UNIVERSIDAD DE COSTA RICA
SISTEMA DE ESTUDIOS DE POSGRADO

EVALUACIÓN DE UNA APLICACIÓN DE APRENDIZAJE DE MÁQUINA PARA LA CLASIFICACIÓN AUTOMÁTICA DE RECETAS CULINARIAS

Trabajo final de investigación aplicada sometido a la consideración de la Comisión del Programa de Estudios de Posgrado en Computación para optar al grado y título de Maestría Profesional en Computación

KAREN MIRANDA HERNÁNDEZ

Ciudad Universitaria Rodrigo Facio, Costa Rica

2019

Dedicatoria

Por su apoyo y amor incondicional a mi papá, mamá y esposo.

Agradecimientos

Agradezco al profesor Dr. Edgar Casasola por su tiempo y recomendaciones con las cuales me fue posible finalizar con éxito este proyecto. También por toda su ayuda y recomendaciones a Aurelio Sanabria Rodríguez.

Este trabajo final de investigación aplicada fue aceptado por la Comisión del Programa de Estudios de Posgrado en Computación de la Universidad de Costa Rica, como requisito para optar al grado académico de Maestría Profesional en Computación.

Dr. Jorge Antonio Leoni de León
Representante del Decano Sistema de Estudios de Posgrado

Dr. Edgar Casasola Murillo
Profesor Guía

Dra. Gabriela Marín Raventón
Lectora

MSc. Mario Hernández Delgado
Lector

M.Sc.Marta Eunice Calderón Campos
Representante de la Directora del Programa de Posgrado en Computación e Informática

---

Karen Miranda Hernández
Sustentante

Índice general

Resumen

Tomando provecho de técnicas de procesamiento de lenguaje natural nace este proyecto, el cual se propuso como objetivo realizar una comparación de la precisión entre un proyecto semiautomático, que utiliza contextos definicionales para la clasificación de un corpus de recetas obtenidas desde web, y un sistema automático, implementado en este proyecto. Para el desarrollo de este sistema automático se utilizaron máquinas de soporte vectorial, una técnica de aprendizaje de máquina supervisado que debe de ser entrenado con documentos, que deben ser previamente preprocesados y etiquetados con las categorías "receta" y "no receta". Este proyecto comparó los resultados obtenidos de la nueva implementación con máquinas de soporte vectorial con los resultados de una implementación semiautomática basada en conocimiento experto y observó una ligera mejoría en la precisión con la nueva implementación.

Palabras clave

Clasificación automática, recetas costarricenses, máquinas de soporte vectorial, bolsa de palabras.

Índice de cuadros

Índice de figuras

Capítulo 1

Introducción

En un trabajo en conjunto entre los investigadores del Instituto de Investigaciones Lingüísticas (INIL) y el Programa de Posgrado en Computación e Informática se desarrolló el proyecto "Análisis de contexto definicionales en corpus de gastronomía tradicional en Costa Rica", conocido como CODEGAT. Este proyecto tenía como objetivo el procesamiento de contextos definicionales, que son fragmentos de texto que sirven para reconocer el significado de los términos y, además, establecer relaciones semánticas [Alarcón, 2009]. Para la identificación de los contextos definicionales fue necesario realizar una serie de tareas, entre ellas la clasificación de los documentos en las categorías "receta" y "no receta" para definir sobre cuáles documentos se podía trabajar. Era necesario que esta tarea se realizará de manera automátizada para obtener la mayor cantidad de documentos posible. De este trabajo surge la primer fase del artículo [Corrales et al., 2018] (disponible en el apéndice A) que lleva a cabo la selección manual de verbos para realizar una clasificación semiautomática de los documentos entre las categorías receta y no receta.

Ahora, en este proyecto de investigación se plantea si es posible obtener una mejora en la precisión de identificación de recetas. Por lo tanto, se plantea la hipótesis de que, al utilizar una representación de los documentos con base en características obtenidas a partir de los verbos, mediante el aprendizaje de máquina, sería posible obtener una mejor precisión en la clasificación automática de documentos .

1.1. Antecedentes

El proyecto de investigación CODEGAT es un proyecto adscrito al INIL de la Universidad de Costa Rica. CODEGAT nace con el objetivo de desarrollar a largo plazo una aplicación que, al recibir como entrada texto, pueda identificar de manera automática, en este texto, tanto el proceso como los insumos necesarios para elaborar algún producto, y determinar el proceso que se siguió. Debido a lo ambicioso de este proyecto, se ha requerido un análisis profundo. De esta manera, sus investigadores decidieron utilizar la receta de cocina con texto representativo de un instructivo procedimental.

Ahora bien, para evitar el procesamiento de texto de manera manual, y, además, poder incrementar la cantidad de textos a procesar, CODEGAT buscó apoyo especializado en el programa de Posgrado en Computación e Informática de la Universidad de Costa Rica, específicamente en el programa de maestría profesional. La idea era aprovechar los conocimientos y experiencia tanto de docentes como estudiantes en el área de Procesamiento de Lenguaje Natural. Dicha unión tuvo como resultado un trabajo colaborativo que se realizó en dos fases distintas. La primera fase consistió en la identificación y delimitación semi-automática de los textos que contienen una receta, basándose en los verbos contenidos en el texto. En la segunda fase se toman estos resultados para delimitar tanto la sección de ingredientes y el procedimiento. La investigación para este proyecto de graduación se concentra en la identificación de recetas (descrita anteriormente en la primera fase).

Es importante destacar que, para esta investigación, se utilizó la recolección automática de documentos con el objetivo de procesar un gran volumen de recetas, al obtener un corpus compuesto por una mayor variedad de documentos.

La identificación de la aparación o no de una receta en un documento se realizó mediante el uso de un programa semiautomático de software. La identificación de recetas es la que se asocia con la primera fase del proyecto CODEGAT. A continuación, en la figura 1.1, se muestra el proceso ejecutado en la primera fase de esa investigación precedente.

Por otro lado, la segunda fase determinó el inicio de la receta mediante la identificación de los ingredientes y el procedimiento, lo cual se convierte en un problema de recuperación de información que tiene particularidades. El proceso que se siguió para resolver

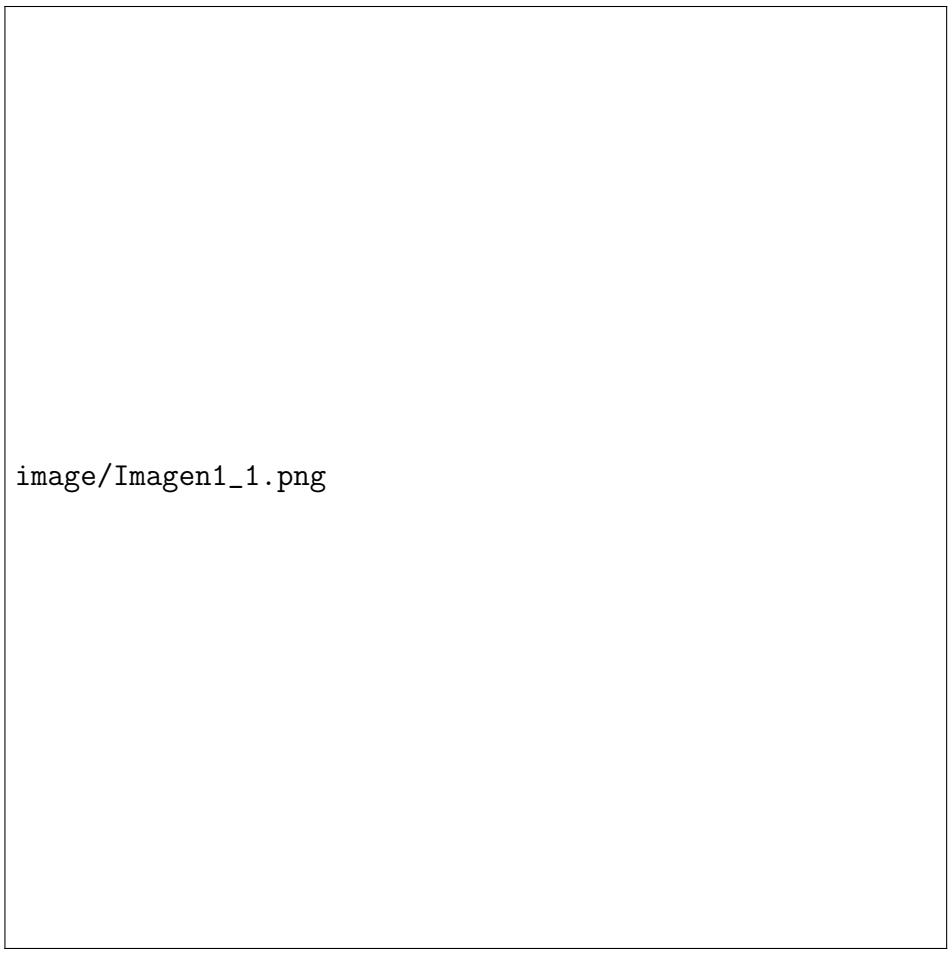


Figura 1.1: Proceso de la primera fase [Corrales et al., 2018]

esas fases se encuentra descrito en [Corrales et al., 2018], cuyo trabajo se realizó como el proyecto del laboratorio para el curso de Procesamiento de Lenguaje Natural, parte de la Maestría Profesional en Computación e Informática.

Lo que se pretende con el presente trabajo final de investigación aplicada es utilizar una segunda técnica, que utiliza el aprendizaje de máquina para la clasificación de textos en recetas. Posteriormente, se pretende proceder a comparar los resultados obtenidos siguiendo esta segunda técnica con los obtenidos con el procedimiento descrito en la primera fase de [Corrales et al., 2018].

1.2. Justificación

Gracias a las colaboraciones iniciales del proyecto de investigación CODEGAT por parte de estudiantes y docentes en el curso de Procesamiento de Lenguaje Natural de

la Maestría Profesional de Computación e Informática, descritas en los antecedentes, nace este proyecto de graduación aplicada. Estas colaboraciones mostraron una gran necesidad de procesamiento de grandes cantidades de documentos, los cuales eran procesados de manera manual, limitando de gran manera la cantidad de documentos y la búsqueda de valores de interés en el texto. La informática, en su campo de procesamiento de lenguaje natural, se involucra en el análisis de contextos definicionales ante la necesidad de analizar grandes volúmenes de texto para clasificar y sistematizar los datos definicionales aplicados a un dominio restringido. Con este proyecto se pretende encontrar una solución automática mediante aprendizaje automático. El método se va a comparar con respecto a los resultados obtenidos en el pasado para el mismo conjunto de datos de CODEGAT. En el siguiente capítulo se definen algunos conceptos involucrados en este trabajo: contextos definicionales, clasificación mediante aprendizaje de máquina, aprendizaje de máquina supervisado, máquinas de soporte vectorial y validación cruzada.

Capítulo 2

Marco Teórico

A continuación se presentan los conceptos necesarios para entender la implementación de este proyecto de investigación. Los conceptos se encuentran en tres categorias distintas: recolección de datos, clasificación y validación. Para la recolección de datos se utilzó una araña dirigida, por lo tanto ese será el primer concepto definido en esta sección. La clasificación de los documentos en este proyecto se utilizó una técnica de aprendizaje de máquina supervisada conocida como máquinas de soporte vectorial por lo que se vera iniciará con el concepto de aprendizaje de máquina para continuar con aprendizaje de máquina supervisa y finalizar con las máquinas de soporte vectorial. Para la validación de los datos se definió el término validación cruzada, que corresponde a la técnica utilizada para validar los datos.

2.1. Araña dirigida

Una araña o crawler dirigido es un programa que visita sitios web para leer sus páginas y otra información (que pueda estar presente en el código de la misma) para guardar esta información en documentos. Es necesario destacar que estas semillas definen el contexto de los documentos que se descargarán ya que todos estos documentos provienen de los enlaces obtenidos por a partir de estas semillas [Rouse, 2005].

El proceso que sigue una araña se describe a continuación en la figura 2.1. La araña utiliza como entrada una lista de enlaces semilla que permiten las descargas iniciales de documentos. Posteriormente la araña procede a descargar los documentos iniciales y obtener de cada uno de estos documentos todos los enlaces que contienen para agregarlos a una lista de enlaces no visitados que seguirá visitando hasta que cumpla con la cantidad de iteraciones o cantidad de documentos que definidos previamente.

2.2. Clasificación mediante aprendizaje de máquina

El término “clasificación”, en procesamiento de lenguaje natural, corresponde a elegir la etiqueta correcta para una entrada de texto [Bird, 2015]. En el caso de este proyecto, las etiquetas posibles corresponden a receta y no receta. Por lo tanto, al

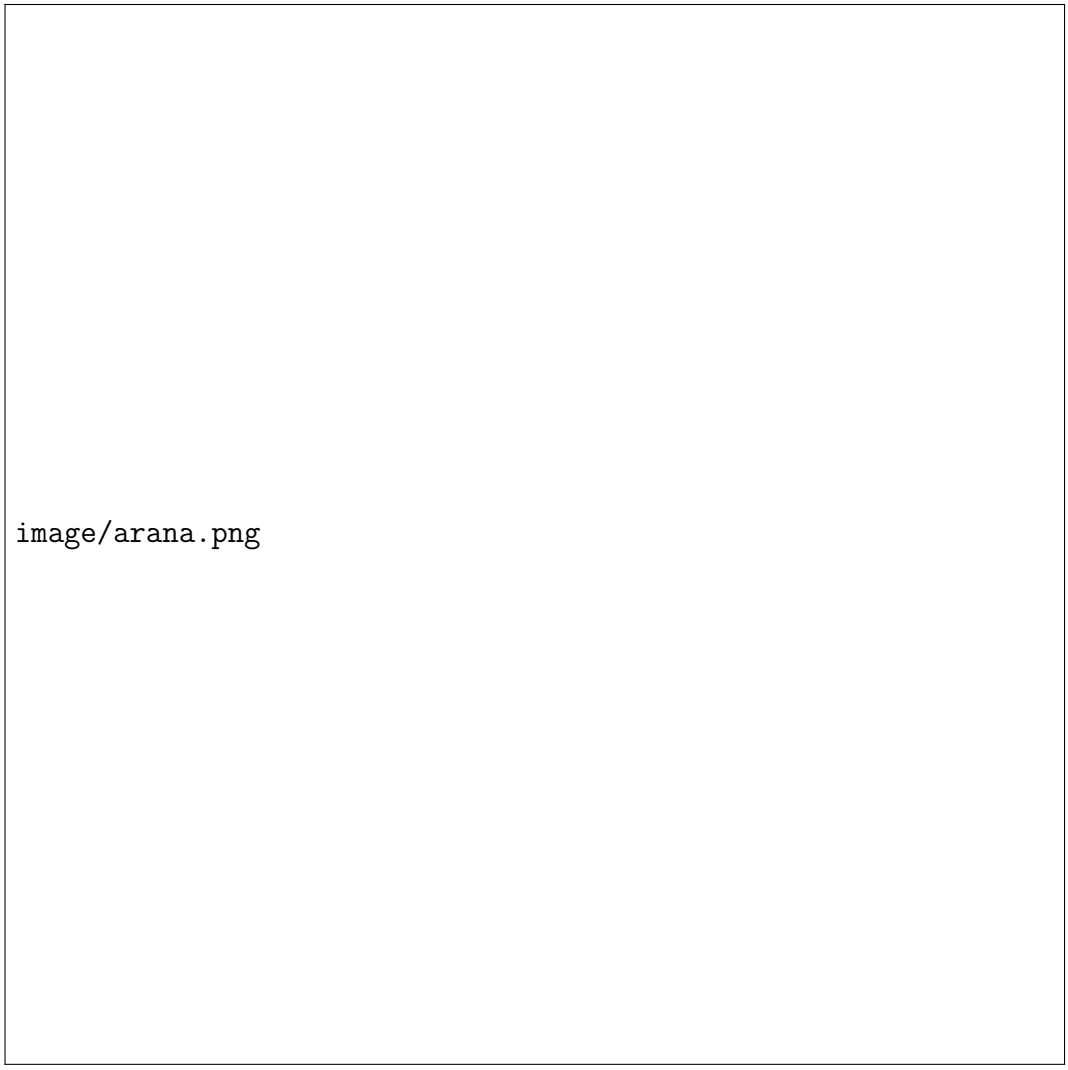


Figura 2.1: Flujo de una araña dirigida

referirnos a la clasificación automática de documentos, queremos decir etiquetar cada documento como receta o no receta.

Por otro lado, aprendizaje de máquina es una tecnología del área de Procesamiento de Lenguaje Natural de la Informática que utiliza técnicas estadísticas para el reconocimiento de patrones [Bird, 2015]. El aprendizaje de máquina tiene una gran cantidad de aplicaciones, entre ellas, la identificación de partes del discurso, entidades, sentimientos, así como la clasificación [Redmore, 2018]. El enfoque de este proyecto corresponde al uso de aprendizaje de máquina para la clasificación de textos.

Existen dos categorías de aprendizaje de máquina: supervisado y no supervisado. Es necesario destacar que el tipo que se utilizará para este proyecto de investigación corresponde a la categoría supervisada. A continuación se procederá a describir la técnica

supervisada.

2.3. Aprendizaje de máquina supervisado

Este tipo de aprendizaje tiene una etapa de entrenamiento en la que se procesan documentos que ya se encuentran correctamente etiquetados. Este proceso se utiliza para mejorar el modelo estadístico que contiene el algoritmo. Inclusive, es posible agregar un modelo de retroalimentación que permita al algoritmo mejorar [Redmore, 2018].

El aprendizaje de máquina tiene dos fases: entrenamiento y predicción. La fase de entrenamiento consiste en utilizar un extractor de características que tiene como objetivo la obtención de un conjunto de características al convertir cada valor de entrada. Estos conjuntos de características capturan la información básica que debería usarse para clasificar cada entrada. Los pares de conjuntos de características y etiquetas se introducen en el algoritmo como datos de entrenamiento para que el aprendizaje automático logre generar un modelo [Bird, 2015].

Una vez concluida la fase de entrenamiento, se continúa con la fase de predicción, que en nuestro caso procede a clasificar los documentos. Durante esta etapa, el mismo extractor de características se usa para convertir entradas invisibles en conjuntos de características. Estos conjuntos de características se introducen en el modelo, y luego se procede a generar las etiquetas pronosticadas (receta y no receta) [Bird, 2015].

A continuación, la figura 2.2 muestra el flujo de las fases del aprendizaje de máquina supervisado.

Algunos de los métodos de aprendizaje de máquina supervisado son las máquinas de soporte vectorial, las redes bayesianas, los clasificadores de entropía máxima, los campos aleatorios condicionales y las redes neuronales con aprendizaje profundo. En el caso de este proyecto de investigación, se utilizará el modelo de máquinas de soporte vectorial.

2.4. Máquinas de soporte vectorial

Las máquinas de soporte vectorial se categorizan como un modelo de aprendizaje de máquina supervisado, diseñadas inicialmente con el objetivo de resolver problemas de clasificación binaria [Carmona, 2014]. Sin embargo, actualmente han evolucionado hasta tener aplicación en distintos problemas la regresión, agrupamiento y multiclasificación, siendo este último de los más relevantes [Resendiz, 2006]. Además, se utilizan con éxito

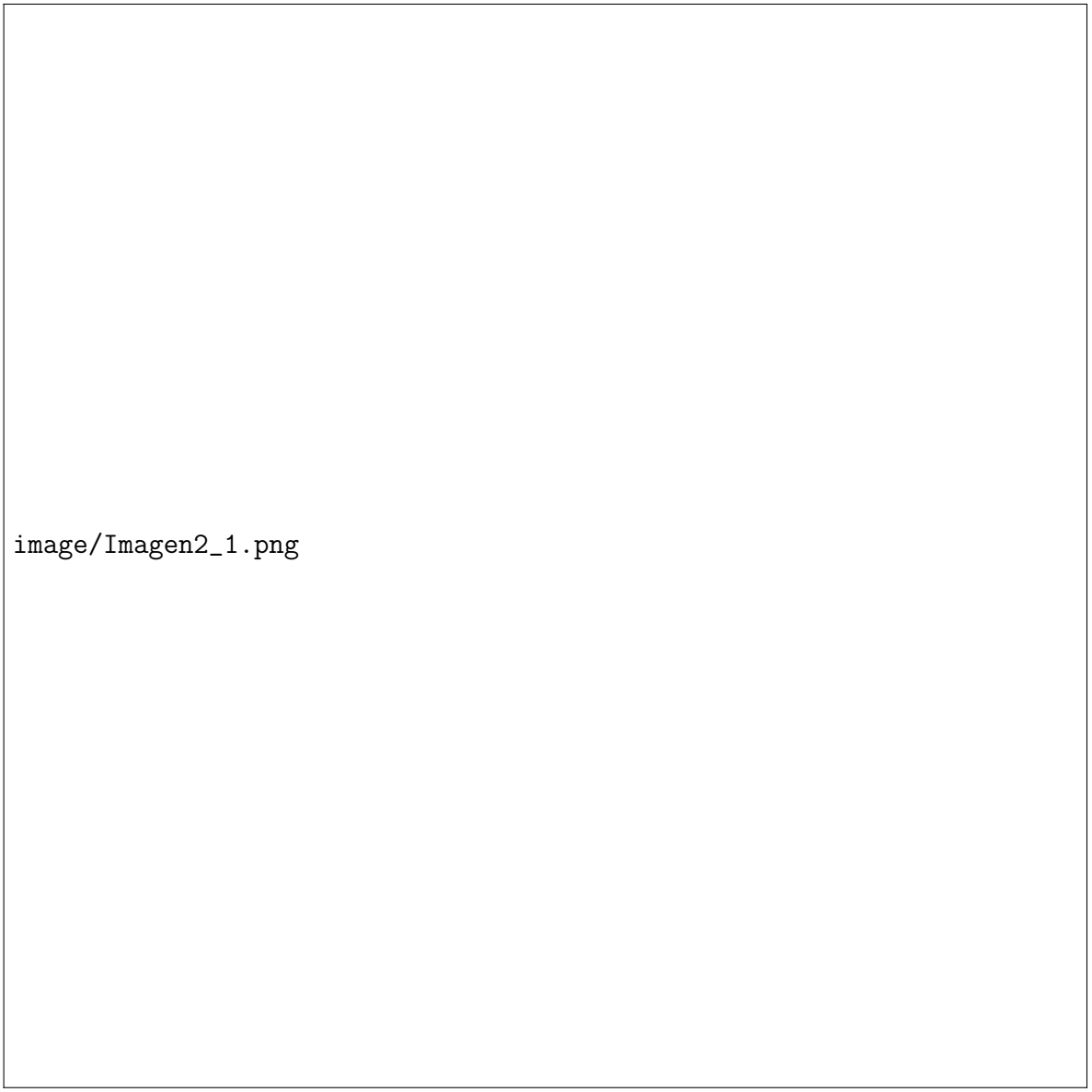


Figura 2.2: Fases del aprendizaje de máquina supervisado [Moujahid, 2016]

en áreas como visión artificial, reconocimiento de caracteres, categorización de texto e hipertexto, clasificación de proteínas, procesamiento de lenguaje natural y análisis de series temporales [Carmona, 2014]. Debido a estas características fue que se decidió utilizar las máquinas de soporte vectorial para el método de clasificación automática de textos entre recetas y no recetas.

Primeramente, para el uso de máquinas de soporte vectorial es necesario transformar los objetos a clasificar en vectores. Una vez realizado este paso de preprocesamiento, se continúa con el algoritmo de las máquinas de soporte vectorial que van a permitir la

clasificación. Este algoritmo se enfoca en el problema general de aprender a discriminar entre miembros positivos y negativos de una clase de vectores de hasta n-dimensiones. Esto se logra mediante una función matemática conocida como kernel, en la que los datos originales se redimensionan para buscar una separación lineal óptima de los mismos. De manera general, las Máquinas de Soporte Vectorial permiten encontrar un hiperplano óptimo que separe las clases [Resendiz, 2006]. En la figura 2.3 se puede observar de manera gráfica cómo ocurre la separación lineal que permite la clasificación de los vectores entre miembros positivos y negativos.

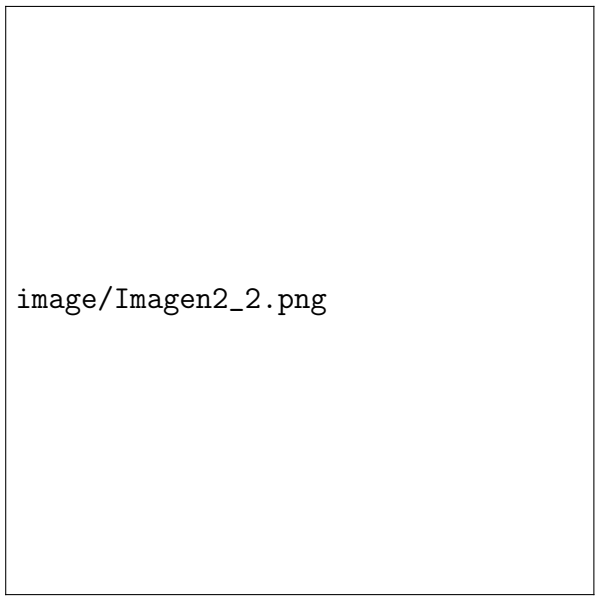


Figura 2.3: Ejemplo de clasificación de vectores mediante máquinas de soporte vectorial [Carmona, 2014]

Es importante destacar que la figura 2.3 muestra el caso más simple de separación que existe en máquinas de soporte vectorial, el cual consiste en una separación lineal. Sin embargo, también es posible realizar la separación de clases mediante una función no lineal, la cual se conoce como kernel trick. Esta función busca encontrar patrones y relaciones para alcanzar una alta precisión en su modelo final de aprendizaje automático.

2.5. Método de validación cruzada

La validación cruzada corresponde a un método que se utiliza en aprendizaje de máquina para estimar la precisión de sistemas con datos con los que no se ha entrenado el modelo de manera estadística, que permite generalizar los resultados a conjuntos

de datos independientes. Es decir, se utiliza una muestra para estimar cómo se va desempeñar el modelo en general para hacer predicciones sobre datos que no se usaron durante el entrenamiento del modelo [Arlot y Celisse, 2009] .

Este método es ampliamente utilizado debido a la facilidad de implementación y también su simplicidad, que da como resultado estimaciones que generalmente tienen un sesgo más bajo que otros métodos [Brownlee, 2018]. El procedimiento general es el siguiente:

1. Mezclar el conjunto de datos al azar.
2. Dividir el conjunto de datos en k grupos.
3. Para cada grupo:
    - Tomar el grupo como un conjunto de datos de entrenamiento o prueba.
    - Entrenar el modelo con el conjunto de datos de entrenamiento y evaluar con el conjunto de datos de prueba.
    - Conservar el puntaje de evaluación y descartar el modelo.
4. Sumarizar la precisión del modelo utilizando los resultados de las k iteraciones.

Es importante destacar que, en cada iteración, la muestra de datos (ya sea de entrenamiento o pruebas) se asigna a un grupo individual y permanece en ese grupo durante la duración del procedimiento. Esto significa que a cada muestra se le da la oportunidad de ser utilizada en el conjunto de prueba una vez y utilizada para entrenar el modelo k-1 veces [Brownlee, 2018].

Una vez que se tienen claros los conceptos anteriores, es posible tener una idea clara del trabajo que se requiere para este proyecto de investigación, por lo que el siguiente capítulo procede a limitar el alcance y resultados que se pretenden lograr.

Capítulo 3

Planteamiento del problema

3.1. Pregunta de investigación

La tarea que se pretende resolver con este proyecto final de investigación aplicada corresponde a la identificación automática de documentos para la clasificación entre receta y no receta. Para esto se pretende utilizar la tecnología de aprendizaje de máquina. De modo específico, se busca aplicar esta técnica con el fin de que esta se pueda comparar con el método utilizado en la primera fase de [Corrales et al., 2018]. La comparación se realizará con el fin de encontrar la técnica con mayor precisión entre los métodos descritos anteriormente.

3.2. Objetivos

Con base en la experiencia adquirida en las colaboraciones anteriores descritas en los antecedentes, y teniendo claro que se quería dar continuidad a estos aportes, se procedió a definir que la primera prioridad de este proyecto sería evaluar los resultados obtenidos con el sistema semi automático obtenido en la fase #1 de [Corrales et al., 2018]. Debido a estas razones, se definen a continuación los objetivos que permitieron delimitar este proyecto.

3.2.1. Objetivo General

Evaluar la precisión de una técnica basada en aprendizaje de máquina para la identificación de documentos que contienen recetas de cocina costarricenses a partir de documentos provenientes de la web.

3.2.2. Objetivos Específicos

1. Desarrollar un mecanismo para la construcción de un conjunto de datos para el entrenamiento y evaluación de un clasificador automático de texto de recetas de cocina basado en un crawler.

2. Generar un modelo de clasificación automática de documentos que contienen recetas de cocina con máquinas de soporte vectorial.

3. Comparar la precisión de la técnica de la fase 1 descrita en [Corrales et al., 2018] contra el modelo de máquina de soporte vectorial mediante el modelo de validación cruzada.

3.2.3. Alcance y limitaciones

Los documentos utilizados tanto para el entrenamiento como para las pruebas de este proyecto son obtenidos de la web mediante la implementación de un crawler que toma como insumos una lista de URLs conocidos como semillas. Los documentos utilizados para el entrenamiento fueron extraídos en setiembre del 2018 y los documentos de prueba en agosto del 2016.

Es importante destacar el origen de los documentos utilizados ya que juegan un rol importante en el desarrollo de este proyecto, el cual está definido a continuación en la metodología.

Capítulo 4

Metodología

En el figura 4.1 se muestra un diagrama que contiene las actividades realizadas en este trabajo. Este diagrama contiene seis tareas que nos permitieron lograr los objetivos del trabajo. La primera tarea correspondió a la construcción del *crawler* dirigido. Una vez concluida esta aplicación, es posible continuar con la segunda tarea, la recolección de documentos desde la web, mediante el uso del crawler, que proveyó los datos necesarios para las pruebas del modelo obtenido en el paso tres. La tercera tarea comprendió preprocesamiento de datos necesario para que estos puedan ser procesados para la cuarta tarea, que consistió en el entrenamiento del modelo de clasificación utilizando máquinas de soporte vectorial.

Una vez realizados las tareas descritas anteriormente se procedió con las dos tareas finales que incluyeron a la ejecución de las pruebas y comparación de resultados, siendo esta última tarea el paso que permitió derivar las conclusiones del proyecto.

4.1. Construcción de la araña dirigida

La construcción de la araña dirigida fue necesaria para que tomara como parámetros una lista de semillas URL. Con esta lista, la aplicación fue capaz de explorar todos los posibles documentos enlazados. Este programa permitió la descarga de documentos basados en parámetros de entrada, conocidos como semillas, que consisten en una lista base de URL, dados anteriormente por expertos del área de lingüística. El objetivo de estos parámetros fue obtener documentos que se encontraran en el contexto de cocina, sin que esto asegure que todos los documentos son recetas, sino con temas relevantes al de este proyecto de investigación

A continuación en la tabla 4.1 se muestran algunos URL. (La lista completa de los URL semilla utilizados se encuentra en el Anexo 1).

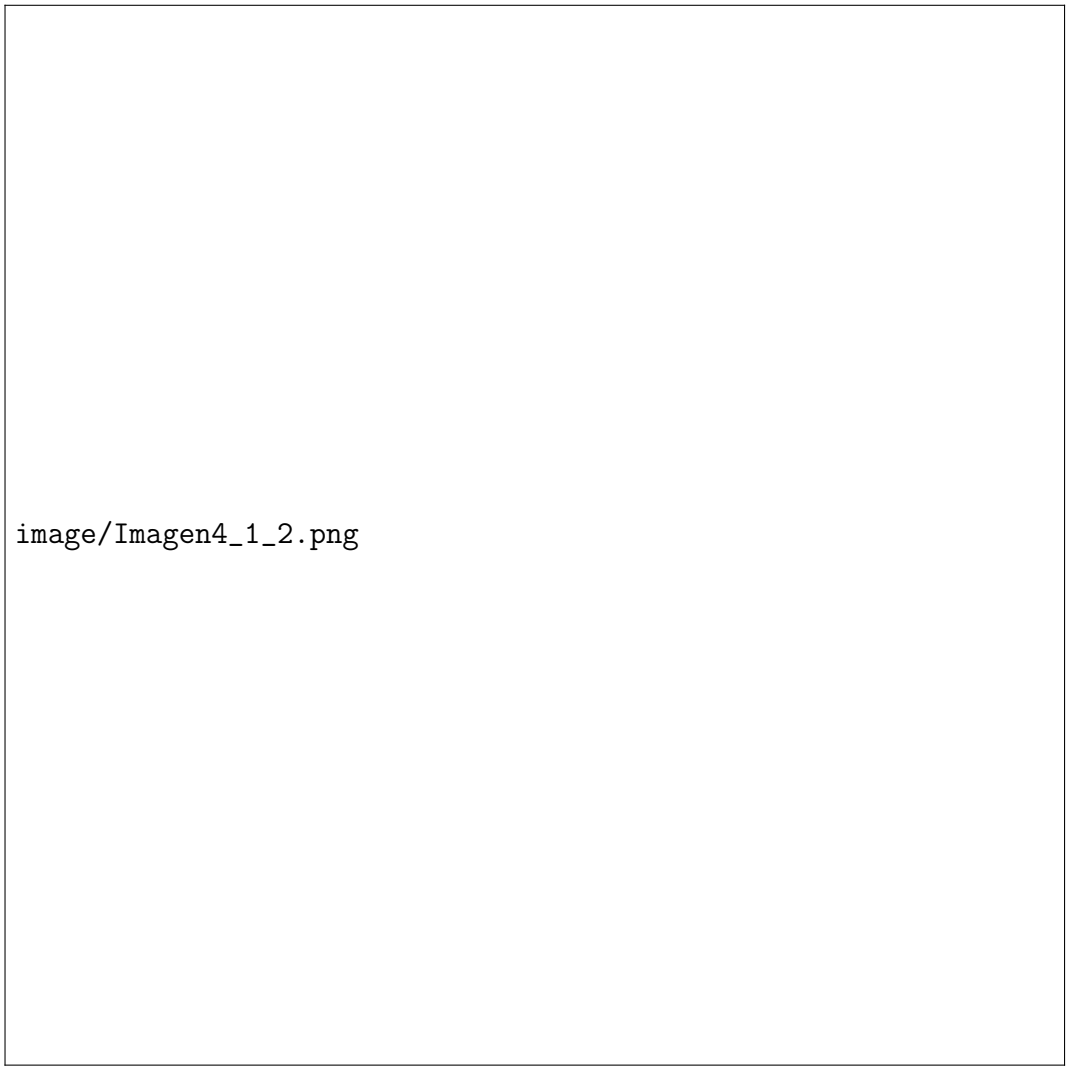


Figura 4.1: Diagrama de la metodología del trabajo

Cuadro 4.1: Muestra de semillas

Semillas	
1	http://quinua.pe/quinua-valor-nutricional/
2	https://cookpad.com/co/recetas/120897-ensalada-de-frutas-para-ninos
3	ttps://recetastipicascr.com/recetas/olla-de-carne/
4	https://www.flores.ninja/palmito/

4.2. Recolección de documentos

Para la obtención de documentos en la red se utilizó el programa crawler, el cual fue desarrollado como primer tarea para este trabajo. Se obtuvieron desde la web 5998 documentos en total, también realizado por mi, cada uno de los cuales con el mismo formato, lo cual facilitaría el procesamiento necesario para los programas clasificadores. El proceso de descarga de los documentos se hizo mediante una ejecución de la araña que tenía como parámetros descargar menos de 6000 documentos, al final dos documentos se encontraban vacios, por lo que fueron descartados y se obtuvo el número final de documentos, 5998. Es necesario aclarar que se limitó la descarga de documentos a 5998 documentos debido a que la clasificación realizada por el sistema de detección de recetas costarricenses semiautomática en la fase #1 de [Corrales et al., 2018] fue de 508 documentos. Por esta razón, fue necesario tener como mínimo 5000 documentos, con el objetivo de tener una relación similar a 90/10 para la evaluación de resultados en este trabajo, en el que se utilizó cerca de un 90 % de los documentos para el conjunto de entrenamiento y 10 % para las pruebas.

Los documentos obtenidos en la web debieron tener un preprocesamiento en el que se elimina cualquier etiqueta o código HTML, dejando como resultado únicamente texto. Un ejemplo de documento se puede observar en la figura 4.2.

4.3. Preprocesamiento de documentos

A pesar de que los documentos son legibles a nivel humano, como se observa en la figura 4.2, aún es necesario realizar algunas operaciones adicionales a estos documentos para generar el modelo. Estas operaciones se conocen como preprocesamiento. El primer paso en este proceso fue el proceder a etiquetar cada uno los de documentos. Es importante destacar que este proceso se realizó tanto para los documentos destinados para entrenar el modelo como para los documentos de prueba. El proceso de etiquetamiento se realizó de manera manual y se registró en un archivo de texto en el que se indicó el nombre del documento, el cual corresponde a un número único que incrementa de uno en uno, seguido por un guión para luego mostrar la clasificación del mismo. El formato es el siguiente:

```
# documento - clasificación: Receta | NOreceta
```

Una sección del documento que contiene la clasificación de todos los documentos se observa a continuación en el cuadro 4.2

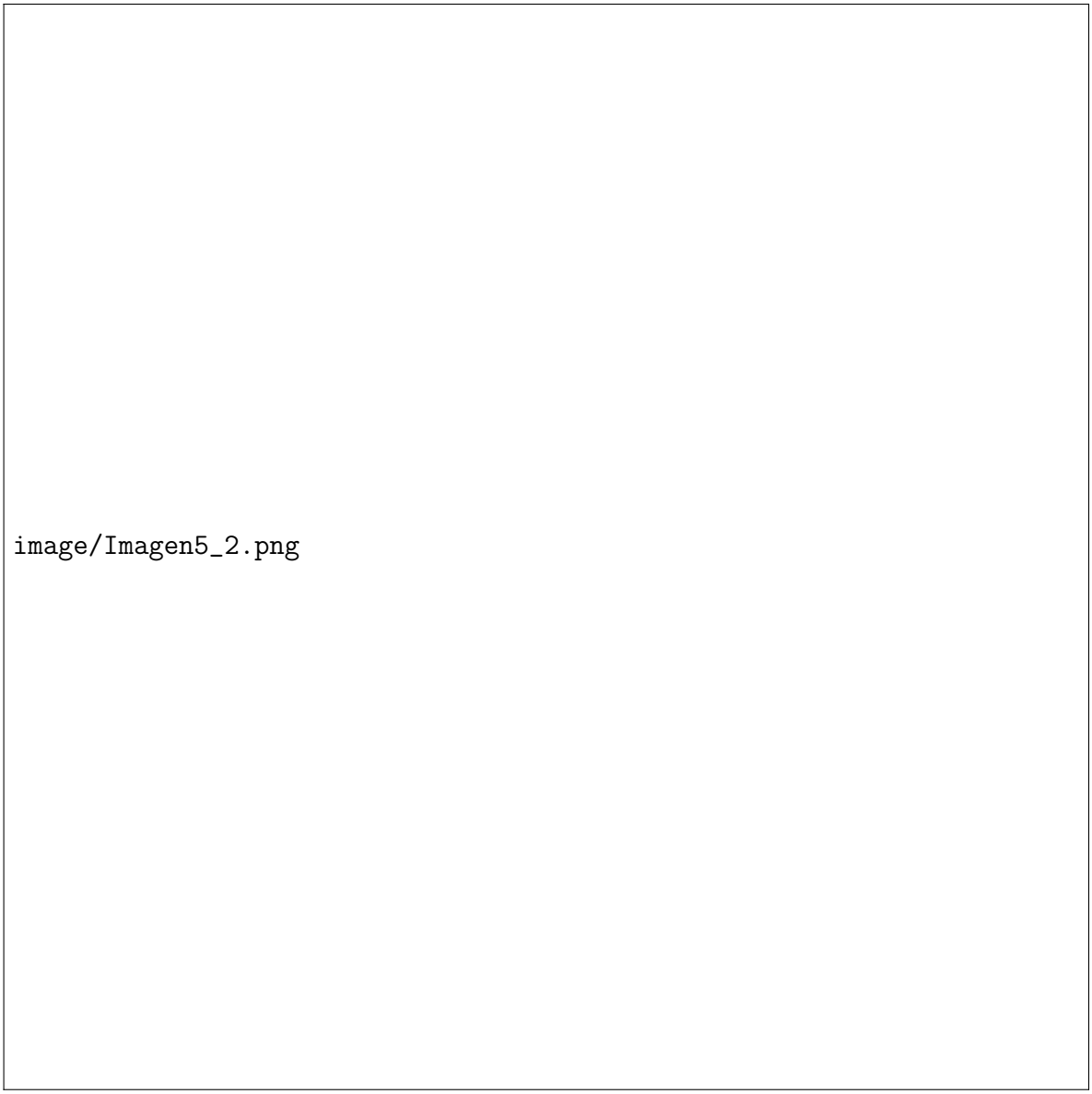


Figura 4.2: Ejemplo de documento

Sin embargo, a pesar de que el documento ilustrado en el cuadro 4.2 es muy claro para entender la clasificación de cada uno de los documentos, no contiene el formato ni toda la información necesaria para entrenar el modelo de máquina de soporte vectorial. Por esta razón, se creó un archivo de tipo CSV formado por dos columnas, en las que cada tupla contiene toda la información de un documento. La primera columna contiene todo el documento, en la que se eliminaron todos los saltos de línea, además de ciertos caracteres especiales como tabulaciones y comillas, ya que todo el documento se agrupó en comillas. Por último, la segunda columna contiene la clasificación del documento.

Cuadro 4.2: Ejemplo de etiquetado de documentos

Documento	**Etiqueta**
0	Receta
1	Receta
2	NO receta
3	NO receta
4	Receta
5	NO receta
6	NO receta
7	NO receta
8	NO receta
9	Receta
10	NO receta

En la figura 4.3 se puede observar una porción del documento final que se utilizó para entrenar el modelo.

4.3.1. Preprocesamiento de datos: WEKA

WEKA es la herramienta que contiene una colección de algoritmos de aprendizaje de máquina para el procesamiento de datos. Es por esta razón, que se utilizó WEKA para la implementación del modelo. Sin embargo, antes de poder proceder con el entrenamiento de modelo, era necesario realizar una serie de pasos que permitiera que WEKA procese los datos de manera correcta. El primer paso fue agregar el documento de datos de entrenamiento, que contenía todos los documentos y sus clasificaciones (mostrado en 4.3).

En total se debieron realizar 3 conversiones distintas que se encuentran en la sección de datos de WEKA. Para la segunda columna, que contiene la clase correspondiente a cada documento, fue necesario aplicar el filtro de tipo no supervisado `renameAtribute`, el cual permite cambiar el nombre de la columna, ya que por defecto tomó el nombre Receta, y esto genera un conflicto, debido a la existencia de esta palabra a lo largo del documento. La aplicación de este filtro se evidencia en la figura 4.4. Es importante eliminar este conflicto ya que no permite aplicar los filtros necesarios a la primera columna que contiene el documento completo.

Los filtros realizados para la primera columna del documento también se en-

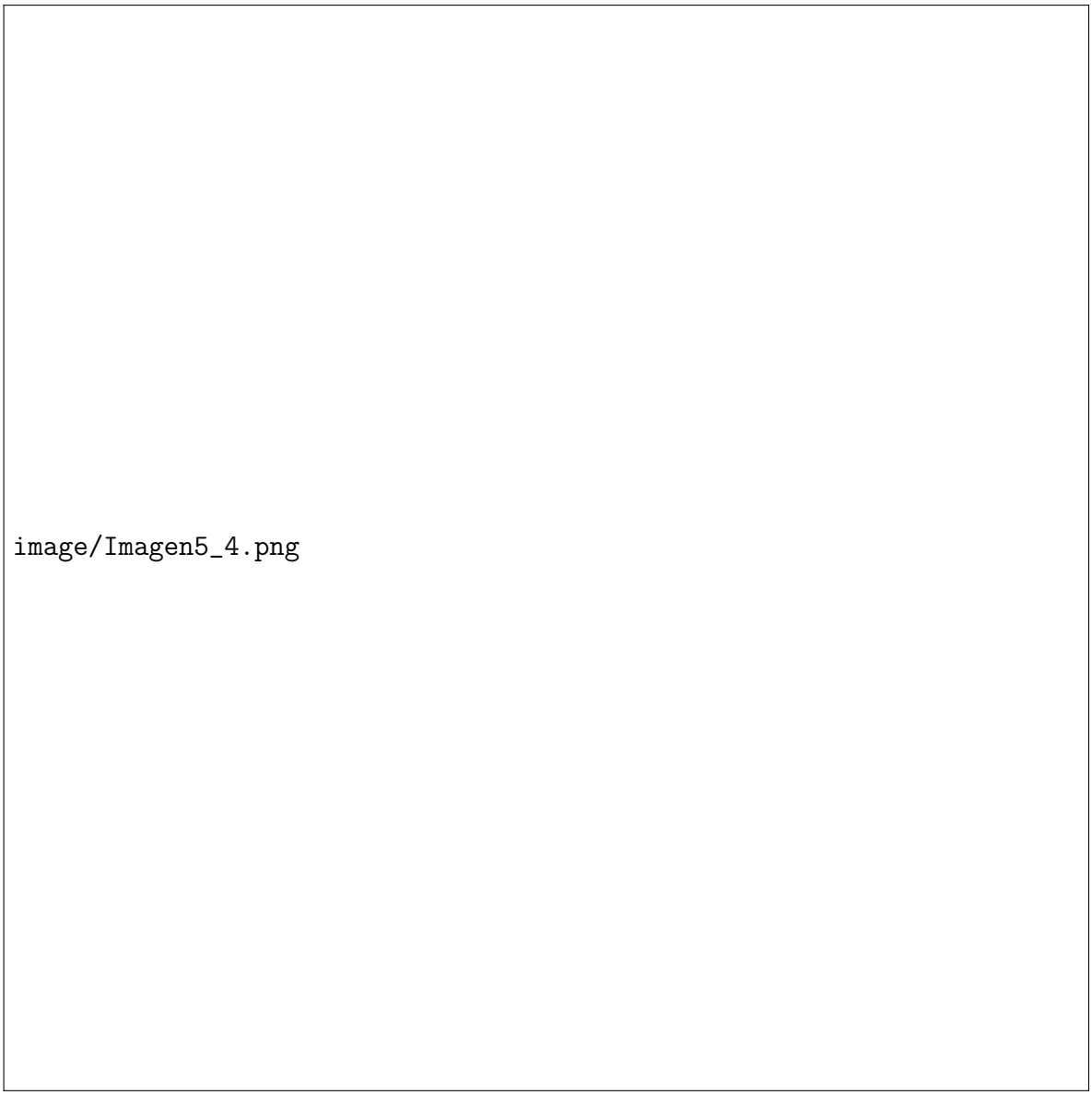


Figura 4.3: Documentos para entrenamiento

image/FiltroClaseReceta.png

Figura 4.4: Aplicación de filtro `renameAtribute`

cuentran en la sección no supervisada y corresponden a: `NominalToString` y `StringToWordVector`. Es importante destacar que la importancia del filtro StringToWordVector es que convierte cada documento en una bolsa de palabras. La bolsa de palabras corresponde a una manera de representación de documentos que permite procesar cada una de las palabras contenidas en el documento de manera individual.

4.4. Entrenamiento del modelo

Fue posible agregar este plugin a WEKA mediante la función del menú `Tools` y seleccionar en esta la opción llamada `Package manager`. Para utilizar máquinas de soporte vectorial, se utilizó un plugin de WEKA llamado libSVM. Este administrador de paquetes muestra una lista de todos los plugins accesibles para agregar en WEKA, como se muestra en la figura 4.5. Una vez seleccionado el paquete que contiene libSVM,

solo se procedió a presionar el botón `Install`.

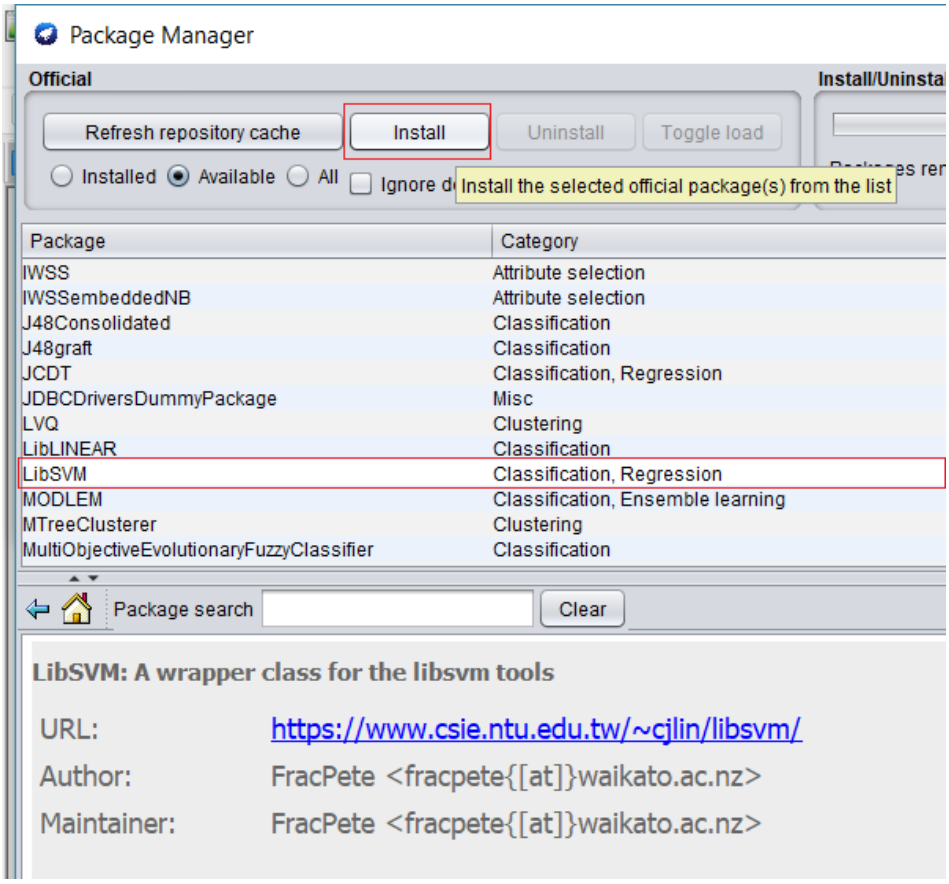


Figura 4.5: Instalación de plugin libsvm en WEKA

Además, como se puede observar en la figura 4.6, para la evaluación de la precisión del modelo se utilizó la técnica de validación cruzada, con diez iteraciones debido a que corresponde a la configuracion por defecto de WEKA, y se consideró como la más apropiada para validar el modelo, ya que 10 iteraciones proporician suficientes pruebas.

Una vez finalizadas las diez iteraciones se obtiene como resultado el modelo. Este modelo se procede a guardar como un archivo tipo MODEL con el fin de utilizarlo posteriormente. La forma que provee WEKA para almacenar este archivo se encuentra al dar clic derecho a la lista de resultados y presionar la opción salvar modelo, como se observa en la figura 4.7

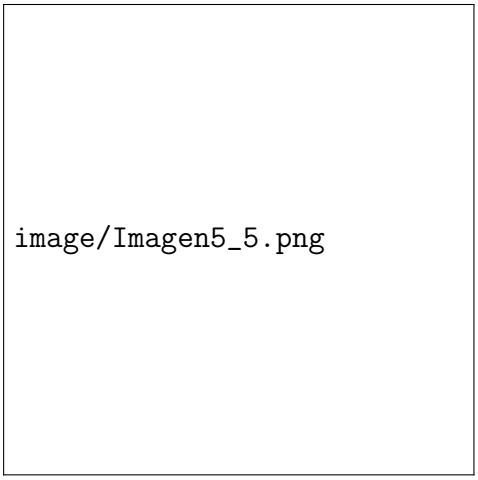


Figura 4.6: Parámetros de evaluación del modelo

Como parte del entrenamiento del modelo se obtiene una matriz de confusión, como se evidencia en el cuadro 4.3. Este cuadro muestra cómo del total de 5998 documentos, 5867 fueron clasificados de manera correcta durante el entrenamiento, y tan solo 131 documentos clasificados de manera incorrecta.

Cuadro 4.3: Matriz de confusión del modelo de entrenamiento

	Receta	NOreceta
Receta	1011	43
NOreceta	88	4856

Ademas, también se utilizaron 5998 documentos en total obtenidos de la web para el entrenamiento del modelo. El resultado de la validación cruzada del modelo se puede observar en la figura 4.8, en la que se puede notar que el porcentaje de aciertos es de muy buena calidad (este resultado es esperable, ya que es entrenamiento) con un 97.8 % correspondiente a los 5867 documentos procesados de manera correcta.

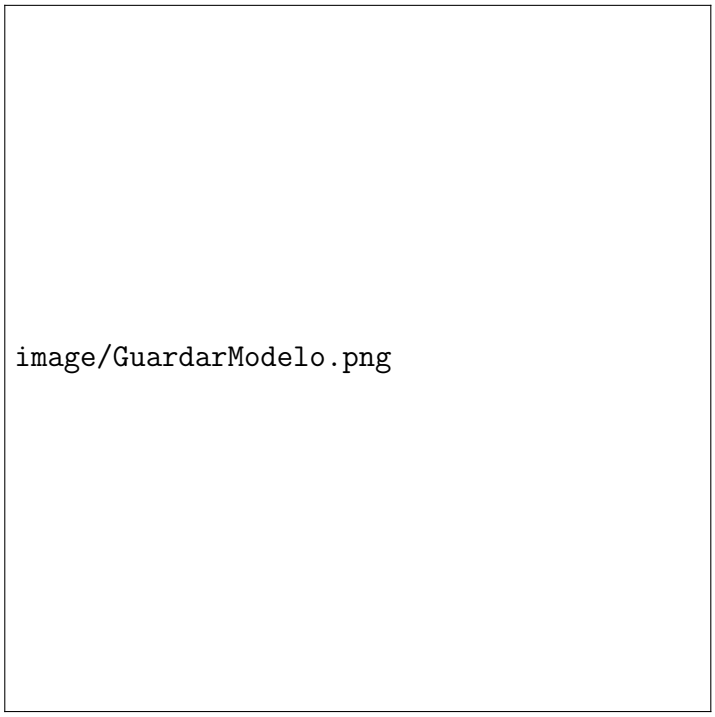


Figura 4.7: Opción para guardar el modelo

image/EntrenamientoModelo_2.png

Figura 4.8: Resultados del entrenamiento del modelo

Capítulo 5

Resultados

Una vez entrenado el modelo, se procedió a realizar la pruebas. Para iniciar, se procede a cargar el documento que contenía el mismo formato que se utilizó para los datos de entrenamiento. Una vez que el documento con los datos de prueba se carga, se deben aplicar todos los filtros que se aplicaron en la etapa de preprocesamiento. El siguiente paso consistió en cargar el modelo que se generó con WEKA. Para ejecutar los resultados, se ingresa a la sección de clasificación y se procede a seleccionar la función de tipo `misc` con la opción `inputMappedClassifier`. Una vez elegida esta función, se debieron ingresar los parámetros: classifier y el modelo, como se puede observar en la figura 5.1 a continuación.

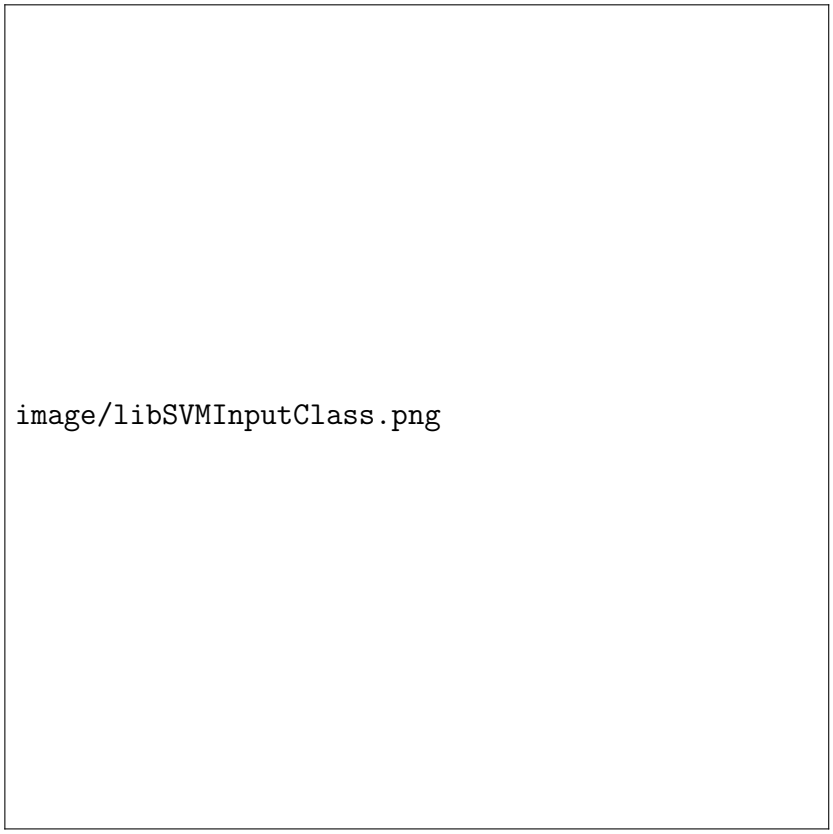


Figura 5.1: Selección del modelo y función clasificadora

Ahora, debido a que se utiliza el modelo de representación de datos *bolsa de palabras*, es necesario un paso extra. Esto se debe a que la representación del documento

bolsa de palabras procesa cada una de las palabras existentes en el documento. Sin embargo, al clasificar nuevos documentos, una vez finalizado el entrenamiento, pueden existir palabras nuevas (es decir, que no estaban presentes en los documentos de entrenamiento) y esta situación puede generar que los documentos no se clasifiquen de la manera correcta. Por esta razón, se decidió utilizar la opción de predicción de texto. Este proceso evitó la afectación de resultados en WEKA aún al encontrar nuevas palabras (desconocidas para el entrenamiento) en el documento de pruebas. Esto se debe de que WEKA le asigna un peso a cada palabra, y al encontrar nuevas palabras, sin un peso asignado se pueden ver afectados los resultados. Para hacer esto, se seleccionó el parámetro `PlainText` en la opción *Output predictions*. Este proceso se puede observar en la figura 5.2.

Figura 5.2: Agregar predicción a WEKA

Una vez realizados estos pasos, solo es necesario presionar el botón *start* en WEKA, para obtener los resultados. A continuación se presentan los resultados de la clasificación de los mismos documentos para la implementación realizada en la fase #1 de [Corrales et al., 2018] y la nueva implementación automatizada utilizando máquinas de

soporte vectorial.

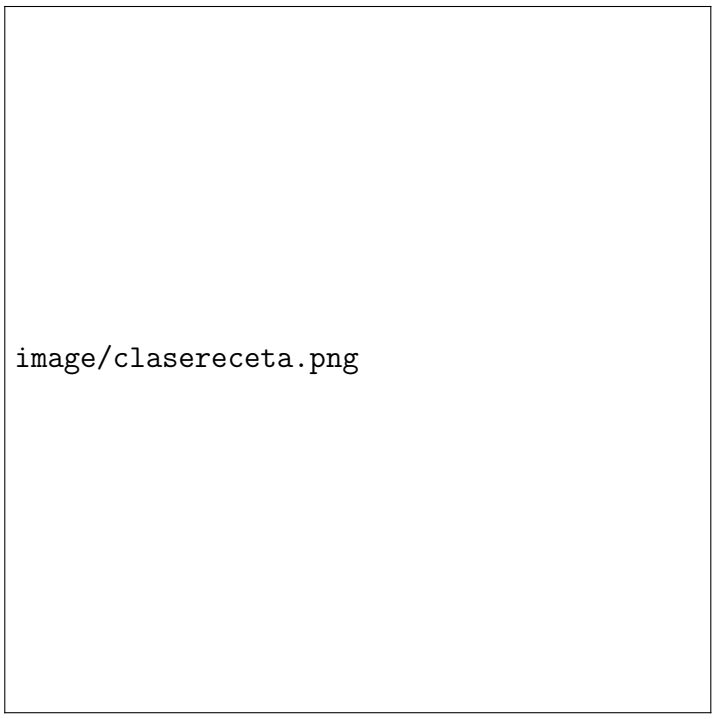


Figura 5.3: Comparación de datos clase: Receta

Como se puede observar las figuras 5.3 y 5.4, fueron clasificados un total de 508 documentos, de los cuales 374 documentos fueron clasificados correctamente y los restantes 134 documentos de manera incorrecta. Con respecto al porcentaje de precisión, se obtuvo un 73.6 %, mientras que [Corrales et al., 2018] obtuvo 66.5 %, por lo que se puede constatar una ligera mejoría al implementar las máquinas de soporte vectorial. Esto apesar de que en la 5.3 se muestra como la fase#1 logró clasificar una mayor cantidad de documentos con la clase receta, la nueva implementación utilizando máquinas de soporte vectorial muestra una mejoría en la clasificación de los documentos que no contienen una receta, es decir evita falsos positivos.

Ahora bien, dado que no se puede concluir que las pruebas son exitosas si no se utiliza un medio para validar los resultados, para este efecto se utilizó la t de *student*. La hipótesis nula que se presenta es que la nueva implementación con máquinas de soporte vectorial presenta una mejoría en la precisión con respecto a la fase #1 de [Corrales et al., 2018]. Esta demostración se muestra en el cuadro 5.1 en el que se despliegan las precisiones obtenidas para cierto documento tanto en la implementacion de [Corrales et al., 2018] como la nueva implementacion que utiliza máquinas de soporte vectorial. Cuando el documento fue etiquetado de manera correcta, la precisión se especifica con 1; de lo contrario, con el valor 0. Además, es necesario especificar que se

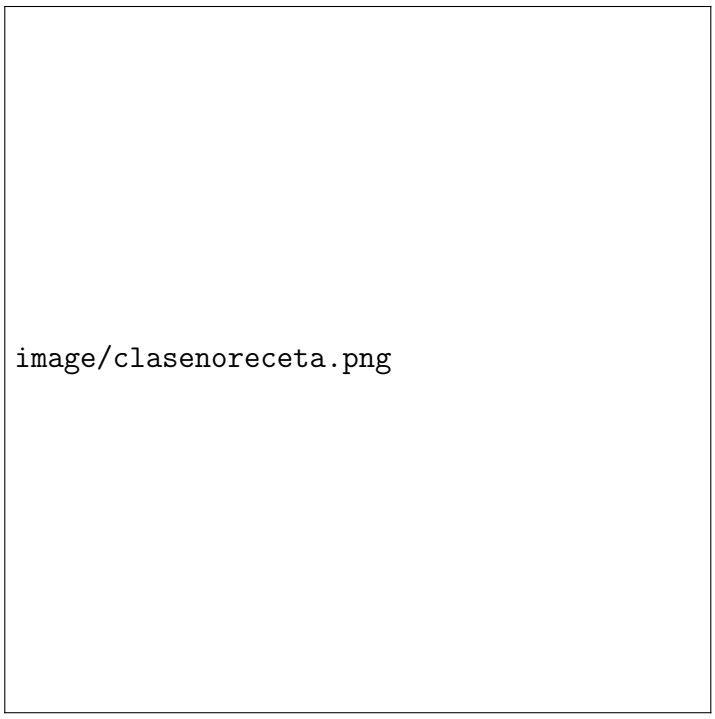


Figura 5.4: Comparación de datos clase: NOreceta

utilizaron $n-1$ grados de libertad en dos muestras independientes con varianzas distintas y que para el nivel de significancia se definió un 10 %, que representa el porcentaje de error aceptado.

A continuación se muestra la demostración que se realizó para validar si es posible rechazar la hipotesis nula. Al definir $t_0 < t_{\alpha/2;n_1+n_2-2}$ se encontraría evidencia para rechazar la hipotesis. En el que t_0 corresponde al valor de la prueba estadística realizada con las pruebas independientes y $t_{\alpha/2;n_1+n_2-2}$ corresponde al valor crítico de la t de *student* utilizando una significancia de 10 %. Como se muestra en el cuadro 5.1, para los documentos clasificados con máquinas de soporte vectorial se obtuvo un promedio $\mu = 0{,}72$ con una desviación estándar $s_1 = 0{,}453$. Mientras que para los documentos de [Corrales et al., 2018] obtuvieron un promedio de $\mu = 0{,}56$ y una desviación estándar $s_2 = 0{,}501$, como se puede observar en la fórmula (5.1).

$$t_0 = \frac{\mu_1 - \mu_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} = \frac{0{,}72 - 0{,}56}{\sqrt{\frac{0{,}453^2}{50} + \frac{0{,}501^2}{50}}} = 1{,}6733 \tag{5.1}$$

Una vez que se obtuvo t_0, se debe de obtener el valor $t_{\alpha/2;n_1+n_2-2} = t\{0{,}05; 98\} = 1{,}660$ entonces $t_{\alpha/2} < t_0$ (5.1) por lo que se puede concluir que la hipótesis nula se acepta.

Es decir, que se observa una mejoría en los procentajes de precisión con respecto a [Corrales et al., 2018], mostrando que la nueva implementacion que utiliza máquinas de soporte vectorial da mejores resultados en la clasificación automatizada de recetas de cocina.

Cuadro 5.1: Precisión

Documento	**Precisión MSV**	**Precisión Fase #1**
366	1	0
464	1	1
112	1	0
410	1	1
42	1	0
288	0	1
15	1	0
289	1	1
297	1	0
259	1	1
290	0	1
416	1	1
499	0	0
469	1	1
99	0	1
3	1	0
204	0	1
40	1	0
490	1	1
167	0	0
130	1	1
386	1	0
145	0	1
294	1	1
202	1	1
41	0	0
390	1	1
406	0	0
332	1	0
37	1	0
298	1	1
407	1	0
146	1	1
185	0	0
476	1	1
291	1	0
279	1	1
421	1	0
444	1	1
318	1	0
435	0	1
495	1	1
248	1	1
144	1	1
180	0	0
489	1	1
56	1	0
471	0	1
92	1	0
392	0	1
Media	**0.72**	**0.56**

Capítulo 6

Conclusiones y trabajo futuro

Una vez completadas cada una de las fases requeridas para este proyecto, quedan como resultado lecciones aprendidas así como recomendaciones para continuar el trabajo en este proyecto, por lo que a continuación se describen las conclusiones y trabajo futuro.

6.1. Conclusiones

Este trabajo muestra el beneficio que se obtiene tras el esfuerzo multidisciplinario entre la lingüística y la computación e informática , en el que se propone una solución de automatización como alternativa computacional a la clasificación manual de documentos en recetas de cocina en el campo de lingüística. La extracción de reglas por parte de un experto conlleva una mayor cantidad de esfuerzo, en cambio la clasificación de datos para un sistema supervisado requiere una menor cantidad de tiempo por parte de los expertos. Por lo tanto, en este sentido el presente trabajo ofrece un aporte importante ya que se trata de un sistema de aprendizaje supervisado.

Este proyecto se propuso implementar un sistema que clasifica en automático recetas costarricenses utilizando máquinas de soporte vectorial con el objetivo de evaluar la precisión de una técnica basada en aprendizaje de máquina para la identificación de documentos de recetas costarricenses a partir de documentos provenientes de la web. Como se evidenció en la sección de la metodología y la de resultados, se logró cumplir con el objetivo general.

Además, se logró cumplir con los objetivos secundarios, al desarrollar una araña dirigida para la recolección de datos (objetivo 1). Además se generó un modelo que permitió la clasificación de documentos para determinar si contienen o no una receta de cocina (objetivo 2) .Por último se realizó una comparación de la precisión entre la implementación de fase 1 y la nueva implementación que utilizó máquinas de soporte vectorial (objetivo 3).

6.2. Trabajo futuro

Después de esta implementación aún existen algunas mejoras que se pueden realizar, por lo que entre mis recomendaciones incluyen utilizar distintos corpus, ya que al analizar los datos que se utilizaron para el entrenamiento del modelo estaban desbalanceados, conteniendo una mayor cantidad de documentos sin recetas. Esto pudo ocasionar que el modelo no fuera entrenado de la manera mas precisa. Además se podría evaluar el desempeño del modelo en corpus más extensos y diversos, así como realizar el etiquetado de datos por expertos, ya que esto aportará más datos al proyecto. Otra recomendación sería utilizar otras técnicas más modernas ahora accesibles en el procesamiento de lenguaje natural como las redes neuronales, que de acuerdo a la literatura parecen prometedores en el campo de la clasificación automática de texto.

Bibliografía

[Alarcón, 2009] Alarcón, R. (2009). Descripción y evaluación de un sistema basado en reglas para la extracción automática de contextos definitorios. En *Tesis Doctoral UPF*, p. 23.

[Arlot y Celisse, 2009] Arlot, S. y Celisse, A. (2009). A survey of cross-validation procedures for model selection. *Revista Káñina, Universidad de Costa Rica*, 4(40-79).

[Bird, 2015] Bird, S. (2015). Natural language processing with python.

[Brownlee, 2018] Brownlee, J. (2018). A gentle introduction to k-fold cross-validation. En *Statistical Methods*, p. 1.

[Carmona, 2014] Carmona, E. (2014). Tutorial sobre máquinas de vectores soporte (svm). En *Universidad Nacional de Educación a Distancia de Madrid (UNED)*, pp. 3–5.

[Corrales et al., 2018] Corrales, S., Miranda, K., Casasola, E., Leoni, A., y Hernández, M. (2018). Análisis de texto para la identificación automática de marcadores lingüísticos definicionales en recetas de gastronomía de costa rica. *Revista Káñina, Universidad de Costa Rica*, 42(3).

[Moujahid, 2016] Moujahid, A. (2016). Una introducción práctica a la profunda aprendizaje con caffe y python.

[Redmore, 2018] Redmore, S. (2018). Machine learning vs. natural language processing.

[Resendiz, 2006] Resendiz, J. (2006). Las máquinas de soporte vectorial para identificación en línea. En *Maestría, Control Automático. Instituto Politecnico Nacional*, pp. 7–22.

[Rouse, 2005] Rouse, M. (2005). Crawler.

A P É N D I C E S

Apéndice A

Lista completa de semillas

Semillas	
1	https://sevilla.abc.es/gurme/las-mejores-recetas/te-presentamos-10-recetas-de-pollo-al-horno/
2	https://cookpad.com/co/recetas/120897-ensalada-de-frutas-para-ninos
3	http://laestrella.com.pa/vida-de-hoy/salud/6-tips-para-preparar-lonchera-escolar-nutritiva-saludable/23846923
4	https://www.directoalpaladar.com/directo-al-paladar/siete-recetas-con-quinoa-una-para-cada-dia-de-la-semana http://recetasconquinoa.es/
5	https://www.yazio.com/es/alimentos/banana.html
6	http://cr.emedemujer.com/cocina/recetas/3-ideas-para-desayunar-recetas/
7	https://www.aceitedecoco.org/2014/04/los-10-beneficios-cientificamente-probados-de-consumir-aceite-de-coco/
8	https://cookpad.com/mx/buscar/desayunos%20de%20costa%20rica
9	http://www.lostiempos.com/doble-click/vida/20180311/consejos-meriendas-saludables-escuela
10	https://www.hola.com/cocina/2017062396249/receta-pollo-en-salsa-blanca-hk/
11	http://quinua.pe/quinua-valor-nutricional/
12	https://saborgourmet.com/sopa-de-platano-verde/
13	https://www.mycolombianrecipes.com/es/sopa-de-platano
14	http://www.diabetesforecast.org/2011/mar/es/el-uso-de-aceites-en-la-cocina.html
15	https://www.flores.ninja/palmito/
16	https://es.tastemade.com/recetas/Recetas-de-Pastas
17	http://www.chopchopmag.org/recipe/huevos-revueltos-con-espinacas
18	https://www.composicionnutricional.com/alimentos/KALE-RAW-7
19	https://www.saboresenlinea.com/recetas/gallo-pinto
20	https://www.buzzfeed.com/gretaalvarez/deliciosdas-recetas-de-pasta
21	http://www.mabelamaro.com/salud/2012/08/09/propiedades-nutritivas-de-la-banana/
22	https://www.alimmenta.com/dietas/adelgazamiento/adelgazar-10-kilos/

23	https://www.cuerpomente.com/blogs/gastronomia-consciente/aceites-para-cocinar-como-elegir-como-usar_1442
24	https://www.guiadelnino.com/recetas-para-ninos-y-bebes/frutas/ensalada-de-fruta-en-un-cochecito
25	https://expansion.mx/lifestyle/2015/05/14/por-que-tienes-hambre-todo-el-tiempo
26	https://www.organicfacts.net/beneficios-de-salud/acites/los-beneficios-del-aceite-de-coco-para-la-salud.html?lang=es
27	http://nutricionycocina.es/propiedades-nutricionales-de-la-quinoa/
28	https://www.cocinadelirante.com/guarnicion/recetas-de-piernas-de-pollo-al-horno
29	http://recetasdecostarica.blogspot.com/2012/03/picadillo-de-palmito.html
30	https://cnnespanol.cnn.com/2017/06/04/debemos-comer-tres-comidas-grandes-o-muchas-comidas-pequenas/
31	https://www.superalimentos.pro/aceite-de-coco/
32	https://www.elconfidencial.com/alma-corazon-vida/2016-11-13/4-situaciones-en-las-que-nunca-jamas-debes-usar-aceite-de-oliva_1288284/
33	https://www.cocinacaserayfacil.net/recetas-cocina-faciles-rapidas/
34	https://recetastipicascr.com/recetas/olla-de-carne/

Apéndice B

Análisis de texto para la identificación automática de marcadores lingüísticos definicionales en recetas de gastronomía de Costa Rica

ANÁLISIS DE TEXTO PARA LA IDENTIFICACIÓN AUTOMÁTICA DE MARCADORES LINGÜÍSTICOS DEFINICIONALES EN RECETAS DE GASTRONOMÍA DE COSTA RICA

Text analysis for automatic identification of definitional linguistic markers in Costa Rican gastronomy recipes

Sharon Corrales[1], Karen Miranda[2]**, Édgar Casasola[3]***, Antonio Leoni[4]****, Mario Hernández[5]******

RESUMEN

El análisis de contextos definicionales permite clasificar y sistematizar las informaciones definicionales pertenecientes a un dominio específico y, posteriormente, identificar estándares de las formas en que se definen las palabras y términos en tal dominio. En este artículo se describe el proceso realizado para automatizar el análisis de contextos definicionales en el dominio gastronómico de Costa Rica. La labor se realizó mediante el uso de herramientas computacionales para el procesamiento de lenguaje natural. La automatización permite el análisis sobre grandes volúmenes de datos y obtener resultados en menos tiempo del requerido por el análisis manual. Ahora bien, el procedimiento consta de dos módulos, uno de clasificación de documentos en textos con recetas o sin ellas, y un segundo módulo de identificación de los ingredientes de cocina con base en patrones lingüísticos formales.

Palabras clave: análisis lingüístico de recetas, análisis de contextos definicionales, patrones definicionales, marcadores definicionales, procesamiento del lenguaje natural.

ABSTRACT

The analysis of definitional contexts allows to classify and systematize the definitional information belonging to a specific domain, and then to identify standards for the forms in which words and terms are defined in this domain. This paper describes the process implemented to automate the analysis of definitional contexts in the gastronomy domain in Costa Rica. The automation was done by using computational tools for natural language processing. The automation enables analysis of large quantities of data and results in less time than required by manual analysis. Automation consists of two modules, the first one is for the classification of documents in texts with or without recipes and the second one is for the identification of recipe ingredients based on formal linguistic patterns.

Key words: linguistic analysis of recipes, analysis of definitional contexts, definitional patterns, definitional markers, natural language processing.

[1]* Estudiante de la Maestría en Computación, UCR. Correo e.: sharoncm.1691@gmail.com.

[2]** Estudiante de la Maestría en Computación, UCR. Correo e.: karenmh09@gmail.com.

[3]*** Escuela de Computación e Informática y Posgrado en Computación, UCR. Correo e.: casasola@gmail.com.

[4]**** Escuela de Filología, Lingüística y Literatura y Posgrado en Lingüística, UCR. Correo e.: a.leoni@me.com.

[5]***** Programa Estudios de Lexicografía, UCR. Correo electrónico: pdfmario@gmail.com.

1. Introducción

El análisis de contextos definicionales o definitorios (en adelante, análisis de CD) es una línea de investigación que permite, en primer lugar, clasificar y sistematizar las informaciones definicionales relativas a un dominio restringido. Posteriormente, esa organización conceptual puede servir tanto para la recuperación de relaciones semánticas definitorias a partir de textos como para la estandarización de las formulaciones definicionales del dominio de especialidad estudiado (cf. Alarcón 2003; Alcina y Valero 2008; Sierra, Alarcón y Aguilar 2006).

A causa del interés en las posibilidades de este tipo de estudios, surge el proyecto "Análisis de contextos definicionales en corpus de gastronomía tradicional en Costa Rica (CODEGAT)", investigación que pretende examinar la información gastronómica presente en textos de recetas costarricenses con el fin último de aportar a la sistematización del conocimiento gastronómico socializado.

Parte importante de esa sistematización es la adecuada identificación de la lista de productos/ingredientes que serán objeto de las diversas acciones y procesos, así como la precisa descripción de las tareas paralelas y secuenciales en las que aquellos se utilizarán. Desde el enfoque del análisis de CD, que es el que aquí seguimos, lo fundamental es identificar las formas recurrentes que se utilizan efectivamente en los textos para la expresión de las relaciones conceptuales pertinentes (cf. Sierra 2009, Valero 2009, Valero y Alcina 2009, Soler 2005, Sierra y Alarcón 2002). A esas formas recurrentes se les llama, en esta perspectiva investigativa, "patrones definicionales", cada uno de los cuales asocia una **clase de contenidos semánticos** con una **clase de formas que sirven para introducirlos** dentro de la cadena textual (y que funcionan como marcadores, señalizadores, indicadores).

A pesar de tener ya varios lustros en desarrollo, la línea de análisis de CD no cuenta aún con paradigmas metodológicos de empleo universal[1]. Sin embargo, una característica esencial de su planteamiento es la automatización de los procedimientos de identificación y validación de los patrones definitorios propuestos, así como de la recuperación de las relaciones conceptuales pertinentes. Esta automatización permite trabajar sobre grandes volúmenes de datos y obtener resultados en menos tiempo que el requerido por el análisis manual. Debido a lo anterior, el equipo de trabajo de CODEGAT incluye tanto a lingüistas como a especialistas con conocimientos en procesamiento del lenguaje natural.

Ahora bien, en relación con el proceso del análisis de texto, este se dividió en dos módulos (v. figura 1). El primer módulo corresponde a la clasificación de los documentos en aquellos que contienen información de recetas y aquellos que no la contienen. Una vez así clasificados, se toman solamente los documentos contenedores de recetas y se aplica el segundo módulo. En este, las palabras de cada documento son etiquetadas según su categoría gramatical. Posteriormente, sobre el texto etiquetado se buscan marcadores lingüísticos y se genera un documento de resultado el cual contiene marcados los ingredientes dentro de la receta.

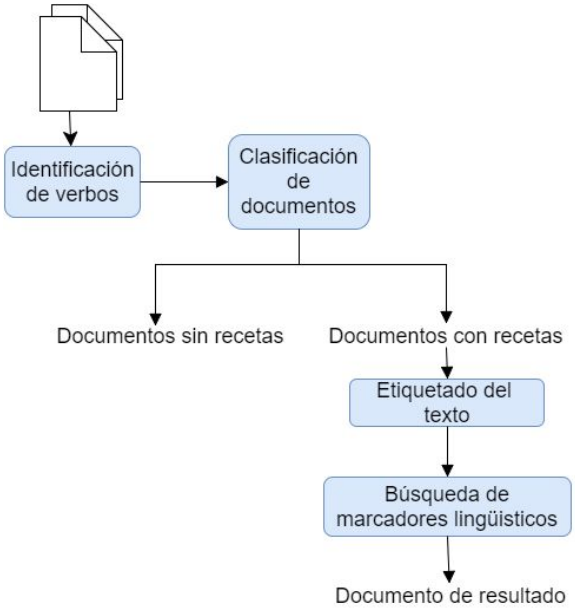


FIGURA 1
Descripción del proceso realizado

Antes de aplicar el procedimiento de dos módulos, se realizó una descarga masiva de documentos mediante una araña de búsqueda. Esta araña estaba guiada por hipervinculados brindados por los lingüistas del equipo.

En cuanto a las herramientas computacionales utilizadas en el proceso descrito en la figura 1, estas corresponden a agentes automáticos de recolección de información, expresiones regulares y etiquetador de partes del discurso:

· **Agente automático de recolección de información:** Proceso que se ejecuta de forma continua, sigue enlaces y busca adelante por información inferida (cf. Casasola y Gauch 1997).

· **Expresión regular:** Secuencia de caracteres utilizada como patrón para describir, manipular y realizar búsquedas dentro de texto. Es una herramienta sumamente flexible y eficiente para procesamiento de texto (cf. Fitzgerald 2012, Friedl 2006, Habibi 2004).

· **Etiquetador de partes del discurso (POS tagger):** Software para asignar a cada palabra dentro del texto una etiqueta con base en la función que asume en la oración (referida especialmente a la clase léxica o la clase morfológica). Este etiquetado es importante en el área de recuperación de información y procesamiento de lenguaje natural porque encapsula datos propios de la palabra (número, género, tiempo verbal, entre otros), así como de sus palabras vecinas (cf. Hasan, UzZaman y Khan 2007).

2. Procesamiento

2.1. Clasificación de documentos

Este módulo es el encargado de analizar los documentos para clasificarlos en dos categorías: documentos con recetas y documentos sin recetas. La implementación de la lógica del módulo se realizó en dos etapas.

2.1.1. Etapa 1

En esta etapa se utilizaron únicamente documentos de recetas previamente analizados por los lingüistas involucrados en la investigación. Estos documentos se analizaron mediante el uso de un etiquetador de partes del discurso (POS tagger) para identificar de manera automática los verbos presentes en los textos. El resultado del proceso mostraba la lista de los verbos identificados, con su correspondiente forma en infinitivo (lema) y su frecuencia absoluta de aparición en los textos analizados.

A partir de este resultado, se consideró el papel desempeñado por cada verbo en las recetas de cocina. De esta manera, se clasificaron los verbos en dos

categorías de significancia (media y alta) basándose en qué tanto es exclusivo cada verbo del dominio de la gastronomía y las recetas de cocina, lo cual es útil para una identificación automática de recetas. Por ejemplo, algunos verbos encontrados en los textos que se estudiaron son de uso común en otros dominios y, por lo tanto, no pueden asociarse de manera única al contexto de cocina. Sin embargo, otros verbos son claramente exclusivos el discurso gastronómico, hecho que permite pensar en la posibilidad de utilizarlos instrumentalmente para la identificación de recetas de cocina dentro de un corpus textual inicialmente indiferenciado.

- **Significancia media**: Verbos comunes en diversos dominios y, por tanto, no exclusivos de los contextos de recetas. Por ejemplo: cocinar, servir, hacer, mezclar.

- **Significancia alta**: Verbos que pueden asociarse comúnmente al contexto de la gastronomía y las recetas de cocina. Por ejemplo: amasar y picar.

2.1.2. Etapa 2

Para la segunda etapa, se tomó como base de la clasificación de documentos el resultado obtenido en la etapa 1 correspondiente a la significancia de los verbos. Además, se procedió a la creación de un agente automático de recolección de información (v. Casasola y Gauch 1997) para la descarga masiva de documentos de internet. Este agente utiliza inicialmente un conjunto de páginas web a las que se les conoce como *semillas* del agente. Estas semillas son visitadas y cualquier enlace incluido en ellas es agregado a la lista de páginas web por visitar y descargar. Por otro lado, para asegurarse de que la mayoría de los documentos descargados sean gastronómicos, este agente confronta los documentos por descargar contra una lista de páginas web ya validadas. Esta lista de páginas válidas fue previamente brindada por los lingüistas del equipo, quienes las analizaron según criterios de contenido y las valoraron como páginas con gran densidad de recetas de cocina.

Por otro lado, para la clasificación automática de documentos se construyó un programa que recibe como parámetro la ubicación de la carpeta donde se encuentran los documentos descargados por el agente y procede a analizar cada uno de ellos. El primer paso del proceso es el etiquetado del texto de los documentos mediante un POS tagger para identificar los verbos presentes.

Posteriormente, se verifica si los verbos identificados se encuentran clasificados en la lista de verbos de significancia media o alta. Si el documento contiene al menos un verbo de significancia alta o si contiene al menos cuatro verbos de significancia media, el documento es clasificado como contenedor de recetas. Por lo tanto, todo verbo no clasificado previamente tendrá impacto nulo en la clasificación del documento.

Con respecto a la elección de un mínimo de cuatro verbos de la categoría de significancia media, esto se debe a que cada uno de estos verbos por sí solo no permite asegurar que el documento contiene recetas. Sin embargo, la aparición de varios de estos verbos en un mismo documento aumenta la posibilidad de que tal documento efectivamente contenga descripciones culinarias.

Por último, el programa genera un archivo de resultado en el cual se indica la clasificación asignada a cada documento analizado. Este archivo de resultado presenta, además, los verbos detectados en el texto del documento durante el análisis, tal como se ejemplifica en la tabla 1.

TABLA 1
Ejemplos de resultado de clasificación de documentos

RESULTADO

El archivo 0 es receta. Verbos: [cocer] Verbos [comer, cocinar, adornar, servir, probar]
El archivo 102 es NO receta. Verbos: [ ] Verbos: [servir]
El archivo 135 es receta. Verbos: [hornear, batir] Verbos: [ ]
El archivo 179 es receta. Verbos: [ ] Verbos: [moler, majar, arrollar, pelar, enfriar, cocinar, cortar]

Por otro lado, con el fin de evaluar la precisión del programa, se realizó una clasificación manual de los documentos, de modo que se pudieran comparar los resultados automáticos contra los manuales. Según la matriz de confusión resultante (tabla 2), la precisión del sistema resultó ser del 77%.

TABLA 2
Matriz de confusión de la clasificación de documentos: clasificación obtenida contra clasificación real

		Clasificación obtenida	
		Receta	No-Receta
Clasificación real	Receta	301	82
	No-Receta	88	37

2.2. Identificación de ingredientes

Este módulo trabaja únicamente sobre los documentos clasificados en el módulo anterior como documentos con recetas. Antes de iniciar el procesamiento propio de este componente, se aplica sobre el texto un preprocesamiento para normalizarlo y estandarizarlo. La normalización consiste en la eliminación o sustitución de los caracteres especiales, conversión de todo el texto a minúscula, eliminación de espacios múltiples entre palabras, entre otros. Por su parte, la estandarización consiste en la transformación de caracteres especiales de representación de fracciones (por ejemplo, ½, ¼) a su forma normal, es decir, mediante caracteres individuales (por ejemplo, 1/2, 1/4, respectivamente). Es conveniente señalar que en un inicio este preprocesamiento no se realizaba. Sin embargo, la falta de estandarización en el texto provocaba problemas en la identificación de los marcadores lingüísticos.

Ahora bien, el proceso realizado en este segundo módulo ha tenido dos etapas. La mayor diferencia entre ambas es el paso de usar expresiones regulares basadas en reglas con palabras específicas a expresiones regulares basadas en reglas con categorías gramaticales.

2.2.1. Etapa 1

Los lingüistas del equipo brindaron un corpus de prueba (CP) constituido por texto plano rico en información gastronómica (y, especialmente, denso en recetas). El corpus CP fue extraído de internet por medio de un proceso automatizado y, posteriormente, depurado y prenormalizado de forma masiva para ser utilizado como corpus anónimo. También ofrecieron una lista de formas lingüísticas postuladas como marcadores definicionales de ingredientes –en la figura 2 se pueden ver algunos ejemplos–.

```
2_barras_de_
2_botellas_de_
2_cabezas_de_
2_[X]_grandes
2_[X]_medianas
2_[X]_pequeñas
2_[X]_picadas
2_[X]_tiernos
2_cucharadas_de_
2_ cucharaditas_de_
2_dientes_de_
2_hojas_de_
2_[X]_batidos
```

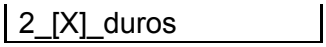


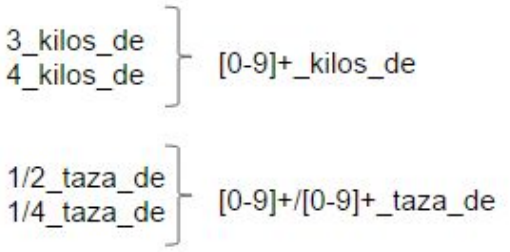
FIGURA 2
Ejemplos de marcadores lingüísticos iniciales

Estos candidatos a marcadores habían sido previamente identificados mediante un proceso manual de análisis de un corpus base (CB) –diferente del corpus de prueba CP ya mencionado–; luego fueron generalizados (o pregeneralizados) utilizando una simbología que pudiera servir de transición hacia la posterior formulación mediante expresiones regulares. La simbología de esa generalización inicial se puede observar en la tabla 3.

TABLA 3
Simbología de los marcadores lingüísticos iniciales

Notación	Significado
[X]	Conjunto obligatorio y variable de 1 o más letras
[[X]]	Conjunto opcional y variable de 1 o más letras
[[elemento]]	Elemento opcional
_	Espacio en blanco

Los marcadores lingüísticos brindados fueron, entonces, transformados por medio de simbología utilizada por las expresiones lingüísticas del lenguaje de programación seleccionado para la automatización. Este paso a expresiones regulares permitió abstraer los patrones iniciales y, por ende, se disminuyó la cantidad de patrones por evaluar. La abstracción se logró en su mayoría al pasar de números específicos (0, 1, 2, 3, 4, 5,... n) a una expresión regular de un conjunto de números, como en:

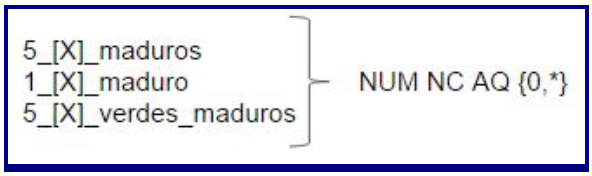


Estas expresiones regulares se buscaron en el texto para identificar los puntos de inserción de ingredientes en las recetas de cocina. Por último, se generaba un documento de resultados en el que se se señalaban los marcadores encontrados y se desplegaba la frecuencia absoluta de aparición de cada uno de ellos. De esta manera se lograron identificar también marcadores que no resultaban útiles para la investigación, debido a su baja o nula frecuencia absoluta de aparición. Además, se logró identificar los casos de superposición de patrones; una vez identificadas y evaluadas estas superposiciones, se eliminaron las expresiones regulares que producían las redundancias.

A pesar de lograr identificar la mayoría de los ingredientes de cocina por medio de marcadores, también había aquellos que no lograban ser detectados. Una vez analizados los resultados, se observó que muchos de los casos de ingredientes no identificados se debían a falta de coincidencia entre el género o número gramatical de las formas que aparecían en el texto y el género o número de las formas postuladas por los marcadores. Además, no todas las medidas y sus diversas formas de escribirse estaban consideradas en los marcadores. A partir de la revisión de esos resultados, se llegó a la conclusión de que era necesario generalizar de manera un poco diferente los marcadores, de modo que siempre incluyeran al menos todas las posibles inflexiones de género y número.

2.2.2. Etapa 2

Esta segunda versión del módulo se desarrolló con el fin de solucionar el problema de las inflexiones nominales (número) y adjetivales (género y número), además de otras generalizaciones pertinentes. Para esto se empleó un etiquetador de partes del discurso (POS tagger) con un modelo correspondiente al lenguaje español. Este etiquetador permitió pasar de las expresiones regulares de la primera etapa a expresiones regulares basadas en categorías gramaticales.



Para la construcción de estas nuevas expresiones regulares se utilizaron los marcadores brindados inicialmente. Además, las categorías de interés corresponden únicamente a los valores numéricos, sustantivos comunes, adjetivos calificativos, signos de puntuación y preposiciones. Esto permitió disminuir más la cantidad de marcadores por evaluar en el texto, ya que las categorías gramaticales generalizaron valores específicos.

Primeramente, este módulo toma cada uno de los documentos y etiqueta su texto según las partes del discurso. En el siguiente paso, se analiza el texto etiquetado para identificar la presencia de los marcadores definicionales de interés. Por último, el proceso genera un documento de resultados por cada documento analizado. En este documento de resultados se presentan señalados los marcadores definicionales e ingredientes encontrados en el documento.

En cuanto a los resultados, estos se evaluaron cuantitativamente según la cantidad de ingredientes identificados correctamente con respecto al total de los ingredientes en los documentos analizados. Este módulo identificó correctamente y en forma automática el 53% de los ingredientes.

3. Conclusiones

El uso de herramientas de computación para el procesamiento automático de texto demostró ser de utilidad para la recolección, clasificación automática e identificación de ingredientes de recetas de gastronomía con base en marcadores lingüísticos. En relación con el POS tagger, este permitió identificar los verbos dentro del texto para hacer una discriminación automática de documentos, así como considerar las inflexiones nominales y adjetivales en los marcadores lingüísticos. Además, el uso de esta herramienta permitió disminuir la cantidad de tiempo en el desarrollo del proceso, ya que no se requirió realizar manualmente la clasificación en categorías gramaticales de cada una de las palabras en los documentos.

Asimismo, la combinación entre expresiones regulares y categorías gramaticales permitió la generalización y expansión de los marcadores que los lingüistas del equipo habían brindado pregeneralizados.

Trabajo futuro. La investigación presentada en este artículo proyecta extenderse y optimizarse para la obtención de mejores resultados. El plan es dividir las tareas computacionales en dos temas a ser desarrollados como trabajos finales de investigación aplicada (TFIA) en la Maestría en Computación. El primero se refiere a la clasificación automática de documentos utilizando aprendizaje de máquina y marcadores lingüísticos. Este trabajo se enfocará en la clasificación de textos para diferenciar entre archivos que tienen información gastronómica (específicamente, recetas) y archivos que no la tienen; esto requerirá trabajar en conjunto con los

expertos en el área de lingüística para brindar pesos a los verbos utilizados para la clasificación.

Por otro lado, el segundo tema corresponde al análisis automático de textos de recetas de cocina para la identificación de procesos paralelos y secuenciales. Este trabajo podrá utilizar como base el proceso realizado para la identificación de los ingredientes en las recetas de cocina por medio de patrones definicionales, y requerirá además contar con un listado de marcadores lingüísticos definitorios asociados a los diversos pasos/tareas/etapas de los procedimientos culinarios.

Notas

1. Precisamente por la búsqueda de modelos metodológicos eficaces con vista en los objetivos del análisis de CD, parte importante de los estudios enmarcados en esta línea de investigación es la propuesta y continuo afinamiento de los procedimientos aplicados a las diversas etapas.

Referencias

Alarcón, Rodrigo. (2003). Análisis lingüístico de contextos definitorios en textos de especialidad. Tesis de licenciatura: Universidad Nacional Autónoma de México.

Alcina, Amparo y Esperanza Valero. (2008): "Análisis de las definiciones del diccionario cerámico científico-práctico. Sugerencias para la elaboración de patrones de definición". En: *Debate Terminológico*, 4. http://seer.ufrgs.br/index.php/riterm/article/download/23841/13830. Consulta: 20/02/2017.

Casasola, Édgar y Susan Gauch. (1997). Intelligent Information Agents for the World Wide Web. [Information and Telecommunication Technology Center, Technical report ITTC-FY97-111100-1]. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.3628&rep=rep1&type=pdf. Consulta: 22-02-2017.

Elleithy, Khaled (ed.). (2007). *Advances and Innovations in Systems, Computing Sciences and Software Engineering*. Dordrecht, Holanda: Springer.

Fitzgerald, Michael. (2012). *Introducing Regular Expressions*. California: O'Reilly Media Inc.

Friedl, Jeffrey E. F. (2006). *Mastering Regular Expressions* (3a. ed.). California: O'Reilly Media Inc.

Habibi, Mehran. (2004). *Java Regular Expressions: Taming the java.util.regex Engine*. Nueva York: Apress Media, LLC.

Hasan, Fahim Muhammad, Naushad UzZaman y Mumit Khan. (2007). "Comparison of different POS Tagging Techniques (N-Gram, HMM and Brill's tagger) for Bangla". En: Elleithy (ed.): 121-126.

Sierra, Gerardo. (2009). "Extracción de contextos definitorios en textos de especialidad a partir del reconocimiento de patrones lingüísticos". En: *linguaMATICA*, 2: 13-37.

Sierra, Gerardo y Rodrigo Alarcón. (2002). "Identification of recurrent patterns to extract definitory contexts". En: *Lecture Notes in Computer Science*, 2276: 436-438.

Sierra, Gerardo, Rodrigo Alarcón y César Aguilar. (2006). "Extracción automática de contextos definitorios en textos especializados". En: *Revista de Procesamiento de Lenguaje Natural*, 37: 351-352.

Sierra, Gerardo, Mara Pozzi y Juan Manuel Torres (eds.). (2009). Proceedings. (1st) International Workshop on Definition Extraction, 18 de setiembre de 2009, Borovets, Bulgaria. https://aclweb.org/anthology/W/W09/W09-4400.pdf. Consulta: 20/02/2017.

Soler, Victoria. (2005). Patrones lingüísticos para la búsqueda de información conceptual en el corpus textual especializado de la cerámica TXTCera. http://repositori.uji.es/xmlui/bitstream/handle/10234/79115/forum_2004_50.pdf?sequence=1. Consulta: 20/02/2017.

Valero, Esperanza. (2009). Los marcadores lingüísticos en las definiciones del grupo conceptual 'procesos de fabricación cerámica'. http://repositori.uji.es/xmlui/bitstream/handle/10234/78051/forum_2008_22.pdf?sequence=1. Consulta: 20/02/2017.

Valero, Esperanza y Amparo Alcina. (2009). "Linguistic realization of conceptual features in terminographic dictionary definitions". En: Sierra, Pozzi y Torres (eds.): 54–60.