



UNIVERSIDAD DE GRANADA

TRABAJO FIN DE MÁSTER
MÁSTER EN INGENIERÍA INFORMÁTICA

Aprendizaje predictivo en Nutrición

Autor

Andrea Morales Garzón

Directores

Juan Gómez Romero
María José Martín Bautista



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

—
Granada, 10 de julio de 2020

Aprendizaje predictivo en Nutrición

Andrea Morales Garzón

Palabras clave: Word Embedding, Food Computing, Alineamiento de datos, Lógica Difusa, Procesamiento de Lenguaje Natural

Resumen

La integración de datos heterogéneos es una tarea indispensable en múltiples dominios y, en especial, en el área de Food Computing. Los grandes volúmenes de datos que intervienen, el vocabulario hiperespecializado, los factores multiculturales y las asunciones en el lenguaje alimenticio la convierten en una tarea difícil de abordar. En este trabajo se propone un método general basado en modelos predictivos del lenguaje para alinear fuentes de datos a partir de las descripciones textuales de sus elementos. En concreto, se ha desarrollado un sistema que combina un modelo de tipo *Word Embedding* para representar los datos textuales y un procedimiento de mapeo entre ítems basado en medidas de distancia sintácticas y semánticas, tanto clásicas y difusas. Con el fin de reflejar sus capacidades y su versatilidad, el sistema se ha aplicado para resolver un problema de interés actual: la adaptación automatizada de recetas a restricciones alimenticias. Las pruebas empíricas realizadas muestran que esta aproximación es apropiada para resolver el problema, especialmente cuando se combinan modelos semánticos de dominio específico con medidas de distancia difusas.

Predicting Learning in the Nutrition field

Andrea Morales Garzón

Keywords: Word Embedding, Food Computing, Data Alignment, Fuzzy Logic, Natural Language Processing

Abstract

Heterogeneous data integration has become an essential task in the Food Computing area and many others. The large volumes of data involved, the hyper-specialized vocabulary, the multicultural factors and the assumptions in food language turn it into a difficult problem to address. In this work, we propose a general method based on predictive language models to map data sources from the textual description of their elements. Specifically, we have developed a computational system that combines a *Word Embedding* model to represent textual data and a mapping procedure that applies different types of distance metrics (syntactic/semantic, crisp/fuzzy) to match them. In order to illustrate its capabilities and the potential of this system, it has been applied to solve the problem of automated adaptation of recipes to food restrictions. Experiments show that this approach is appropriate for solving the problem, particularly when domain-specific models are combined with fuzzy distance metrics.

Agradecimientos

En primer lugar, quiero expresarle mi gratitud a mis tutores, María José y Juan, por su supervisión y guía a lo largo de este trayecto. Gracias por vuestra experiencia, ejemplo, dedicación, paciencia y saber hacer. Lo que podría haber quedado en un trabajo de máster, ha resultado ser una experiencia mucho más que enriquecedora en mi formación.

Por otra parte, este trabajo tampoco habría sido posible sin el soporte del proyecto europeo Stance4Health. Querría agradecerle a mis superiores en este proyecto el abrirme la vía al, hasta entonces desconocido para mí, mundo del Food Computing y de la Nutrición. Asimismo toda la experiencia que me ha aportado y todas las personas que he conocido. Cada uno de ellos aporta su grano de arena a mi experiencia profesional.

En último lugar, a mi madre, por ser un apoyo constante en mi vida en todas sus facetas, incluido el tránscurso de este trabajo. Eres mi persona favorita.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Concepto de Food Computing	2
1.3. Nuestro problema a resolver	3
1.4. Objetivos	5
1.5. Contenido de la memoria en relación a los objetivos	6
2. Planificación y metodología	7
2.1. Especificación de requisitos	7
2.2. Actividades	9
2.3. Planificación	10
2.4. Estimación de costes	11
3. Antecedentes	13
3.1. Food Computing	13
3.2. Técnicas predictivas en Food Computing	16
3.2.1. Aproximaciones generales	16
3.2.2. Modelos predictivos del lenguaje	18
3.2.3. Identificación de términos alimenticios textuales	19
4. Arquitectura del sistema	21
4.1. Descripción del sistema	21
4.2. Módulos del sistema	22
4.2.1. Módulo de Procesamiento de Lenguaje Natural	22
4.2.2. Módulo de Mapeo	23
4.2.3. Módulo de Consultas Adaptadas	25
4.3. Sistema para adaptación de dietas en Food Computing	26
5. Word Embedding	27
5.1. Introducción a los Word Embeddings	27
5.2. Word embeddings generales vs específicos	29
5.3. Metodología	31
5.3.1. Datos utilizados	31
5.3.2. Preprocesamiento de los datos textuales	34

5.3.3. Implementación de Word2vec	37
5.3.4. Entrenamiento del modelo	39
6. Mapeo de datos	41
6.1. Procedimiento de mapeo	41
6.2. Medidas de distancia implementadas	42
6.2.1. Distancia sintáctica entre descripciones	42
6.2.2. Distancia semántica entre descripciones	44
6.2.3. Distancia híbrida entre descripciones	45
6.2.4. Fuzzificación de las medidas de distancia	45
6.3. Elección de la medida de distancia	50
7. Diseño y desarrollo de la aplicación	51
7.1. Recomendación de recetas	51
7.2. Adaptación de recetas según restricciones	53
7.3. Aplicación móvil	54
7.3.1. Arquitectura	54
7.3.2. Tecnologías utilizadas	55
7.3.3. Fuentes de datos	55
7.3.4. Sistema para Adaptación de Recetas	58
7.3.5. Interfaz de Programación de Aplicaciones (API)	58
7.3.6. Aplicación	59
7.3.7. Siguientes pasos en el desarrollo de la aplicación móvil	62
8. Experimentación y resultados	65
8.1. Diseño experimental	65
8.1.1. Bases de datos utilizadas	66
8.2. Resultados y discusión	68
8.2.1. Resultados del Módulo de PLN	68
8.2.2. Resultados del Módulo de Mapeo	71
8.3. Resultados del Módulo de Consultas Adaptadas	80
9. Conclusiones	85
9.1. Conclusiones	85
9.2. Trabajo futuro	87
A. Manual de usuario	97

Índice de figuras

1.1.	Food Computing	2
1.2.	Datos heterogéneos en Food Computing	4
2.1.	Diagrama de Gantt	11
4.1.	Arquitectura del sistema	21
4.2.	Arquitectura del sistema: módulo de NLP	23
4.3.	Arquitectura del sistema: módulo de Mapping	24
4.4.	Arquitectura del sistema: módulo de Consultas Adaptadas . .	25
4.5.	Arquitectura del sistema aplicada al problema de adaptación de dietas en Food Computing	26
5.1.	Estructura de las recetas en el dataset de archive.org	34
5.2.	Procedimiento para formar el corpus de recetas	35
5.3.	Modelo CBOW donde el contexto es una palabra [62]	38
5.4.	Modelos del algoritmo Word2vec [62]	39
6.1.	Intersección entre dos conjuntos	43
6.2.	Funcionamiento de la distancia Word Mover's [43]	44
6.3.	Intersección entre dos conjuntos	46
6.4.	Función de distancia difusa Jaccard	47
7.1.	Consulta sobre los datos fusionados	53
7.2.	Mapeo de datos de fuentes heterogéneas	54
7.3.	Arquitectura de la aplicación	55
7.4.	Diagrama de casos de uso de la aplicación	60
7.5.	Diagrama HTA de la aplicación	60
7.6.	Diagrama Conceptual de la aplicación	61
7.7.	Diagrama Wireflow de la aplicación	61
7.8.	Pantallas de la aplicación: funcionamiento básico	62
8.1.	Visualización del modelo: elementos similares (ejemplo 1) . .	68
8.2.	Visualización del modelo: localización espacial de los items (ej.1)	69
8.3.	Visualización del modelo (ejemplo 2)	70

8.4. Medida de distancia híbrida: comportamiento del parámetro w	75
8.5. Comparación entre Word Embedding genéricos y preentrenados	80
8.6. Adaptación de recetas a restricciones vegetarianas y veganas	81
8.7. Adaptación de una receta vegana I	82
8.8. Adaptación de una receta vegana II	82
8.9. Adaptación de una receta vegetariana	83
A.1. Pantallas de inicio a la aplicación I	97
A.2. Pantallas de inicio a la aplicación II	98
A.3. Pantallas para adaptar recetas	99
A.4. Pantallas de navegación de recetas	99
A.5. Pantallas de la receta adaptada	100

Índice de Tablas

2.1. Actividades llevadas a cabo en el proyecto	11
2.2. Estimación de costes en p.m. por descomposición de actividades	12
5.1. Similitudes para <i>ajo</i> con modelos de W.E.	31
5.2. Ejemplo de receta: Noodles en 5 minutos	32
5.3. Corpus de recetas: origen y número de recetas	33
7.1. Descripción de los campos en la colección de recetas originales	57
8.1. Algunos ejemplos de la base de datos i-Diet	67
8.2. Algunos ejemplos de la base de datos USDA	67
8.3. Resultados del mapeo (%) con las medidas de distancia . . .	72
8.4. Resultados obtenidos con la distancia de Jaccard	72
8.5. Resultados obtenidos con la distancia de Word's Mover . . .	73
8.6. Resultados obtenidos con la distancia híbrida	74
8.7. Resultados obtenidos con la distancia de Jaccard difusa . . .	76
8.8. Resultados obtenidos con la distancia entre documentos difusos	77
8.9. Comparación entre Jaccard (J) y Jaccard difuso (\tilde{J})	78
8.10. Comparación entre Word Mover's (WMD) y Distancia difusa entre documentos (\tilde{D})	79
8.11. Resultados del mapeo (%) para distintos modelos de W.E. .	80

Capítulo 1

Introducción

En este capítulo se introduce el problema que se va a resolver en este proyecto, así como la motivación e impacto del mismo. Se presenta el concepto de Food Computing, el área de estudio en la que se engloba este trabajo.

1.1. Motivación

La nutrición es esencial para el desarrollo de la vida humana. Nuestros hábitos alimenticios tienen un impacto directo en nuestra salud, y por tanto, en nuestra calidad de vida. En los últimos años, se viene observando una tendencia al alza de enfermedades como la obesidad y la diabetes, a la par de un aumento de enfermedades cardiovasculares, todas ellas con estrecha relación con nuestra alimentación diaria [23]. Esta involución nutricional tiene un origen complejo, en el que la interacción de factores culturales y socioeconómicos juegan un papel fundamental. Incluso en aquellas zonas donde la cultura gastronómica consta de características saludables (como es el caso de la dieta mediterránea), se observa una deriva nutricional hacia modelos menos recomendables y con impacto negativo en la salud.

Estos problemas, unidos a la irrupción de la tecnología en la vida diaria, han sido uno de los puntos de partida del desarrollo de sistemas relacionados con la nutrición y el bienestar. Con las técnicas adecuadas y un correcto tratamiento y análisis, los datos generados por estos sistemas se pueden utilizar para una mayor comprensión de esta área de estudio, especialmente desde un punto de vista dietético centrado en hábitos saludables.

1.2. Concepto de Food Computing

El uso de nuevas tecnologías para la interpretación y comprensión de grandes volúmenes de datos se aplica actualmente en múltiples áreas, entre las que se encuentra la nutrición. En ella, se trabaja con datos relativos a la alimentación, también conocidos como *Food Data*. En este contexto, se introduce el concepto de *Food Computing*, el cual engloba la totalidad de técnicas y modalidades relativas a tareas de computación que utilizan datos alimenticios, de naturaleza heterogénea y procedentes de distintas fuentes de información [52]. Su finalidad es mejorar la calidad de vida de la población, así como entender de manera más profunda el comportamiento humano en lo que a esta área concierne.

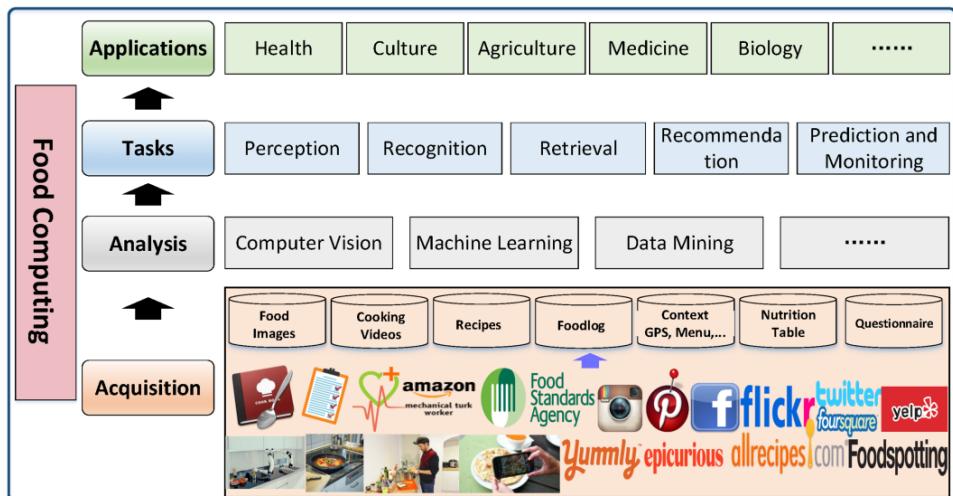


Figura 1.1: Food Computing

En términos generales, las tareas llevadas a cabo dentro del cuadro del Food Computing se caracterizan por partir de un procedimiento de extracción de información, el cual deriva en el desarrollo de herramientas cuyas tareas se pueden englobar en labores de reconocimiento de imágenes, percepción, recomendación, predicción y recuperación de información. De estas tareas destaca especialmente el aprendizaje predictivo, por su papel fundamental en la resolución de problemas en Food Computing [52]. Es tal su relevancia que no solo se corresponde con una de las tareas fundamentales abordadas en este ámbito, sino que también cobra protagonismo en las otras mencionadas, bien ocupándose de funcionalidades muy específicas, bien a nivel global, con el fin de dotar de *inteligencia* a la solución (p.ej., capacidad de decisión, identificación de entidades, detección de comportamientos anómalos o previsión de comportamientos futuros que pueda tener un sistema).

La aplicación de estas tareas no solo afecta a la nutrición sino que también utilizan conocimiento específico de otras áreas para abordar problemas relacionados con la alimentación tales como la medicina, la biología, la gastronomía la agronomía o la cultura. En la Figura 1.1 se muestra un esquema simplificado de todo lo que engloba el concepto de Food Computing [52]. En ella se aprecia cómo las técnicas predictivas tienen una gran presencia en este campo, ya que además de formar una de las tareas principales, también se encuentran de forma intrínseca en las labores de análisis (Visión por Computador, Aprendizaje Automático y Minería de textos entre otras).

Uno de los mayores desafíos a abordar cuando se trabaja con problemas de Food Computing es la dificultad intrínseca que conlleva el tratamiento de la información. Con ello, no solo nos referimos a la gran cantidad de datos que se generan hoy en día dentro de este contexto, sino también a la dificultad añadida de trabajar con su distinta naturaleza y origen. Abordar un problema de Food Computing suele llevar asociado la necesidad de trabajar de forma simultánea con datos de distintos contextos, como pueden ser la tecnología de alimentos, la nutrición y las redes sociales. En una primera instancia esto puede resultar inviable, ya que se tratan de fuentes de información muy heterogéneas entre sí, las cuales no se encuentran agregadas y requieren un tratamiento previo que permita poder trabajar con todos los datos de forma simultánea. En esta línea, se pueden utilizar modelos predictivos de lenguaje para identificar elementos equivalentes que se encuentren en dos o más conjuntos de datos, y así poder integrar distintas fuentes y trabajar con ellas como si de una única colección de datos se tratase.

1.3. Nuestro problema a resolver

Tal y como se ha introducido previamente, en el mundo de la alimentación es difícil trabajar con todos los datos requeridos, puesto que las fuentes de información no suelen proporcionarlos de manera directa. Un ejemplo claro son las páginas web de recetas, fuentes de información protagonistas en este ámbito. Normalmente, trabajar con estas recetas desde un punto de vista nutricional requiere de un uso conjunto de sus ingredientes con una base de datos de composición de alimentos para así poder extraer su información nutricional.

Para trabajar de forma conjunta con ambas fuentes de información se requiere un proceso intermedio que haga de intermediario y facilite el tratamiento de datos de distinta naturaleza y origen. Esta idea se refleja en la Figura 1.2, donde se puede ver cómo la información alimenticia se nutre de múltiples fuentes de información, dando lugar a un único elemento que se ha denominado en dicha figura como *item food*, que representa al objeto fusionado que se aspira a obtener. Esta dificultad no es específica de Food

Computing, sino que se encuentra presente en cualquier área de estudio que combine información de distinto tipo y necesite trabajar con ella de forma conjunta, como puede ser por ejemplo, en el área de medicina al trabajar con datos biomédicos de usuarios y bases de datos de conocimiento o en el área legal, donde fuentes de conocimiento legislativo experto se combinan con sucesos o eventos concretos.



Figura 1.2: Datos heterogéneos en Food Computing

En este trabajo, se desarrolla una herramienta que permite fusionar información heterogénea procedente de diferentes fuentes de datos con el fin de trabajar con ellas simultáneamente. Para ello, se identificarán elementos equivalentes entre las distintas fuentes, para así poder integrarlos y utilizarlos de manera combinada. El problema de identificación de términos equivalentes se puede abordar desde un punto de vista predictivo a partir de varios enfoques. En nuestro caso, lo abordaremos desde la perspectiva del Procesamiento de Lenguaje Natural, utilizando un modelo predictivo de Word Embedding (Word2vec) entrenado con textos de recetas para aprender representaciones de palabras. Con el fin de detectar correspondencias entre los datos, se llevará a cabo un procedimiento de mapeo que utilice dichas representaciones para medir la similitud entre elementos.

Se mostrará su funcionamiento dentro de un sistema que permita su uso para abordar algún problema concreto con los datos ya combinados. En nuestro caso, lo haremos con una aplicación dentro del área de la nutrición, desarrollando un sistema de adaptación de recetas a restricciones alimenti-

cias de usuario, como pueden ser intolerancias a algún alimento, alergias, o incluso restricciones derivadas de dietas como puede ser la vegetariana o la vegana. Este sistema permitirá recomendar, en base a dichas restricciones, posibles variaciones de recetas a través de modificaciones de sus ingredientes. La adaptación de recetas se llevará a cabo gracias al uso previo de la herramienta de fusión de información heterogénea, la cual permitirá combinar recetas de cocina con la información nutricional de sus ingredientes para poder detectar las restricciones alimenticias que deban solventarse.

De las tareas de Food Computing mencionadas anteriormente, con este problema se aborda principalmente un problema de **Predicción** para la creación de un modelo de lenguaje. Se lleva a cabo el entrenamiento de un modelo predictivo de tratamiento del lenguaje basado en un Word Embedding entrenado sobre un conjunto de recetas que permita identificar términos alimenticios. Por otra parte, también se engloba en otras tareas de Food Computing: sin seguir las estructuras tradicionales y estandarizadas de sistemas basados en recomendación, proporciona opciones de recetas personalizadas a unas preferencias de usuario concretas, resolviendo así un problema en el ámbito de la **Recomendación** en Food Computing. Problemas de **Recuperación** e **Identificación** también se encuentran intrínsecos en este trabajo, puesto que se llevan a cabo tanto la obtención del corpus y fuentes de datos utilizadas para resolver este problema como una tarea de identificación de elementos alimenticios en distintas bases de datos.

1.4. Objetivos

Para poder abordar el diseño e implementación de un sistema que permite poner solución al problema descrito en el apartado anterior, se ha marcado como objetivo principal el estudio, diseño e implementación de técnicas predictivas para resolver un problema de Procesamiento de Lenguaje Natural. De este objetivo general se derivan los siguientes objetivos específicos:

1. *Identificar los elementos a los que representan los datos que intervienen en el problema:* asignar una representación interna a los datos a través de sus descripciones textuales, con la que poder detectar equivalencias para así realizar la agregación entre las fuentes de datos.
2. *Fusionar información heterogénea:* desarrollar una herramienta que permita agregar datos de distinta naturaleza y procedencia que necesiten ser tratados de manera conjunta a la hora de abordar algún problema concreto.
3. *Mostrar la eficacia y el alcance de la herramienta desarrollada:* argumentar la utilidad de la herramienta con un problema real y de interés

actual: la sustitución de elementos por otros similares mediante el modelo predictivo y las representaciones textuales que genera.

1.5. Contenido de la memoria en relación a los objetivos

La memoria se ha redactado en función de los tres grandes pilares en los que se basa este trabajo, los cuales hacen referencia a los tres objetivos específicos detallados en este primer capítulo. En el Capítulo 2 se especifican los requisitos y la planificación por actividades llevada a cabo, las cuales están directamente ligadas a los objetivos. En el Capítulo 3 se exponen los antecedentes que preceden a este trabajo; se profundiza en el concepto de *Food Computing* así como en la revisión bibliográfica de las técnicas de predicción en los problemas abordados en este campo. En el Capítulo 4 se describe de forma general la arquitectura del sistema diseñado para abordar el problema. Se especifican los tres pilares en los que se basa el proyecto como módulos que permiten construir la solución. Estos módulos se relatan de forma detallada en los Capítulos 5, 6 y 7, abarcando el diseño y la implementación llevada a cabo para cada uno de ellos. En el Capítulo 8 se detalla la experimentación y resultados obtenidos en cada uno de los módulos. Por último, en el Capítulo 9, se exponen todas las conclusiones obtenidas a lo largo del trabajo, así como las posibles líneas que se podrían seguir explorando a partir de este trabajo.

Capítulo 2

Planificación y metodología

Este capítulo presenta los requisitos que debe satisfacer el sistema que se desarrolla en este trabajo, así como su posterior desglose en las actividades a realizar. Por último, se presenta la planificación realizada en base a dichas actividades.

2.1. Especificación de requisitos

A continuación se enumeran las necesidades que debe satisfacer el sistema desarrollado de acuerdo a los objetivos detallados en la Sección 1.4.

Calidad

1. El modelo de lenguaje debe contener un vocabulario lo suficientemente extenso como para poder trabajar con el lenguaje en el área de estudio escogida.
2. El modelo de lenguaje tiene que permitir identificar elementos equivalentes entre las fuentes de datos que intervienen en el problema.
3. Los mapeos obtenidos deben ser lo suficientemente fiables como para detectar los ingredientes en recetas y poder acceder a su información nutricional.
4. Las recetas adaptadas deben satisfacer las restricciones que se especifiquen.
5. La adaptación de recetas debe poder ser interpretable. Para ello, se debe mostrar qué ingredientes no cumplen las restricciones y por tanto deben ser modificados.

6. Las imágenes de recetas utilizadas en el sistema deben ser lo más representativas posibles.
7. La aplicación final debe poseer interfaces gráficas bien formadas que permitan una navegación intuitiva por parte del usuario.

Escalabilidad

1. El sistema debe ser capaz de controlar grandes volúmenes de datos.
2. El modelo de lenguaje debe poder utilizarse de manera independiente al resto de elementos del sistema. Debe permitir el uso del conocimiento aprendido a otros problemas de Food Computing, así como su reutilización en otros modelos predictivos.
3. La implementación del modelo de lenguaje tiene que posibilitar la opción de realizar, en caso que se requiera, nuevas tareas de entrenamiento con otros corpus distintos del utilizado en este trabajo.
4. El mapeo entre fuentes de datos debe ser independiente de la cantidad de elementos contenidos en cada una de ellas.
5. La adaptación de recetas según restricciones debe ser extensible a nuevas recetas futuras que se añadan al sistema.
6. La implementación de la adaptación de recetas debe permitir la incorporación de nuevas restricciones en el futuro.

Facilidad de mantenimiento

1. El sistema se organizará en distintos módulos, donde cada uno implementará una funcionalidad diferente con el objetivo de que el proceso de modificaciones, pruebas y validación sea más sencillo.

Facilidad de uso

1. Los usuarios no tienen por qué tener experiencia o conocimientos específicos sobre informática o sobre el proyecto: cualquier usuario debe poder hacer uso de la aplicación final sin dificultad.
2. La aplicación final debe contar con un pequeño tutorial de la aplicación, para orientar al usuario acerca de cómo funciona el sistema.
3. Se debe proporcionar un manual de usuario de la aplicación final que permita entender el flujo de datos entre las distintas pantallas implementadas.

4. La aplicación final debe proporcionar mensajes intuitivos para el usuario que le permitan utilizar la aplicación sin dificultad.

Rendimiento

1. El flujo de datos entre los distintos módulos debe ser el mínimo necesario para optimizar el tiempo de ejecución del sistema.
2. En la aplicación final no habrá excesiva redundancia en cuanto a la información con el objetivo de acelerar las consultas.
3. En la aplicación final solo tendremos acceso a la información de las recetas en cuestión cuando sean seleccionadas.

Robustez

1. El modelo de lenguaje debe proporcionar representaciones para cualquier descripción, controlando las palabras que se queden fuera del vocabulario del modelo.
2. Los mapeos debe contemplar la posibilidad de tratar con descripciones con palabras no contempladas en la representación del modelo de lenguaje.

2.2. Actividades

Para poder abordar los requisitos detallados en la sección anterior, se ha elaborado la siguiente distribución del trabajo en tres actividades principales, cada una de ellas relativas a los tres grandes bloques que se abarcan en este proyecto. Para el desarrollo de los distintos bloques, se han detallado también las actividades específicas que se llevarán a cabo en cada uno de ellos.

1. Diseño y desarrollo de un módulo de Representación del Lenguaje con técnicas predictivas para trabajar con problemas de Procesamiento de Lenguaje Natural en un dominio concreto.
 - a) Recopilación masiva de datos y creación del conjunto de entrenamiento del modelo con la colección de datos recopilada.
 - b) Elaborar técnicas de preprocesamiento de textos para poder trabajar las descripciones textuales de los elementos que se pretenden fusionar.

- c) Diseño e implementación de un modelo de Word Embedding y ajuste de hiperparámetros.
 - d) Visualización del modelo entrenado con ejemplos concretos para verificar su funcionalidad.
2. Diseño y desarrollo de un módulo de mapeo de datos que permita fusionar información heterogénea a partir de las descripciones de los elementos a fusionar.
 - a) Diseño de medidas de distancia entre descripciones textuales utilizando su representación textual (previamente preprocesada).
 - b) Diseño de medidas de distancia entre descripciones textuales utilizando su representación vectorial obtenida con el modelo de lenguaje.
 - c) Diseño e implementación de un módulo de mapeo con las medidas de distancia diseñadas.
 - d) Experimentación y análisis de la eficacia de las medidas de distancia con un problema de mapeo entre dos bases de datos.
 3. Experimentación del sistema mediante el desarrollo de una aplicación de adaptación de dietas de usuarios en función de sus restricciones alimenticias.
 - a) Implementar un sistema para aplicar la herramienta de fusión de datos desarrollada.
 - b) Desarrollar un módulo de consultas adaptadas que permita obtener recetas adecuadas a restricciones.
 - c) Detectar ingredientes de recetas no sujetos a las restricciones indicadas y substituirlos por alimentos que sí las satisfagan.
 - d) Implementación de una aplicación móvil para realizar consultas adaptadas sobre las recetas.

2.3. Planificación

La planificación se realizó en función de las actividades definidas en la Sección 2.2. En la Tabla 2.1 se puede ver la duración asociada a cada una de estas actividades. Tal y como se aprecia en dicha tabla, la planificación se ha estimado de forma semanal. Nótese que se ha establecido un identificador para cada una de las actividades definidas en este proyecto. De esta forma, podemos establecer las dependencias entre las distintas actividades (ver columna *Predecesores* en Tabla 2.1) para poder generar el diagrama de Gantt (ver Figura 2.1).

ID	Descripción	Duración (Sem.)	Predecesores
A	Actividad 1a	2	-
B	Actividad 1b	3	A
C	Actividad 1c	8	B
D	Actividad 1d	1	C
E	Actividad 2a	4	B
F	Actividad 2b	4	C
G	Actividad 2c	2	E,F
H	Actividad 2d	7	G
I	Actividad 3a	2	G
J	Actividad 3b	2	I
K	Actividad 3c	2	J
L	Actividad 3d	18	K
M	Redacción de la memoria	27	-

Tabla 2.1: Actividades llevadas a cabo en el proyecto

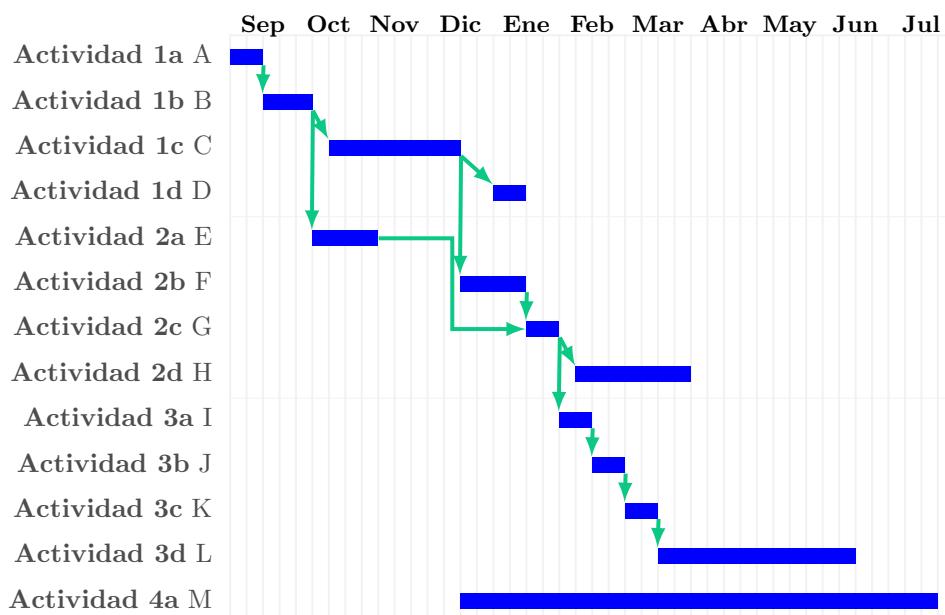


Figura 2.1: Diagrama de Gantt

2.4. Estimación de costes

Para poder estimar los costes de este proyecto debemos distinguir entre los costes de personal y de recursos computacionales utilizados. Para el cálculo de los costes derivados del personal que colabora en este proyecto, se

ha llevado a cabo una estimación por descomposición en actividades medida en *p.m.*¹. En la Tabla 2.2 se muestra una estimación del esfuerzo que sería necesario para abordar las distintas actividades en las que se divide este proyecto.

Actividad	Plan	Análisis	Diseño	Desarrollo	Test	Total
Actividad 1	0.4	0.4	1.4	1.1	0.7	4.0
Actividad 1a	0.1	0.0	0.1	0.4	0.0	0.6
Actividad 1b	0.1	0.1	0.5	0.1	0.2	1.0
Actividad 1c	0.1	0.2	0.7	0.4	0.2	1.6
Actividad 1d	0.1	0.1	0.1	0.2	0.3	0.8
Actividad 2	0.4	1.2	0.7	0.35	0.9	3.55
Actividad 2a	0.1	0.4	0.2	0.0	0.2	0.9
Actividad 2b	0.1	0.5	0.2	0.0	0.2	1.0
Actividad 2c	0.1	0.2	0.2	0.25	0.2	0.95
Actividad 2d	0.1	0.1	0.1	0.1	0.3	0.7
Actividad 3	0.3	0.35	1.0	0.95	0.75	3.35
Actividad 3a	0.1	0.1	0.1	0.1	0.1	0.5
Actividad 3b	0.1	0.1	0.2	0.15	0.2	0.75
Actividad 3c	0.0	0.05	0.1	0.1	0.15	0.4
Actividad 3d	0.1	0.1	0.6	0.6	0.3	1.7
Total	1.1	1.95	3.1	2.4	2.35	10.9

Tabla 2.2: Estimación de costes en p.m. por descomposición de actividades

A estos costes de personal habría que añadirle los asociados a los recursos computacionales necesarios. En este punto hay que considerar la capacidad de procesamiento y de memoria del ordenador para poder entrenar y alojar un modelo predictivo además de una aplicación multiplataforma. En nuestro caso, hemos utilizado un ordenador portátil de gama alta, y el coste asociado a su uso se estima en 700€.

- **Costes por recursos computacionales:** 700€
- **Costes laborales:** 1600€/p.m.
- **Estimación:** 700€ + 1600€/p.m. * 10.9 p.m. = 18140€

Mediante la estimación por descomposición de actividades el coste del proyecto se estima en 18140€.

¹p.m. (persona-mes): si un proyecto toma *x* p.m. significa que si se pudieran contratar *x* personas, el proyecto se terminaría en 1 mes o bien, significa que si sólo contratamos a una persona entonces el proyecto se terminaría en *x* meses,

Capítulo 3

Antecedentes

Este capítulo presenta los antecedentes que preceden a este trabajo, así como las líneas de investigación y los trabajos previos centrados en tareas predictivas en el campo de Food Computing.

3.1. Food Computing

Como ya se ha introducido en el Capítulo 1, el desarrollo de las nuevas tecnologías y el incremento del interés de la población acerca de la alimentación saludable ha contribuido al aumento en la cantidad de datos generados en este campo. Este aumento ha dado pie a su tratamiento con algoritmos que puedan procesar grandes cantidades de información con el fin resolver problemas de interés para la población. Aquí tienen especial protagonismo aquellos que son resultado de la interacción entre usuarios en redes sociales o comunidades de usuarios con interés específico en la cocina para predecir valores de sobrepeso y diabetes en la población [31, 2]. Por otro lado, las páginas web y comunidades de usuarios cuyo objetivo específico es compartir recetas (como pueden ser AllRecipes¹ o Yummly²), son hoy en día el origen de la mayor parte de colecciones de datos alimenticios en Food Computing [52]. En estas páginas, encontramos multitud de recetas con contenido tanto textual, multimedia, como en muchos casos, nutricional. Debido al gran impacto que tienen, múltiples estudios parten de estos datos para la implementación de herramientas o sistemas en Food Computing [40, 34].

Sin embargo, si nos fijamos en estas últimas fuentes de datos mencionadas, cada vez es mayor la tendencia hacia recetas con una fuerte componente no saludables. Hoy en día, la concienciación de la población ha dado lugar al desarrollo de movimientos de vida sana basados en la alimentación, los cu-

¹www.allrecipes.com

²www.yummly.com

les llevan a las personas a interesarse cada vez más por estas recetas. Aquí reside el interés en el desarrollo de herramientas informáticas que puedan adaptar recetas a necesidades de usuarios, ya sea facilitando dietas completas (o bien recetas) adaptadas al usuario, así como con la creación de las llamadas pseudo-recetas, las cuales se centran en generar recetas en base a unas especificaciones dadas, ya sea de forma completa (generación de recetas completas utilizando por ejemplo redes de sabores y grafos de ingredientes con recetas) o parcial (modificando ingredientes concretos dentro de recetas) [17]. En este trabajo se profundiza en esta segunda línea.

El otro gran foco de información reside en las bases de datos de composición de alimentos, las cuales permiten acceder a la información nutricional de ingredientes, recetas, o incluso platos preparados [47], siendo la base de los estudios alimenticios desde el punto de vista de ciencias como la química y la tecnología de alimentos. En este punto destacan bases de datos como la proporcionada por USDA (United States Department of Agriculture) [32], la cual es una base de datos de referencia a nivel mundial en cuanto a composición nutricional se refiere [60]. También tienen relevancia las bases de datos de composición nutricional de países europeos disponibles en EuroFIR (European Food Information Resource Network) [20]. Ya sea de manera individual o conjunta, estos dos grandes focos de información forman el punto de partida de las principales vías de investigación en Food Computing: cocinas internacionales (y el análisis de dietas y recetas derivadas de dichas cocinas) y estudio de composición de alimentos (mayormente orientado a la tecnología de alimentos). Por tanto, las tareas abarcadas en este área tienen como origen uno de estos dos tipos de fuentes de información (o ambos).

A pesar de la gran extensión de tareas que abarca Food Computing, estas se pueden clasificar en cinco grandes grupos en función de los objetivos que persiguen: percepción, reconocimiento de imágenes, recomendación, predicción y recuperación de información, detalladas a lo largo de esta sección.

Percepción

La percepción humana acerca de los alimentos influye de forma directa en los hábitos alimenticios. Por ello, esta tarea es ampliamente estudiada en Food Computing desde el punto de vista de la neurociencia y las ciencias cognitivas. Aquí toma importancia el *factor sensorial* de los alimentos donde se incluyen el sabor, la textura o incluso características superficiales del alimento (p.ej., el color o el brillo). La impresión y las sensaciones que provocan los alimentos influyen en la opinión humana de los que los consumen, existiendo una relación entre éstas y los hábitos alimenticios de la gente. En los últimos años han empezado a surgir líneas de investigación que enfocan estos problemas con Aprendizaje Automático y Redes Neuronales [56].

Reconocimiento de imágenes

El reconocimiento o identificación de elementos del mundo culinario es otro de los problemas más abordado en este ámbito. En esta línea, el reconocimiento de los denominados *item food*, se ha vuelto una tarea esencial en Food Computing. En concreto, el procesamiento de imágenes es sin duda el mayor foco de atracción en este contexto, el cual ya ha sido abordado de múltiples formas: etiquetado de recetas, identificación de ingredientes y grupos alimenticios en grandes bancos de imágenes, etc. En los últimos años, ha aumentado exponencialmente la resolución de estos problemas a partir de técnicas predictivas. Las últimas tendencias residen en combinar la información obtenida a partir del procesamiento de estas imágenes con información de otras fuentes, dando lugar a un conjunto de información de mayor completitud que permita alcanzar mayor precisión en los resultados.

Recomendación

Los sistemas de recomendación forman una de las áreas más explotadas en el campo de la alimentación estos últimos años. Conllevan el desafío de información compleja y polifacética, y esto es lo que la diferencia de las tareas de recomendación centradas en otra áreas, donde puedan existir distintos estándares o componentes más objetivas y, por tanto, más sencillas de calcular e interpretar. Principalmente, han abarcado dos vías que merece la pena destacar. Por una parte, la recomendación basada en las preferencias del usuario en cuestión, teniendo en cuenta para ello sus gustos, sus rutinas, así como patrones en cuanto alimentación que a simple vista puedan ser más complicados de detectar. Por otra parte, se le suma a estos sistemas de recomendación la inmersión en el mundo de la nutrición y vida sana, dando lugar a sistemas que persiguen proporcionar asesoramiento dietético personalizado para el usuario, el cual conlleva una fuerte componente saludable. Está muy ligada a las tareas de predicción, sobre todo, para la parte relativa a la recomendación en función de las preferencias del usuario, lo que suele llevar asociado tareas predictivas [52]. En los últimos años, se ha producido un auge en el desarrollo de sistemas de recomendación centrados en la generación de dietas personalizadas teniendo como requisito que sean saludables [74].

Predicción

La gran cantidad de datos que se produce en este área ha dado lugar a que se empleen técnicas predictivas para resolver problemas de Food Computing. Multitud de parámetros, sobre todo relativos a los alimentos, se han estudiado desde el punto de vista de la predicción. A pesar de los grandes

avances que se han hecho en los últimos años, la complejidad intrínseca en los datos, así como las relaciones entre ellos, hacen que realmente este tipo de tareas tenga éxito en entornos muy controlados y restringidos, y para una cantidad concreta de elementos. En escenarios reales, la variedad de casuísticas y factores a considerar es tan grande que, junto con la falta de estandarización, hace inevitable tener que acotar el problema, reduciendo así las probabilidades de éxito que tendría su aplicación en otras fuentes externas. Entre muchas otras, se ha hecho especial hincapié en la evaluación de propiedades de alimentos, seguridad alimentaria y aspectos culturales.

Por otra parte, tal y como se ha mencionado en los apartados de tareas relativas a Recomendación y Reconocimiento, su aplicación no queda reducida a un campo concreto de Food Computing, sino que se ve aplicada en múltiples ámbitos. En la Sección 3.2 se profundiza en los problemas abarcados en literatura que incluyan técnicas predictivas para su resolución.

Recuperación de información

Las tareas relativas a la recuperación de datos alimenticios pueden no tener aplicaciones directas, pero sí son de vital importancia para que el resto de tareas que se han detallado anteriormente en esta sección puedan desarrollarse de manera apropiada y proporcionen resultados de calidad: la recopilación de un dataset extenso de imágenes de platos de comida puede no tener un gran impacto en sí mismo, pero la existencia de estos facilita el diseño y desarrollo de técnicas que sí precisen de grandes cantidades de datos, por ejemplo, para entrenar o validar los modelos obtenidos [48]. Es por ello que un motor de recuperación de información culinaria se hace indispensable para poder trabajar con estas grandes colecciones de datos.

3.2. Técnicas predictivas en Food Computing

3.2.1. Aproximaciones generales

Desde el punto de vista de la predicción en Food Computing se ha estudiado en amplitud patrones intrínsecos en los alimentos con objetivos muy dispares. En concreto, se ha hecho especial hincapié en la evaluación de propiedades, sobre todo relacionadas con componentes bioactivos y características psicoquímicas de los mismos. En [21] se recopilan algunos de los estudios llevados a cabo en este ámbito, como la evaluación de las características antioxidantes de los aceites o el determinar la tasa de fermentación de las semillas de cacao en función de medidas de aminoácidos y de cambios de color en los alimentos. Además de estudiar sus propiedades, el análisis y evaluación de los alimentos en el mercado también han tenido un gran pro-

tagonismo en las tareas predictivas realizadas en este campo. Se han llevado a cabo estudios realizados en base a monitorización con tratamientos térmicos, como medio para determinar estados de buena calidad en productos alimenticios. También se pueden consultar en la literatura de Food Computing algunos estudios acerca de la conductividad térmica de productos de panadería, así como de patrones en base al nivel de deshidratación de frutas, estableciendo una relación con la pérdida de agua. Por otra parte, se han realizado estudios que buscan predecir la relación entre la carga de bacterias y la concentración de las mismas en determinados vegetales, concretamente en el tomate y en las hojas de lechuga [21].

Otro campo explotado desde estas técnicas ha sido el de la seguridad alimentaria, donde se han llevado a cabo estudios que buscan determinar el tiempo de caducidad de determinados productos, así como predecir el estado ideal de refrigeración de platos cocinados, o la calidad de los alimentos en función de su conservación en frío. Los problemas de predicción también han cobrado protagonismo en lo que concierne a tareas de clasificación. En [21] se propone una metodología para clasificar distintos tipos de aceite, vinagre, o incluso de variedades de queso. También se han estudiado clasificadores de grupos alimenticios a partir de parámetros como pueden ser la claridad, el color, el grado de fermentación o incluso la acidez[52].

Por otra parte, las técnicas basadas en el uso de Redes Neuronales también tienen presencia en el mundo culinario, abarcando principalmente tres vías, que se podrían resumir en predicción de parámetros, clasificación y estudios de calidad. Asimismo, las Series Temporales también tienen cabida en este sector, aunque principalmente han tenido presencia desde un punto de vista médico. En [9] se hace uso de estas técnicas para interpretar datos procedentes de la monitorización de pacientes con diabetes, intentando evaluar la salida obtenida con una terapia concreta. Otros estudios exitosos se centran en el estudio de los precios u otras características económicas dentro del mundo de la comida [80]. En esta área, también entran estudios relativos a la detección de enfermedades en plantas (en vista a la detección de brotes). En [59] estudian la detección de la enfermedad *Blast* en la hoja del arroz a partir de técnicas predictivas en procesamiento de imágenes.

Si nos centramos en el análisis a nivel de receta y no de alimento, es posible aprovechar la detección de ingredientes y los métodos de predicción de la cocina para comparar los alimentos en función de sus componentes [70]. De manera similar, se puede identificar el país de origen de la receta utilizando términos extraídos del texto [54]. Asimismo, otro campo de estudio en el que se han centrado diferentes problemas de predicción ha sido en los aspectos multiculturales relativos al mundo culinario, analizando datos de distintas zonas geográficas a nivel mundial [65]. La relación entre propiedades de ingredientes, influencia de la región y hábitos alimenticios han sido objeto de

estudio en Food Computing durante los últimos años [53] en lo referente a enfermedades relacionadas con la alimentación. También se han analizado patrones de combinación de ingredientes en distintas regiones [11] para la búsqueda de equivalencias de cocina regional de unas zonas geográficas a otras totalmente dispares [40]. Sin embargo, los trabajos en este área se han visto limitados por la ausencia de herramientas que permitan gestionar de forma conjunta bases de datos de composición nutricional de distinta procedencia geográfica. Este problema ha sido expuesto en múltiples trabajos [47], dando lugar a una necesidad de herramientas de fusión de información de distintas fuentes de datos. La falta de estandarización en los datos y la ausencia de una base de datos estandarizada impide realizar estudios más allá de entornos específicos muy controlados.

3.2.2. Modelos predictivos del lenguaje

Si nos centramos en el uso de técnicas predictivas en Food Computing para el procesamiento de información textual, no han tenido tanto recorrido en comparación con las tareas predictivas mencionadas en la sección anterior. Uno de los trabajos más destacados es *Food2Vec*, donde se utiliza un modelo de Word Embedding entrenado con las listas de ingredientes incluidos en las recetas [7]. Otro modelo de Word Embedding también entrenado en el ámbito culinario es *Recipe2Vec* [13], que aunque codifica todo el texto, se centra en la comparación y recuperación de recetas (además, de no ser código abierto). Sí que se han expuesto las ventajas de un modelo de fusión de información heterogénea en Food Computing, en el que se integren imágenes, textos e información nutricional procedente de recetas [66]. Sin embargo, este último modelo está enfocado a tareas de reconocimiento de imágenes, y los datos textuales que contienen son minoría y no bastan por sí solos para desarrollar un modelo de Procesamiento de Lenguaje Natural.

Además, debemos tener en cuenta que los textos, sobre todo procedentes de recetas suelen incluir marcas de alimentos, ya que, una de las propiedades más características del lenguaje alimenticio es la utilización de manera indistinta de marcas y los nombres de alimentos representados por dichas marcas. En este tipo de textos, a menudo encontraremos marcas sustituyendo a los propios ingredientes. Además, la información de estos productos comerciales también aparece en las bases de datos, por la propia flexibilidad que tienen a la hora de almacenar su información. En consecuencia, nuestro modelo de lenguaje debe permitir lidiar con este tipo de términos. En esta línea, destacamos los trabajos de [27], en los que, con un modelo de Word Embedding identificaron, entre otros, nombres de marcas pertenecientes a suplementos dietéticos. Asimismo, se han utilizado modelos de Word Embedding para abordar problemas de evaluación de la calidad de menús de restaurantes, utilizando funciones de puntuación sobre las representaciones

vectoriales de platos en los menús de dichos establecimientos [16].

Más allá de la aplicación de modelos de Word Embedding a recetas y cocinas del mundo, el uso de estos modelos también se ha empleado para detectar similitud a nivel de alimento. Como trabajo destacado en este contexto, se han utilizado representaciones obtenidas con modelos de Word Embedding para enriquecer la red de sabores entre los distintos ingredientes involucrados en recetas, y así poder crear recetas sustitutas en función de factores sensoriales [67]. En este trabajo, se parte de la hipótesis de que los ingredientes se pueden representar en un espacio de sabores [4], y que a partir de esas representaciones se puede pasar de unos grafos receta-ingredientes a otros, teniendo en cuenta dicha información cognitiva.

Desde una perspectiva más amplia, otros trabajos de investigación estudiaron la relación entre los ingredientes y los métodos de cocción a partir de las descripciones de los datos de los alimentos y redes de sabores. Este último problema ha sido ampliamente abordado en la literatura, sobre todo mediante la aplicación de técnicas personalizadas de procesamiento estadístico de lenguaje natural [73, 18, 15].

3.2.3. Identificación de términos alimenticios textuales

El problema de mapear términos entre dos fuentes de datos dadas (en nuestro caso, alimenticias) se puede ver como un problema de identificación de términos equivalentes entre dichas fuentes de datos. En las secciones anteriores ya se ha introducido la necesidad de fusionar fuentes de datos de composición nutricional de distinta procedencia. Esta dificultad también ha sido ratificada por otros estudios llevados a cabo sobre estas fuentes de información, los cuales confirman la necesidad de utilizar herramientas informáticas que permitan automatizar la unificación de bases de datos de composición nutricional de alimentos [8].

Estas bases de datos se caracterizan por una alta complejidad, nivel de detalle, y periodicidad con la que nueva información de este campo se actualiza (o directamente se genera información nueva), formando el llamado *agujero negro* de la nutrición [8]. Por ello, el esfuerzo en fusionar fuentes de datos de esta naturaleza se ha centrado fundamentalmente en el mapeo entre atributos de distintas bases de datos nutricionales, con el objetivo de unificar las características nutricionales de los alimentos [37] y así llevar a cabo comparativas a nivel nutricional entre ellas, en este caso, por medio de una ontología.

El uso de ontologías para poder trabajar de forma simultánea con más de una base de datos de composición nutricional ha tenido un largo recorri-

do [71, 26, 5]. Un ejemplo de ello es EuroFIR³ (European Food Information Resource Network), cuya ontología permite, a través de su buscador *FoodExplorer*, mostrar en un formato armonizado el contenido e información nutricional de múltiples alimentos procedentes de bases de datos europeas [69]. En este ámbito tienen especial protagonismo las ontologías *LanguaL* [36] y *FoodEx2* [1], por su completitud así como por el reconocimiento de los organismos oficiales que las mantienen: US Food and Drug Administration (FDA)⁴ en el caso de *LanguaL* y European Food Safety Authority (EFSA)⁵ en el caso de *FoodEx2*. Sin embargo, la falta de estandarización y los niveles de detalle superficiales alcanzados por dichas ontologías (representan agrupaciones por grupos y subgrupos alimenticios), impiden poder ser utilizadas para la unificación e identificación a nivel de alimento. Esto hace que las herramientas disponibles de mapeo a nivel de ontología resulten insuficientes para mapear alimentos muy concretos en estas bases de datos [45].

Desde un punto de vista de tratamiento de información textual, en Food Computing destacan dos vías principales para abordar esta dificultad: el uso de expresiones regulares para extraer información de los alimentos [16], y utilizar métricas para calcular la distancia entre palabras o textos cortos a partir de modelos de Word Embedding [28, 41], dado el buen funcionamiento que ha tenido a la hora de detectar términos en fuentes de datos externas [27]. En este contexto, las técnicas de similitud que involucran Lógica Difusa pueden aplicarse para obtener los mapeos entre dos fuentes de datos dadas [75]. Estas técnicas ya han sido probadas anteriormente en el campo de Food Computing para mapear datos alimenticios procedentes de cuestionarios de frecuencia de consumo de alimentos [22].

Más allá de medidas de similitud tradicionales como pueden ser la medida de *Jaccard* o la similitud *Coseno* [72], a la hora de detectar equivalencias entre descripciones textuales cortas y modelos de Word Embedding, es destacable mencionar la medida de distancia *Word's Mover*, la cual ha demostrado ser especialmente adecuada para identificar equivalencias a partir de los vectores de documentos cortos obtenidos con Word2Vec [78, 3].

³<http://www.eurofir.org/>

⁴<https://www.fda.gov/>

⁵<https://www.efsa.europa.eu/>

Capítulo 4

Arquitectura del sistema

Este capítulo recoge la descripción de la arquitectura del sistema que se ha desarrollado en este trabajo.

4.1. Descripción del sistema

Como hemos introducido previamente, en este trabajo se va a desarrollar una herramienta que permita trabajar de forma simultánea con datos heterogéneos de distinta procedencia, que posteriormente puedan ser utilizados de forma conjunta en alguna aplicación o problema concreto: en nuestro caso, Food Computing.

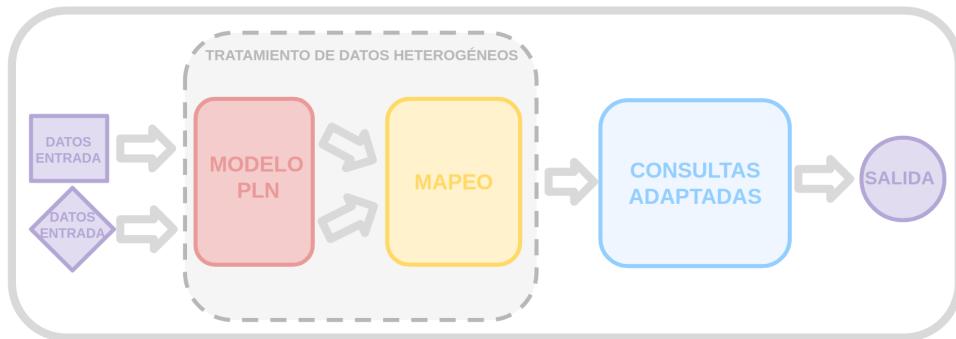


Figura 4.1: Arquitectura del sistema

Desde un punto de vista computacional, la resolución de un problema de esta naturaleza se puede modelar como un sistema que funciona en dos fases: una primera fase que se encarga de fusionar los datos de entrada y devolverlos en una estructura que permite poder trabajar con ellos como un todo, y una segunda fase la cual utiliza dicha información fusionada, ya

correctamente preprocesada, en el desarrollo de una aplicación que resuelva un problema concreto. Esta idea se puede ver reflejada en la Figura 4.1, la cual describe de forma simplificada nuestra propuesta de arquitectura general. En dicha figura se puede observar cómo, a partir de unos datos de entrada heterogéneos entre sí, éstos pasan por un procedimiento que permite trabajar con ellos de forma conjunta (ilustrado en la figura como *Tratamiento de datos heterogéneos*) y poder ser utilizados para llevar a cabo consultas adaptadas a la información una vez fusionada (también apreciable en dicha figura).

4.2. Módulos del sistema

Tal y como se puede apreciar en el esquema de la arquitectura del sistema (ver Figura 4.1), las distintas tareas que se llevan a cabo se pueden organizar en módulos independientes, cada uno de ellos ocupándose de una tarea específica que permita cumplir con todos los requisitos del sistema. En dicha figura se pueden apreciar dos primeros módulos, *Modelo de Procesamiento de Lenguaje Natural (PLN)* y *Mapeo*, los cuales se utilizan para el tratamiento con los datos de entrada heterogéneos. Un último módulo denominado *Consultas Adaptadas*, engloba la funcionalidad de un motor de consultas sobre los datos de las distintas fuentes una vez que ya están fusionados.

Cada uno de estos módulos es independiente de los demás, y se encarga de una labor específica necesaria para el funcionamiento global del sistema. La funcionalidad de cada uno de ellos, así como su papel en el sistema, se detalla a continuación.

4.2.1. Módulo de Procesamiento de Lenguaje Natural

Este módulo hace referencia al modelo de Procesamiento de Lenguaje Natural (PLN) que se ha implementado para el tratamiento de información textual en este trabajo. Permite obtener una representación de las descripciones textuales de los distintos datos que se introducen como entrada al sistema (ver objetos *Datos entrada* en la Figura 4.1). En este punto, se debe tener en cuenta que no hay restricciones previas sobre los datos de entrada más allá de que deban tener descripciones textuales del mismo ámbito para poder llevar a cabo la fusión entre ellos. Por ejemplo, combinar una receta con una base de datos de alimentos a través de las descripciones textuales de los ingredientes de la receta y las de los alimentos de la base de datos de composición nutricional.

Los datos de entrada son procesados en este módulo, para así obtener la

representación en el modelo de lenguaje de la descripción textual de cada uno de los datos introducidos. Estos datos pueden ser datos individuales (p.ej., una receta), o un conjunto de ellos (p.ej., colecciones de recetas o bases de datos de alimentos). El componente principal de este módulo, y por el cual se obtienen las representaciones textuales que utiliza el sistema, es un modelo de procesamiento de lenguaje formado por un Word Embedding entrenado en el dominio específico del problema que se pretende resolver.

En el Capítulo 5 se describe de manera detallada el desarrollo e implementación del modelo de lenguaje. Finalmente, como salida de este módulo, se devuelven las representaciones de las descripciones textuales de los datos que se han proporcionado como entrada al sistema. Las representaciones que se obtienen como salida de este módulo son con las que se trabajará en los módulos restantes que engloban el funcionamiento del sistema. En la Figura 4.2 se puede ver en mayor detalle el contenido de este módulo. En ella se aprecia el modelo de lenguaje basado en Word Embedding.

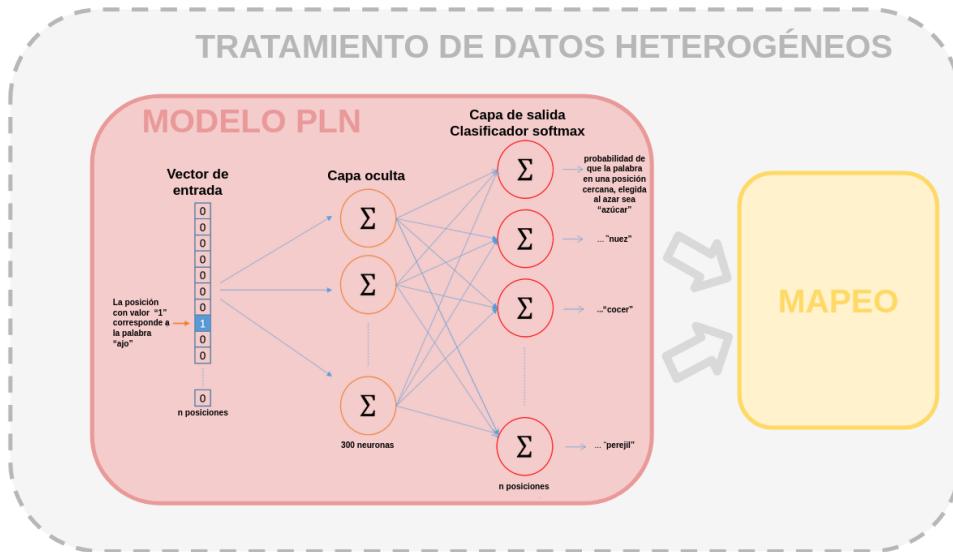


Figura 4.2: Arquitectura del sistema: módulo de NLP

4.2.2. Módulo de Mapeo

Este módulo se encarga de identificar las posibles equivalencias entre los datos de entrada, los cuales, como ya se ha mencionado anteriormente, son de naturaleza heterogénea y distinta procedencia. Para ello, se encarga de obtener la mejor correspondencia posible entre los datos proporcionados como entrada al sistema. Este cálculo se realiza en dos pasos:

1. En primer lugar se calcula la similitud entre las descripciones textua-

les de los datos heterogéneos que se han proporcionado. Para ello, se calcula la distancia entre las representaciones que devuelve el módulo de procesamiento del lenguaje de los datos de entrada que se quieren fusionar. Por ejemplo, supongamos que queremos conocer los valores nutricionales de un alimento concreto, y para ello queremos mapear dicho alimento a una base de datos de composición nutricional. En este primer paso, se calcularía la similitud entre el alimento en cuestión y cada uno de los alimentos que se encuentran en la base de datos (los cuales son candidatos a ser el equivalente de dicho alimento en esa base de datos).

2. En segundo y último lugar, se obtiene, de entre todos los posibles candidatos, aquella correspondencia con la que se obtenga la mayor similitud. Volviendo al ejemplo del punto anterior, en este segundo paso nos quedaríamos aquel elemento de la base de datos de composición nutricional cuya similitud con el alimento a fusionar sea máxima.

El funcionamiento que se ha detallado en los dos puntos anteriores se puede ver en la Figura 4.3, la cual muestra de forma esquematizada las distintas tareas que se abordan en este módulo. Como se puede apreciar en dicha figura, a partir de las representaciones del módulo de procesamiento del lenguaje se establecen correspondencias entre los datos de entrada utilizando el módulo de Mapeo, los cuales permiten trabajar de forma conjunta con los distintos datos que se introducen al sistema.

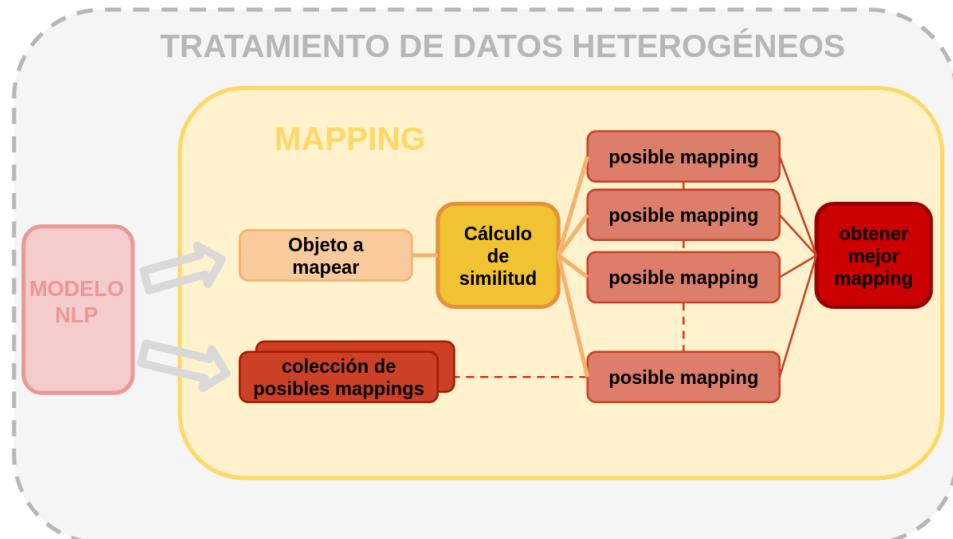


Figura 4.3: Arquitectura del sistema: módulo de Mapping

Debido a la labor general que abarcan tanto este módulo como el de

procesamiento de lenguaje, su funcionalidad se puede ver de forma conjunta a más alto nivel, como una única tarea de combinación de información heterogénea llevada a cabo por dos módulos internos independientes. Esta tarea se encarga de establecer las relaciones necesarias entre los datos de entrada para poder trabajar con ellos de forma agregada y así un tercer módulo (en nuestro caso, un módulo de consultas) pueda utilizar dichos datos de forma conjunta. En el Capítulo 6, se detalla el comportamiento de este módulo.

4.2.3. Módulo de Consultas Adaptadas

A diferencia de los módulos anteriores, los cuales tienen una funcionalidad orientada al tratamiento de las fuentes de datos heterogéneas, la de este último se basa en facilitar la realización de operaciones de consulta que requieran el uso de dichas fuentes de datos de forma simultánea. En otras palabras, a través de este módulo se facilitan los datos, previamente agregados, que resuelvan los requisitos necesarios para el problema en cuestión que se quiera resolver. Estos requisitos dependerán de la tarea específica para la que se quieran utilizar los datos, la cual es independiente tanto de la arquitectura del sistema como del área de aplicación.



Figura 4.4: Arquitectura del sistema: módulo de Consultas Adaptadas

En la Figura 4.4 se muestra de manera esquematizada la funcionalidad que se implementa en el interior de este módulo. En ella se aprecia cómo se permite obtener la información ya fusionada de manera homogénea por medio de consultas específicas. Estas consultas dependen completamente del problema que se pretende resolver y tal y como se puede ver en dicha figura, aquí pueden incluirse otras bases de datos o algunas tareas adicionales de preprocesamiento para adecuar los datos al uso o tipo de consulta específica llevar a cabo. Este módulo representa el objetivo último del sistema que se ha implementado, y se corresponde con la propia salida del mismo (ver objeto *Salida* en Figura 4.1). Su funcionamiento se describe de forma detallada en el Capítulo 7.

4.3. Sistema para adaptación de dietas en Food Computing

En este trabajo, se ha implementado una aplicación simple que permite testear el comportamiento y alcance de la herramienta de fusión de datos heterogéneos que se ha desarrollado, probando su funcionamiento en un sistema real como el descrito en los apartados anteriores. Para ello, se ha abordado un problema de gran relevancia en el mundo de la nutrición y el asesoramiento dietético, el cual consiste en aplicar restricciones alimenticias a recetas, adecuando para ello sus ingredientes por otros más idóneos que sí cumplen con dichas restricciones: dada una receta (y sus ingredientes) y una restricción alimenticia concreta, se adaptarán sus ingredientes para que la receta satisfaga las especificaciones proporcionadas. Por ejemplo, una receta con carne podría ser convertida en una receta vegetariana, modificando los ingredientes pertinentes por otros que sí cumplen las restricciones indicadas.

Este sistema se ha implementado como núcleo de una aplicación móvil que gestiona una colección de recetas, permitiendo adaptarlas a una restricción alimenticia concreta seleccionada por el usuario. Para poder abordar esta tarea, se hace uso de la lista de ingredientes correspondiente a la receta en cuestión y de una base de datos de composición de alimentos. Estos datos, de naturaleza claramente heterogénea, necesitan fusionarse para poder conocer las características nutricionales de la receta, y así poder aplicar las restricciones pertinentes. Por ello, se agregan estos datos por medio de la herramienta de fusión de datos heterogéneos, mapeando los ingredientes de la receta con la base de datos de composición nutricional de alimentos de forma que podamos realizar las sustituciones con un respaldo nutricional. Esta secuencia de tareas puede verse en la Figura 4.5, donde se muestra la arquitectura global del sistema descrito anteriormente, pero ya aplicada al problema en cuestión.

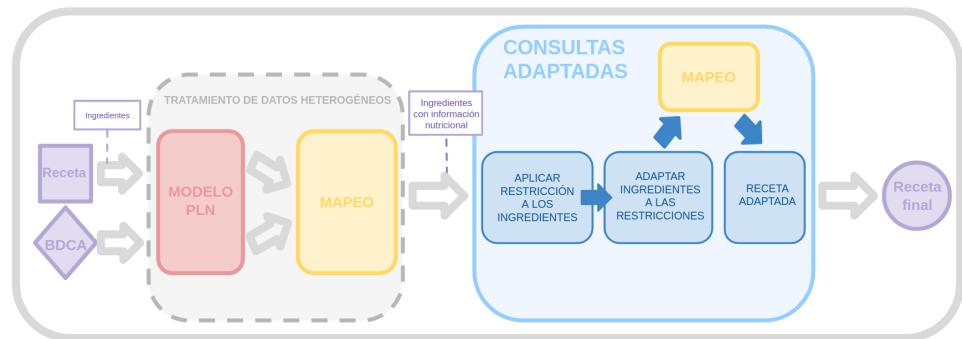


Figura 4.5: Arquitectura del sistema aplicada al problema de adaptación de dietas en Food Computing

Capítulo 5

Word Embedding

Este capítulo introduce el concepto de Word Embedding y describe de forma detallada el modelo desarrollado para este proyecto.

5.1. Introducción a los Word Embeddings

La comprensión del lenguaje natural a partir de datos textuales es una de las áreas más estudiadas dentro del ámbito de la Inteligencia Artificial. Esto se debe a la necesidad de una forma adecuada para la representación de información textual de manera que ésta pueda ser tratada de forma automática. En la literatura, se han abordado técnicas para resolver este problema bajo distintas perspectivas, las cuales se centran en obtener una representación de palabras o un conjunto de ellas (generalmente frases o párrafos), que nos permitan trabajar con la información que contienen. La forma más sencilla de representar información textual como vector es mediante *one-hot embedding*, donde el mapeo de los datos textuales se realiza con un vector donde cada casilla se asocia al índice de cada palabra y contiene el número de veces que aparece dicha palabra en el texto. Sin embargo, esta representación resulta poco escalable y no funciona bien cuando se trabaja con vocabularios de gran tamaño, ni tampoco tiene en cuenta las relaciones que existen entre una palabra y las que le rodean (lo que resulta indispensable al trabajar con información textual). Este último problema es el que recogen las técnicas basadas en *n-gramas* [39], las cuales intentan tener en cuenta, no sólo una palabra, sino también aquellas que se encuentran cercanas en el texto con el objetivo de modelar la información de manera más realista. Esta técnica no sólo ha sido aplicada a nivel de palabra, sino que también se han aplicado técnicas basadas en n-gramas a otros niveles, como puede ser al de frase o de carácter [76]. Usar n-gramas conlleva la dificultad de determinar qué cantidad de información debe de ser la que se tenga en cuenta para una palabra

en cuestión, ya que es difícil determinar (y sobretodo, generalizar) el tamaño de la ventana que se considera a cada lado del elemento a representar. Aun así, en las tareas que involucran Procesamiento de Lenguaje Natural, existen otras dificultades añadidas como pueden ser la ambigüedad del lenguaje, el tratamiento de anáforas o la posibilidad de trabajar con palabras que no se hayan contemplado a la hora de diseñar el modelo del lenguaje en cuestión [46, 68]. Estos problemas han llevado al desarrollo de herramientas más sofisticadas como pueden ser los modelos con métodos basados en Modelo de Espacio Vectorial (VSM) o Asignación Latente de Dirichlet (LDA) [6].

El concepto de Word Embedding se introduce con los Modelos de Lenguaje basados en Redes Neuronales (Neural Networks Language Model) basándose en la idea de que es más probable que dos palabras sean similares si se usan en contextos parecidos. La motivación detrás de esta idea reside en que dos palabras con un significado similar deben usarse de forma parecida, y por tanto, su representación debe de ser parecida también. Este tipo de modelos se caracterizan por ser modelos no supervisados, y algunos de las representaciones más conocidas son las unidades LSTM (Long-Short Term Memory), *Redes Neuronales Recurrentes* (RNN) o *Word2Vector* [46]. El éxito de estos modelos se debe a los buenos resultados que se han obtenido al aplicarlos en problemas que requieran trabajar con información textual. Como consecuencia de ello, se utilizan en multitud de aplicaciones de diferentes campos de estudio y se ha destacado su eficacia respecto a otros métodos a la hora de trabajar con textos [44]. En nuestro caso, el objetivo es definir una representación que nos permita codificar alimentos procedentes de textos de recetas, para así identificar ingredientes y a su vez permitirnos detectar alimentos equivalentes (o incluso de uso equivalente). Este objetivo lo podemos lograr con modelos de Word Embedding puesto que tienen en cuenta el contexto que acompaña a las apariciones de cada palabra en el conjunto de entrenamiento de forma que se mantenga su significado en la codificación, a partir de cuándo, cómo y con qué otras palabras suele aparecer.

Esta importancia del contexto a la hora de obtener la codificación de los datos textuales nos permitirá trabajar con los ingredientes de las recetas de manera más precisa, ya que, muchos de estos ingredientes serán similares (o su uso será parecido), o incluso se caracterizarán por poder nombrarse de más de una forma distinta. Como ejemplo de ello tenemos numerosos alimentos para los que se emplean más de un nombre de manera indistinta: sepia y choco, judías y habichuelas, pero y manzana, o incluso galleta y cokie. Además, por las características del lenguaje culinario, hay que tener en cuenta que las diferencias culturales y las marcas de alimentación se suelen ver involucradas con frecuencia en las recetas de cocina. Muchas veces se utiliza la marca del alimento en sustitución del propio alimento (p.ej., *Danone* y yogur, pan *Bimbo* y pan de molde), y otras veces, dependiendo

del origen geográfico de la receta, un mismo alimento puede ser referido de distintas formas (p.ej., boniato y batata, melocotón y durazno, etc). El utilizar representaciones que sean capaces de tener en cuenta esta semántica (como es el caso de los modelos de Word Embedding) nos permitirá trabajar de una forma más adecuada con estos problemas y solventar dificultades procedentes del lenguaje como pueden ser las diferencias culturales mencionadas anteriormente, así como las asunciones de conocimiento que se hacen al hablar en términos de recetas y cocina en general.

5.2. Word embeddings generales vs específicos en Food Computing

Hoy en día, existen modelos de Word Embedding de carácter general ya entrenados sobre grandes cantidades de datos textuales procedentes de grandes bases de datos de documentos. Este es el caso, por ejemplo, del modelo Word2vec entrenado por Google¹, el cual incluye representaciones vectoriales de palabras para un vocabulario de 3 millones de palabras y frases entrenado a partir de 100 billones de palabras procedentes de un dataset de Google News. Estos modelos permiten abordar de manera eficaz con problemas de Procesamiento de Lenguaje Natural de manera superficial y sin profundizar en campos de conocimiento muy específicos.

Sin embargo, en nuestro caso, al estar trabajando con un dominio tan específico como es el de la nutrición, surge la necesidad de utilizar modelos entrenados sobre dicho dominio concreto. Esto se debe principalmente a la gran cantidad de vocabulario especializado que es difícil de encontrar en un modelo de carácter general, y que limitaría la potencia de éste al no ser capaz de representar las descripciones alimenticias de manera correcta. A la hora de tratar con las descripciones textuales de los alimentos como si se tratases de documentos cortos, cada una de las palabras pertenecientes al documento es representada usando el modelo de Word Embedding entrenado. Dado que el vocabulario del modelo no engloba todas las palabras de ese lenguaje, muchas de ellas no podrán representarse con este modelo. Estas palabras, denominadas *out-of-vocabulary* (OOV) [77], suponen un reto en las tareas de Procesamiento de Lenguaje Natural, y existen en la literatura múltiples formas de abordar este problema [14]. Cuando una de estas palabras no tiene representación en el modelo, es omitida (o en algunos casos reemplazada), introduciendo ambigüedad en la representación final de dicho documento.

En nuestro caso, donde gran parte del vocabulario referente a los alimentos es raramente usado fuera del contexto culinario o nutricional, el uso de un modelo de carácter general intensificaría el problema previamente co-

¹<https://code.google.com/archive/p/word2vec/>

mentado, dando lugar a representaciones muy precarias. Para ilustrar este problema, pongamos como ejemplo que queremos modelar la descripción alimenticia “*Tartaletas de escalibada con anchoas*”. Teniendo en cuenta el tratamiento de las palabras OOV previamente comentado, supongamos que el modelo no capaz de detectar la palabra *Tartaleta*, y por tanto ésta sea omitida a la hora de obtener la representación vectorial de dicha frase. De esta forma, dejaríamos de considerar la descripción original y pasaríamos a tener “*Escalibada con anchoas*”. De igual forma podría pasar con *Escalibada*, pasando a trabajar con “*Tartaletas de anchoas*”, o incluso si no pudiéramos modelar ninguna de las dos palabras mencionadas, pasando a tener ”*Anchoas*”(descripción que no se asemeja a la original). Con este ejemplo se pretende ilustrar que sin un modelo que tenga en cuenta de manera exhaustiva lenguaje alimenticio, no seríamos capaces de trabajar con representaciones precisas de descripciones de alimentos. Como consecuencia, no podríamos trabajar con los datos de una manera adecuada y fiable.

De igual forma, tampoco podríamos trabajar de manera precisa con el vocabulario de un modelo genérico de Word Embedding. Por ejemplo, al intentar obtener las palabras más similares del vocabulario, en un modelo genérico no obtendríamos el nivel de exactitud que podríamos conseguir con un modelo generado a partir de contenido específico. Ejemplo de ello se muestra en la Tabla 5.1, donde se muestran los resultados de alimentos más parecidos a uno previamente proporcionado obtenidos con un Word Embedding de propósito general y con uno específico de alimentación (ver columnas *Alimento*). De igual forma, en la Tabla 5.1, podemos ver en las columnas *Similitud* el grado de similitud (cuyo valor se encuentra entre 0 y 1, siendo 1 la similitud máxima entre dos elementos) de los alimentos más similares obtenidos por cada modelo de Word Embedding. En los resultados expuestos en dicha tabla, se observa cómo para una misma palabra (en este caso *ajo*) se obtienen resultados más cercanos a la palabra original con el modelo correspondiente a un Word Embedding especializado en alimentación. En el Capítulo 8 (Sección 8.2.2) se muestra una comparativa del uso de Word Embedding específico y genérico en este problema que corrobora estas afirmaciones.

Por último, es importante tener en cuenta la naturaleza del dominio en el que estamos trabajando, donde las recetas y los textos relativos a la alimentación suelen mantener un vocabulario *cerrado*. Con ello nos referimos a que, suelen utilizarse siempre los mismos verbos (o sinónimos de los mismos), y que las recetas suelen utilizar en su mayor parte un gran número de ingredientes comunes. Con ello, pretendemos ilustrar que, con una gran cantidad de recetas, podríamos obtener de forma sencilla una muestra representativa de los diferentes alimentos que se ven involucrados en la cocina.

Por esta facilidad a la hora de modelar el vocabulario involucrado en

Tabla 5.1: Similitudes para *ajo* con modelos de W.E.

W.E. (General)		W.E. (Específico)	
Alimento	Similitud	Alimento	Similitud
hinojo	0.7304	diente de ajo	0.5661
orégano	0.7030	cebolla	0.5097
perejil	0.7022	chalota	0.4825
laurel	0.6971	ajo en polvo	0.4796
cilantro	0.6914	cúrcuma	0.4525
cebolla	0.6913	comino	0.4479

el mundo culinario, así como por todas las otras razones previamente expuestas, se ha decidido utilizar un modelo de Word Embedding entrenado de manera específica sobre datos textuales procedentes de repositorios relacionados con el mundo de la alimentación, para así poder trabajar con su vocabulario de la forma más exhaustiva posible.

5.3. Metodología

Para abordar el diseño e implementación del modelo de Word Embedding, se han distinguido cuatro pasos principales, explicados a lo largo de este capítulo. En primer lugar definiremos los datos utilizados y la propia creación del corpus para el entrenamiento del modelo (Subsección 5.3.1), el preprocesamiento llevado a cabo a dichos datos (Subsección 5.3.2) y la implementación y entrenamiento del modelo (Subsección 5.3.3 y Subsección 5.3.4 respectivamente).

5.3.1. Datos utilizados

Se ha decidido utilizar como conjunto de entrenamiento una colección de textos de preparación de recetas en inglés para entrenar el modelo de Word Embedding. Esta decisión ha venido condicionada por la gran cantidad de contenido de esta naturaleza que se puede encontrar en Internet, así como por la existencia de grandes repositorios de recetas cuyos datos contienen información textual que podemos procesar.

En este tipo de textos podemos encontrar los ingredientes de las recetas así como el uso que se le da a estos, con qué tipo de verbos se combinan, con qué ingredientes se mezclan, y una gran cantidad de información intrínseca que puede facilitar la representación de un ingrediente, así como la identificación del mismo a posteriori. Esto se debe a que gracias a la información que se encuentra en los textos de preparación de las recetas, podríamos iden-

tificar otros ingredientes con el mismo comportamiento, que aparezcan junto a los mismos ingredientes, o se utilicen los mismos verbos. Esta funcionalidad da lugar a múltiples aplicaciones que pueden ser aplicadas sobre los ingredientes, como la identificación de ingredientes sustitutos o equivalentes (por ejemplo, “*Tartaleta*” y “*Pasta de hojaldre*”).

En un principio vamos a trabajar con recetas cuyo texto se encuentra en inglés. Las razones de esta decisión son las siguientes:

- La mayor parte de los repositorios disponibles de recetas trabajan con textos de preparación e ingredientes en inglés.
- Las bases de datos de referencia de composición nutricional de alimentos está en inglés. Por otra parte, las bases de datos cuyo idioma original difiere del inglés, incluyen traducciones de los campos textuales a éste, lo que nos permitiría trabajar con dichas bases de datos de igual forma.
- Debido a la internacionalización de las recetas, disponemos de una amplia selección de recetas en inglés que cubren las distintas culturas culinarias independientemente de la región geográfica de las que provengan. De esta forma, podremos trabajar en inglés con recetas de todo el mundo, sin encontrar restricciones de idioma.

Esta decisión tiene como consecuencia que el modelo entrenado nos obligará a trabajar con contenido en inglés. Sin embargo, tal y como se ha expuesto en los puntos anteriores, gracias a la internacionalización de las recetas que existe hoy en día, no tendremos problemas referentes a restricciones en cuanto al contenido de las mismas, sino que de esta forma podremos trabajar en nutrición de forma internacional sin centrarnos en países concretos.

Tabla 5.2: Ejemplo de receta: Noodles en 5 minutos

5 minutes noodles
Pour 700ml freshly boiled water into a saucepan, add the stock cube and stir well to dissolve . Add the noodles, spring onions, peas and chicken, bring to the boil and cook for 5 minutes, or until the noodles and peas are cooked and the chicken is hot through.

En la Tabla 5.2 se muestra un ejemplo de las recetas que utilizaremos para entrenar el modelo de Word Embedding. En ella, se han señalado los verbos típicos que encontraremos en este tipo de recetas, como muestra del vocabulario tan concreto (y también cerrado, sobretodo en el caso de los verbos) con el que estaremos trabajando.

Para formar el corpus de recetas con el que entrenaremos el modelo de Word Embedding hemos utilizado una colección de recetas publicada por archive.org². En esta colección se encuentran recopiladas un total de 267,071 recetas correspondientes a los sitios web enumerados a continuación (la distribución de recetas según sitio web se puede observar en la Tabla 5.3).

- *AllRecipes.com*³: red social centrada en el mundo de cocina en la que se puede compartir y encontrar recetas, trucos de cocina, fotos y vídeos con el objetivo de inspirar a usuarios para crear nuevas recetas. Su página de recetas es una de las más visitadas del mundo.
- *BBC Food Recipe*⁴: colección de recetas procedentes de los programas y chefs de la BBC, la principal emisora de servicio público del mundo. En ella se pueden encontrar recetas clasificadas por estación, festividades, e incluso por ingredientes.
- *Epicurious*⁵: es una marca digital para consumidores centrada en la comida y el arte culinario. Tiene más de 300,000 recetas, así como vídeos y recetas y consejos para la cocina del día a día.
- *CookStr*⁶: sitio web de recetas cuyo objetivo se centra en organizar los libros de cocina de éxito así como las recetas del mundo y hacerlos universalmente accesibles.

Tabla 5.3: Corpus de recetas: origen y número de recetas

Procedencia	Número de recetas
BBC Food Recipe	10,679
Epicurious	20,111
Cookstr	225,602
AllRecipes	10,679
Total de recetas	267,071

Dicha colección se distribuye en ficheros de extensión .json clasificados por el sitio web de procedencia de la receta. De esta forma, disponemos de cuatro ficheros con recetas, donde cada uno de ellos está formado por la lista

²<https://archive.org/download/recipes-en-201706>

³Sitio web de AllRecipes: <https://www.allrecipes.com/>

⁴Sitio web de BBC Recipes: <https://www.bbc.co.uk/food/recipes>

⁵Sitio web de Epicurious: <https://www.epicurious.com/>

⁶Sitio web de CookStr: <https://www.cookstr.com/>

total de recetas obtenidas de dicho sitio web. En la Figura 5.1 se muestra la estructura de las recetas que forman este repositorio.

```
{
  "ingredients": [
    "1 1/2 teaspoons cumin seeds",
    "3/4 cup ketchup",
    "1 1/2 teaspoons chopped canned chipotle chilies plus 1
      tablespoon spicy tomato sauce reserved from can",
    "1 1/2 tablespoons fresh lime juice",
    "1 tablespoon tequila"
  ],
  "picture_link": null,
  "instructions": "Stir cumin seeds in heavy small saucepan
    over medium heat until fragrant and seeds darken, about
    1 minute. Transfer to plate and cool. Grind seeds in
    spice grinder or in mortar with pestle. Return cumin to
    same saucepan. Whisk in ketchup, chipotle chilies with 1
    tablespoon spicy tomato sauce, lime juice and tequila.
    Simmer over medium-low heat until ketchup thickens
    slightly, stirring occasionally, about 5 minutes. Serve
    warm or at room temperature. (Can be made 1 week ahead.
    Cover and chill.)\nStir cumin seeds in heavy small
    saucepan over medium heat until fragrant and seeds
    darken, about 1 minute. Transfer to plate and cool.
    Grind seeds in spice grinder or in mortar with pestle
    .\nReturn cumin to same saucepan. Whisk in ketchup,
    chipotle chilies with 1 tablespoon spicy tomato sauce,
    lime juice and tequila. Simmer over medium-low heat
    until ketchup thickens slightly, stirring occasionally,
    about 5 minutes. Serve warm or at room temperature. (Can
    be made 1 week ahead. Cover and chill.)",
  "title": "Cumin-Chipotle Ketchup "
}
```

Figura 5.1: Estructura de las recetas en el dataset de archive.org

5.3.2. Preprocesamiento de los datos textuales

Como se puede apreciar en la Figura 5.1, los datos textuales de las recetas no se encuentran en un formato directamente manipulable para entrenar un modelo de lenguaje. Por ello, es necesario aplicar técnicas de preprocesamiento de datos que permitan su limpieza y adecuación para poder ser usados como datos de entrada en el entrenamiento del modelo de Word Embedding. En este paso, así como en todo el proceso de creación del modelo, haremos uso de la librería de Topic Modelling Gensim⁷. Esta librería de código abierto proporciona funciones para trabajar con problemas de Procesamiento de Lenguaje Natural en el lenguaje de programación Python, y en particular, para entrenar y trabajar con modelos de Word Embedding. Con las funciones implementadas en dicha librería, podremos tanto entrenar como utilizar con facilidad el modelo, así como llevar a cabo las tareas de

⁷<https://radimrehurek.com/gensim/>

limpieza y preprocesamiento de los datos que utilicemos.

Creación del corpus

En primer lugar, hay que crear un corpus que recoja todos los datos textuales con los que se va a trabajar. Como ya se ha mencionado en secciones previas, los datos se encuentran distribuidos en cuatro ficheros, cada uno de ellos con una lista de las recetas obtenidas de los sitios web de recetas previamente mencionados. Por tanto, se requiere un preprocesamiento previo de cada uno de los ficheros correspondientes de forma que podamos disponer de una sola colección de datos que contenga únicamente la información textual que vamos a utilizar. Dicho de otra forma, debemos reestructurar la información para poder trabajar con un corpus con todos los datos textuales procedentes de los textos de preparación de las recetas que tenemos.

Para solventar esta dificultad, se ha generado un fichero por cada receta con el texto de su preparación, de forma que únicamente trabajemos con ficheros cuyos datos son los que nos interesa preprocesar. De manera automatizada, se lee cada uno de los ficheros previamente generados para formar el corpus de recetas. En la Figura 5.2 se aprecia de forma esquemáticaizada el procedimiento para montar este corpus.

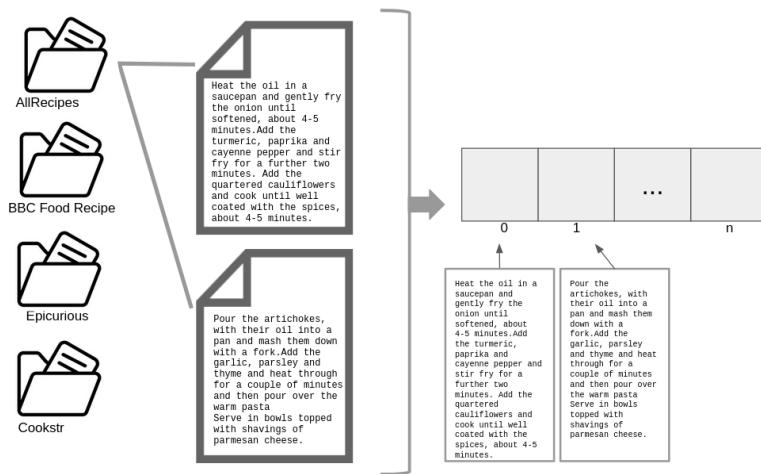


Figura 5.2: Procedimiento para formar el corpus de recetas

Limpieza del corpus

Como se ha podido observar en ejemplos previos, el texto de preparación de las recetas se encuentra sin ningún tipo de procesamiento previo, y por ello necesita un tratamiento adecuado de la información textual para que el modelo de Word Embedding proporcione resultados de calidad. En nuestro caso,

para el mencionado corpus de recetas se ha llevado a cabo la limpieza y tokenización que se presenta a continuación. Para complementar los pasos, utilizaremos como ejemplo el siguiente texto de preparación que se corresponde con una de las recetas del dataset: “Combine Ranch, salsa, tomatoes and chilies, if desired. Chill 1 hour. Serve with tortilla chips.”

1. **Tokenización del texto.** Se obtienen los tokens del fichero. Al generar los tokens se lleva a cabo la eliminación de símbolos (incluidos los de puntuación), dígitos, y pasar texto a minúsculas. Como resultado, se obtiene el texto como una lista de tokens con las palabras del fichero formadas por caracteres alfabéticos. Tras aplicar estos cambios, el ejemplo quedaría de la siguiente manera:

```
[‘combine’, ‘ranch’, ‘salsa’, ‘tomatoes’, ‘and’,
‘chilies’, ‘if’, ‘desired’, ‘chill’, ‘hour’, ‘serve’,
‘with’, ‘tortilla’, ‘chips’]
```

2. **Eliminación de las llamadas *stopwords*.** Las stopwords son palabras muy comunes del vocabulario, que no van a proporcionar información útil, y por tanto es recomendable prescindir de ellas [64]. Ejemplos de stopwords en inglés pueden ser “the”, “a”, “my”, “your”, etc. Para este caso, hemos utilizado la lista de stopwords en inglés proporcionada por la librería *Gensim*. En este punto, el fichero tomado como ejemplo quedaría como se muestra a continuación:

```
[‘combine’, ‘ranch’, ‘salsa’, ‘tomatoes’, ‘chilies’,
‘desired’, ‘chill’, ‘hour’, ‘serve’, ‘tortilla’, ‘chips’]
```

En el ejemplo se aprecia cómo se han eliminado aquellas palabras incluidas en el vocabulario de stopwords (en este caso, las stopwords existentes en el ejemplo son ‘the’, ‘if’ y ‘with’).

3. **Aplicación de lematización.** Por último, aplicaremos una normalización lingüística al corpus conocida como lematización. Mediante esta técnica, se reducen todas las palabras a su raíz, de forma que se transforman las distintas variaciones morfológicas de una palabra a una única forma común. Esto ocurrirá, por ejemplo, con los verbos, donde nos encontraremos un mismo verbo en sus distintas formas verbales cuando todos ellos hacen referencia a una misma acción (así forzaremos a tener una única forma de aparición de los mismos). Tras aplicar la normalización comentada, el fichero quedaría de la siguiente forma:

```
[‘combin’, ‘ranch’, ‘salsa’, ‘tomato’, ‘chili’, ‘desir’,  
‘chill’, ‘hour’, ‘serv’, ‘tortilla’, ‘chip’]
```

En dicho ejemplo se puede observar el caso comentado sobre los verbos, los cuales se han llevado a su raíz (como ocurre en el caso de ‘serve’, que pasa a ser ‘serv’). También podemos ver que como resultado de dicha normalización se ha eliminado el plural de las palabras (‘tomatoes’ pasa a ser ‘tomato’).

Detección de bigramas

Por otra parte, en el lenguaje natural es normal que aparezcan palabras compuestas. El lenguaje culinario no es una excepción, y también existen palabras que suelen aparecer siempre juntas, como puede ser “*pimienta negra*”, “*vino tinto*.º” “*azúcar moreno*”. Este tipo de casos, donde hay palabras que tienen más sentido ser tratadas como una sola que de manera independiente, da lugar al uso de técnicas de bigramas, que permiten tratar palabras (o en nuestro caso, tokens) de manera colectiva como si de una única se tratase.

Para llevar esta idea a la práctica, en primer lugar es necesario entrenar el modelo de bigramas a partir de todos los textos que forman el corpus, para así poder detectar qué palabras son las que aparecen juntas una cantidad de veces considerable y poder representarlas en el texto como si fueran una sola. Esto se lleva a cabo sobre los datos una vez limpiados de la forma descrita en la sección anterior. Para este paso, también se ha hecho uso de las utilidades proporcionadas por la librería Gensim.

Una vez entrenado el modelo de bigramas, éste es aplicado sobre el mismo corpus para sustituir aquellos tokens de los que se detecten bigramas. A continuación se muestra cómo quedaría el ejemplo utilizado en el apartado anterior tras la detección de bigramas.

```
[‘combin’, ‘ranch’, ‘salsa’, ‘tomato’, ‘chili’, ‘desir’,  
‘chill’, ‘hour’, ‘serv’, ‘tortilla_chip’]
```

Tal y como se puede ver, los token ‘tortilla’ y ‘chip’ pasarían a ser un token único (‘tortilla_chip’), lo cual es de esperar porque son dos palabras que, en las recetas, suelen encontrarse siempre juntas ya que se trata de un alimento cuya descripción textual es una palabra compuesta.

5.3.3. Implementación de Word2vec

Para construir el modelo lingüístico a partir del corpus de recetas descrito en las secciones anteriores se pueden utilizar distintos algoritmos no supervisados estandarizados. Entre los más conocidos, se encuentran Word2Vec [50,

51], Glove [58] o fasttext [10]. En este trabajo, se ha optado por utilizar el algoritmo Word2vec para entrenar el Word Embedding. De este algoritmo existen diferentes implementaciones; las más conocidas son las denominadas *Continuous Bag of Words* and *Skip-Gram* [62]. A continuación, se explica de forma detallada en qué consisten estas implementaciones.

Modelo Continuous Bag of Words (CBOW)

Esta implementación de Word2vec se centra en predecir una palabra a partir del contexto, cuya estructura puede apreciarse en la Figura 5.3. La capa de entrada de la red (X) representará el contexto, la capa oculta (H) hace referencia al Embedding entrenado y la capa salida (Y) representará la palabra de la que se quiere obtener su representación en el modelo. Este contexto puede estar formado por una o más palabras. En el caso de que el contexto esté formado por una única palabra, estaríamos en la versión simplificada de este modelo (que es la mostrada en la Figura 5.3). Independientemente del tamaño tenido en cuenta en el contexto, cada una de las palabras que formen parte del mismo se encuentran codificadas mediante el método *one-hot-enconding* explicado en secciones anteriores. Este vector tendrá como tamaño el del vocabulario (notado por V), y estará completo a valor 0 exceptuando las posiciones que se correspondan con la palabra (o palabras) que se utilicen en el contexto, cuyo valor será 1.

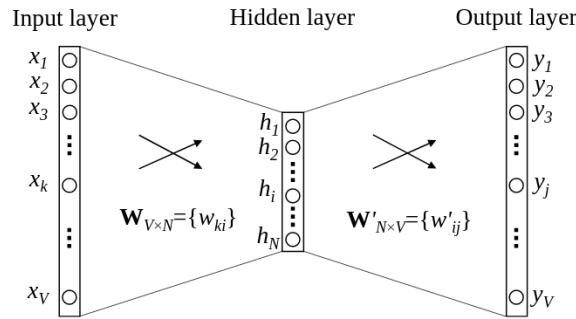


Figura 5.3: Modelo CBOW donde el contexto es una palabra [62]

Modelo Skip-Gram

En este modelo, se lleva a cabo el procedimiento inverso. A pesar de que la capa oculta (H) sigue representando el Embedding, en este caso, la capa de entrada del modelo (X) es la palabra que corresponde a la capa de salida en el modelo CBOW (es decir, el target). Tal y como se puede ver en la Figura 5.4b, la capa de salida (Y) de este modelo está formada por el

contexto correspondiente a la palabra proporcionada como entrada. En otras palabras, mientras que en el modelo CBOW lo que se predice es la palabra, en el modelo Skip-Gram lo que se pretende predecir es el contexto de dicha palabra. En la selección de Figuras 5.4 se puede ver de forma gráfica esta diferencia existente entre los modelos.

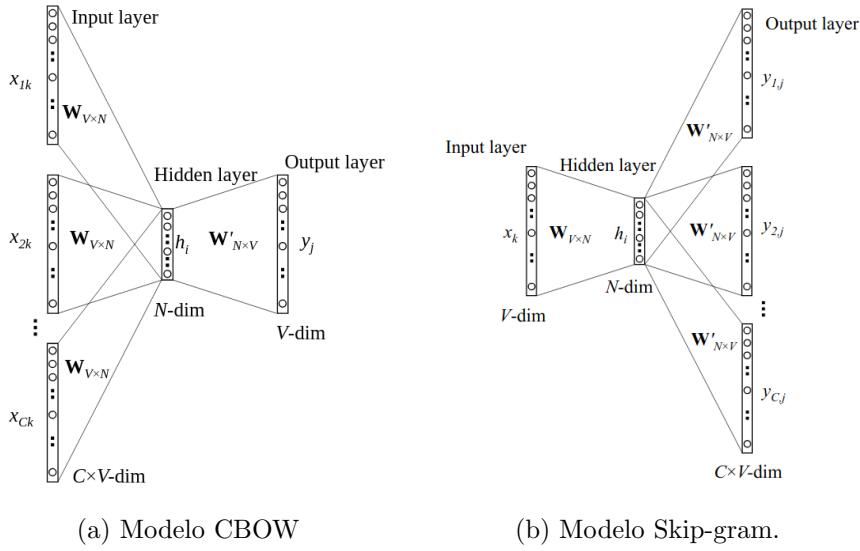


Figura 5.4: Modelos del algoritmo Word2vec [62]

Como es de esperar, ambos modelos tienen sus ventajas y desventajas, y cada uno de ellos es más o menos apropiado en función del problema en cuestión. En líneas generales, el modelo CBOW suele funcionar mejor con palabras del vocabulario que aparecen con frecuencia, mientras que Skip-Gram es capaz de obtener buenas representaciones para vocabulario poco frecuente en el corpus utilizado para entrenar el modelo de Word Embedding.

5.3.4. Entrenamiento del modelo

Para el entrenamiento del modelo de Word Embedding a partir del corpus de recetas hemos decidido utilizar la implementación de Word2vec *Continuous Bag of Words* (CBOW). La razón principal reside en su implementación, que permite obtener buenas representaciones para palabras de uso frecuente. Esto es especialmente adecuado para nuestro problema dado que en el vocabulario culinario hay una gran cantidad de palabras que aparecen con mucha frecuencia en los textos. Al igual que con la creación del corpus y su preprocessamiento, hemos utilizado las funcionalidades de la librería Gensim para llevar a cabo el entrenamiento del modelo. En el Capítulo 8 se detalla la experimentación y el ajuste de hiperparámetros realizado.

Capítulo 6

Mapeo de datos

Este capítulo presenta el procedimiento y las distintas medidas implementadas para calcular la similitud entre dos descripciones de forma que podamos distinguir cuándo dos elementos son equivalentes.

6.1. Procedimiento de mapeo

Ser capaces de identificar un alimento en una base de datos de composición nutricional es una tarea esencial si queremos realizar algún tipo de tarea automática que conlleve consultar valores nutricionales de alimentos o ingredientes en recetas. Esta tarea se puede simplificar a la decisión de si dos alimentos son equivalentes o no en función de lo parecidas que sean sus descripciones nutricionales.

Para conocer la similitud entre dos descripciones textuales de alimentos se han implementado distintas medidas de distancia, detalladas en la Sección 6.2. Con dichas medidas de distancia, se permitirá mapear alimentos entre bases de datos con el objetivo de identificar equivalencias y agregar sus atributos. Para poder llevar a cabo esta identificación, es necesario conocer la distancia de la descripción textual del ingrediente en respecto a las descripciones de los posibles alimentos hacia los que podamos realizar el emparejamiento. El mapeo más adecuado (y el que finalmente devuelve la herramienta como alimento equivalente) será aquel con el obtengamos la distancia mínima de entre todos los alimentos de la base de datos a mapear. De esta forma, cuanto más cercano a 0 sea el valor de distancia entre dos alimentos dados, más parecidos serán esos alimentos y viceversa. Finalmente, elegiremos la medida que mejor se ajuste a las funcionalidades del sistema implementado, en función del resultado obtenido con cada una de ellas.

Es importante recalcar, que estas descripciones no tienen por qué co-

rresponderse con una única palabra; en algunos casos sí lo será (p.ej., *patata* o *zanahoria*) pero en otros casos, la descripción del ingrediente puede ser más extensa (p.ej., *pimiento asado en conserva* o *salsa mahonesa reducida en calorías*). Por ello, cada una de estas descripciones va a ser tratada como un documento corto, formado únicamente por dicha descripción. Con esta consideración, podremos tener en cuenta toda la información de la descripción del ingrediente, tanto de una forma global, como de manera individual con las palabras que lo forman. Manteniendo la coherencia con la implementación del modelo de Word Embedding, a las descripciones textuales se les debe aplicar previamente las tareas de preprocesamiento y limpieza de datos detalladas en el Capítulo 5. Por ello, en este punto hablaremos de *token* en lugar de *palabra*, como forma de referirnos al contenido de la descripción correctamente procesado.

6.2. Medidas de distancia implementadas

Sea S_i la descripción textual correspondiente a un alimento, y sea $T_i = \{t_1, \dots, t_n\}$ el conjunto de tokens obtenido como resultado de las tareas de preprocesamiento aplicadas a dicha descripción. Por ejemplo, consideremos el elemento k con la representación textual $S_k = "Canned fish, average"$. Su conjunto T_k correspondiente sería $\{ "can", "fish", "average" \}$. Teniendo en cuenta esta nomenclatura, se han implementado distintas medidas de distancia entre descripciones textuales de alimentos, detalladas a continuación.

6.2.1. Distancia sintáctica entre descripciones

En primer lugar, se ha optado por implementar una medida de distancia basada en la concordancia entre dos descripciones textuales, que nos permita distinguir si dos descripciones son más parecidas o no en base a la comparación a nivel de caracteres de su representación textual.

Distancia de Jaccard

La distancia de Jaccard (ver Fórmula 6.1) permite medir el grado de intersección entre dos conjuntos [75]. En este caso, cada descripción alimenticia formará un conjunto, y los tokens de dicha descripción serán los elementos de dicho conjunto. Esta medida devuelve un valor comprendido entre 0 y 1, donde el 0 hace referencia a dos conjuntos iguales mientras que el 1 a dos conjuntos totalmente disjuntos. Este valor es calculado a partir de la intersección entre ellos. Para poder medir el grado de la intersección de los conjuntos, se debe determinar cuáles, de entre todos los elementos de dos conjuntos dados, pertenecen a la intersección de ambos. Para que uno

de los tokens forme parte de la intersección de dichos conjuntos, debe aparecer de forma exacta en ambos conjuntos. En nuestro caso, dos elementos pertenecerán a la intersección en función del parecido léxico entre ellos.

$$J(S_1, S_2) = 1 - \frac{|T_1 \cap T_2|}{|T_1| + |T_2| - |T_1 \cap T_2|} \quad (6.1)$$

Para ver el funcionamiento de esta medida de distancia, se va a mostrar como ejemplo la medida de distancia de Jaccard entre dos descripciones textuales de alimentos S_1 y S_2 , que se corresponden con ‘Coconut oil’ y ‘Palm seed oil’ respectivamente. En la Figura 6.1a, se aprecia cómo quedarían representados como conjuntos las descripciones denotadas por S_1 y S_2 . En dicha imagen se puede ver cómo la descripción ‘Coconut oil’ se representa como un conjunto con dos elementos: **coconut** y **oil**, mientras que ‘Palm seed oil’ se representa como conjunto formado por tres elementos (**palm**, **seed** y **oil**). En la Figura 6.1b se puede ver la intersección entre los dos conjuntos. Como se puede observar, el conjunto intersección está formado por un único elemento, **oil**, el cual es el único que aparece en ambos documentos (y por tanto en sus conjuntos). Aplicando sobre estos conjuntos la medida de Jaccard expuesta en la Fórmula 6.1, obtendríamos que la distancia entre ambos conjuntos es $J(S_1, S_2) = 1 - \frac{1}{2+3-1} = 0.75$.

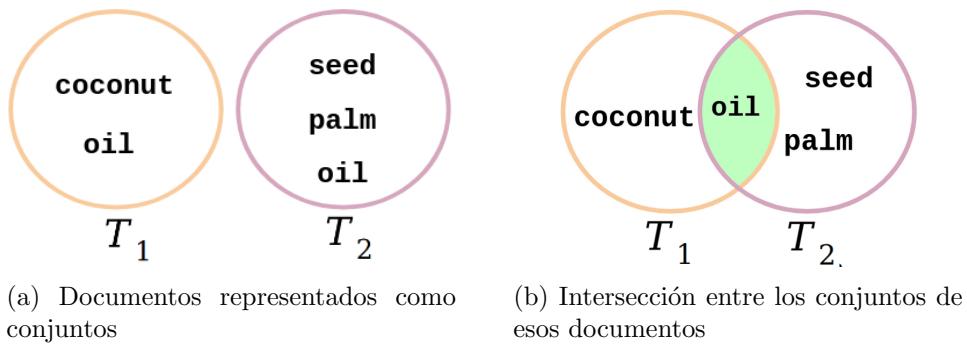


Figura 6.1: Intersección entre dos conjuntos

Distancia Levenshtein

La distancia de Levenshtein es otra medida de concordancia entre descripciones textuales, y se define como el mínimo número de operaciones o movimientos que son necesarios para transformar una secuencia de caracteres en otra [79]. Esta medida de distancia permite tanto trabajar de forma normalizada (por ejemplo, en el rango 0 a 1) como sin normalizar.

Aplicada a nuestro problema en cuestión, hace referencia al número de movimientos necesarios para pasar de la descripción de un ingrediente a la de otro. Los movimientos permitidos vienen enumerados a continuación:

- Eliminar un carácter: ABC → AB, AC, BC
- Añadir un nuevo carácter: ABC → ABCD, EABC, AEBC
- Sustituir un carácter por otro: ABC → ABE, ADC, FBC

6.2.2. Distancia semántica entre descripciones

En este caso, se ha optado por una medida de distancia que utilice la representación numérica obtenida por el modelo de Word Embedding. Dado que esta representación se obtiene utilizando el contexto de cada palabra, trae de forma implícita la representación de su semántica. El utilizar medidas de distancia entre las representaciones numéricas de las palabras (o documentos) nos permitirá detectar si dos palabras son más o menos parecidas en función del contexto en el que se usen.

Distancia Word Mover's

La medida de distancia Word Mover's trata los documentos cortos como representaciones de una nube de puntos, donde cada punto representa una palabra del documento expresada de forma vectorial a partir del modelo de Word Embedding. La distancia entre dos nubes se cuantifica como el coste que las palabras de un documento de texto deben asumir para coincidir exactamente con la nube de puntos del otro documento de texto [43]. En nuestro caso, cada descripción alimenticia formará un documento corto.

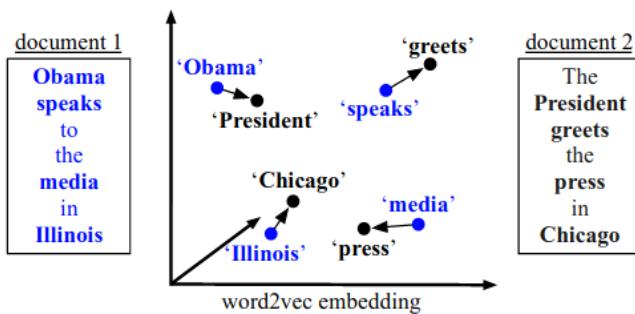


Figura 6.2: Funcionamiento de la distancia Word Mover's [43]

Para calcular la distancia entre dos palabras individuales, se utiliza una distancia euclídea entre las representaciones vectoriales correspondientes.

De esta forma, se mantiene la semántica a la hora de obtener la similitud entre las palabras. En la Figura 6.2 se puede apreciar de manera gráfica el funcionamiento de esta medida de distancia.

6.2.3. Distancia híbrida entre descripciones

En los apartados anteriores se han detallado medidas de distancias que tienen en cuenta la distancia entre descripciones desde un punto de vista puramente léxico o desde una aproximación basada en la semántica.

Con el objetivo de obtener una mayor precisión en los resultados, se introduce una medida híbrida que tiene como objetivo combinar la capacidad sintáctica y la semántica de las descripciones textuales. Esta función de distancia se formula como una combinación ponderada de las distancias de Jaccard y Word Mover's (ver Fórmula 6.2.3). Con esta medida, se pretende beneficiar aquellas descripciones textuales que sean similares desde un punto de vista sintáctico, sin olvidar la importancia de la semántica a la hora de alcanzar el resultado más preciso posible.

$$HDISTANCE(t_1, t_2) = wJ(t_1, t_2) + (1 - w)WMD(t_1, t_2) \quad (6.2)$$

donde $w \in \mathbb{R}$ and $0 \leq w \leq 1$

6.2.4. Fuzzificación de las medidas de distancia

Cuando se trabaja con datos textuales, una de las cosas que hay que tener en cuenta es que hay que hacer frente a los problemas derivados de la ambigüedad del lenguaje. En capítulos anteriores ya se ha introducido la importancia de la semántica en el tratamiento de datos textuales. Esta importancia viene derivada de dificultades comunes que suelen aparecer al trabajar con información textual, como puede ser el uso de sinónimos, palabras muy similares, o incluso hacer frente a distintos niveles de detalle en las descripciones con las que trabajamos.

Para lidiar con esta vaguedad existente en el lenguaje, se ha propuesto implementar medidas de distancia con Lógica Difusa que permitan dotar a nuestra herramienta de una mayor flexibilidad y robustez para hacer frente a este tipo de desafíos. En concreto, se han fuzzificado dos medidas de las detalladas en este capítulo, que se corresponden con Jaccard y Word's Mover. En ambas medidas se parte de la descripción textual como un conjunto de tokens, que se corresponden con la lista de palabras que forman la descripción textual a las cuales se les ha aplicado las tareas de preprocesamiento explicadas en el capítulo anterior.

Distancia difusa de Jaccard

En este capítulo, se ha explicado el funcionamiento de la medida de distancia de Jaccard, la cual computa la distancia en base a la cantidad de elementos que forman parte de la intersección entre conjuntos. Dicha medida establece la intersección de forma estricta: un elemento pertenece al conjunto intersección solo si se encuentra de manera exacta en ambos conjuntos. Teniendo esto en cuenta, la medida de Jaccard no contempla que dos elementos similares se valoren a la hora de medir la similitud entre dos conjuntos.

Por la ambigüedad existente en el lenguaje y las múltiples formas de expresar una descripción alimenticia, se ha optado por implementar una versión difusa de la medida de Jaccard, que permita tener en cuenta el parecido entre los elementos de ambos conjuntos de manera proporcional al grado de similitud que tengan. De esta forma se valorará de forma positiva el parecido léxico entre los elementos que, sin llegar a formar parte de la propia intersección, sí tienen un grado de parecido notable que incita a ser considerados en el cálculo de la medida de distancia.

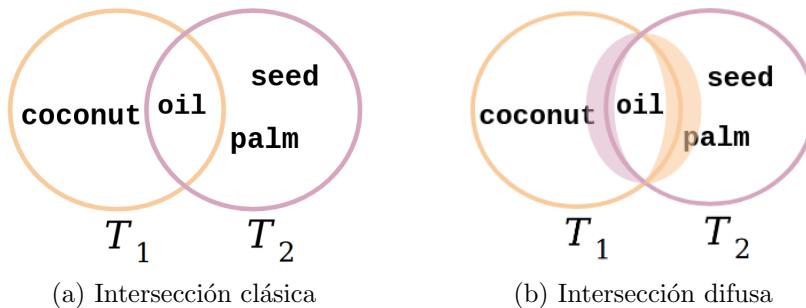


Figura 6.3: Intersección entre dos conjuntos

La versión difusa de Jaccard implementada se aprecia en la Fórmula 6.3 [75]. Consiste en una combinación de similitudes tanto a nivel de token como de carácter, con el objetivo de determinar el conjunto de intersección difusa que existe entre dos conjuntos (entendiendo cada descripción textual como un conjunto). Para conocer la distancia a nivel de token se utiliza la medida Jaccard que se ha descrito anteriormente, y mediante un umbral δ se determina cuáles de estos tokens forman parte del conjunto de intersección difuso. El valor del umbral se ha ajustado empíricamente al valor de 0.2.

$$\tilde{J}_\delta(S_1, S_2) = \frac{|T_1 \tilde{\cap}_\delta T_2|}{|T_1| + |T_2| - |T_1 \tilde{\cap}_\delta T_2|} \quad (6.3)$$

$$\delta = 0.2$$

Para ilustrar el cálculo de esta medida, vamos a usar el mismo ejemplo que se empleó para la medida Jaccard (ver Figura 6.1) para ejemplificar cómo funciona su versión difusa. En la Figura 6.3 se puede ver el valor de distancias a nivel de token entre ambos conjuntos. Por simplificación, únicamente se muestran aquellas distancias cuyo valor es distinto de 1, y que por tanto, tienen alguna posibilidad de poder formar parte de la intersección difusa. En este caso, las líneas continuas, representan aquellos elementos cuya medida de distancia supera el umbral y por tanto se incluyen en la intersección difusa. Como se puede ver, en este caso el único elemento que lo supera es *oil*, por lo que la intersección tradicional y la difusa serían la misma: $J(S_1, S_2) = 1 - \frac{1}{2+3-1} = 0.75$.

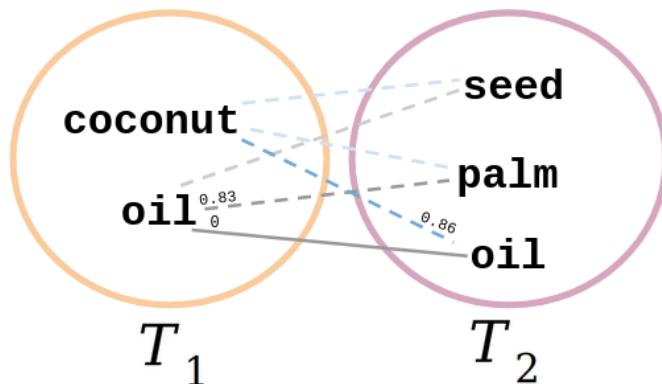


Figura 6.4: Función de distancia difusa Jaccard

Ahora imaginemos que, en un hipotético caso, la distancia entre *coconut* y *palm* es 0.1 porque son extremadamente similares, y por tanto entraría dentro del conjunto de la intersección difusa al no superar el umbral. Si aplicamos la fórmula de la distancia difusa de Jaccard, se quedaría tal que $J(S_1, S_2) = 1 - \frac{1+0.9}{2+3-1-0.9} = 0.39$, un valor inferior que el obtenido con la versión clásica, que sí representaría esta valoraría la similitud entre los elementos.

Distancia difusa entre documentos

Se ha diseñado una medida de la distancia entre documentos cortos partiendo de un enfoque difuso, considerando cada documento como el conjunto de tokens que se obtienen del preprocesamiento de la descripción textual.

En las medidas que trabajan con conjuntos que se han utilizado hasta ahora, (entre ellas incluida la medida difusa de Jaccard) se hace hincapié en el parecido de los elementos entre conjuntos uno a uno. Sin embargo, parece interesante medir la distancia de un elemento a un conjunto completo,

en lugar de distancia entre elemento y elemento. Esto permitiría obtener una medida que valore el papel de cada elemento al nivel de la descripción textual, y no elemento a elemento lo cual podría resultar poco apropiado, puesto que no se estaría teniendo en cuenta a la descripción como un todo. Por ello, se ha propuesto en este trabajo la medida de distancia \tilde{D} , la cual se centra en esta afirmación y cuyo cálculo se expone en la Fórmula 6.4. Para poder obtener el valor de distancia \tilde{D} entre las descripciones textuales S_1 y S_2 se tienen en cuenta los siguientes puntos:

1. Obtener el conjunto unión ($T_1 \cup T_2$) de los conjuntos T_1 y T_2 , los cuales se corresponden respectivamente con el conjunto de tokens que forman las descripciones S_1 y S_2 .
2. Calcular, para cada elemento presente en los conjuntos T_1 y T_2 (el cual se encuentra en el conjunto $T_1 \cup T_2$) su grado de pertenencia al conjunto contrario. Para calcular el grado de pertenencia de un elemento a un conjunto, se utiliza la función $\mu_{T_i}(x)$, donde i hace referencia al conjunto y x al elemento del que se quiere obtener el grado de pertenencia a dicho conjunto i .
3. Calcular el valor de la intersección difusa de los dos conjuntos. Para calcular el grado de pertenencia de un elemento a un conjunto, se utiliza la función $\mu_{T_i}(x)$ comentada en el punto anterior.
4. El valor de distancia se corresponde con el resultado de la sumatoria en el paso 2 dividido entre el valor de la intersección en el paso 3.

Como se puede apreciar en el cálculo de dicha medida, se hace uso de la función de pertenencia $\mu_{T_i}(x)$ la cual permite obtener el grado de pertenencia del elemento x al conjunto T_i , donde x hace referencia a un elemento de un conjunto obtenido a partir de una descripción textual, y T_i es el conjunto que representa a la descripción textual S_i .

$$\tilde{D}(S_1, S_2) = \frac{\sum_{x \in T_1 \cup T_2} \min(\mu_{T_1}(x) \times \min(\mu_{T_2}(x)))}{\sum_{x \in T_1} (\mu_{T_1}(x)) + \sum_{x \in T_2} (\mu_{T_2}(x)) - \sum_{x \in T_1 \cup T_2} \min(\mu_{T_1}(x) \times \min(\mu_{T_2}(x)))} \quad (6.4)$$

$$\mu_{T_i}(x) = \begin{cases} 1 & d_E(t_i, x) = 0 \\ 0 & d_E(t_i, x) = \infty \\ \text{sigmoid}\left(\frac{1}{d_E(t_i, x)}\right) & 0 < d_E(t_i, x) < \infty \end{cases} \quad (6.5)$$

donde $d_E(t_i, x)$ es la distancia euclídea entre t_i y x

En la práctica, al determinar el grado de pertenencia de un elemento a un conjunto, estamos haciendo referencia al parecido entre los elementos de descripciones. Para ello, se ha diseñado una función de pertenencia, la cual se puede observar en la fórmula Fórmula 6.2.4. Para la definición de esta función, En primer lugar hay que determinar la medida de distancia entre dos elementos textuales procedentes de descripciones. En este caso, la distancia entre dos elementos o tokens procedentes de las descripciones se calcula como la distancia euclídea entre los vectores de los tokens de ambos conjuntos. Estos vectores corresponden a la representación numérica obtenida del modelo de Word Embedding previamente entrenado. En función del resultado que se obtenga con la distancia euclídea, se consideran tres casuísticas, las cuales se contemplan en la función de pertenencia:

1. Si el valor de distancia euclídea entre los vectores de dos palabras del vocabulario es 0, las dos palabras son idénticas, por lo que el grado de pertenencia es máximo. En nuestro caso, este valor es 1 y representa la máxima similitud. Esta situación se encuentra contemplada en el primer caso de la función de pertenencia (ver fórmula 6.4).
2. El segundo caso contemplado en la función de pertenencia (ver fórmula 6.4) hace referencia a aquellas situaciones donde el valor de distancia euclídea sea ∞ . Que se obtenga dicho valor significa que no se ha podido determinar la distancia entre elementos porque no se tiene representación vectorial de alguno de ellos, o incluso de ambos. En este caso al no poder determinarse un valor de distancia, no hay pertenencia posible y como resultado, el grado de pertenencia es 0, que representa similitud nula.
3. Si el valor de distancia euclídea se encuentra en el intervalo $(0, \infty)$, estamos en el último caso contemplado por la función de pertenencia. Puesto que estamos trabajando con grados de pertenencia, para poder acotar el problema debemos trasladar este valor de distancia al intervalo $(0,1)$. Para ello, se ha utilizado el valor de la función *sigmoide* de la inversa de la distancia. Con este cálculo, conseguimos trabajar con valores acotados en el intervalo $(0,1)$ que representen la similitud entre dos elementos.

Con esta medida, se premia a aquellos elementos que aparecen en ambos conjuntos, a la vez que se combina con el parecido existente entre los tokens que no forman parte de la intersección. Además, con la función de pertenencia utilizada se tiene en cuenta la carga semántica que contempla el modelo de Word Embedding descrito en la sección anterior, dotando así de una mayor flexibilidad a la herramienta.

6.3. Elección de la medida de distancia

Para analizar el funcionamiento de este módulo y evaluar la calidad de los resultados obtenidos con las distintas métricas, se ha estudiado el comportamiento de cada una de las medidas expuestas en la sección anterior aplicando este módulo a un problema de mapeo de datos entre dos bases de datos de composición nutricional, detallado en el Capítulo 8. El objetivo de esta tarea de mapeo de datos es estudiar cómo se comportan las distintas medidas de distancia utilizadas, y poder concluir cuál de ellas se adapta mejor al problema y es capaz de identificar elementos equivalentes con mayor precisión.

Capítulo 7

Diseño y desarrollo de la aplicación para adaptación de recetas

Este capítulo recoge el comportamiento del módulo de Consultas adaptadas, así como la descripción del prototipo de aplicación que se ha implementado para ilustrar el funcionamiento del sistema de adaptación de dietas con la herramienta de fusión de datos heterogéneos.

7.1. Recomendación de recetas

La recomendación de las recetas es un campo concreto de estudio dentro de los sistemas de recomendación en Food Computing, el cual se centra fundamentalmente en la recomendación de dietas cuyas recetas cumplen una serie de características nutricionales. Entre otras muchas especificaciones, destacan las dietas saludables o aquellas que persiguen un objetivo concreto, como puede ser la pérdida de peso o la definición de masa muscular.

Dentro del amplio campo de estudio de recomendación de recetas, destacan dos formas principales: la recomendación de recetas ya existentes, y la recomendación de recetas a partir de generación automática de recetas [17]. En el primer caso hablamos de recomendación de recetas ya existentes donde los sistemas, en base a unas preferencias concretas, realizan un filtrado hasta proporcionar la receta o conjunto de ellas que mejor se adapten a los requerimientos del usuario. Se han desarrollado una gran cantidad de sistemas de recomendación basados en recetas desde el punto de vista de Recomendación basada en el contenido, en función de la puntuación y opinión que los

usuarios tienen de los ingredientes que las forman [29, 30]. En cuanto a los sistemas de recomendación dependiente del contexto, se han desarrollado sistemas que aconsejan recetas en base al género, tiempo, aficiones, localización u otros aspectos culturales relacionados con los usuarios. También hay distintos sistemas de recomendación que tienen en cuenta distintas combinaciones de sabores, o incluso patrones de combinación de ingredientes en las distintas recetas, y otros basados en otro tipo de datos que en principio puedan parecer menos relevantes, como la rutina diaria de usuarios obtenida a partir de redes sociales como Twitter [61]. En el segundo caso hablamos de sistemas generadores de recetas automáticas: a partir del estudio de las relaciones que existen entre ingredientes, recetas y factores multiculturales en las cocinas, se pueden generar versiones de recetas que permitan mantener dicha coherencia intrínseca contenida en las recetas [40]. En este capítulo, nos centramos en este segundo caso.

Además, es relevante destacar que la incorporación de aspectos saludables en los sistemas de recomendación de dietas tiene especial importancia, y, debido a la gran cantidad de literatura centrada en la incorporación de este factor a los sistemas de recomendación, merece ser destacada en este apartado. En los últimos años, se ha producido un auge en el desarrollo de sistemas de recomendación centrados en la generación automática de dietas personalizadas teniendo como requisito que sean saludables [12, 74].

Aplicaciones móviles de recomendación de recetas

Utilizar aplicaciones móviles para resolver problemas de Food Computing ha resultado exitoso, mostrando su efectividad en ensayos focalizados a la adecuación del consumidor hacia dietas más saludables [35]. Los sistemas de Food Computing cada vez se encuentran más inmersos en las aplicaciones móviles, dando lugar a múltiples aplicaciones con funcionalidades muy diversas. El desarrollo de las comunidades de usuarios online en las que se comparten millones de recetas (AllRecipes y Yummly) ha contribuido a la disponibilidad de su información a través de aplicaciones móviles más allá de mantener su servicio vía web. Sin embargo, estos servicios están muy focalizados a la búsqueda de recetas, y no a su recomendación y adaptación a los usuarios, lo cual es una de las funcionalidades más demandadas desde estas aplicaciones [17]. En esta línea, han surgido sistemas centrados en la recomendación de recetas a través de dispositivos móviles [19, 38, 42, 49].

La falta de información (normalmente nutricional) muy concreta o específica que se suele proporcionar en las recetas procedentes de esas fuentes de datos ha originado que éstas no tenga cabida en que los sistemas de recomendación de recetas ya existentes, ya que las características de estos conjuntos de datos no son lo suficientemente detalladas como para poder

trabajar con restricciones muy específicas que no se proporcionen en ámbitos poco especializados. Es por ello que la generación y completación de recetas ha sido y es la vía de recomendación de recetas más utilizada en este ámbito, utilizando para ello las relaciones existentes entre ingredientes y recetas [24]. En este contexto podemos destacar el framework *NutRec*, cuyo motor principal en un sistema de recomendación basado en la búsqueda de recetas similares a una pseudo-receta generada automáticamente a partir de las especificaciones concretas del usuario.

7.2. Adaptación de recetas según restricciones

Como ya se ha ido introduciendo a lo largo del desarrollo de este proyecto, para ilustrar el funcionamiento del sistema desarrollado se ha implementado un módulo de Consultas Adaptadas, el cual, en nuestro caso, permite adecuar recetas en función de unas restricciones alimenticias dadas. Para ello, se permite seleccionar una receta y una restricción con el objetivo de detectar sus ingredientes y modificarlos en caso de que la información nutricional de alguno (o varios) de ellos impida su uso en algún tipo de dieta (como puede ser la vegetariana).

En este último módulo del sistema se hace uso de fuentes de datos heterogéneas que se fusionan mediante los dos módulos anteriores explicados en el Capítulo 5 y 6. De esta forma, con una única consulta podremos obtener todos los datos necesarios para que este módulo pueda funcionar correctamente. En la Figura 7.1 se visualiza esta misma idea.

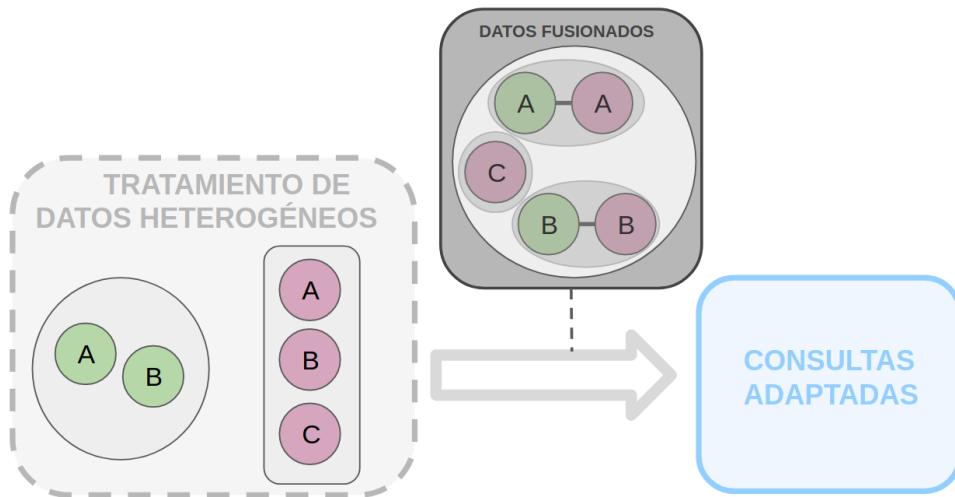


Figura 7.1: Consulta sobre los datos fusionados

Centrándonos en nuestro problema de adaptación de dietas, se utilizan

dos fuentes de datos: una primera formada por recetas, y una segunda base de datos con información nutricional, ambas detalladas en el apartado 7.3.3. Al conectar las recetas con la base de datos nutricional (a través de sus ingredientes), podremos extraer dicha información y obtener las características nutricionales de las recetas (ver Figura 7.2). De esta forma, se podrá comprobar si los ingredientes cumplen o no las restricciones impuestas (y en su caso, modificarlos por otros más adecuados). Es en este punto es donde toma relevancia el tratamiento de datos heterogéneos, que nos permite obtener información especializada que no viene incluida en los datos de las recetas.

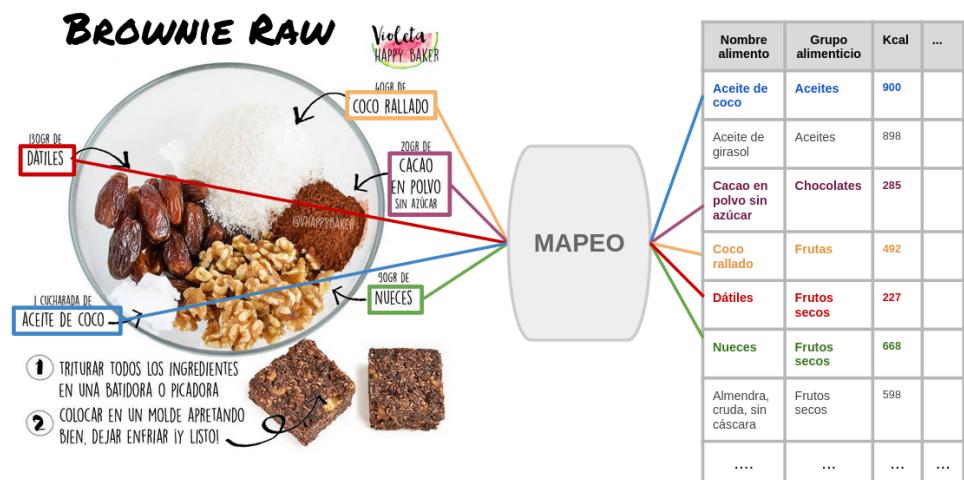


Figura 7.2: Mapeo de datos de fuentes heterogéneas

7.3. Aplicación móvil

Para ilustrar cómo funciona la solución diseñada para el problema de adaptación de recetas, se ha diseñado un prototipo funcional de aplicación móvil que permita ver su comportamiento de una forma más cómoda y realista, coherente a la línea que se sigue hoy en día con este tipo de pseudo-recomendaciones.

7.3.1. Arquitectura

En la Figura 7.3 se muestra la arquitectura global de la aplicación de adaptación de recetas. A través de la interfaz móvil, se realizan consultas al sistema de adaptación de recetas realizando consultas a la API, para obtener recetas adecuadas a las restricciones. A su vez, se hace uso de una base de datos de recetas, también accesible desde la API.

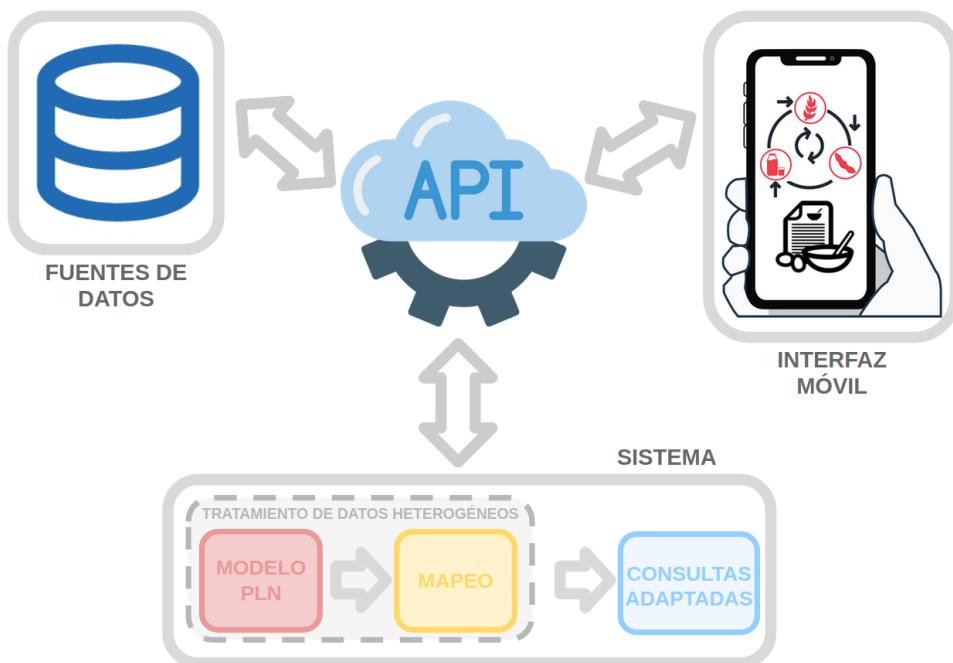


Figura 7.3: Arquitectura de la aplicación

7.3.2. Tecnologías utilizadas

A continuación se enumeran las tecnologías utilizadas para la implementación y puesta en funcionamiento del sistema descrito.

1. Para la Interfaz de Programación de Aplicaciones (API) se ha utilizado el lenguaje de programación Python con la librería Flask (<https://flask.palletsprojects.com/en/1.1.x/>).
2. Para la base de datos se ha utilizado MongoDB (www.mongodb.com/es), un sistema de base de datos no estructuradas, orientado a colecciones de documentos.
3. Para la implementación de la aplicación, se ha utilizado IONIC (<https://ionicframework.com/>), un kit de desarrollo Software de código abierto, el cual permite la creación de aplicaciones a nivel web y móvil (tanto para el sistema operativo Android como iOS).

7.3.3. Fuentes de datos

Para el funcionamiento correcto del módulo, es necesario utilizar una base de datos que permita gestionar las recetas así como una base de datos

de composición alimenticia para obtener la información nutricional de los alimentos.

Base de datos de recetas

En nuestro caso, hemos utilizado recetas procedentes de una colección de Food.com¹ proporcionada por Kaggle. Esta colección se llama *Food.com Recipes and Interactions*, que tal y como indica su nombre, está orientada al estudio de recetas e interacciones entre usuarios². Puesto que no necesitamos las interacciones de los usuarios, únicamente hemos importado el fichero correspondiente a las recetas sin preprocesar (RAW_recipes.csv³).

La elección de este conjunto de datos viene dada por el origen de las recetas que lo forman. Esta colección no contiene ninguna receta procedente de las páginas web de las que se obtiene el conjunto de recetas del corpus con el que se entrena el modelo de Word Embedding. De esta forma, aseguramos que nuestros resultados con este módulo son válidos y no están sesgados por las recetas que utilizamos, pues no se han empleado para la construcción del modelo del lenguaje. Además, el contenido de dichas recetas está en inglés (recordemos que el modelo de lenguaje implementado está en dicho idioma) e incluyen todos los atributos de recetas requeridos por el módulo.

Para gestionar la base de datos de recetas, hemos utilizado tres colecciones de datos:

- **Colección de recetas originales:** esta colección almacena todas las recetas utilizadas para el sistema de adaptación de recetas. Antes de insertar las recetas en la base de datos, hemos prescindido de aquellas columnas del conjunto de Kaggle que no nos aportan información útil, quedándonos únicamente con aquellas que sí nos interesa utilizar en el Módulo de Consultas Adaptadas. Asimismo, hemos añadido un campo *Imagen*, para mejorar la visualización en el prototipo. En la Tabla 7.1 se muestra el contenido de las recetas de esta colección.
- **Colección de recetas adaptadas:** esta colección almacena recetas ya adaptadas a una restricción mediante el Módulo de Consultas Adaptadas. Contiene la misma estructura de atributos detallada en la sección anterior (ver Tabla 7.1), con dos diferencias: en primer lugar, añade un atributo para almacenar la restricción aplicada sobre la receta y, en segundo lugar, los atributos *Ingredientes* y *Pasos en la aplicación* contienen dos nuevos campos, que almacenan si existe una modificación

¹www.food.com

²www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions

³www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions#RAW_recipes.csv

y, en caso afirmativo, de cuál se trata.

- **Colección de etiquetas de recetas:** una de las ventajas del conjunto de datos utilizado es que las recetas contienen un campo *Etiquetas* que permite realizar clasificaciones sobre ellas. En nuestro caso, hemos definido una colección de etiquetas con algunas de las utilizadas en las recetas con el fin de poder clasificarlas en base a las etiquetas que nos sean de utilidad y realizar consultas por medio de ellas. Esta colección contiene la etiqueta, con un nombre e imagen añadida por nosotros para mejorar la de visualización en el prototipo.

Atributo	Contenido	Tipo
Descripción	Descripción de la receta, consejos y algunos comentarios extra	cadena de caracteres
Imagen	Imagen de la receta	cadena de caracteres
Nombre	Nombre de la receta	cadena de caracteres
Ingredientes	Lista de ingredientes de la receta	lista de cadena de caracteres
Minutos	Tiempo de preparación de la receta en minutos	cadena de caracteres
Nutrientes	Lista con los valores nutricionales de la receta en el siguiente orden: kilocalorías, grasas totales, azúcar, sodio, proteína, grasas saturadas y carbohidratos	lista de números de coma flotante
Pasos en la preparación	Lista con los pasos para preparar la receta	lista de cadenas de caracteres
Etiquetas	Lista de etiquetas asociada a la receta (p.ej., “30 minutes or less”)	lista de cadenas de caracteres

Tabla 7.1: Descripción de los campos en la colección de recetas originales

Base de datos de Composición de Alimentos

Como se ha comentado previamente, este módulo requiere un procedimiento previo de tratamiento de datos heterogéneos de las recetas con una base de datos de composición de alimentos para obtener información nutricional de los ingredientes que forman las recetas. En este módulo se ha utilizado la Base de Datos de Composición de Alimentos i-Diet [33], que se encuentra detallada en el Capítulo 8 (Apartado 8.1.1).

La elección de i-Diet se debe principalmente a dos de sus características principales. Por un lado se trata de una base de datos de alimentos muy

depurada que además de ser utilizada por nutricionistas, se emplea en sistemas informáticos de nutrición y dietética [33]. Por otro lado, está orientada a su uso conjunto con platos (lo cual es difícil de encontrar en este ámbito), por lo que contiene una cantidad representativa de ingredientes que suelen aparecer en recetas de cocina.

7.3.4. Sistema para Adaptación de Recetas

Para poder llevar a cabo la adaptación de las recetas en base a una restricción alimenticia dada se siguen los siguientes pasos:

1. En primer lugar, se debe seleccionar una receta junto con la restricción alimenticia a aplicar, los cuales son los datos de entrada al sistema. En nuestro caso, esta receta se selecciona de entre una lista proporcionada. De esta receta, obtenemos sus ingredientes, de los cuales obtenemos sus correspondientes representaciones vectoriales con el Módulo de Procesamiento de Lenguaje Natural.
2. Una vez tenemos las representaciones vectoriales de los ingredientes se realiza un mapeo de estos alimentos hacia la base de datos de composición nutricional, para obtener información específica de cada uno de dichos alimentos. Este paso se lleva a cabo en el Módulo de Mapeo.
3. Con toda la información de los ingredientes ya disponible, realizamos una consulta para ver cuáles de ellos cumplen con las restricciones alimenticias proporcionadas (actualmente, el sistema incorpora dos restricciones a elegir: dieta vegana y dieta vegetariana). Para cada alimento que no la cumpla, se vuelve a recurrir al módulo de mapeo, para recomendar un posible alimento que pueda sustituirlo. En este paso, al usar las representaciones capaces de capturar la información semántica de los ingredientes, se posibilita que la opción proporcionada se adapte a dicha receta, puesto que este segundo mapeo se realiza únicamente sobre los elementos permitidos devolviendo el más cercano al ingrediente a modificar. Finalmente, se proporciona como la salida, la receta con los cambios correspondientes. En nuestro caso, en vez de proporcionar una única alternativa, hemos decidido facilitar varias opciones de reemplazo para cada ingrediente incompatible. Con ello, se da la posibilidad de elegir entre uno de ellos o realizar alguna modificación previa a la adaptación final de la receta.

7.3.5. Interfaz de Programación de Aplicaciones (API)

La Interfaz de Programación de Aplicaciones permite acceder a través de peticiones a las operaciones CRUD sobre las recetas, además de conectarse

al sistema de adaptación de recetas y poder realizar las adaptaciones que se requieran (ver Figura 7.3).

Operaciones CRUD

Se ha realizado el diseño e implementación de una API REST completamente funcional, que se conecta a las colecciones de la base de datos para realizar operaciones sobre los datos almacenados. Para ello, se han definido rutas con todas las consultas necesarias para el funcionamiento de la aplicación tanto a nivel de receta como de colección: lista completa de recetas, recetas que incluyan alguna etiqueta concreta, etc. Para poder realizar peticiones a nivel de receta, se ha llevado a cabo el diseño del correspondiente modelo de datos de Receta, para describir el objeto Receta y poder trabajar con la estructura de los datos en las colecciones de recetas. Este objeto Receta se utiliza para llevar a cabo las operaciones CRUD correspondientes para crear, leer, actualizar y eliminar recetas de la base de datos.

Adaptación de recetas

Se han implementado las rutas necesarias para poder realizar peticiones al Sistema de Adaptación de recetas, y así obtener recetas adaptadas acorde a alguna restricción concreta. Para ello, a través de la API se cargan los modelos predictivos necesarios para poder llevar a cabo dichas acciones.

7.3.6. Aplicación

Diseño conceptual

Previa a la implementación del prototipo que se utiliza en esta aplicación, se ha llevado a cabo el diseño conceptual de la aplicación. En primer lugar, las capacidades de la aplicación se pueden ver en el diagrama de casos que se ha obtenido para esta aplicación (ver Figura 7.4). Si nos centramos en la arquitectura de la información, se han descrito las tareas usando un diagrama de tareas HTA (ver Figura 7.5). Los posibles planes para el diagrama HTA se enumeran a continuación:

1. **Plan 0:** Si sólo quiere consultar recetas
 - a) Hacer 1-2-3-4-5-6-8 (7 opcional)
2. **Plan 1:** Si quiere adaptar recetas
 - a) Hacer 1-2-3-4-5-6-8 (7 opcional)
 - b) Si quiere guardar la receta hacer 6.3

El contenido de la información se ha descrito mediante un diagrama de conceptos (ver Figura 7.6), en el que se puede apreciar las relaciones existentes entre las recetas (originales y adaptadas) y el usuario en cuestión. Por último, se han descrito los mapas de flujo para el usuario de la aplicación móvil a través de un diagrama WireFlow que se puede ver en la Figura 7.7.

En base a estos diagramas, se puede apreciar que la funcionalidad implementada en la aplicación se corresponde con una tarea: la adaptación de recetas. Por ello, queremos resaltar que estos diagramas (y por tanto, la aplicación) podrían integrarse en el diseño de una aplicación móvil de dietética más general, en la que entre otras muchas tareas, una de las facilitadas fuera la adaptación en base a restricciones del usuario.

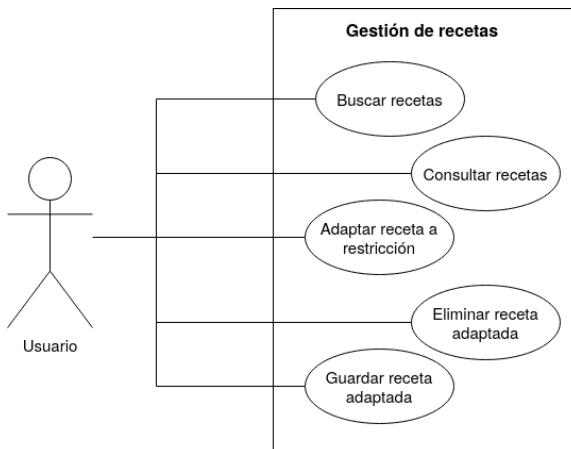


Figura 7.4: Diagrama de casos de uso de la aplicación

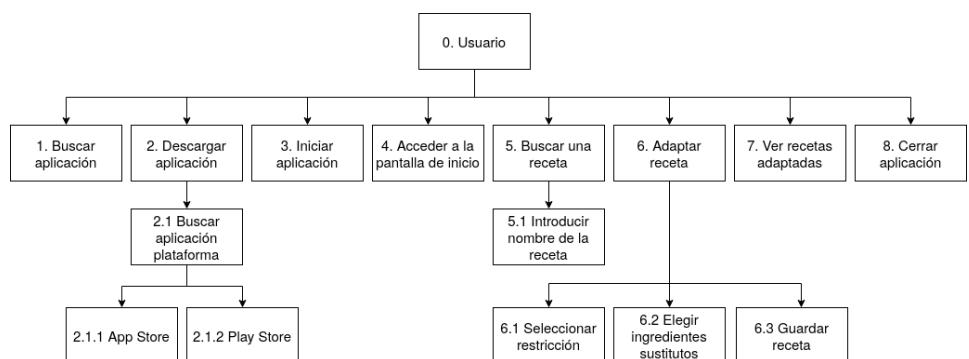


Figura 7.5: Diagrama HTA de la aplicación

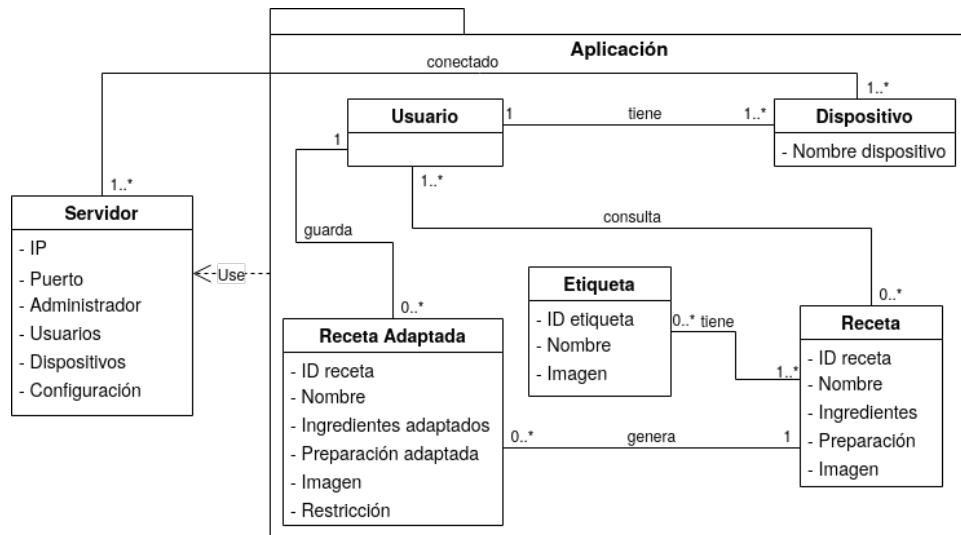


Figura 7.6: Diagrama Conceptual de la aplicación

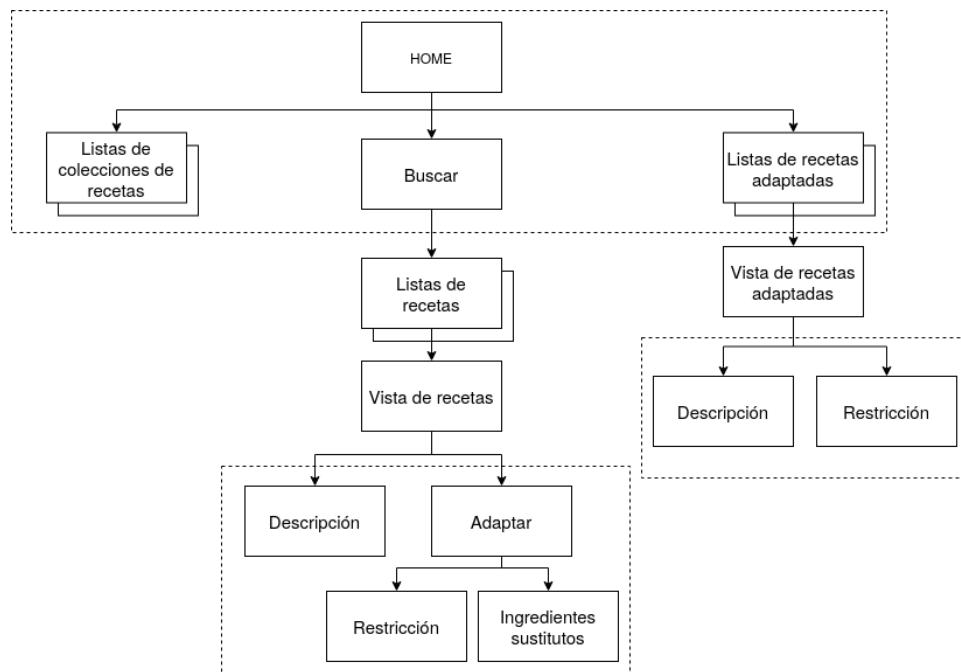


Figura 7.7: Diagrama Wireflow de la aplicación

Funcionamiento de la aplicación

En la selección de Figuras 7.8 se puede ver el comportamiento básico de la aplicación, el cual consiste en acceder a una receta y obtener su versión

adaptada. Tal y como se puede ver, a partir de la navegación por colecciones de recetas (Figura 7.8a), se puede acceder a una receta específica (Figura 7.8b) y obtener una adaptación de la misma (Figura 7.8c). Estas tres pantallas de la aplicación tan sólo forman una versión simplificada del funcionamiento de la aplicación. En el Apéndice A se puede ver de forma detallada toda la estructura de navegación en esta aplicación a través de los flujos de datos entre las distintas pantallas implementadas.

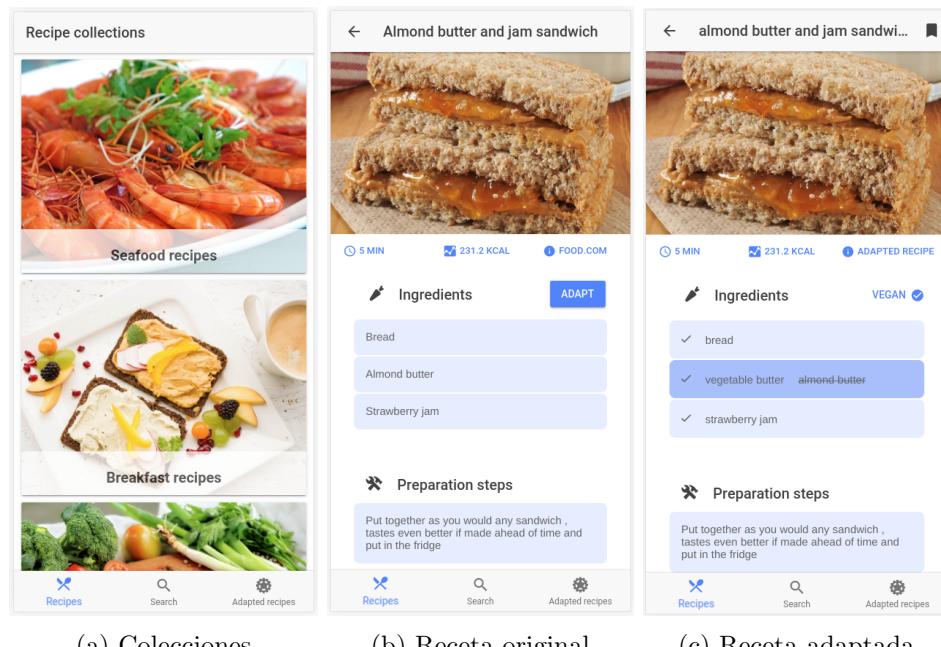


Figura 7.8: Pantallas de la aplicación: funcionamiento básico

7.3.7. Siguientes pasos en el desarrollo de la aplicación móvil

Tal y como se ha visto en la sección anterior, a pesar de que la funcionalidad básica está implementada, aún queda un amplio recorrido para obtener una versión final de la interfaz gráfica de esta aplicación móvil, ya que, tal y como se ha descrito, se encuentra en una primera iteración del proceso de desarrollo. Con este prototipo pretendemos centrarnos en la funcionalidad, no sólo con el objetivo de ver el funcionamiento de la inteligencia por debajo de la aplicación, sino también para ver las posibilidades que tienen los sistemas adaptados tal como el que se ha implementado en este trabajo.

Para continuar con el diseño de la interfaz, los pasos a seguir parten de la evaluación de la aplicación con y sin usuarios: en primer lugar una evaluación general, con lista de chequeos y los consecuentes informes de evaluación heurística, seguidos del test de usabilidad de la aplicación y finalmente, un

test de evaluación con usuarios para analizar el éxito y dificultades al interactuar con la aplicación. A través de estas evaluaciones, se podrán realizar las siguientes iteraciones, más centradas en el diseño y ultimación de las funcionalidades de la aplicación con lo que esperan los usuarios finales.

Capítulo 8

Experimentación y resultados

Este capítulo recoge la experimentación realizada en los tres módulos en los que se divide este trabajo, así como los resultados obtenidos y una breve discusión sobre los mismos.

8.1. Diseño experimental

Para llevar a cabo la experimentación del sistema que se ha desarrollado se ha optado por analizar el funcionamiento de cada uno de los módulos del sistema de manera independiente. Todos los experimentos realizados se han ejecutado de forma local, utilizando para ello mi ordenador personal. La organización de los experimentos se han llevado a cabo de la siguiente forma:

1. Para estudiar los resultados obtenidos con el modelo de lenguaje (Sección 8.2.1 de este capítulo), se ha decidido llevarlo a cabo desde el punto de vista de la visualización de las representaciones generadas con el modelo de Word Embedding. Para ello, se ha decidido representar las relaciones entre las palabras mediante la obtención de aquellas que son más similares a una dada en el vocabulario. Esta decisión se ha tomado debido a la dificultad de analizar las representaciones originales por la alta dimensionalidad de las mismas. Para ello, se ha experimentado con visualizaciones del vocabulario del modelo. Con la ayuda de estas experimentaciones, hemos ajustado los parámetros de entrenamiento del modelo para optimizar su funcionamiento. El modelo final ha sido entrenado durante 30 épocas con vectores de dimensión 300 con las palabras que aparecen 3 o más veces en el corpus de en-

trenamiento. Para el contexto de cada palabra se ha fijado el tamaño de ventana a 5. Finalmente, el modelo obtenido tiene un vocabulario de 11,288 palabras.

2. Para la experimentación con el módulo de Mapeo (Sección 8.2.2), se ha realizado una validación de las medidas de distancia implementadas para determinar cuál es la más efectiva en nuestro problema, y que por tanto, usaremos en nuestra solución. Concretamente, buscaremos las correspondencias de los elementos en la base de datos de i-Diet [33] con los elementos en USDA [32] (ya introducida en el Capítulo 3). Ambas bases de datos, se utilizan en el proyecto Stance4Health¹, donde tiene cabida esta tarea de mapeo (más información se puede encontrar en [55]). Stance4Health, tiene como objetivo desarrollar un servicio de nutrición personalizado para optimizar la actividad de la microbiota intestinal, haciendo uso para ello de múltiples bases de datos de composición nutricional de distinta procedencia, entre las que se encuentran *i-Diet* y *USDA*, ambas explicadas con mayor detalle en la Sección 8.1.1. Los mapeos obtenidos han sido contrastados con las correspondencias reales para obtener el porcentaje de acierto de cada una de las medidas de distancia. La decisión de qué medida es la más adecuada para el sistema se ha tomado en función del porcentaje de aciertos obtenido con cada una de ellas.
3. Por último, los experimentos realizados con la aplicación de adaptación de recetas (Sección 8.3) se han orientado al estudio del comportamiento de la inteligencia por debajo de la aplicación, llevada a cabo en el Módulo de Consultas Adaptadas. Para ello, se muestra de forma ejemplificada su comportamiento con resultados representativos que engloban su comportamiento general. Para este módulo no se requiere el uso de ninguna otra base de datos adicional a las utilizadas en el prototipo de la aplicación (explicadas en el Capítulo 7).

8.1.1. Bases de datos utilizadas

Base de datos de Composición Nutricional i-Diet

La base de datos i-Diet [33] es una base de datos de composición nutricional de origen español. Esta base de datos, está formada por un total de 734 alimentos y consta de 75 atributos, entre los que se encuentra su descripción en español y en inglés, grupo alimenticio y los valores correspondientes de macronutrientes y micronutrientes.

¹Stance4Health (Smart Technologies for Personalised Nutrition and Consumer Engagement) es un proyecto financiado por la Unión Europea por el programa de investigación e innovación Horizon 2020. Más información: <https://www.stance4health.com>.

ID	Descripción (ESP)	Descripción (ENG)	Grupo alimenticio	...
96	Cebolla	Onion	HORTALIZAS BULBOSAS	...
290	Manzana	Apple	FRUTAS	...

Tabla 8.1: Algunos ejemplos de la base de datos i-Diet

En la Tabla 8.1 se puede ver la estructura simplificada de esta base de datos, con dos ejemplos de alimentos contenidos en ella. En nuestro caso, para los mapeos utilizaremos la columna “*Descripción ENG*”, la cual contiene la descripción en inglés de cada alimento almacenado en la base de datos (recordemos que el modelo de lenguaje está dicho idioma).

Base de datos de Composición Nutricional USDA

La base de datos de Alimentos y Nutrientes para Estudios Dietéticos (FNDDS) del Departamento de Agricultura de los Estados Unidos (USDA)², es una base de datos de composición nutricional de referencia, creada con el objetivo de obtener los valores nutricionales a partir de las cantidades de alimentos consumidos en Estados Unidos. Esta base de datos esta formada por distintas tablas; alimentos y bebidas, nutrientes, ingredientes, valores nutricionales de ingredientes y porciones y unidades de medida de alimentos. En concreto, en este trabajo hemos utilizado únicamente una de las tablas contenidas en esta base de datos: *Nutrient Values*, la cual nos permite acceder a los datos nutricionales de una gran cantidad de alimentos consumidos en Estados Unidos. Esta tabla está formada por 8690 elementos de los que se tienen 69 atributos con la descripción en inglés, el código de la categoría de alimentos, macronutrientes y micronutrientes. En la Tabla 8.2 se muestran dos ejemplos de esta base de datos, con una estructura simplificada de la misma.

Código del alimento	Descripción alimenticia principal	Código de categoría WWEIA	Descripción de categoría WWEIA	...
75117020	Onions, mature, raw	6414	Onions	...
63101210	Apple, cooked or canned, with syrup	6002	Apples	...

Tabla 8.2: Algunos ejemplos de la base de datos USDA

²<https://data.nal.usda.gov/dataset/food-and-nutrient-database-dietary-studies-fndds>

8.2. Resultados y discusión

8.2.1. Resultados del Módulo de Procesamiento de Lenguaje Natural

A priori, comprender el comportamiento de un modelo de Word Embedding puede resultar complejo, debido a la alta dimensionalidad de los vectores numéricos que se obtienen de las palabras con las se trabaja. Por ello, se ha optado por utilizar el algoritmo de Aprendizaje Automático t-SNE (*t-Distributed Stochastic Neighbor Embedding*) para obtener visualizaciones de dichas representaciones de palabras. Con este algoritmo, se obtiene una representación bidimensional de dichos vectores, favoreciendo que aquellos que sean más similares aparezcan más cercanos en el plano, y aquellos más diferentes queden espacialmente más alejados entre ellos. De esta forma, podremos interpretar los resultados obtenidos con dicho modelo, así como analizar las posibles relaciones que puedan existir entre el conjunto de palabras contenidas en el vocabulario resultante del entrenamiento del modelo.

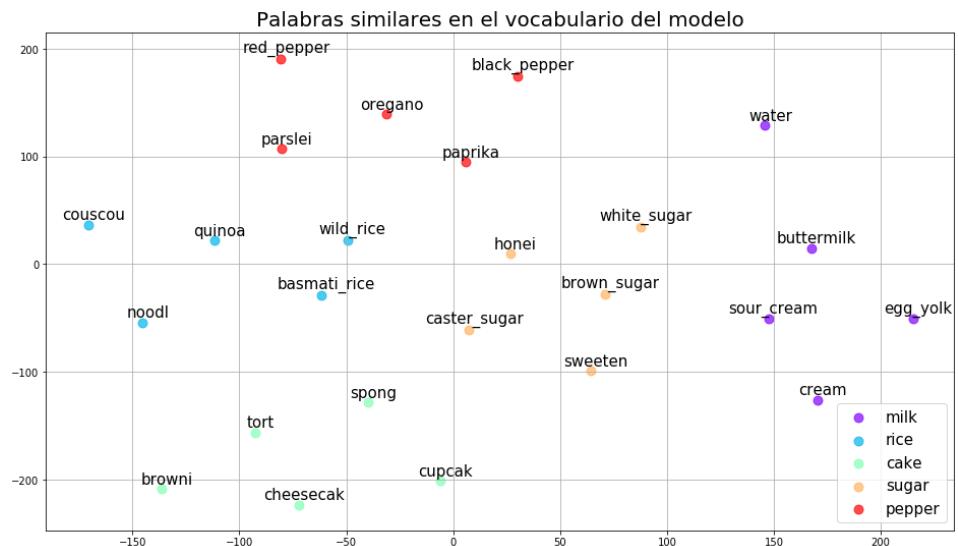


Figura 8.1: Visualización del modelo: elementos similares (ejemplo 1)

En nuestro caso, para analizar la calidad de las representaciones que obtenemos con el modelo de procesamiento de lenguaje vamos a visualizar posibles relaciones de similitud entre palabras del vocabulario del modelo con una de las principales aplicaciones de los modelos de Word Embedding: detectar elementos similares dentro del vocabulario. Esta funcionalidad viene implementada dentro de la librería de Procesamiento de Lenguaje Natural que utilizamos para el entrenamiento y utilización del modelo (*Topic Mo-*

delling Gensim³). De esta forma, para una palabra dada, podremos obtener aquellas más similares que se encuentren representadas dentro del vocabulario del modelo de lenguaje.

En la Figura 8.1 se pueden ver, para los alimentos *leche*, *arroz*, *tarta*, *azúcar* y *pimiento*, las palabras del vocabulario del modelo más similares para dichos elementos. Como se puede observar, dichas palabras se muestran en inglés tras el proceso de lematización aplicado al conjunto de entrenamiento para la creación del modelo (tal y como se explica en el Capítulo 5). En dicha figura se muestra cómo, por ejemplo, los elementos más similares a *azúcar* son azúcar moreno, azúcar blanco, endulzar, etc; o cómo para *arroz*, el modelo devuelve *arroz salvaje*, *arroz basmati*, *quinoa* o *couscous*. En este último ejemplo se aprecia la gran potencia que aportan estos modelos, donde detecta que el *couscous* es similar al arroz, llegando a ese punto únicamente a través del modelo predictivo entrenado con las recetas. Con este caso se demuestra cómo el modelo entrenado ha sido capaz de capturar la semántica de los datos con los que se ha entrenado, bajo la idea de que alimentos utilizados en contextos y forma similares, guardan también una relación de parecido. Además, debido a su gran valor en cuanto a similitud, incluso podría ser un sustitutivo del mismo, ya que el uso, preparación, cocinado y combinación con otros ingredientes en recetas es similar entre uno y otro.

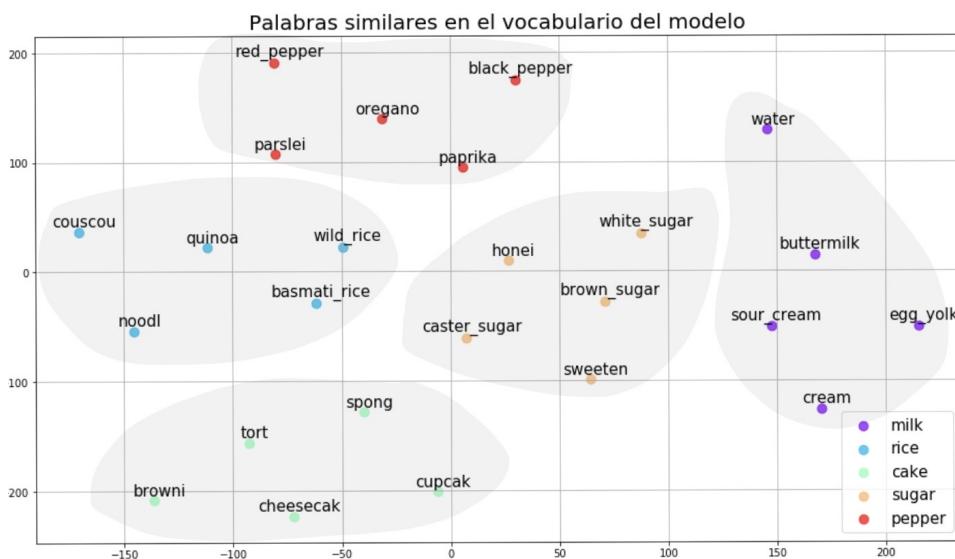


Figura 8.2: Visualización del modelo: localización espacial de los items (ej.1)

Por otra parte, si en la Figura 8.1 nos fijamos en el vocabulario más similar obtenido para *Leche*, se puede ver cómo se han obtenido como similares

³<https://radimrehurek.com/gensim/>

yema de huevo y mantequilla. Esto se puede deber a que gran parte de las recetas que incluyan el ingrediente *leche* sean postres, y aparezca acompañado o con un uso similar a los últimos mencionados. Por ello, es importante hacer ver que las representaciones obtenidas pueden estar sesgadas por las características de las recetas que se utilicen. Aquí se hace necesario utilizar grandes conjuntos de recetas, que permitan abarcar una gran cantidad de combinaciones de ingredientes, así como preparaciones de los mismos, para que el resultado sea lo más realista posible. En este caso, sí podemos concluir que se ha obtenido una buena representación, ya que otro de los elementos más similares es *agua*, lo cual nos permite ver que se están teniendo en cuenta otras características en su representación interna (como en este caso, que es similar a otros elementos del vocabulario que también son bebidas). Además, en dicha imagen se puede ver cómo los elementos similares a cada una de las descripciones textuales elegidas (nombres de alimentos) se encuentran cercanas en cuanto a su representación espacial. Este hecho se aprecia de forma más clara en la Figura 8.2, donde se han delimitado sobre la imagen aquellos elementos más similares a cada uno de los escogidos.

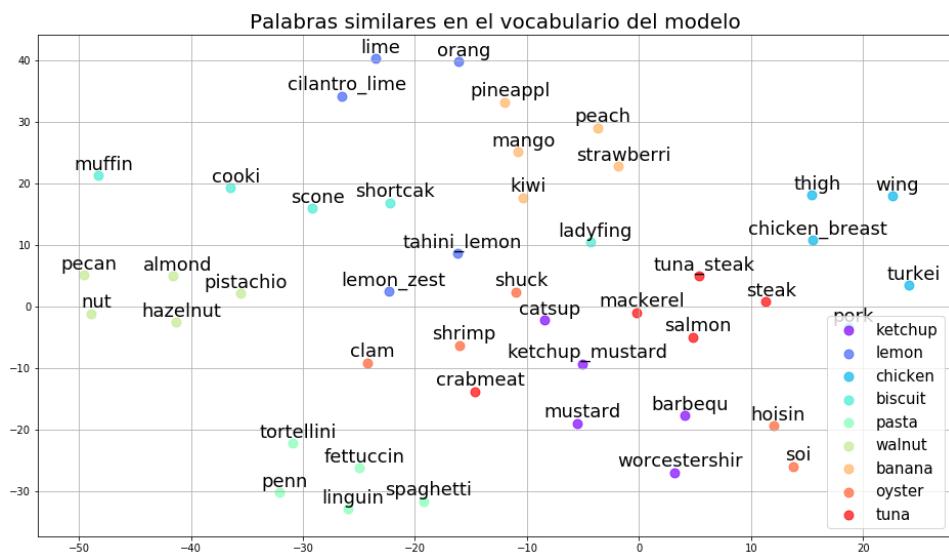


Figura 8.3: Visualización del modelo (ejemplo 2)

Si obtenemos una visualización con mayor número de alimentos, vemos que al aumentar la cantidad de elementos en la visualización no es tan sencillo encontrar una “separación espacial” tan clara como la de la Figura 8.2. Esto se debe a que estamos en un contexto en el que un ingrediente puede tener múltiples usos y en gran cantidad de contextos distintos (p.ej, el limón se utiliza en postres pero también en pescados o carnes). Este hecho se aprecia en la Figura 8.3. En esta figura podemos contrastar cómo

de nuevo, alimentos con usos muy similares son detectados por el modelo sin necesidad de ser el mismo alimento o derivado del mismo: pollo y pavo (a la derecha en azul en la figura) o atún y salmón (en rojo a la derecha), son ejemplos de ello.

Otro caso relevante en el que la semántica de los alimentos ha sido tenida en cuenta dentro del modelo se aprecia en los elementos similares obtenidos para *Limón* y para *Plátano*. Podemos ver que a pesar de que ambos, *Limón* y *Plátano*, son tipos de frutas, el modelo no se ha limitado a proporcionar como similares otras frutas, sino que en el caso del limón, ha sido capaz de representar características relacionadas con sabores: en este caso, todos los elementos más similares a *Limón* también son cítricos, mientras que en los más parecidos a *Plátano*, no hay ninguno con componente cítrico.

Con estos ejemplos, hemos podido comprobar la calidad de las representaciones obtenidas, puesto que capturan de forma precisa las semejanzas semánticas y sintácticas entre los alimentos, llegando incluso a ser capaz de representar sabores debido a las relaciones intrínsecas entre los alimentos. Estas relaciones se obtienen gracias al conjunto de entrenamiento utilizado que, al ser de instrucciones de preparación de recetas, permiten capturar información del contexto en el que se utilizan dichos alimentos, con cuáles se combinan y qué tipo de preparación y técnicas de cocinado se utilizan en ellos. Además, capturar la semántica de los alimentos en la representación del modelo de lenguaje teniendo en cuenta el uso de los mismos en el ámbito culinario no sólo permite detectar equivalentes, sino que abre una vía hacia la generación, alteración y adecuación de recetas teniendo en cuenta múltiples casuísticas.

8.2.2. Resultados del Módulo de Mapeo

Tal y como se detalla en el Capítulo 6, para analizar el funcionamiento de las distintas métricas de distancia utilizadas, se ha realizado un mapeo entre dos bases de datos nutricionales: i-Diet y USDA. Para cada elemento de i-Diet, se aplica el procedimiento de mapeo ya detallado para obtener su equivalente en USDA, utilizando las distintas medidas implementadas. En la Tabla 8.3 se pueden ver los resultados alcanzados con estas medidas de distancia. Para una mayor comprensión del modelo, hemos tenido en cuenta distintos niveles de cobertura de los resultados obtenidos. En la columna *Top 1*, se muestra el porcentaje de elementos en la base de datos de i-Diet para los que se ha obtenido el mejor mapeo posible en la base de datos USDA. En el caso de la columna *Top 2*, el valor obtenido se corresponde con el porcentaje de elementos en i-Diet cuyo mejor mapeo posible se encuentra entre los dos mejores resultados obtenidos con el procedimiento, y sucesivamente con el resto de columnas. En dicha tabla, se puede ver cómo las aproximaciones

difusas nos permiten obtener los mejores resultados, superando las medidas de distancia semántica, sintáctica, o combinación de ambas. En concreto, la medida de *Distancia difusa entre documentos* nos permite detectar mayor número de equivalencias. En los siguientes apartados se profundiza en los mapeos obtenidos con las distintas métricas para obtener una idea más concreta del comportamiento y eficacia de las mismas.

Medida de distancia	Top 1	Top 2	Top 3	Top 5	Top 10
Distancia Jaccard	16.75	20.16	22.20	25.20	27.52
Word Mover's Distance	30.65	35.55	36.92	40.87	44.82
Distancia híbrida	32.15	37.12	40.19	43.05	47.41
Distancia Jaccard difusa	23.84	29.70	33.37	39.23	45.64
Distancia entre documentos difusa	35.55	40.46	43.46	47.00	53.26

Tabla 8.3: Resultados del mapeo (%) con las medidas de distancia

Distancia sintáctica entre descripciones

Distancia de Jaccard

En la Tabla 8.4 se pueden ver algunos de los mapeos más representativos obtenidos con el procedimiento de mapeo usando la medida de distancia Jaccard. Como ya se ha introducido, esta medida de concordancia utiliza el conjunto de tokens preprocesados obtenidos a partir de cada una de las descripciones alimenticias para asignar un valor de distancia en función de los tokens que pertenecen al conjunto intersección.

Alimento a mapear (i-Diet)	Alimento mapeado (USDA)	Mejor mapeo posible (USDA)	Valor distancia
(1) Peanut oil	Peanut oil	Peanut oil	0.0 ✓
(2) Cauliflower	Cauliflower, raw	Cauliflower, raw	0.199 ✓
(3) Anchovy	Anchovy, canned	Anchovy, canned	0.125 ✓
(4) Chicory	Brioche	Chicory, beverage	0.16 ✗
(5) Avocado	Vodka	Avocado, raw	0.33 ✗
(6) Swett potatoes	Stewed potatoes	Sweet potato NFS	0.0 ✗

Tabla 8.4: Resultados obtenidos con la distancia de Jaccard

Esta concordancia a nivel de cadena de caracteres se refleja en los resultados obtenidos, ya que en las tres primeras filas se obtienen mapeos adecuados debido al parecido sintáctico entre los elementos de ambas bases de datos (las descripciones son muy similares desde el punto de vista de caracteres utilizados). De igual forma, se puede apreciar de las filas (3) a (6), puesto que los resultados del mapeo no tienen ningún tipo de relación,

más allá de la sintáctica, con los elementos mapeados (además de hacerlo con un valor de distancia mínimo, para nada representativo). A su vez, se puede ver en la fila (6) la falta de robustez con la que contamos con esta métrica, puesto que un mínimo error tipográfico en las descripciones lleva a un mapeo erróneo.

Distancia semántica entre descripciones

Distancia Word Mover's

Dado que no siempre vamos a contar con descripciones muy similares para elementos equivalentes, utilizar la semántica intrínseca en estas descripciones puede ayudar a obtener mejores resultados (tal y como se puede observar en la Tabla 8.3). En el caso de la distancia *Word Mover's*, en la Tabla 8.5 se pueden ver algunos de los mapeos más representativos obtenidos con esta métrica.

Alimento a mapear (i-Diet)	Alimento mapeado (USDA)	Mejor mapeo posible (USDA)	Valor distancia
(1) Sweet wine	Wine, dessert, sweet	Wine, dessert, sweet	13.207 ✓
(2) Tomato paste	Tomato, catsup	Tomato, catsup	15.453 ✓
(3) Pate liver not specified	Liver paste or pate, chicken	Liver paste or pate, chicken	17.555 ✓
(4) Sausage Bratwurst	Deer Sausage	Bratwurst	10.043 ✗
(5) Cocoa and hazelnut butter, Nocilla, Nutela	Almond Butter	No matches	19.028 ✗
(6) Sobrasada mallorquina	No matches	No matches	∞ ✗

Tabla 8.5: Resultados obtenidos con la distancia de Word's Mover

Además de poder resolver sin problema mapeos donde las descripciones son similares (véase fila (1) en la Tabla 8.5), también es capaz de solventar otros casos con una mayor dificultad, como puede ser el mapeo mostrado en la fila (2) de dicha tabla. En este caso podemos ver, cómo a pesar de utilizar una marca alimenticia, el mapeo se resuelve de forma correcta. Este ejemplo tiene una alta relevancia, ya que estamos mapeando bases de datos de distintas culturas culinarias y las marcas comerciales no tienen por qué coincidir en ambas zonas geográficas. Con las representaciones del modelo de Word Embedding estamos consiguiendo lidiar con esta complejidad añadida.

Otro detalle destacable es el que se expone en los ejemplos (4) y (5), donde se puede apreciar que en ambas bases de datos, existen distintos niveles de detalle en las descripciones de los alimentos, lo que añade más complejidad al problema de mapeo. Si nos fijamos en estos ejemplos, uno resulta en un mapeo correcto mientras que el otro no, y a pesar de ello, los valores de distancia obtenidos no resultan esclarecedores como para poder determinar una relación entre valor de distancia y precisión del mapeo. Aquí aparece la necesidad de lidiar con la vaguedad del lenguaje que estas diferencias de detalle provocan, para así poder aumentar la robustez y precisión de la tarea de mapeo. Tal y como se detallará más adelante en este capítulo, para lidiar con este tipo de obstáculos se hará uso de una aproximación difusa de esta medida de distancia.

Por otra parte, con el ejemplo en la fila (5) se remarca de nuevo cómo el uso de modelos de Word Embedding pueden permitir detectar posibles equivalencias, como es este caso, el cual es especialmente interesante, puesto que en USDA no hay ningún mapeo posible (debido a las diferencias culturales en la cocina). Por último, destacar el ejemplo en la fila (6), donde el elemento a mapear no está traducido al idioma en el que hemos implementado el modelo de Procesamiento de Lenguaje Natural. En este caso, esto supone un problema añadido a la hora de realizar el mapeo, porque el uso de malas traducciones empobrecen las representaciones que podamos obtener.

Distancia híbrida entre descripciones

En vista a mejorar los resultados obtenidos con las métricas previas, se ha diseñado una medida que tiene en cuenta ambas aproximaciones a través de una combinación ponderada de las medidas Jaccard y Word's Mover.

w	Top 1	Top 2	Top 3	Top 5	Top 10
(1)	16.75	20.16	22.20	25.20	27.52
(2)	30.65	35.55	36.92	40.87	44.82
(3)	25.06	29.01	30.92	33.51	37.87
(4)	25.88	29.70	31.88	35.42	39.23
(5)	26.83	30.79	33.10	36.23	41.82
(6)	27.79	31.60	33.65	39.64	44.14
(7)	29.42	34.46	36.23	41.82	46.18
(8)	30.38	35.55	37.46	42.23	46.73
(9)	31.60	36.23	38.82	43.18	47.00
(10) 0.25	32.01	37.32	38.82	42.64	47.13
(11)	32.15	37.19	40.19	43.05	46.59
(12)	32.69	37.05	40.05	42.37	46.59

Tabla 8.6: Resultados obtenidos con la distancia híbrida

En la Tabla 8.6 se puede ver la experimentación realizada con esta medida híbrida, teniendo en cuenta ambas métricas en diferentes proporciones. La prueba (10), destacada en azul, es la que en vista a los resultados, ofrece los mapeos de mejor calidad de entre todas las combinaciones.

En la Figura 8.4 se aprecia la información de la Tabla 8.6. En esta figura se muestra la eficacia de los mapeos alcanzados con la distancia híbrida en función del valor asociado al parámetro w . Tal y como se observa, la calidad de los mapeos solo mejora cuando se tiene en cuenta de forma mínima la información sintáctica obtenida con la distancia de Jaccard (ponderada por el valor del parámetro w), resultando contraproducente en la mayor parte de los resultados obtenidos. Sin embargo, los resultados obtenidos con esta medida híbrida dejan ver que ponderar la información sintáctica y semántica no obtiene resultados destacables, puesto que el aumento de los porcentajes de acierto no es muy representativo, sobre todo si los comparamos con los obtenidos con la métrica *Word Mover's Distance*.

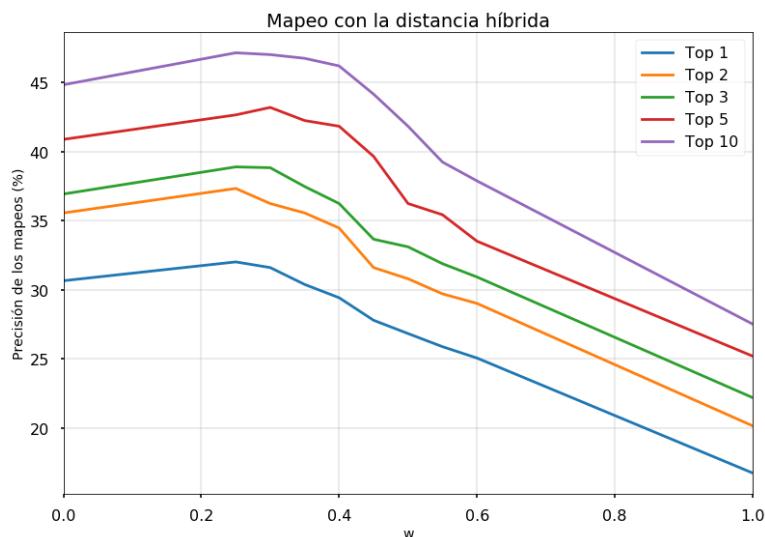


Figura 8.4: Medida de distancia híbrida: comportamiento del parámetro w

Medidas de distancia difusas

Tal y como se ha podido ver en los resultados alcanzados con las métricas anteriores, se hace necesario involucrar otras técnicas que permitan un tratamiento de la información textual capaz de lidiar con distintos niveles de detalle en las descripciones, así como con la ambigüedad intrínseca del lenguaje para poder llevar a cabo un procedimiento de mapeo más robusto. En este caso, tal y como se comentó previamente, se han obtenido versiones difusas de Jaccard y de Word Mover's.

Jaccard difuso

Como se ha podido observar en la Tabla 8.3, los resultados obtenidos con la medida de Jaccard difusa mejoran los conseguidos con la versión clásica, debido a una mayor permisividad al considerar que los elementos que se comparan no tienen por qué ser exactamente iguales a la hora de detectar equivalencias (aunque sí suficientemente parecidos). Sin embargo, a pesar de obtener mejores resultados que con la distancia de Jaccard, sigue habiendo una fuerte dependencia entre el parecido de palabras concretas entre las descripciones, como se puede apreciar en la fila (4) en la Tabla 8.7, que muestra algunos de los mapeos obtenidos con la medida difusa de Jaccard. Esto es debido en su mayoría a las implicaciones que conlleva trabajar con descripciones sintácticas a la hora de detectar posibles equivalencias.

Alimento a mapear (i-Diet)	Alimento mapeado (USDA)	Mejor mapeo posible (USDA)	Valor distancia
(1) Avocado	Avocado, raw	Avocado, raw	0.5 ✓
(2) Watercress	Watercress, raw	Watercress, raw	0 ✓
(3) Crab	Crab, cooked, NS as to cooking method	Crab, cooked, NS as to cooking method	0.333 ✓
(4) Soybean sprouts	Sprouts, NFS	Bean sprouts, raw	0.666 ✗
(5) Broccoli	Licorice	Broccoli, raw	0.16 ✗
(6) Peanut, roasted	Peanuts, honey roasted	Peanuts, roasted, salted	0.333 ✗

Tabla 8.7: Resultados obtenidos con la distancia de Jaccard difusa

Distancia difusa entre documentos

En vista a los resultados obtenidos con esta medida de distancia (ver Tabla 8.3), con esta medida es con la que alcanzamos los mejores resultados. Esto se debe mayoritariamente a que con las medidas detalladas hasta ahora, se le da mucha importancia a cada elemento considerado dentro de la descripción, ya que todas las palabras tienen la misma importancia en la descripción global. Sin embargo, si nos referimos al procesamiento de lenguaje natural, suponer que cada elemento de la descripción tiene la misma relevancia no es una buena práctica, ya que favorece a evaluar la similitud entre descripciones valorándola elemento a elemento y no como un todo. Con esta métrica le estamos dando mayor importancia al conjunto en sí, puesto que a la hora de calcular el papel que tiene cada elemento en la descripción, lo hacemos en función de su parecido semántico con la otra descripción a un nivel global. Con ello, permitimos una mayor flexibilidad a la hora de encontrar equivalencias en la base de datos.

Resultados concretos con esta medida de distancia se pueden ver en la Tabla 8.8. En términos generales, se puede observar cómo se obtiene mayor rigor con esta técnica, pues mejora mapeos erróneos vistos en apartados anteriores (como en las filas 1 y 5). Además, se ha observado un comportamiento generalizado en las correspondencias obtenidas, y es que mantiene la semántica de los elementos a mapear. Con ello nos referimos a que con la semántica capturada con el modelo somos capaces de identificar los alimentos a nivel de ingrediente o alimento principal en la descripción y las dificultades que obtenemos se deben en su mayoría a los problemas derivados de distintos niveles de detalle entre ambas bases de datos. Esto es lo ocurre con el mapeo (4) y (5) en la Tabla 8.8. Esto último es relevante en nuestro problema a resolver, puesto que esta semántica intrínseca nos permitirá detectar alimentos equivalentes entre sí cuando queramos modificar alimentos concretos de las recetas.

Alimento a mapear (i-Diet)	Alimento mapeado (USDA)	Mejor mapeo posible (USDA)	Valor distancia
(1) Sausage Bratwurst	Bratwurst	Bratwurst	0.5 ✓
(2) Chicory	Chicory beverage	Chicory beverage	0.285 ✓
(3) Chocolate and cream pudding, Chamburcy	Pie, chocolate cream	Pie, chocolate cream	0.555 ✓
(4) Swett potatoes	Potato, NFS	Sweet potato NFS	0.0 ✗
(5) Cocoa and hazelnut butter, Nocilla, Nutela	Hazelnuts	No matches	0.8 ✗
(6) Sobrasada mallorquina	No matches	No matches	∞ ✗

Tabla 8.8: Resultados obtenidos con la distancia entre documentos difusos

Por último, respecto a los valores de precisión obtenidos con esta métrica (ver 8.3), debemos tener en cuenta que los resultados son de suficientemente calidad y coherencia teniendo en cuenta que el modelo de procesamiento de lenguaje natural utilizado se basa en aprendizaje predictivo no supervisado. Para comprender la calidad de los resultados obtenidos, hay que valorar que el procedimiento de mapeo obtiene para cada uno de los elementos de i-Diet, el mejor mapeo posible de entre todos los elementos en USDA (un total de 8606), realizando el cálculo de distancia para todos los posibles emparejamientos. Si nos fijamos en las columnas *Top 10*, estas demuestran la tendencia a que a mayor flexibilidad en la medida de precisión utilizada seguimos siendo capaces de encontrar mapeos adecuados, y que el acierto en el mapeo no es fruto del azar.

Comparativa entre medidas difusas y no difusas

Tal y como se observó anteriormente, los mapeos obtenidos con las medidas de distancia Jaccard y Word's Mover son mejorados por sus respectivas versiones difusas (ver Tabla 8.3).

Si nos centramos exclusivamente en el caso de Jaccard, esta mejora se debe principalmente a que los mapeos obtenidos con la medida clásica de Jaccard se basan en la comparación íntegra de las descripciones desde un punto de vista morfológico. Con su versión difusa se valoran las descripciones con un mayor grado de flexibilidad, permitiendo así que el alimento principal que aparece en las descripciones se mantenga en los mapeos resultantes. Este hecho se puede apreciar en la Tabla 8.9, la cual recoge una comparativa de algunos mapeos obtenidos con ambas versiones (Jaccard y Jaccard difuso). En dicha tabla podemos ver cómo en las correspondencias obtenidas con la versión difusa se mantiene el concepto de *Aceite*. Sin embargo, con la versión clásica de Jaccard, se tiende a proporcionar un mapeo con el ingrediente mayoritario, lo cual no tiene que derivar en un mapeo correcto (que en este caso, sería el vegetal del que se obtiene dicho aceite).

Alimento a mapear (i-Diet)	Top 1	Top 2	Top 3	Mejor mapeo posible (USDA)
J Peanut oil	Peanut oil	Peanut, boiled	Tuna pot pie	Peanut oil ✓
\tilde{J} Peanut oil	Peanut oil	Walnut oil	Flaxseed oil	Peanut oil ✓
J Wheat germ oil	Wheat germ oil	Wheat germ, plain	Roll, whole grain white	Wheat germ oil ✓
\tilde{J} Wheat germ oil	Wheat germ oil	Sunflower oil	Sesame oil	Wheat germ oil ✓
J Sunflower oil	Sunflower oil	Sunflower seeds, NFS,	Safflower oil	Sunflower oil ✓
\tilde{J} Sunflower oil	Sunflower oil	Soybean and sunflower oil	Canola, soybean and sunflower oil	Sunflower oil ✓

Tabla 8.9: Comparación entre Jaccard (J) y Jaccard difuso (\tilde{J})

En el caso de Word Mover's, debemos valorar que la métrica computa la distancia de cada elemento en una descripción hacia el elemento más parecido en la otra descripción. Con la versión difusa que hemos diseñado inspirándonos en esta medida, valoramos el parecido de cada elemento al conjunto de la intersección generado por ambas descripciones. Esto nos permite valorar la distancia de los elementos hacia descripciones completas, para así valorar las ambigüedades y características propias del lenguaje natural.

presentes en ellas. En la Tabla 8.10 se puede apreciar el efecto de considerar la descripción de forma más global. En los ejemplos de dicha tabla, se puede ver cómo los mapeos alcanzados con la medida de distancia difusa entre documentos (\tilde{D}) proporciona resultados de más calidad.

	Alimento amapear (i-Diet)	Top 1	Top 2	Top 3	Mejor mapeo posible (USDA)	
WMD \tilde{D}	Cherry	Chips, rice	Cherries, dried	Ceviche	Cherry	✗
	Cherry	Cherries, frozen	Cherries, dried	Cobbler, cherry	Cherry	✓
WMD \tilde{D}	Garlic	Garlic, raw	Roll, garlic	Garlic, sauce	Garlic, raw	✓
	Garlic	Garlic, sauce	Garlic, sauce	Garlic, cooked	Garlic, raw	✗
WMD \tilde{D}	Lemon	Lo mein, NFS	Salmon, smoked	Lo mein, with beef	Lemon, raw	✗
	Lemon	Lemon, raw	Cassaba melon, raw	Lemon butter sauce	Lemon, raw	✓

Tabla 8.10: Comparación entre Word Mover's (WMD) y Distancia difusa entre documentos (\tilde{D})

Otras medidas de distancia

Además de las medidas de distancia ya comentadas en esta sección, también se han hecho experimentos adicionales con otras medidas de distancia. Este es el caso de la medida de distancia sintáctica Levenshtein, así como otras pruebas híbridas combinando las medidas difusas. Con estas métricas, no se han obtenido resultados relevantes que añadan detalles adicionales a lo explicado en apartados anteriores, por lo que no se entra en detalle en este capítulo.

Influencia del Word Embedding utilizado

Los resultados obtenidos con el procedimiento de mapeo también nos pueden ayudar a analizar la eficacia de utilizar un Word Embedding específico para este problema y no uno genérico ya entrenado (tal y como se mostró en el Capítulo 5). En la Tabla 8.11 se muestra el porcentaje de acierto de los mapeos obtenidos con un modelo de Word Embedding específico (el entrenado por nosotros) y uno genérico (un modelo de Word Embedding entrenado con el conjunto de datos de Google News). Para ello, se ha

utilizado la medida de distancia con mejor comportamiento (distancia de documentos difusos).

Modelo de W.E.	Top 1	Top 2	Top 3	Top 5	Top 10
W.E. Google	5.85	6.26	6.40	7.22	7.76
W.E. Recetas	35.55	40.46	43.46	47.00	53.26

Tabla 8.11: Resultados del mapeo (%) para distintos modelos de W.E.

En la Figura 8.5 se pueden ver los resultados de la Tabla 8.11. Con ella, podemos ratificar que el uso de un modelo específico en este problema es lo que nos ha llevado a obtener buenas representaciones (y por tanto mapeos) que no podríamos haber obtenido con uno genérico.

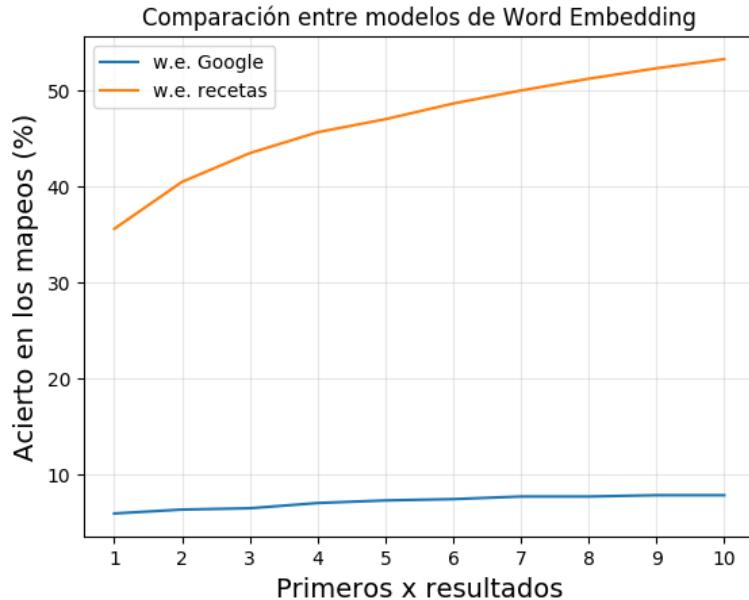


Figura 8.5: Comparación entre Word Embedding genéricos y preentrenados

8.3. Resultados del Módulo de Consultas Adaptadas

Para poder analizar los resultados obtenidos con este módulo, en este apartado se muestran algunos ejemplos representativos de adaptaciones de recetas visualizadas a través del prototipo de la aplicación. En primer lugar, vamos a ver cómo se comporta el sistema de adaptación de recetas con las restricciones incorporadas en el sistema. Para ello, se va a ilustrar con una

receta cuyos ingredientes no satisfacen ni la dieta vegetariana ni la vegana (ver Figura 8.6a). Tal y como se muestra en la Figura 8.6b, el módulo de Consultas Adaptadas es capaz de detectar aquellos ingredientes no aptos para una dieta vegetariana (en este caso, el ingrediente *Bacon*). De igual forma ocurre con la opción vegana, cuyos ingredientes no veganos son detectados también a través del módulo de Consultas Adaptadas (ver Figura 8.6c).

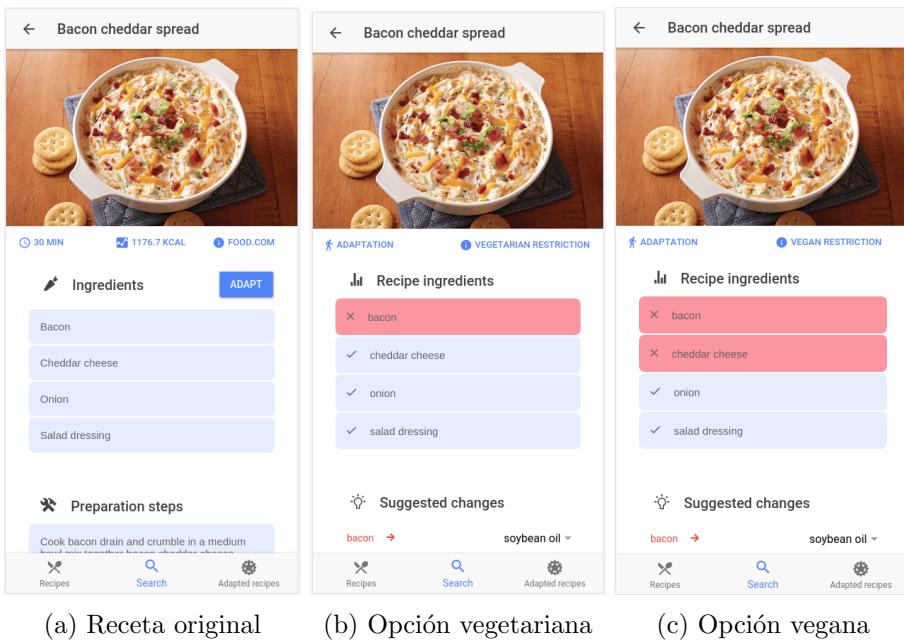


Figura 8.6: Adaptación de recetas a restricciones vegetarianas y veganas

Si nos centramos en adaptaciones a dietas veganas, en la Figura 8.7a se puede ver una receta con ingredientes no compatibles con dicha dieta. De hecho, al aplicar la restricción de dieta vegana (ver Figura 8.7b), podemos ver cómo el sistema evalúa correctamente los ingredientes al detectar aquellos no compatibles con la restricción. Si modificamos el ingrediente no compatible (en este caso es *Mantequilla*), se facilitan algunas alternativas para poder sustituir dicho ingrediente. Podemos observar en la Figura 8.7c algunas de las opciones para reemplazar la mantequilla: entre ellas, se propone la mantequilla vegetal, la cual es apta para estas dietas. Sin embargo, no siempre obtenemos adaptaciones adecuadas para las recetas. Un ejemplo de ello ocurre con la receta mostrada en la Figura 8.8a. Tal y como se detecta con el módulo de Consultas Adaptadas, esta receta contiene *Leche*, ingrediente no apto para una receta vegana (ver Figura 8.8b). Sin embargo, ninguna las alternativas propuestas es adecuada para la receta en cuestión (ver Figura 8.8c). Esto se debe a que no siempre se consigue un alimento alternativo totalmente adecuado para la receta, más aún si se trata de restricciones tan

estrictas que limitan los alimentos permitidos a un conjunto muy escaso que impide encontrar opciones que mantengan la esencia del plato.

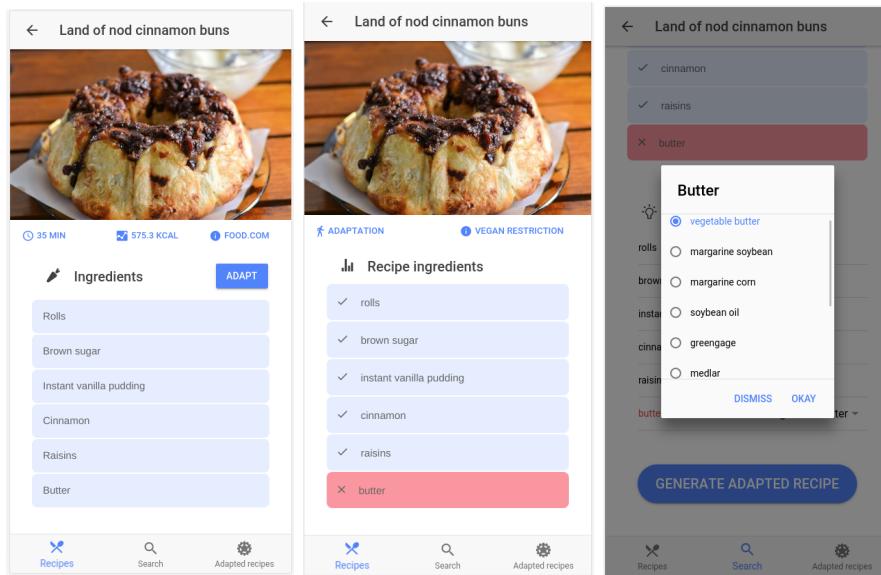


Figura 8.7: Adaptación de una receta vegana I

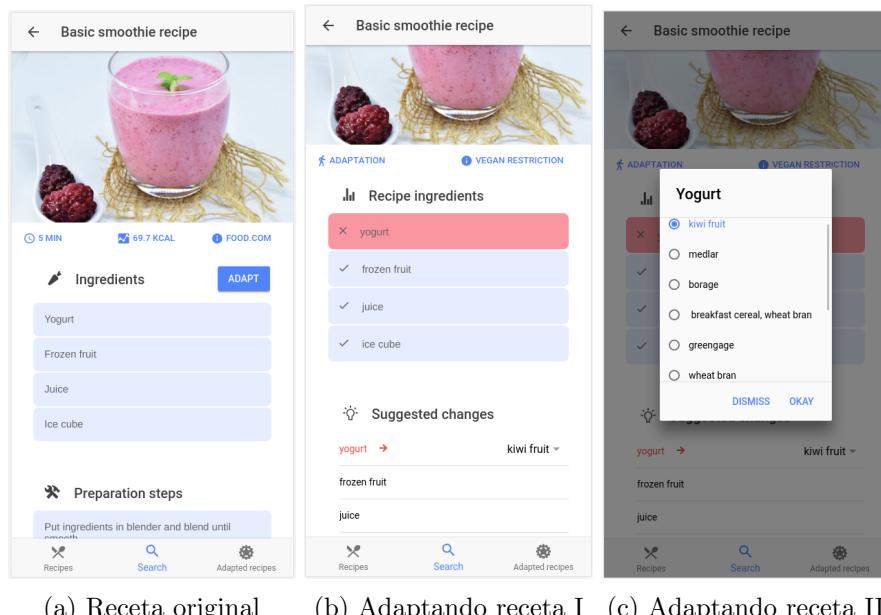


Figura 8.8: Adaptación de una receta vegana II

Por último, en las Figuras 8.9a, 8.9b y 8.9c se muestra un ejemplo de adaptación sujeta a las restricciones de la receta vegetariana. Se puede ver que los elementos que se deben adecuar a dicha dieta son detectados correctamente (en este caso el ingrediente *Pollo*), y algunas de las sugerencias para su sustitución. Se puede apreciar que las alternativas son coherentes, puesto que se sugiere utilizar verdura o incluso algún tipo de fruta en su lugar, los cuales son substitutos bastante intuitivos para esta dieta.

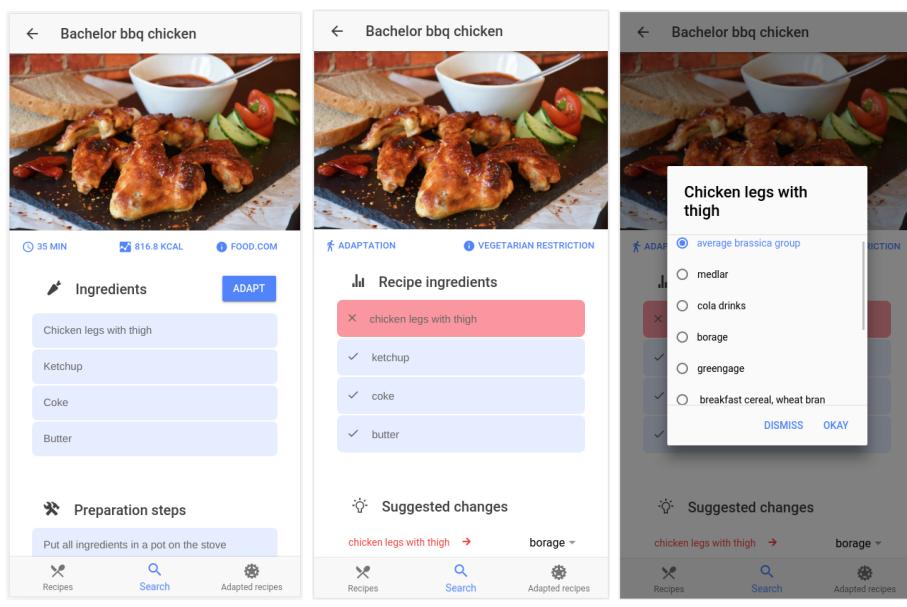


Figura 8.9: Adaptación de una receta vegetariana

Capítulo 9

Conclusiones

Este capítulo recoge los principales aspectos concluidos del desarrollo del documento y las futuras líneas de trabajo con las que continuar este proyecto.

9.1. Conclusiones

Con las representaciones obtenidas con el modelo de Word Embedding entrenado en este trabajo hemos podido ver que utilizar técnicas predictivas de este tipo para obtener representaciones textuales nos ha permitido capturar la semántica de la terminología alimenticia, dando lugar a aproximaciones más precisas que nos permitan tratar con problemas comunes en el lenguaje culinario, como el uso de marcas y sinónimos alimenticios. Al contrario que en otros modelos de Word Embedding en Food Computing, en nuestro caso no sólo utilizamos los ingredientes de las recetas, también usamos el texto que describe las instrucciones de cocción para el entrenamiento del modelo. Con ello, conseguimos obtener codificaciones cercanas para los ingredientes que aparecen juntos en las recetas, pero también para aquellos que están involucrados en preparaciones similares. Esto es útil para la comparación de elementos a nivel multicultural, y también para detectar posibles alternativas o substitutos de alimentos en base a las instrucciones de cocinado empleadas en las recetas. Sin embargo, es importante tener en cuenta que al estar utilizando recetas en el corpus de entrenamiento, corremos el riesgo de obtener muy buenas representaciones de alimentos con mucho protagonismo en el mundo culinario pero pobres o incluso inexistentes para aquellos elementos que no suelen aparecer en las recetas de este tipo de páginas web. Además, es importante valorar que, en este tipo de bases de datos nutricionales, se tienen en cuenta especies florales que no suelen ser consumidas por la población: con nuestro conjunto de entrenamiento, no

seríamos capaz de obtener representaciones para este tipo de elementos. Por ello, un conjunto de entrenamiento más amplio, capaz de abarcar alimentos más allá de las recetas, podría subsanar este posible sesgo introducido en el modelo.

Como se ha podido comprobar con las experimentaciones realizadas, las consideraciones tenidas en cuenta al definir medidas que permitan detectar equivalencias cambian totalmente el comportamiento del módulo de mapeo. Por ello, se han valorado distintos enfoques abarcando tanto información sintáctica y semántica como la vaguedad en el lenguaje que pueda existir en las fuentes de datos utilizadas. Aun así, la similitud entre textos cortos sigue siendo un desafío debido a la dificultad añadida de tratar con su breve contenido, donde se incrementa mucho la importancia que tiene cada palabra perteneciente a la descripción. A pesar de que la información sintáctica y semántica tienen relevancia en este punto (y deben ser tenidas en cuenta en los mapeos) el uso de una métrica que permita darle más importancia a la descripción en sí que a los elementos concretos que forman parte de la misma ha sido determinante para ser capaces de detectar equivalencias entre las bases de datos. No obstante, en los casos de mapeos incorrectos, los resultados han seguido mostrando la calidad y robustez de la herramienta, permitiendo detectar buenas aproximaciones e ingredientes principales. Por otra parte, al analizar los mejores posibles mapeos para cada elemento de i-Diet, hemos podido darnos cuenta cómo no sólo el mejor mapeo posible es de calidad, sino que las 10 primeras alternativas siguen mostrando una mejora muy sustancial en los resultados. Estos mejores mapeos obtenidos con nuestra estrategia, podrían ser combinados con otro tipo de técnicas, o incluso enriquecidos con mayor información del conjunto de datos con el objetivo de alcanzar resultados más exactos.

Con el prototipo implementado para la aplicación de recetas adaptadas hemos podido ver la potencia que podría tener esta aplicación más allá de una solución a un problema de mapeo. Se han podido ver las ventajas de utilizar el modelo predictivo implementado para poder adaptar dietas, puesto que la semántica capturada en el modelo permite alcanzar buenas alternativas a los elementos restringidos. Por otra parte, se ha permitido ver la interpretabilidad de este módulo, ya que paso a paso se indican aquellos ingredientes no adecuados y que deben ser sustituidos por otros. Esta interpretabilidad nos ha permitido percarnos de la dificultad de algunas adaptaciones, en las que las restricciones limitan las posibilidades de mapeo a alimentos que son difícilmente combinables con los ingredientes de la receta. Esto no deja de ser un problema fuera del ámbito de la computación, ya que la versión vegana o vegetariana de un plato no tiene por qué ser una solución intuitiva de por sí.

En los capítulos correspondientes a cada uno de los módulos implemen-

tados, así como en la experimentación llevada a cabo con ellos, se ha podido apreciar que se han tenido en cuenta con éxito los requisitos definidos en el Capítulo 2. Sin embargo, dada la etapa de desarrollo en la que se encuentra la aplicación móvil (como comentamos en el Capítulo 7 se trata de un prototipo funcional), el nivel de cumplimiento de los requerimientos relativos a la interfaz podría mejorarse con la realimentación que se obtiene de las pruebas y evaluación con usuarios en etapas posteriores de desarrollo.

Por último, puntualizar que las distintas tareas de Food Computing involucradas en el problema que hemos abordado en este trabajo (Predicción, Recomendación, Identificación y Recuperación de información) son tareas generales que se pueden encontrar en otros campos de estudio que no tienen por qué estar centrados en el ámbito culinario y de la nutrición. Al tratarse de tareas comúnmente abordadas en cualquier campo, justifican aún más la independencia del sistema desarrollado al área de Food Computing, posibilitando su aplicación en otros muchos contextos. Con ello, queremos destacar la relevancia de esta aproximación para resolver problemas que lidien con información heterogénea, puesto que todo el procedimiento puede ser aplicable a otro campo de estudio, cambiando las fuentes de datos y el modelo de Word Embedding por uno específico en el área en cuestión o incluso por uno genérico si la situación lo propiciase.

9.2. Trabajo futuro

A lo largo del trabajo, hemos podido ver las grandes posibilidades que brindan los modelos predictivos a la hora de trabajar con modelos de lenguaje. En este punto se abren nuevas vías que pretendemos estudiar para poder obtener un modelo del lenguaje más sofisticado. Como es de esperar en un problema de predicción, una vía de mejora reside en la calidad de los datos utilizados. Las dificultades que nos hemos encontrado a causa de traducciones inexactas e incorrectas en los datos han derivado en representaciones del lenguaje imprecisas. Una mejora de las traducciones de los alimentos, así como de errores tipográficos, permitiría reducir el ruido añadido, y obtener mejores resultados: cuanta mayor calidad tengan los datos textuales con los que trabajemos, con mayor facilidad encontraremos buenas representaciones con el modelo. Por ello, también planteamos el uso de técnicas de Traducción Automática (Machine Translation) para mejorar la calidad de estos y así reducir el ruido añadido en los datos. Además, aumentar la cantidad de datos alimenticios abarcados en el conjunto de entrenamiento para el Word Embedding nos permitirá obtener un vocabulario más amplio. Para ello, estudiaremos obtener un corpus de entrenamiento más extenso más allá de la información que nos puedan proporcionar las recetas, para así poder recoger tanto vocabulario alimenticio como nos sea posible. Además

de aumentar el tamaño del corpus de entrenamiento, como trabajo futuro se plantea utilizar otros modelos de lenguaje más actuales como BERT [25], así como aplicar Trasferencia de Aprendizaje [57] (Transfer Learning) con el modelo ya implementado con el objetivo de obtener representaciones aún más precisas que nos permitan convertir los mapeos aproximados (aunque erróneos) en correctos.

Por otra parte, trabajar con modelos de Word Embedding en Food Computing nos ha llevado a comprobar la importancia de la influencia cultural existente al utilizar las distintas cocinas del mundo. Esta dificultad abre nuevas vías de estudio con modelos de lenguaje en Food Computing, tales como abordar problemas de modelos de Word Embedding multilingües, más conocidos como *Cross-Lingual Word Embedding* [63] en vistas a poder lidiar de forma más sofisticada con las características culturales contenidas en los datos con los que hemos trabajado en este proyecto. Con ello, podríamos detectar posibles transformaciones lineales de los datos alimenticios entre zonas geográficas y abordar de forma más sofisticada las dificultades intrínsecas al trabajar con bases de datos nutricionales procedentes de distintas culturas.

A su vez, confiamos en que transformar este problema en uno multimodal que tenga en cuenta el resto de información contenida acerca de cada alimento (como puede ser el grupo de alimento o incluso los macronutrientes) nos permita obtener un procedimiento de mapeo más robusto, así como una mayor precisión en los resultados. La información que acompaña a los alimentos nos puede ayudar a mejorar la calidad de los mapeos, ya que estaríamos teniendo en cuenta descripciones más completas de los elementos a mapear. Para ello, pretendemos estudiar el problema desde dos perspectivas: en primer lugar, considerar un modelo predictivo que combine la representación vectorial de las descripciones con el resto de datos asociados a cada elemento; y en segundo lugar, considerar estos datos adicionales a la descripción textual en la función de distancia utilizada para medir la similitud entre los alimentos.

Bibliografía

- [1] European Food Safety Authority (EFSA). “The food classification and description system FoodEx 2 (revision 2)”. En: *EFSA Supporting Publications* 12.5 (2015), 804E.
- [2] Sofiane Abbar, Yelena Mejova e Ingmar Weber. “You tweet what you eat: Studying food consumption through twitter”. En: *Conference on Human Factors in Computing Systems - Proceedings* 2015-April (2015), págs. 3197-3206. DOI: 10.1145/2702123.2702153. arXiv: arXiv:1412.4361v2.
- [3] Farida Chowdhury Adnan Ahmad Md Amin. “Bengali Document Clustering Using Word Movers Distance”. En: sep. de 2018, págs. 1-6. DOI: 10.1109/ICBSLP.2018.8554598.
- [4] Yong-Yeol Ahn y col. “Flavor network and the principles of food pairing”. En: *Scientific reports* 1 (2011), pág. 196.
- [5] S. Albukhitan y T. Helmy. “Multilingual Food and Health Ontology Learning Using Semi-Structured and Structured Web Data Sources”. En: *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*. Vol. 3. 2012, págs. 231-235.
- [6] Felipe Almeida y Geraldo Xexéo. *Word Embeddings: A Survey*. 2019. arXiv: 1901.09069 [cs.CL].
- [7] Jaan Altosaar. *Augmented cooking with machine intelligence*. URL: <https://jaan.io/food2vec-augmented-cooking-machine-intelligence/>.
- [8] Albert-László Barabási, Giulia Menichetti y Joseph Loscalzo. “The unmapped chemical complexity of our diet”. En: *Nature Food* (2019), págs. 1-5.
- [9] Riccardo Bellazzi y col. “Mining biomedical time series by combining structural analysis and temporal abstractions”. En: *Proceedings / AMIA ... Annual Symposium. AMIA Symposium* 1 (feb. de 1998), págs. 160-4.
- [10] Piotr Bojanowski y col. “Enriching word vectors with subword information. CoRR abs/1607.04606 (2016)”. En: *arXiv preprint arXiv:1607.04606* (2016).

- [11] Lukas Bossard, Matthieu Guillaumin y Luc Van Gool. “Food-101 - Mining discriminative components with random forests”. En: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8694 LNCS.PART 6 (2014), págs. 446-461. ISSN: 16113349. DOI: 10.1007/978-3-319-10599-4_29.
- [12] Sergio Pérez Burilo y col. “Nutrición personalizada inteligente”. En: *Alimentaria: Revista de tecnología e higiene de los alimentos* 500 (2019), págs. 25-29.
- [13] BuzzFeed. *Recipe2Vec: How Word2Vec helped us discover related Tasty recipes*. URL: <https://pydata.org/nyc2017/schedule/presentation/65/>.
- [14] Jose Camacho-Collados y Mohammad Taher Pilehvar. “From word to sense embeddings: A survey on vector representations of meaning”. En: *Journal of Artificial Intelligence Research* 63 (2018), págs. 743-788.
- [15] Minsuk Chang y col. “RecipeScape: An Interactive Tool for Analyzing Cooking Instructions at Scale”. En: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. 2018, pág. 451.
- [16] Chih-Yu Chao y col. “Dish Discovery via Word Embeddings on Restaurant Reviews.” En: *RecSys Posters*. 2016.
- [17] Meng Chen y col. “Eating healthier: Exploring nutrition information for healthier recipe recommendation”. En: *Information Processing & Management* (2019), pág. 102051.
- [18] Yuzhe Chen. “A Statistical Machine Learning Approach to Generating Graph Structures from Food Recipes”. En: August (2017).
- [19] Teh Lee Cheng, Umi Kalsom Yusof y Mohd Nor Akmal Khalid. “Content-based filtering algorithm for mobile recipe application”. En: *2014 8th Malaysian Software Engineering Conference (MySEC)*. IEEE. 2014, págs. 183-188.
- [20] SM Church. “EuroFIR synthesis report No 7: Food composition explained”. En: *Nutrition Bulletin* 34.3 (2009), págs. 250-272.
- [21] Diofanor Acevedo Correa, Piedad Margarita Montero Castillo y Raul Jose Martelo. “Neural networks in food industry”. En: *Contemporary Engineering Sciences* 11.37 (2018), págs. 1807-1826. ISSN: 13136569. DOI: 10.12988/ces.2018.84141.
- [22] Michael Crowe y col. “Data Mapping From Food Diaries to Augment the Amount and Frequency of Foods Measured Using Short Food Questionnaires”. En: *Frontiers in Nutrition* 5 (2018), pág. 82. ISSN: 2296-861X. DOI: 10.3389/fnut.2018.00082. URL: <https://www.frontiersin.org/article/10.3389/fnut.2018.00082>.

- [23] Santa D’Innocenzo, Carlotta Biagi y Marcello Lanari. “Obesity and the Mediterranean Diet: A Review of Evidence of the Role and Sustainability of the Mediterranean Diet”. En: *Nutrients* 11.6 (2019), pág. 1306.
- [24] Marlies De Clercq y col. “Data-driven recipe completion using machine learning methods”. En: *Trends in Food Science & Technology* 49 (2016), págs. 1-13.
- [25] Jacob Devlin y col. “Bert: Pre-training of deep bidirectional transformers for language understanding”. En: *arXiv preprint arXiv:1810.04805* (2018).
- [26] Damion M Dooley y col. “FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration”. En: *npj Science of Food* 2.1 (2018), págs. 1-10.
- [27] Yadan Fan y col. “Using word embeddings to expand terminology of dietary supplements on clinical notes”. En: *JAMIA Open* 2 (mar. de 2019), págs. 246-253. ISSN: 2574-2531. DOI: 10.1093/jamiaopen/ooz007. eprint: <http://oup.prod.sis.lan/jamiaopen/article-pdf/2/2/246/28854866/ooz007.pdf>. URL: <https://doi.org/10.1093/jamiaopen/ooz007>.
- [28] Mamdouh Farouk. “Measuring Sentences Similarity: A Survey”. En: *ArXiv* (2019).
- [29] Jill Freyne y Shlomo Berkovsky. “Intelligent Food Planning: Personalized Recipe Recommendation”. En: *Proceedings of the 15th International Conference on Intelligent User Interfaces*. IUI ’10. Hong Kong, China: ACM, 2010, págs. 321-324. ISBN: 978-1-60558-515-4. DOI: 10.1145/1719970.1720021. URL: <http://doi.acm.org/10.1145/1719970.1720021>.
- [30] Jill Freyne y col. “Personalized techniques for lifestyle change”. English. En: *Artificial Intelligence in Medicine in Europe*. Ed. por Mor Peleg, Nada Lavrac y Carlo Combi. Lecture Notes in Computer Science. United States: Springer, Springer Nature, 2011, págs. 139-148. ISBN: 9783642222177. DOI: 10.1007/978-3-642-22218-4_18.
- [31] Daniel Fried y col. “Analyzing the language of food on social media”. En: *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014 Section II* (2015), págs. 778-783. DOI: 10.1109/BigData.2014.7004305. arXiv: [arXiv:1409.2195v2](https://arxiv.org/abs/1409.2195v2).
- [32] Susan Gebhardt y col. “USDA national nutrient database for standard reference, release 21”. En: *United States Department of AgricultureAgricultural Research Service* (2008).

- [33] S.L. Gestión de Salud y Nutrición. “i-Diet Food Composition Database, updated from original version of G. Martín Peña FCD”. En: (2019).
- [34] Morgan Harvey, Bernd Ludwig y David Elsweiler. “You Are What You Eat: Learning User Tastes for Rating Prediction”. En: *String Processing and Information Retrieval*. Ed. por Oren Kurland, Moshe Lewenstein y Ely Porat. Cham: Springer International Publishing, 2013, págs. 153-164. ISBN: 978-3-319-02432-5.
- [35] Michelle L Ijpjian y Carol S Johnston. “Smartphone technology facilitates dietary change in healthy adults”. En: *Nutrition* 33 (2017), págs. 343-347.
- [36] Jayne D Ireland y A Møller. “Langual food description: a learning process”. En: *European journal of clinical nutrition* 64.3 (2010), S44-S48.
- [37] Gordana Ispirova y col. “Mapping Food Composition Data from Various Data Sources to a Domain-Specific Ontology”. En: nov. de 2017. DOI: 10.5220/0006504302030210.
- [38] Thienne Johnson y col. “A mobile food recommendation system based on the traffic light diet”. En: *arXiv preprint arXiv:1409.0296* (2014).
- [39] Michael N Jones y Douglas JK Mewhort. “Representing word meaning and order information in a composite holographic lexicon.” En: *Psychological review* 114.1 (2007), pág. 1.
- [40] Masahiro Kazama y col. “A Neural Network System for Transformation of Regional Cuisine Style”. En: *Frontiers in ICT* 5 (2018), pág. 14. ISSN: 2297-198X. DOI: 10.3389/fict.2018.00014. URL: <https://www.frontiersin.org/article/10.3389/fict.2018.00014>.
- [41] Tom Kenter y Maarten De Rijke. “Short text similarity with word embeddings”. En: *International Conference on Information and Knowledge Management, Proceedings* (2015), págs. 1411-1420. DOI: 10.1145/2806416.2806475.
- [42] Hathairat Ketmaneechairat, Chutima Kongketwanich y Thitinun Naijit. “Recommender system for thai food cooking on smartphone”. En: *2017 Twelfth International Conference on Digital Information Management (ICDIM)*. IEEE. 2017, págs. 169-178.
- [43] Matt Kusner y col. “From word embeddings to document distances”. En: *International conference on machine learning*. 2015, págs. 957-966.
- [44] Guy Lev, Benjamin Klein y Lior Wolf. “In Defense of Word Embedding for Generic Text Representation”. En: jun. de 2015, págs. 35-50. ISBN: 978-3-319-19580-3. DOI: 10.1007/978-3-319-19581-0_3.
- [45] Weizhuo Li y col. “Multi-view Embedding for Biomedical Ontology Matching”. En: *Ontology Matching* (2019), pág. 13.

- [46] Yang Li y Tao Yang. “Word embedding for understanding natural language: a survey”. En: *Guide to Big Data Applications*. Springer, 2018, págs. 83-104.
- [47] Ascension Lupiáñez-Barbero, Cintia Blanco y Alberto De Leiva. “Spanish food composition tables and databases: Need for a gold standard for healthcare professionals (review)”. En: *Endocrinología, Diabetes y Nutrición (English ed.)* (jul. de 2018). DOI: 10.1016/j.endien.2018.05.011.
- [48] Javier Marin y col. “Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images”. En: *IEEE transactions on pattern analysis and machine intelligence* (2019).
- [49] Takuma Maruyama, Yoshiyuki Kawano y Keiji Yanai. “Real-time mobile recipe recommendation system using food ingredient recognition”. En: *Proceedings of the 2nd ACM international workshop on Interactive multimedia on mobile and portable devices*. 2012, págs. 27-34.
- [50] Tomas Mikolov y col. “Distributed representations of words and phrases and their compositionality”. En: *Advances in neural information processing systems*. 2013, págs. 3111-3119.
- [51] Tomas Mikolov y col. “Efficient estimation of word representations in vector space”. En: *arXiv preprint arXiv:1301.3781* (2013).
- [52] Weiqing Min y col. “A survey on food computing”. En: *ACM Computing Surveys (CSUR)* 52.5 (2019), págs. 1-36.
- [53] Weiqing Min y col. “You Are What You Eat: Exploring Rich Recipe Information for Cross-Region Food Analysis”. En: *IEEE Transactions on Multimedia* 20.4 (2018), págs. 950-964. ISSN: 15209210. DOI: 10.1109/TMM.2017.2759499.
- [54] Weiqing Min y col. “You Are What You Eat: Exploring Rich Recipe Information for Cross-Region Food Analysis”. En: *IEEE Transactions on Multimedia* 20.4 (2018), págs. 950-964. ISSN: 15209210. DOI: 10.1109/TMM.2017.2759499.
- [55] Andrea Morales-Garzón, Juan Gómez-Romero y María J Martin-Bautista. “A Word Embedding Model for Mapping Food Composition Databases Using Fuzzy Logic”. En: *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer. 2020, págs. 635-647.
- [56] Ferda Ofli y col. “Is saki# delicious? the food perception gap on instagram and its relation to health”. En: *Proceedings of the 26th International Conference on World Wide Web*. 2017, págs. 509-518.
- [57] Sinno Jialin Pan y Qiang Yang. “A survey on transfer learning”. En: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), págs. 1345-1359.

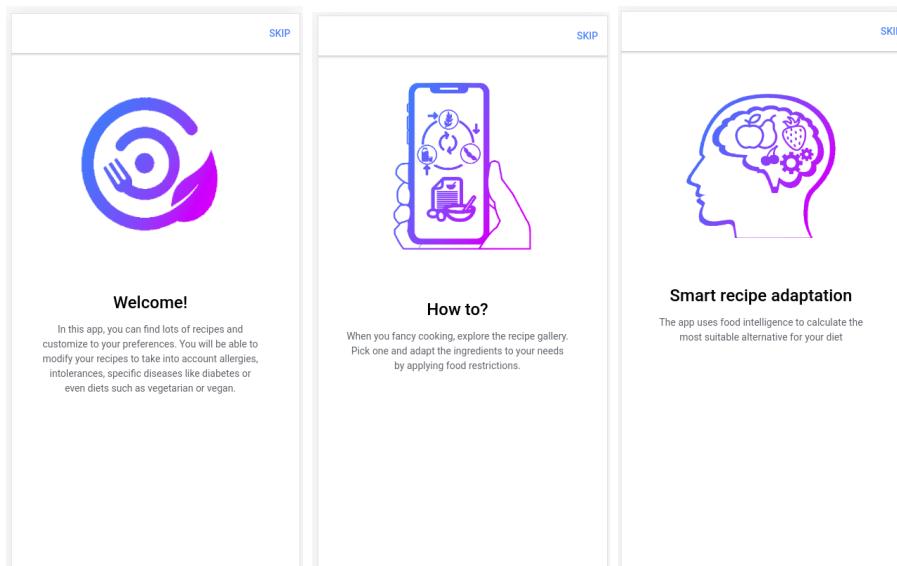
- [58] Jeffrey Pennington, Richard Socher y Christopher D. Manning. “GloVe: Global Vectors for Word Representation”. En: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, págs. 1532-1543. URL: <http://www.aclweb.org/anthology/D14-1162>.
- [59] Pratibha Phaiju. “Towards Food Security through Artificial Neural Network”. En: *Journal of Science and Engineering* 6 (mayo de 2019), págs. 71-77. DOI: 10.3126/jisce.v6i0.23968.
- [60] Nancy Raper y col. “An overview of USDA’s dietary intake data system”. En: *Journal of food composition and analysis* 17.3-4 (2004), págs. 545-555.
- [61] Markus Rokicki y col. “Plate and Prejudice: Gender Differences in Online Cooking”. En: *UMAP*. 2016.
- [62] Xin Rong. “word2vec parameter learning explained”. En: *arXiv preprint arXiv:1411.2738* (2014).
- [63] Sebastian Ruder, Ivan Vulić y Anders Søgaard. “A survey of cross-lingual word embedding models”. En: *Journal of Artificial Intelligence Research* 65 (2019), págs. 569-631.
- [64] Hassan Saif y col. “On stopwords, filtering and data sparsity for sentiment analysis of twitter”. En: (2014).
- [65] Sina Sajadmanesh y col. “Kissing cuisines: Exploring worldwide culinary habits on the web”. En: *26th International World Wide Web Conference 2017, WWW 2017 Companion* (2019), págs. 1013-1021. DOI: 10.1145/3041021.3055137. arXiv: [arXiv:1610.08469v4](https://arxiv.org/abs/1610.08469v4).
- [66] Amaia Salvador y col. “Learning Cross-Modal Embeddings for Cooking Recipes and Food Images”. En: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Jul. de 2017, págs. 3068-3076. DOI: 10.1109/CVPR.2017.327.
- [67] Christopher R. Sauer y Alex Haigh. “Cooking up Food Embeddings Understanding Flavors in the Recipe-Ingredient Graph”. En: 2017.
- [68] IQ Sayed. “Issues in Anaphora Resolution”. En: (ene. de 2003).
- [69] D Sheehan y col. “The European Food Information Resource Network (EuroFIR) internet-deployed database (EuroFIR BASIS): an online composition and biological effects database of plant-based bioactive compounds”. En: *Proceedings of the Nutrition Society* 67.OCE7 (2008).
- [70] Rishabh Singh e Himanshu Arora. “CSE 255 Assignment 2 Cuisine Prediction / Classification based on ingredients”. En: 2015.
- [71] Chakkrit Snae y Michael Bruckner. “FOODS: a food-oriented ontology-driven system”. En: *2008 2nd IEEE International Conference on Digital Ecosystems and Technologies*. IEEE. 2008, págs. 168-176.

- [72] Christian Sternitzke e Isumo Bergmann. “Similarity measures for document mapping: A comparative study on the level of an individual scientist”. En: *Scientometrics* 78.1 (2009), págs. 113-130.
- [73] Jun Takahashi y col. “Implementation of automatic nutrient calculation system for cooking recipes based on text analysis”. En: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2012), págs. 789-794. ISSN: 03029743. DOI: 10.1007/978-3-642-32695-0-74.
- [74] Christoph Trattner y David Elsweiler. “Food Recommender Systems: Important Contributions, Challenges and Future Research Directions”. En: November (2017). arXiv: 1711.02760. URL: <http://arxiv.org/abs/1711.02760>.
- [75] Jiannan Wang, Guoliang Li y Jianhua Fe. “Fast-join: An efficient method for fuzzy token matching based string similarity join”. En: *2011 IEEE 27th International Conference on Data Engineering*. IEEE. 2011, págs. 458-469.
- [76] John Wieting y col. “Charagram: Embedding words and sentences via character n-grams”. En: *arXiv preprint arXiv:1607.02789* (2016).
- [77] M. Won y J. Lee. “Embedding for Out of Vocabulary Words Considering Contextual and Morphosyntactic Information”. En: *2018 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*. 2018, págs. 212-215.
- [78] Tan Yong-mei, Wang Min-da y Niu Shao-zhang. “Chinese Textual Entailment Recognition Via Ordered Word Mover Distance”. En: *Beijing Youidian Daxue Xuebao/Journal of Beijing University of Posts and Telecommunications* (oct. de 2017), págs. 123-128. DOI: 10.13190/j.jbupt.2016-221.
- [79] Li Yujian y Liu Bo. “A normalized Levenshtein distance metric”. En: *IEEE transactions on pattern analysis and machine intelligence* 29.6 (2007), págs. 1091-1095.
- [80] H.F. Zou y col. “An investigation and comparison of artificial neural network and time series models for Chinese food grain price forecasting”. En: *Neurocomputing* 70.16-18 (2007). cited By 96, págs. 2913-2923. DOI: 10.1016/j.neucom.2007.01.009. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-34548167332&doi=10.1016%2fj.neucom.2007.01.009&partnerID=40&md5=b30f2e80b4f61c362485cbbf674d85a6>.

Apéndice A

Manual de usuario

Con el inicio de la ejecución de la aplicación, aparece un tutorial introductorio que informa al usuario acerca del funcionamiento básico de ésta. Las pantallas que forman esta guía introductoria se muestran de manera ordenada en las Figuras A.1a, A.1b, A.1c y A.2a. Para navegar por cada una de estas pantallas, simplemente hay que deslizar la ventana de derecha a izquierda hasta llegar la última pantalla del tutorial, que permite acceder a la pantalla de bienvenida a través del botón CONTINUE (ver Figura A.2b). El botón SKIP en las pantallas A.1a, A.1b y A.1c facilita un acceso directo a la pantalla de bienvenida sin necesidad de recorrer la guía introductoria.



(a) Pantalla tutorial I (b) Pantalla tutorial II (c) Pantalla tutorial III

Figura A.1: Pantallas de inicio a la aplicación I

Una vez que se pulsa el botón **LET'S COOK** en la pantalla de bienvenida, se accede a las distintas colecciones de recetas en las que se organizan las recetas (ver Figura A.2c). Esta pantalla se corresponde con la pantalla de inicio de la aplicación. En dicha pantalla se muestran tres pestañas en la barra inferior: *Recipes* (en la que se muestra la pantalla inicial), *Search* (la cual permite realizar búsquedas sobre todas las recetas de la base de datos) y *Adapted Recipes*, que contiene recetas ya adaptadas que se han querido almacenar para uso posterior.

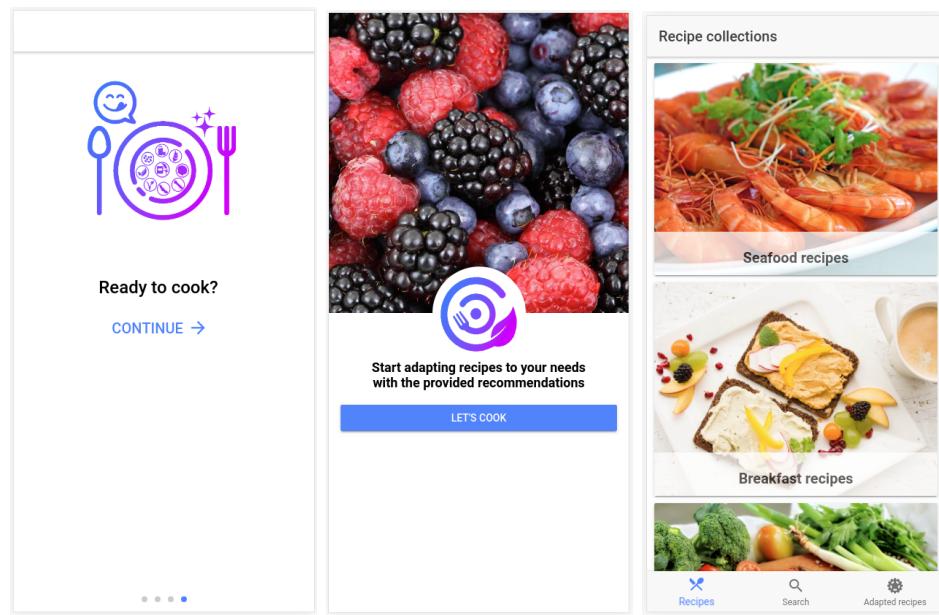
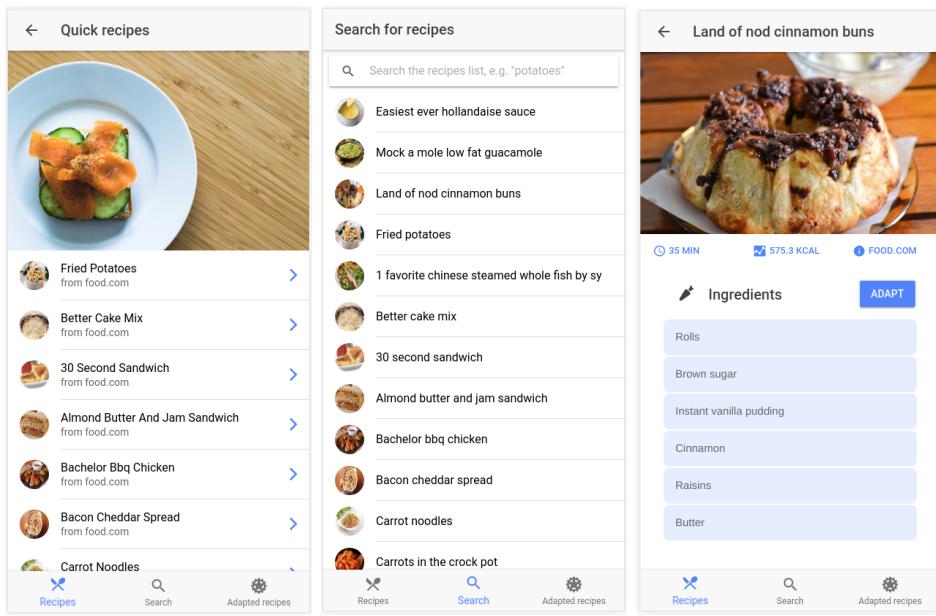


Figura A.2: Pantallas de inicio a la aplicación II

Para acceder a una receta concreta se pueden seguir dos formas: seleccionar una colección de las mostradas en la pantalla principal (ver Figura A.3a) o a través de la pantalla del buscador de recetas (ver Figura A.3b). Una vez que se está en la vista detallada de una receta (ver Figura A.3c) ya podemos proceder a su adaptación mediante el botón azul **ADAPT**. Una vez que se indica la restricción a aplicar sobre la receta (ver Figura A.4a) se facilita una pantalla intermedia que muestra la adecuación de los ingredientes a la restricción (ver Figura A.4b). En esta última pantalla, se facilitan alternativas a cada ingrediente no compatible con la dieta para proceder a su modificación (ver Figura A.4c). Una vez seleccionadas las alternativas a los alimentos que deben modificarse, se muestra la receta adaptada con los cambios tanto los ingredientes como los pasos de preparación que se tendrían que variar (ver Figuras A.5b y A.5a).



(a) Recetas en colección (b) Buscador de recetas (c) Vista de receta

Figura A.3: Pantallas para adaptar recetas

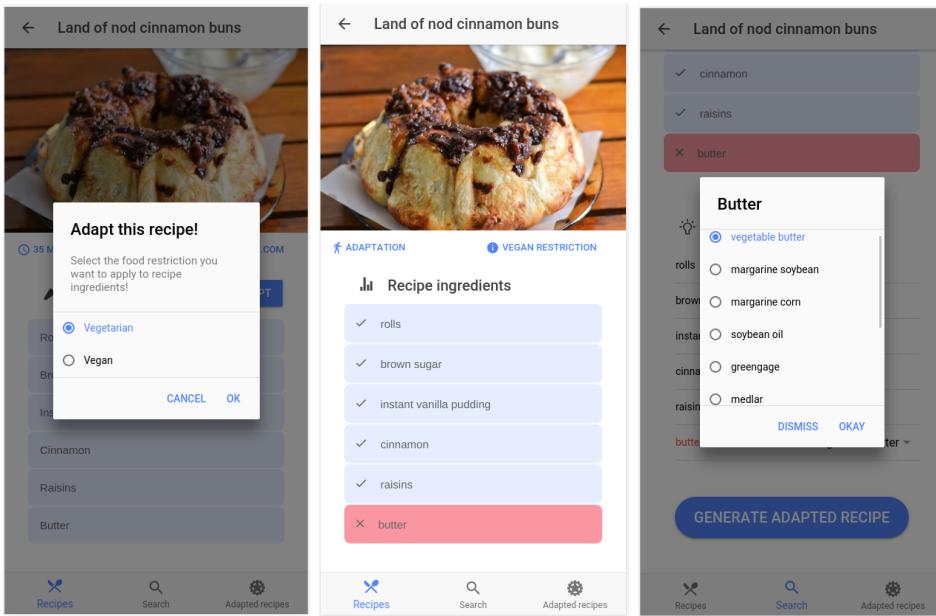


Figura A.4: Pantallas de navegación de recetas

Finalmente, arriba a la derecha en la pantalla de la receta adaptada (ver

Figura A.5a) hay un botón que permite guardar la receta adaptada en una colección dedicada para ello (ver Figura A.5c).

(a) Receta adaptada I

(b) Receta adaptada II

(c) Recetas adaptadas

Figura A.5: Pantallas de la receta adaptada

