkerMain, [main](#), [11](#)

org.fogbeam.example.opennlp.DocumentClassifierMain, [12](#)
[12](#)

[main](#), [12](#)

org.fogbeam.example.opennlp.NameFinderMain, [14](#)
[14](#)

[main](#), [14](#)

org.fogbeam.example.opennlp.ParserMain, [16](#)
[16](#)

[main](#), [16](#)

org.fogbeam.example.opennlp.PartOfSpeechTaggerMain, [17](#)
[17](#)

[main](#), [17](#)

org.fogbeam.example.opennlp.SentenceDetectionMain, [18](#)
[18](#)

[main](#), [18](#)

org.fogbeam.example.opennlp.TokenizerMain, [20](#)
[20](#)

[main](#), [20](#)

org.fogbeam.example.opennlp.training, [9](#)

org.fogbeam.example.opennlp.training.ChunkerTrainer, [12](#)
[12](#)

[main](#), [12](#)

org.fogbeam.example.opennlp.training.DocumentClassifierTrainer,

[13](#)

org.fogbeam.example.opennlp.training.NameFinderTrainer,

[14](#)

[main](#), [15](#)

org.fogbeam.example.opennlp.training.ParserTrainer,

[17](#)

[main](#), [17](#)

org.fogbeam.example.opennlp.training.PartOfSpeechTaggerTrainer,

[18](#)

[main](#), [18](#)

org.fogbeam.example.opennlp.training.SentenceDetectionTrainer,

[19](#)

org.fogbeam.example.opennlp.training.TokenizerTrainer,

[20](#)

[main](#), [20](#)

Práctica 4, [1](#)

README.md, [23](#)

src/main/java/org/fogbeam/example/opennlp/ChunkerMain.java,

[23](#)

src/main/java/org/fogbeam/example/opennlp/DocumentClassifierMain.java,

[23](#)

src/main/java/org/fogbeam/example/opennlp/NameFinderMain.java,

[24](#)

src/main/java/org/fogbeam/example/opennlp/ParserMain.java,

[24](#)

src/main/java/org/fogbeam/example/opennlp/PartOfSpeechTaggerMain.java,

[25](#)

src/main/java/org/fogbeam/example/opennlp/SentenceDetectionMain.java,

[25](#)

src/main/java/org/fogbeam/example/opennlp/TokenizerMain.java,

[26](#)

src/main/java/org/fogbeam/example/opennlp/training/ChunkerTrainer.java,

[26](#)

src/main/java/org/fogbeam/example/opennlp/training/DocumentClassifierT

[27](#)

src/main/java/org/fogbeam/example/opennlp/training/NameFinderTrainer.j

[27](#)

src/main/java/org/fogbeam/example/opennlp/training/ParserTrainer.java,

[28](#)

src/main/java/org/fogbeam/example/opennlp/training/PartOfSpeechTagger

[28](#)

src/main/java/org/fogbeam/example/opennlp/training/SentenceDetectionTr

[29](#)

src/main/java/org/fogbeam/example/opennlp/training/TokenizerTrainer.java,
[29](#)