

UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

**TEXT MINING APLICADO A DOCUMENTACIÓN DE API PARA LA DETECCIÓN DE DIRECTIVAS**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

GABRIEL CORREA GAETE

PROFESOR GUÍA:

ROMAIN ROBBES

MIEMBROS DE LA COMISIÓN:

JORGE PÉREZ R.

BÁRBARA POBLETE L.

SANTIAGO DE CHILE

2015

Índice

[1 Introducción 1](#_Toc425868280)

[1.1 Documentación de API 1](#_Toc425868281)

[1.2 Problema y Motivación 1](#_Toc425868282)

[1.3 Una Solución 1](#_Toc425868283)

[1.4 Objetivos 1](#_Toc425868284)

[1.4.1 Objetivos Generales 1](#_Toc425868285)

[1.4.2 Objetivos Específicos 1](#_Toc425868286)

[1.4.3 Metodología 1](#_Toc425868287)

[2 Marco teórico y trabajos anteriores 1](#_Toc425868288)

[2.1 Importancia de comentarios en APIs y código fuente 1](#_Toc425868289)

[2.2 Destacación de directivas en APIs 1](#_Toc425868290)

[2.2.1 Problema: Las directivas pueden pasar desapercibidas 1](#_Toc425868291)

[2.2.2 eMoose 1](#_Toc425868292)

[2.2.3 Syntax highlighting en editores de texto: Sublime Text 1](#_Toc425868293)

[2.3 Trabajo de Monperrus et al. 2](#_Toc425868294)

[2.3.1 Keywords para detectar directivas 2](#_Toc425868295)

[2.3.2 Taxonomía de directivas 2](#_Toc425868296)

[3 Trabajo Realizado: Text Mining sobre comentarios de API 2](#_Toc425868297)

[3.1 Preparación de los datos 2](#_Toc425868298)

[3.1.1 Datos del trabajo de Monperrus 2](#_Toc425868299)

[3.1.2 Datos necesarios para aplicar Text Mining 2](#_Toc425868300)

[3.1.3 Proceso de adaptación de los datos 2](#_Toc425868301)

[3.2 Implementación de aplicación para la revisión de comentarios 2](#_Toc425868302)

[3.3 Diseño del proceso de Text Mining 3](#_Toc425868303)

[3.3.1 Text Mining en Weka 3](#_Toc425868304)

[3.3.2 Semántica de los datos 3](#_Toc425868305)

[3.3.3 Clasificadores 3](#_Toc425868306)

[3.3.4 Experimentos realizados usando Weka 3](#_Toc425868307)

[3.4 API de Weka 3](#_Toc425868308)

[4 Resultados 3](#_Toc425868309)

[4.1 Resultados de los Experimentos realizados en Weka 4](#_Toc425868310)

[4.1.1 Tablas comparando distintos clasificadores, usando o no stemming, usando o no stopwords 4](#_Toc425868311)

[4.1.2 Tablas adicionales comparando otras opciones y parámetros del filtro y opciones del mejor clasificador 4](#_Toc425868312)

[4.2 Discusión de los resultados 4](#_Toc425868313)

[5 Limitaciones y problemas de validez 4](#_Toc425868314)

[6 Conclusiones y trabajo futuro 4](#_Toc425868315)

[7 Glosario 4](#_Toc425868316)

[8 Bibliografía 4](#_Toc425868317)

[9 Anexo 4](#_Toc425868318)

# Introducción

## Documentación de API

## Problema y Motivación

## Una Solución

Hablar de destacar directivas y de la necesidad de mayor automatización pues la asignación manual de directivas es muy tediosa y toma mucho tiempo.

## Objetivos

(parte similar al informe de CC6908)

### Objetivos Generales

### Objetivos Específicos

### Metodología

# Marco teórico y trabajos anteriores

## Importancia de comentarios en APIs y código fuente

* % de comentarios en código fuente
* Trabajos pasados que apoyan la inclusión de comentarios para ayudar a desarrolladores

## Destacación de directivas en APIs

### Problema: Las directivas pueden pasar desapercibidas

* Mostrar ejemplos de directivas no vistas

### eMoose

* Resultados de este estudio con las directivas destacadas en Eclipse
* Imágenes de eMoose
* Explicación de eMoose y su posible utilidad y poder
* Taxonomia propuesta

### Syntax highlighting en editores de texto: Sublime Text

* Hablar de que se probó que en sublime text es posible destacar frases o palabras claves (mostrar 2 imágenes de ejemplo) (mencionar esto pues puede ser otro camino de solución al destacar directivas)

## Trabajo de Monperrus et al.

Mencionar que el trabajo presenta una buena referencia de comparación al detectar directivas automáticamente, pues el trabajo obtuvo los porcentajes de keywords en cada directiva que encontraron

### Keywords para detectar directivas

### Taxonomía de directivas

# Trabajo Realizado: Text Mining sobre comentarios de API

Explicación de que se necesita adaptar los datos para Weka (.arff)

Y que se necesita revisar manualmente los comentarios -> por eso la implementación de la herramienta Web

## Preparación de los datos

### Datos del trabajo de Monperrus

Los datos vienen en .xml en 2 archivos. Hablar algo más sobre los datos: la fuente, la validez, etc

### Datos necesarios para aplicar Text Mining

* Se necesita tener los datos en csv y arff
* Se necesita tener los comentarios separados por frase
* Se necesita que cada frase tenga un campo nuevo “is\_directive” o “class”
* Ya se tienen las directivas de Monperrus. Faltan igual cantidad de no-directivas para aplicar Machine Learning.
* Por otra parte se decide clasificar manualmente 500\*3 comentarios por cuenta propia para usarlos en Weka

### Proceso de adaptación de los datos

Explicación del proceso usado para adaptar los datos para usarlos en Weka

## Implementación de aplicación para la revisión de comentarios

Estaba pensando en ponerle un nombre a la aplicación. Pensé en: Comments Highlighter (CHI o CHi); Java Api Highlighter (JAH); Jah Api Highlighter (JAH)

* Elección de aplicación HTML5 por tener gran compatibilidad
* Mostrar capacidades de la página, imágenes, input, output, hablar de importancia de hotkeys, hablar de dificultades encontradas para lograr una buena visualización de comentarios
* Extracción de comentarios de varios archivos en una carpeta

## Diseño del proceso de Text Mining

### Text Mining en Weka

Introducir brevemente a Weka, su Explorer, Experimenter, Filtros, Clasificadores, Seleccionadores y API.

### Semántica de los datos

* Texto entendido como un vector, representando la presencia de cada palabra posible
* String to Word verctor (en Weka)
* Unigramas

### Clasificadores

* Presentar lista clasificadores
* Clasificadores seleccionados: J48, RandomForest, SMO, MDNText
* Stemming
* StopWords

### Experimentos realizados usando Weka

* Hablar de que experimentos se hicieron, con qué parámetros (en la siguiente sección estarán los resultados)

## API de Weka

* Sobre Text Mining automatizado usando su API de Java
* Clasificadores que son capaces de seguir aprendiendo automáticamente (no todos tienen esa capacidad en la API de Weka)

# Resultados

Aca aun no se si se debería incluir los resultados de el 1er set de datos que usé (los datos anteriores, que no son la selección de 500 comentarios por librería que tengo pendiente)

## Resultados de los Experimentos realizados en Weka

### Tablas comparando distintos clasificadores, usando o no stemming, usando o no stopwords

### Tablas adicionales comparando otras opciones y parámetros del filtro y opciones del mejor clasificador

## Discusión de los resultados

# Limitaciones y problemas de validez

# Conclusiones y trabajo futuro

# Glosario

# Bibliografía

# Anexo