

ARPIA: Aprendizaje Remoto Personalizado Inteligente Automatizado

Miriam Morata Vegas

Resumen— La adaptación de la enseñanza universitaria al nuevo contexto educativo ha conllevado un cambio en la metodología docente, ahora centrada en el alumno. En este punto, es necesario adaptar la comunicación entre profesores y estudiantes para que ambos se sientan implicados como protagonistas. Parte de esta comunicación es la obtención de feedback de manera continua para poder adaptarse a dificultades en el aprendizaje. En este TFG se explora el uso de tecnologías de aprendizaje computacional para desarrollar un sistema de soporte al aprendizaje (ARPIA) mediante el cual los estudiantes se verán retados con preguntas seleccionadas a partir de los conocimientos que hayan ido demostrando. El desarrollo del sistema de aprendizaje se ha basado en la técnica Bag of Words y se han implementado dos estrategias de clasificación diferentes, una basada en comparación de histogramas y otra en ranking de palabras. Asimismo se ha desarrollado una primera versión de la aplicación web, incluyendo el diseño de la base de datos no relacional y el diseño de la interfaz web. Los resultados preliminares muestran una clasificación prometedora de las preguntas, sobre todo usando el ranking de palabras, obteniendo un accuracy medio de 0,88.

Palabras clave— Soporte aprendizaje, e-learning, aprendizaje computacional, Bag of Words, No SQL, histogramas.

Abstract— The adaptation of university education to the new educational context has implied a change in the teaching methodology, now centered on the student. At this point, it is necessary to adapt the communication between teachers and students so that both feel involved as protagonists. Part of this communication is feedback acquisition in a continuous way so difficulties in learning can be easily detected. This TFG explores the use of computational learning technologies to develop a learning support system (ARPIA) through which students will be challenged with specific questions based on the knowledge they have been demonstrating. The development of the learning system is based on the Bag of Words technique and two different classification strategies have been implemented, one based on histogram comparison and one based on word ranking. A first version of the web application including the design of the non-relational database and the design of the web interface has also been developed. Preliminary results detail a promising classification of the questions, especially using the word ranking, obtaining an average accuracy of 0.88.

Index Terms— Learning support, e-learning, computer learning, Bag of Words, No SQL, histograms.



1) INTRODUCCIÓN

Este documento describe completamente el proceso de creación de un sistema recomendador empleado para mejorar la experiencia docente en el contexto de una asignatura del grado de Ingeniería Informática de la UAB

1.1 Contexto

Actualmente los profesores tienen muy pocas herramientas para recibir feedback sobre los conocimientos adquiridos por los alumnos de la asignatura; los únicos momentos en que se refleja el estado de la clase son los periodos de evaluación tales como exámenes y entrega de prácticas.

Fuera de estos periodos evaluativos, se ha observado una falta de motivación general de los alumnos para estudiar fuera de clase y no hay ninguna plataforma que les acerque a la asignatura de forma dinámica y fácil. Con esta idea en mente, el sistema que se pretende realizar en este TFG está

pensado para ayudar en el proceso de aprendizaje tanto a los alumnos como al profesor, convirtiéndolo en una actividad más eficiente y dinámica, que a la vez generará feedback al profesor.

Este proyecto viene motivado asimismo al observar el interés creciente de los alumnos en la asignatura Compiladores (asignatura de la mención de Computación del grado de Ingeniería Informática de la UAB) al poder participar en pequeños cuestionarios durante los seminarios de problemas usando la herramienta Kahoot! [1,2,3].

El hecho de realizar estos cuestionarios podía suponer un pequeño incremento en la nota final (aunque todos podían llegar a la nota máxima sin realizarlos) y suponía un beneficio adicional para el profesor (al tener feedback inmediato sobre el progreso de los alumnos en la asignatura) y para el alumno (que podía saber si un concepto dado lo había asimilado o no).

El paso siguiente que se ha considerado es implementar esta herramienta (o una similar) también fuera del aula, de modo que el profesor pueda tener feedback de los estudiantes mientras trabajan en la asignatura.

- E-mail de contacte: miriam.morata@e-campus.uab.cat
- Menció realitzada: Computació
- Treball tutoritzat per: Jorge Bernal del Nozal (Ciencias de la Computación)
- Curs 2019/20

1.2 Estado del arte

Existen diferentes tipos de herramientas para obtener feedback a partir de actividades realizadas dentro del aula, de las cuales debemos tomar ejemplo tanto de sus funcionalidades como de sus puntos fuertes y débiles:

1) Kahoot [1]: Permite la creación de encuestas interactivas con participación del público. Se accede a las encuestas mediante un código desde cualquier dispositivo. Cuando las preguntas han sido contestadas, se muestra la respuesta con más votos.

2) Socrative [4]: La aplicación es un gestor de la participación de los estudiantes en el aula en tiempo real. Esta herramienta de apoyo al aprendizaje permite crear ejercicios, quizz e informes para la monitorización del progreso de los alumnos en las aulas.

3) PlayBrighter [5]: Plataforma para la creación de entornos de aprendizaje personalizados la cual requiere que el personal docente genere "misiones" y objetivos que deberán cumplir los estudiantes. Además, dispone de preguntas que se pueden proponer a los alumnos.

Ninguna de estas herramientas integra el conocimiento del usuario para ajustar su experiencia de aprendizaje; es decir, todos los alumnos se enfrentan a las mismas preguntas independientemente de su nivel de conocimientos. Por tanto, surge la necesidad de tener una herramienta que se adapte a cada estudiante y le ayude a conseguir, a su ritmo, los objetivos de aprendizaje de la asignatura.

1.3 Objetivos

El objetivo principal del proyecto es la creación de un módulo de procesamiento del lenguaje para el análisis de preguntas cortas de examen. Este módulo se integra en una primera aplicación denominada ARPIA: Aprendizaje Remoto Personalizado Inteligente Automatizado.

El propósito de esta aplicación es poder proponer al alumno preguntas para poder validar su grado de asimilación de conceptos básicos y también para poder recomendar preguntas adicionales basadas en los aciertos y los fallos de preguntas anteriores. Este módulo se integrará posteriormente dentro de un videojuego, como parte de un TFG que será desarrollado en el segundo semestre del curso 2019-2020.

Los objetivos secundarios de este proyecto están desglosados en la Tabla 1. Definir bien estos objetivos es muy importante para poder concretar y definir las tareas en que se dividirá el proyecto.

Objetivos	Prioridad
Diseño base de datos No SQL	Alta
Creación de interfaz para ver e insertar preguntas	Media
Selección e implementación de la herramienta de aprendizaje computacional más adecuada	Alta
Preparación de una demo del sistema	Baja

Tabla 1 Objetivos del proyecto

1.4 Requisitos del proyecto

Para la realización adecuada del proyecto se han de fijar unos requerimientos de hardware y software. Los requerimientos del sistema se separan en requerimientos mínimos y requerimientos recomendados.

Los requerimientos mínimos de sistema son:

- Sistema Operativo: Windows 7, MacOS, Linux o Android 7.0
- Memoria RAM: 2 GB
- Espacio en disco: 150MB

Los requerimientos recomendados de sistema son:

- Sistema Operativo: Windows 7 o posterior, MacOS o Linux
- Memoria RAM: Mínimo 4 GB
- Espacio en disco: 300MB

Para la realización del proyecto se ha utilizado:

- Windows 10 64-bit
- AMD Ryzen 7 1700X Eight-Core Processor 3.40 GHz
- 16GB de RAM
- MP500 4580 GB y 3000MB/sec

En cuanto a las funcionalidades del sistema, definimos a continuación algunos de los principales requisitos funcionales y no funcionales.

ID	Requerimientos	Tipo
R1	El sistema ha de recomendar preguntas en función del conocimiento del usuario	Funcional
R2	El sistema debe contar con un formulario para introducir preguntas	Funcional
R3	El sistema debe permitir a los usuarios responder preguntas de las asignaturas	Funcional
R4	La aplicación web debe tener un diseño "Responsive" para una adecuada visualización en múltiples dispositivos	No Funcional
R5	Base de datos No SQL	No Funcional
R6	Visualización de las preguntas introducidas	Funcional
R7	Uso de formato JSON para fácil integración de sistema de aprendizaje y BBDD	No Funcional

Tabla 3 Requerimientos del proyecto

1.5 Metodología del proyecto

Para realizar de manera correcta este proyecto se ha apostado por una metodología incremental que incluye reuniones semanales con el tutor. Debido a estos requerimientos, se ha optado, dentro del conjunto de metodologías Agile [6], por la metodología agile Kanban [7].

Esta metodología se basa en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Esta metodología tiene como objetivo gestionar de manera general cómo se van completando las tareas.

1.6 Planificación

El desarrollo de ARPIA se ha dividido en las siguientes etapas:

- 1) Diseño de la base de datos (inicio del proyecto hasta informe de progreso I)
- 2) Diseño de la interfaz web para introducir y visualizar las preguntas del sistema (inicio del proyecto hasta informe de progreso I)
- 3) Introducción de preguntas en la base de datos para validación de ARPIA (inicio del proyecto hasta informe de progreso II)

- 4) Desarrollo del sistema de aprendizaje computacional (informe de progreso II)
- 5) Validación off-line de ARPIA (desde informe de progreso II hasta final del TFG). Esta fase incluye el desarrollo de una pequeña demo del sistema.

1.7 Estructura del documento.

El documento se estructura de la siguiente manera. Tras esta introducción, la sección 2 muestra el desarrollo completo de ARPIA, incluyendo tanto el desarrollo del sistema de aprendizaje computacional como el de la aplicación web de soporte. La sección 3 explica el diseño experimental que se ha seguido para su validación; los resultados de los experimentos realizados se presentan en la sección 4. Finalmente la sección 5 expone las conclusiones principales extraídas del TFG así como indica las líneas de trabajo futuro.

2 DESARROLLO

En esta sección se detalla el desarrollo de la aplicación ARPIA en su primera versión. Previamente a la explicación del desarrollo de la aplicación en sí, se comienza por una descripción de la base de datos que se ha diseñado para contener toda la información que necesita la aplicación. Una vez diseñada la BBDD, se explica cómo se ha diseñado la interfaz de usuario que permite una introducción ágil y sencilla de preguntas en el sistema. Finalmente se detalla el desarrollo del sistema de aprendizaje computacional que se ha escogido para diseñar el sistema recomendador de preguntas.

2.1 Diseño de base de datos

En ARPIA se ha usado NOSQL [8] como implementación de la base de datos, con el objetivo de favorecer la rapidez al acceso de los datos por las técnicas de machine learning en contraposición a las bases de datos SQL [9][10].

Se ha decidido diseñar una única estructura para almacenar toda la información referente a las preguntas y respuestas de una asignatura, de esta manera podemos fácilmente categorizarlas con el objetivo final de crear un sistema recomendador.

La base de datos está orientada a la gestión de múltiples asignaturas; cada una de ellas contará con los parámetros de nombre, temas, cada tema contará con preguntas y cada pregunta a su vez contará con respuestas, categoría (tags) y dificultad asociada, todos estos datos facilitarán la clasificación.

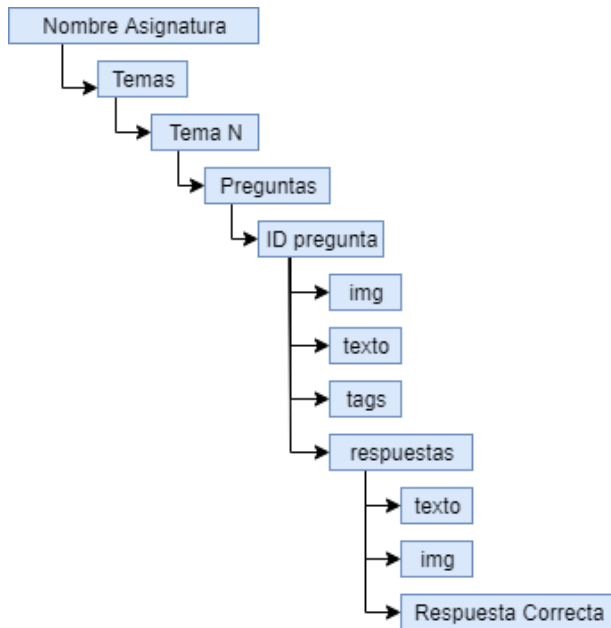


Figura 1 Diseño BBDD

Cada pregunta (y cada respuesta) podrá contener un texto y/o una imagen. Con el fin de dar soporte a la validación automática de las respuestas, cada una de ellas incorporan el atributo de correcta o incorrecta en forma de booleano.

Es importante mencionar que la razón principal del uso de No SQL es que permite adaptar ARPIA a cualquier estructura de examen y no solo aprovecharla para preguntas tipo test de la asignatura objeto de este TFG

2.2 Diseño de interfaz de usuario

Se ha realizado una aplicación web con Angular [11] con la conexión a la base de datos con Firebase [12]. La elección de estas herramientas y no otras (Amazon cloud, Google Cloud) ha venido dada por dos factores principales: 1/ coste económico y 2/ facilidad de integración entre ambas.

El propósito de la interfaz web es poder introducir nuevas preguntas y ver las preguntas ya existentes de una asignatura ya presente en la base de datos. Para ello se han realizado dos vistas diferentes en la interfaz.

La primera vista, que se muestra en la Figura 2, consta del formulario de la nueva pregunta donde podemos incluir el tema de la pregunta, los tags que contiene, la dificultad de esta, el enunciado, la imagen del enunciado y hasta 5 posibles respuestas con texto e imagen y la posibilidad de marcar cuáles son las correctas.

La segunda vista, que se muestra en la Figura 3, consta de un simple listado con las preguntas de la base de datos donde en cada pregunta podemos ver el enunciado, la imagen en caso de tenerla, las respuestas, su imagen en caso de tenerla y si es la respuesta correcta o no.

2.3 Sistema de aprendizaje computacional

La elección del sistema de aprendizaje computacional que

Figura 2. Formulario para introducir preguntas

Figura 3. Visualización de una pregunta

se ha incluido en ARPIA ha venido motivada por dos parámetros: facilidad de adaptación a los datos con los que contamos y facilidad de desarrollo e integración con la aplicación futura.

Dado que estamos trabajando con conjuntos de preguntas de texto, se ha decidido explorar el uso de la técnica Bag of Words [13]. Como es bien conocido, esta técnica se basa en extraer de manera automática 'palabras' de cada uno de los diferentes datos que suponen el conjunto de entrenamiento, obteniendo como resultado un diccionario que representará el conjunto de palabras que aparecen en los datos de entrada.

A continuación se explica paso a paso cómo se ha particularizado la técnica Bag of Words para el problema que nos ocupa.

2.3.1. Creación del diccionario

El primer paso de este proyecto ha sido la extracción de las preguntas de cada tema del fichero JSON donde se han almacenado vía aplicación web. Se ha extraído la pregunta y la clase a la que pertenece, dando un resultado como el que podemos ver en la Figura 4. Se ha optado en convertir el nombre de cada clase en un código numérico para luego

```
['Durante ejecución automática...', 0]
['¿Cuál siguientes opciones entrada Tabla Símbolos LOOS?', 1]
['¿Cuál tamaño bloque activación función h LOOS?', 4]
```

Figura 4. Ejemplo de preguntas extraídas de la BBD

facilitar su tratamiento (los códigos asociados a cada categoría se pueden apreciar en la Tabla 3).

El diccionario se crea a partir de las preguntas que se han categorizado previamente como de entrenamiento. Debido a la aparición de conceptos que se definen con más de una palabra (por ejemplo: autómatas finitos), se ha explorado la creación de dos diccionarios diferentes, uno que consta de palabras individuales (incluyendo el número de aparición de cada una de ellas) y el otro de conjuntos de dos palabras.

Para el diccionario individual de palabras se ha procesado cada pregunta dividiéndola en palabras y se van añadiendo al diccionario; en caso de que ya existan se incrementa el contador de apariciones de cada palabra.

Para los conjuntos, se dividió la pregunta en palabras y se hacen conjuntos de dos en dos, donde cada pareja siempre tendrá el último elemento de la anterior (Figura 5). Cada conjunto se añade al diccionario si no existía antes, siguiendo el mismo criterio mencionado anteriormente para contabilizar las apariciones de estos en el caso de que dicho conjunto ya existiera en el diccionario.

Es importante mencionar que, antes de dividir y añadir las palabras/conjuntos al diccionario se ha realizado un pequeño “preprocesado” de la pregunta, eliminando palabras no significativas tales como preposiciones y artículos y algunas otras que se han detectado que aparecen en todos los temas generando redundancias innecesarias, como puede ser la palabra “LOOS” (Language Orientado a Objetos Simple, nomenclatura específica de Compiladores).

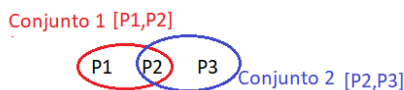


Figura 5 Ejemplo de la creación de conjuntos

2.3.2. Sistema de clasificación

2.3.2.1. Clasificación basada en histogramas

En un primer momento, se ha realizado el sistema de clasificación basándonos en la comparación de histogramas. Para ello se ha definido primero un histograma medio para cada clase a partir del diccionario y la categoría de cada pregunta. Para ello se contabiliza el número de apariciones de las palabras del diccionario según cada tema y se divide por el total de apariciones de todas las palabras asociadas a las preguntas del tema.

Una vez tienes estos histogramas se genera el histograma de la pregunta que se quiere clasificar a partir de todas las palabras del diccionario. Para determinar a qué clase pertenece se calcula la distancia entre histogramas utilizando la distancia euclidiana y la distancia de Manhattan.

- Distancia euclidiana:

$$d_E(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- Distancia de Manhattan:

Se escoge como categoría aquella con menor distancia entre histograma medio e histograma de pregunta.

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^n |p_i - q_i|,$$

2.3.2.2. Clasificación basada en palabras Top

Mediante el diccionario general podemos obtener las palabras con más apariciones contabilizadas a lo largo de las preguntas de entrenamiento. En este segundo sistema de validación asignamos a cada una de estas palabras una clase, que será aquella donde aparece de modo más predominante.

Se ha utilizado un método de votación para determinar la clase a la que pertenece la nueva pregunta. Por cada pregunta de prueba comprobamos las palabras del ranking que aparecen en ella. Para cada una de estas palabras top se contabiliza la clase a la que pertenece, es decir, se suma un voto a la probabilidad de pertenecer a esa clase.

Por ejemplo, teniendo estas palabras top:

- Autómata -> clase 1
- Finito -> clase 1
- Código -> clase 2

Y quisiéramos clasificar la pregunta “Un autómata finito de este código es...”, tendríamos 2 votos para la clase 1 y 1 para la clase 2, por lo que se acabaría clasificando como clase 1.

3 DISEÑO EXPERIMENTAL

3.1. Base de datos.

Para poder validar el sistema ARPIA, se ha contado con un conjunto de exámenes tipo test de la asignatura Compiladores que han sido cedidos por los profesores de la asignatura.

La base de datos de exámenes está dividida en tres tipos: parcial 1, parcial 2 y examen de recuperación. La asignatura en sí está dividida en 5 grandes bloques temáticos:

- Análisis Lexicográfico
- Análisis Sintáctico
- Análisis Semántico
- Organización de Memoria
- Generación de Código

La Tabla 3 muestra el contenido de la base de datos final de preguntas, desglosado por los diferentes bloques temáticos. Como se puede observar en la tabla, todas las categorías están representadas de una manera similar, aunque se observa un cierto desequilibrio con la categoría Análisis Semántico. Dicho desequilibrio viene dado debido a que

en el primer parcial se incluyen los tres primeros temas, siendo el grueso de este Lexicográfico y Análisis Sintáctico.

Código: Tema	Código	N Preguntas
Análisis Lexicográfico	0	50 (20.66%)
Análisis Sintáctico	2	58 (23.96%)
Análisis Semántico	1	38 (15.70%)
Organización de Memoria	4	51 (21.07%)
Generación de Código	3	45 (18.59%)
Total	-	242

Tabla 3. Desglose contenido de la base de datos

3.2. Métricas

Con el fin de poder validar el sistema de aprendizaje computacional, se usarán métricas comunes al problema de clasificación multi-clase. La estructura principal que usaremos para medir la eficacia del sistema será la matriz de confusión por clase que se muestra en la Figura 6.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figura 6. Ejemplo matriz de confusión

A partir de esta matriz de confusión por clase se derivan las métricas que usaremos en los experimentos: accuracy por cada una de las 5 clases y accuracy medio de todas las clases. Adicionalmente, para el experimento de clasificación a partir de las palabras Top, se usan métricas de agregación adicionales como Precision, Recall y F1-score.

3.3. Diseño del experimento.

Con el fin de mitigar el impacto que pueda introducir una distribución prefijada del conjunto de datos entre entrenamiento y test, se han realizado 20 repeticiones de cada experimento. Debido a ello, para cada experimento se mostrarán los valores de las métricas mencionadas previamente, indicando su valor medio y su desviación estándar.

Asimismo cabe resaltar que se ha asegurado que la distribución de preguntas de cada tema en los conjuntos de train y test siga la misma distribución porcentual que en el global de la base de datos del estudio y así evitar que el sistema pudiera aprender incorrectamente.

4 RESULTADOS

4.1. Clasificación por comparación de histogramas

4.1.1. Histogramas de palabras

La Tabla 4 muestra los resultados obtenidos al comparar el histograma medio de cada clase con el obtenido al evaluar la pregunta nueva. Durante la realización de los experimentos se observó que, debido a la longitud del diccionario, el uso de histogramas completos generaba distancias demasiado similares debido a la contribución de palabras con escasa aparición y la disparidad en el número de palabras con que contribuye cada tema al diccionario. Para solucionar esto se ha diseñado el histograma recortado que sólo considera las palabras que contiene la pregunta que se quiere clasificar.

Extraemos las siguientes conclusiones de los resultados. El uso de histogramas recortados conlleva una mejora general en el accuracy de cada clase al usar la distancia Euclidiana (por ejemplo, para Análisis Sintáctico mejora de un 6% a un 38%, mejorando el accuracy global del 6 al 20%). El uso de la distancia de Manhattan lleva también asociado una mejora general de los resultados aunque dicha mejora se aprecia mejor al usar el histograma completo.

4.1.2. Histogramas de conjuntos

Respecto al uso de conjuntos de palabras para crear el diccionario, se observa en la Tabla 5 que no hay diferencia en la comparación de histogramas completos pero sí al usar histogramas recortados. En este último caso se observa una gran mejora asociada al uso de la distancia de Manhattan.

Clases	Histogramas Completos		Histogramas Recortados	
	D. Euclidiana Accuracy	D. Manhattan Accuracy	D. Euclidiana Accuracy	D. Manhattan Accuracy
A. Lex.	0,23	0,55	0,44	0,50
A. Sem.	0,15	0,60	0,30	0,40
A. Sint.	0,06	0,24	0,38	0,46
Gen. Código	0,25	0,57	0,45	0,44
Org. Memoria	0,21	0,52	0,38	0,41
MEDIO	0,18	0,49	0,39	0,44

Tabla 4: Clasificación mediante comparación histogramas de palabras

Aunque estos resultados son prometedores, sí que se observa que hay determinadas clases para las cuales los resultados no son óptimos (especialmente Análisis Lexicográfico). También se observa que el uso de distancias diferentes no afecta de la misma manera a los dos tipos de histogramas: la distancia de Manhattan parece más adecuada para el uso de histogramas completos.

Clases	Histogramas Completos		Histogramas Recortados	
	D. Euclidiana Accuracy	D. Manhattan Accuracy	D. Euclidiana Accuracy	D. Manhattan Accuracy
A. Lex.	0,20	0,20	0,24	0,36
A. Sem.	0,55	0,55	0,45	0,58
A. Sint.	0,45	0,45	0,42	0,69
Gen. Código	0,52	0,52	0,58	0,70
Org. Memoria	0,47	0,47	0,58	0,68
MEDIO	0,44	0,44	0,45	0,60

Tabla 5: Clasificación mediante comparación histogramas de conjuntos

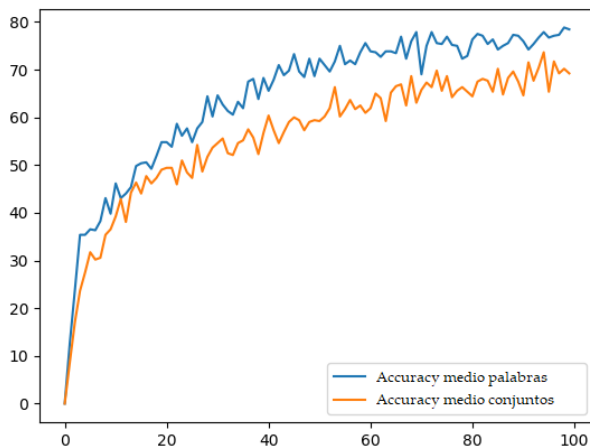


Figura 7. Búsqueda de la mejor k

4.2. Clasificación basada en palabras top

4.2.1. Selección del umbral 'k' para seleccionar el subconjunto de palabras/conjuntos top a estudiar

Para la clasificación de nuevas preguntas según palabras/conjuntos top, se ha realizado un experimento previo para el cual a la hora de clasificar no se han tenido en

cuenta todas las palabras/conjuntos que aparecen en el diccionario sino un subconjunto de ellas (siendo k el número de palabras top a considerar, estando ordenadas por el número total de apariciones de la palabra contabilizadas al crear el diccionario).

Para determinar el mejor número de palabras top se ha probado entre 0 y 100 como podemos ver en la Figura 7, comprobando el resultado del accuracy medio en cada caso. El valor óptimo de K obtenido ha sido de 98 para palabras top palabras y 94 para conjuntos top, número que hemos utilizado para hacer los demás experimentos y sacar resultados.

4.2.2. Ranking basado en aparición de palabras

La Tabla 6 muestra los resultados de clasificación mediante palabras top. Este sistema de clasificación muestra una clara mejora de rendimiento respecto al anterior, obteniendo accuracys de cada clase superiores a 0,8 (accuracy medio de 0,88). Sí que es cierto que hay categorías donde el sistema da un mayor número de errores de clasificación (como Organización de Memoria). Para determinar mejor las clases con mayor interdependencia de palabras top, mostramos la matriz de confusión en la Figura 8.

Clases	Accuracy	Precision	Recall	F1-score
A. Lex.	0,90	0,800	0,800	0,800
A. Sem.	0,90	1,00	0,50	0,67
A. Sint.	0,88	0,71	1,00	0,83
Gen. Código	0,92	0,80	0,89	0,84
Org. Memoria	0,82	0,67	0,55	0,60
MEDIO	0,88	-	-	-

Tabla 6: Clasificación mediante palabras top

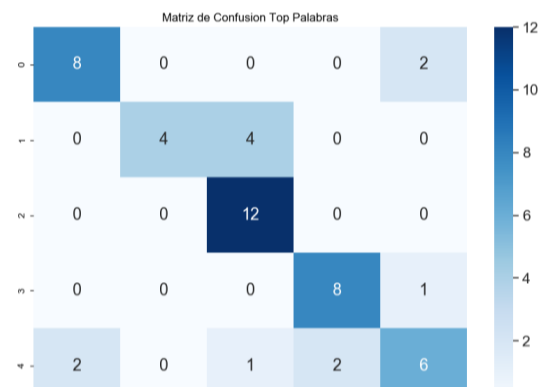


Figura 8. Matriz de confusión asociada al experimento de clasificación basado en palabras top

	A.Lex	A.Sem	A.Sin	G.Cod	O.Mem
símbolos	35.29	26.47	38.24	0.00	0.00
código	0.00	0.00	3.70	85.19	11.11
autómata	100.00	0.00	0.00	0.00	0.00
siguiente	0.00	13.33	56.67	30.00	0.00
gramática	0.00	0.00	100.00	0.00	0.00
compilador	0.00	8.70	0.00	60.87	30.43
scanner	100.00	0.00	0.00	0.00	0.00
expresiones	45.00	0.00	30.00	25.00	0.00
generación	0.00	0.00	0.00	92.86	7.14
terminales	0.00	0.00	100.00	0.00	0.00
función	0.00	0.00	0.00	33.33	66.67

Tabla 7: Ejemplo de las 11 palabras con mayor aparición en el diccionario de entrenamiento

Clases	Accuracy (%)	Precision	Recall	F1-score
A. Lexicográfico	0,88	0,50	1,00	0,67
A. Semántico	0,75	1,00	0,50	0,67
A. Sintáctico	0,88	0,75	0,50	0,60
Generación Código	0,79	0,71	0,56	0,63
O. Memoria	0,83	0,86	0,55	0,67
MEDIO	0,83	-	-	-

Tabla 8: Clasificación mediante conjuntos top

Mediante la observación de la matriz de confusión vemos que hay una clase que no presenta ninguna confusión con otra (análisis sintáctico) y que hay clases con mayor interdependencia como, por ejemplo, Organización de Memoria y Generación de Código. Para poder profundizar más en las razones de dichas confusiones, mostramos en la Tabla 7 el ranking de las 11 palabras top.

Mediante el análisis de este ranking podemos ver que hay palabras que sólo aparecen en una categoría (como autómata o scanner para Análisis Lexicográfico) pero hay otras que aparecen con cierta prevalencia en más de una categoría, como la palabra código que aparece tanto en Organización de Memoria como en Generación de Código. Son estas últimas palabras la causa de no tener una matriz de confusión con una diagonal completa.

4.2.3. Ranking basado en apariciones de conjuntos

La Tabla 8 muestra los resultados de clasificación mediante conjuntos top. Podemos observar una bajada global en el accuracy (tanto por clase como el medio) aunque también podemos observar cómo determinadas clases (como Organización de Memoria) ven incrementados sus resultados especialmente en cuanto a la Precisión.

Tal y como se hizo antes, mostramos en la Figura 9 la matriz de confusión asociada a la clasificación de preguntas mediante conjuntos top. Mediante el análisis de la matriz de confusión vemos de nuevo que hay clases donde el sistema no proporciona ninguna clasificación errónea (en este caso, Análisis Semántico) pero que, en general, hay mayor dependencia entre las clases.



Figura 9: Matriz de confusión conjuntos top

Mostramos en la Tabla 9 el ranking de los 11 conjuntos de palabras más frecuentes. Podemos observar nuevamente como hay conjuntos de palabras exclusivos de una sola clase (Automáta finito o Automáta determinista para el Análisis Lexicográfico) pero que hay otros conjuntos que

		A.Lex	A.Sem	A.Sin	G.Cod	O.Mem
autómata	finito	100.00	0.00	0.00	0.00	0.00
símbolos	terminales	23.33	33.33	43.33	0.00	0.00
generación	código	0.00	0.00	4.35	86.96	8.70
autómata	determinista	100.00	0.00	0.00	0.00	0.00
Durante	ejecución	28.57	0.00	0.00	0.00	71.43
objeto	clase	0.00	10.00	0.00	0.00	90.00
identificador	variable	32.50	0.00	0.00	46.75	20.75
tacionesiones	autómata	100.00	0.00	0.00	0.00	0.00
gramática	siguiente	0.00	0.00	69.79	15.38	14.82
expresiones	regulares	42.86	0.00	35.23	21.43	0.00
terminales	terminales	0.00	0.00	100.00	0.00	0.00

Tabla 9: Ejemplo de los 9 conjuntos con mayor aparición en el diccionario de entrenamiento

aparecen en varias categorías, como símbolos terminales que aparece con cierta prevalencia en tres clases (Análisis Lexicográfico, Sintáctico y Semántico).

Tras analizar los resultados de los diferentes experimentos, hemos podido comprobar que la clasificación de preguntas mediante el uso de palabras top proporciona resultados claramente superiores que la comparación clásica de histogramas.

5 CONCLUSIONES

5.1 Conclusiones generales del proyecto

El objetivo principal de este TFG era estudiar el potencial de las técnicas de aprendizaje computacional para incluirlas como parte de herramientas de soporte al aprendizaje universitario.

Durante el TFG se ha desarrollado una primera versión del sistema, denominada ARPIA, que es capaz de asociar correctamente la categoría a la que pertenece una pregunta nueva. El sistema de aprendizaje se basa en la metodología Bag of Words; para la parte de clasificación se han probado dos estrategias: comparación de histogramas y clasificación mediante la extracción de palabras top.

Los resultados preliminares muestran un claro rendimiento superior del sistema de clasificación basado en palabras top, obteniendo un accuracy medio de 0,88.

Aparte del desarrollo del sistema de aprendizaje computacional, se ha desarrollado una plataforma web con acceso a la base de datos no relacional usada en el proyecto y que permitirá a los usuarios futuros gestionar de manera ágil la introducción y revisión de preguntas.

5.2 Conclusiones personales extraídas del proyecto

El desarrollo de este TFG me ha permitido afrontar un proyecto completo en el que he podido integrar conocimientos adquiridos durante diferentes asignaturas de la carrera y combinarlos con mi interés por las metodologías innovadoras de docencia. Por ejemplo, he utilizado los conocimientos adquiridos en la asignatura de Bases de Datos y Technologies de Desenvolupament per a Internet i Web para el desarrollo de la aplicación web que da soporte a la gestión de preguntas y asignaturas. He podido poner en práctica los conocimientos adquiridos en asignaturas de la mención de Computación (Aprenentatge Computacional y Coneixement Raonament e Incertesa) para el desarrollo del módulo de aprendizaje computacional.

Personalmente estoy contenta de haber podido llevar a cabo todos los objetivos que se plantearon al comenzar el proyecto y haber podido desarrollar una primera versión completa de ARPIA. Pese a que el desarrollo de la parte computacional del código ha sido suave, es importante recalcar que una de las partes más costosas del proyecto ha sido la recolecta de los datos, ya que hay contábamos con diferentes exámenes (no muy abundantes) en diversos formatos. La recolecta de los datos en un formato común ha

sido clave para poder avanzar con el proyecto.

5.3 Trabajo futuro

Pese a que ya se cuenta con una versión funcional de ARPIA, éste proyecto es solo el inicio de la creación de una plataforma educativa donde el aprendizaje se hará mediante pequeños juegos y donde el sistema aprenderá de los usuarios para ofrecer una experiencia personalizada.

Dentro de las siguientes líneas de actuación, la primera que se debería afrontar es la recolección/creación de preguntas para ampliar la base de datos con la que se crea el diccionario. Respecto a éste, se podrían ampliar las anotaciones aprovechando los tags que se han incluido al recolectar los datos iniciales.

Con respecto al sistema de aprendizaje computacional y, una vez teniendo datos suficientes, se podrían explorar otros sistemas de clasificación como árboles de decisión o, llegado el caso, redes neuronales. Otra funcionalidad pendiente es permitir una actualización del diccionario al procesar nuevas preguntas implementando metodologías de active learning, es decir, una vez clasificada la pregunta se contabilizan o añaden al diccionario las palabras de esa pregunta, para ampliar el diccionario y mejorar el rendimiento del sistema a la hora de determinar de qué clase es cada palabra. Asimismo se podrían usar otros sistemas de clasificación más centrados en el procesamiento de lenguaje natural [14].

Finalmente, se ha de avanzar en el desarrollo de la parte lúdica del proyecto de cara a integrar ARPIA dentro de un videojuego de soporte del proceso de aprendizaje.

AGRADECIMIENTOS

Me gustaría expresar mi más sincero agradecimiento a Jorge, por ser el mejor tutor que podría haber tenido. Por aconsejarme y animarme en todo momento, ayudarme cuando lo necesitaba y ponerme presión cuando iba justa. Por estar siempre accesible para responder dudas y mirar un número infinito de versiones de los informes dándome un feedback claro y honesto.

Quiero agradecer a todos mis amigos y compañeros de la universidad que me han apoyado y dado ánimos en momentos difíciles, es muy difícil mencionarlos a todos, pero me gustaría destacar a Marta y Yael por venir a mi casa y obligarme a escribir los informes, porque saben que es la parte más tediosa. A JT, por ofrecerme su cueva como refugio donde poder trabajar tranquila y por ofrecerme consejos y comida basura.

Y sobre todo agradecer a Arnau, por estar ahí para mi tal y como es, por ayudarme incluso antes de empezar el TFG a elegir tutor y tema. Agradecerle que me escuchara y me diera su opinión en todo momento, por estar ahí para mí para consultarle mis dudas sobre cómo llevar el desarrollo del proyecto, por leerse mis informes, por tranquilizarme

en momentos de ansiedad y sobre todo agradecerle el ser un apoyo incondicional para mí.

He tenido la suerte de contar con el apoyo de mucha gente, no podría haber acabado este proyecto sin ellos. Gracias.

BIBLIOGRAFIA

[1] Kahoot! | Learning Games | Make Learning Awesome! (s.f.). Recuperado 6 enero, 2020, de <https://kahoot.com>

[2] Ares, A. M., Bernal, J., Nozal, M. J., Sánchez, F. J., & Bernal, J. (2018, May). Results of the use of Kahoot! gamification tool in a course of Chemistry. In Fourth International Conference on Higher Education Advanc-es.

[3] Sánchez, F. J., & Bernal, J. (2018, July). Use of Software Tools for Real-time Monitoring of Learning Processes: Application to Compilers subject. In 4th International Conference on Higher Education Advances (HEAD'18) (pp. 1359-1366). Editorial Universitat Politècnica de València.

[4] Socrative. (2019, 13 noviembre). Recuperado 6 enero, 2020, de <https://socrative.com>

[5] PlayBrighter | Dave Hall | Freelance web developer | Cambridge. (s.f.). Recuperado 6 enero, 2020, de <https://etianen.com/projects/playbrighter/>

[6] Manifiesto por el Desarrollo Ágil de Software. (s.f.). Recuperado 2 febero, 2020, de <http://agilemanifesto.org/iso/es/manifesto.html>

[7] Metodologías ágiles. Las 3 más usadas actualmente. (2019, 6 marzo). Recuperado 8 enero, 2020, de <https://blog.conectart.com/metodologias-agiles/>

[8] NoSQL Relational Database Management System: Home Page. (s.f.). Recuperado 9 de enero, 2020, de http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/nosql/Home%20Page

[9] The Fundamental Guide to SQL. (2019, 22 septiembre). Recuperado 20 enero, 2020, de <https://www.life-wire.com/sql-fundamentals-1019780>

[10] NoSQL vs SQL: principales diferencias y cuándo elegir cada una de ellas. (2019, 15 julio). 20 enero, 2020, de <https://pandorafms.com/blog/es/nosql-vs-sql-diferencias-y-cuando-elegir-cada-una/>

[11] Angular. (s.f.). Recuperado 14 enero, 2020, de <https://angular.io/>

[12] Firebase. (s.f.). Recuperado 14 enero, 2020, de <https://firebase.google.com/>

[13] Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics, 1(1-4), 43-52.

[14] Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of cnn and rnn for natural language processing. arXiv preprint arXiv:1702.01923.