



UNIVERSIDAD DE JAÉN
Nombre del Centro

Trabajo Fin de Grado

ANÁLISIS DE OPINIONES Y EMOCIONES EN LA WEB 2.0 MEDIANTE INTEGRACIÓN DE RECURSOS

Alumno: Sebastián Moya Pareja

Tutor: Prof. Doña M^a Teresa Martín Valdivia
Prof. Doña Salud M^a Jiménez Zafra

Dpto: Informática

Septiembre, 2017



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Doña M^a Teresa Martín Valdivia y Doña Salud M^a Jiménez Zafra, tutoras del Proyecto Fin de Carrera titulado: Análisis de opiniones y emociones en la Web 2.0 mediante integración de recursos, que presenta Sebastián Moya Pareja, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, SEPTIEMBRE de 2017

El alumno:

Fdo: Sebastián Moya Pareja

Los tutores:

Fdo: Prof. D. M^a Teresa Martín Valdivia
Fdo: Prof. D. Salud M^a Jiménez Zafra

Índice

Índice	4
Resumen	6
Capítulo 1. Introducción	7
1.1. Introducción al proyecto	7
1.2. Motivación	9
1.3. Propósito del proyecto	12
1.4. Objetivos del proyecto	12
1.5. Resultados esperados	13
1.6. Planificación	13
1.6.1. Diagrama de Gantt	13
1.6.2. Estimación de costes	14
1.6.2.1. Costes hardware	15
1.6.2.2. Costes software	15
1.6.2.3. Costes de personal	16
1.6.2.4. Otros costes	17
1.6.2.5. Coste Total	17
1.7. Estructura del proyecto	17
Capítulo 2. Estudio y análisis de las recursos	19
2.1 WordNet-Affect	19
2.2. SentiWordNet	19
2.3. SentimentStrength	20
2.4. SentiStrength	21
2.5. SenticNet	21
2.6. SO-CAL	22
2.7. SEL	22
Capítulo 3. Selección de recursos y propuesta de aplicación	24
3.1 Emociones de Ekman	24
Capítulo 4. Desarrollo del proyecto	27
4.1. Descripción del problema	27
4.2. Objetivos del sistema	27
4.3. Metodología	27
4.4. Historias de usuario	30
4.5. Propuesta de solución	32
4.6. Descripción de la solución	33
4.6.1. Iteración 1	33

4.6.1.1. Extracción tweets offline	33
4.6.1.2. Extracción tweets online	34
4.6.2. Iteración 2	35
4.6.2.1. SEL	35
4.6.2.2. BabelNet	36
4.6.3. Iteración 3	37
4.6.4. Iteración 4	40
4.6.5. Iteración 5	41
4.6.6. Iteración 6	41
4.7. Tecnologías	44
4.7.1. Python	44
4.7.2. Java	44
4.7.3. Flask	45
4.7.4. BabelNet	45
4.7.5. Babelfy	46
4.7.6. MongoDB	46
4.7.7. JavaScript	47
4.7.8. Bootstrap	48
4.7.9 Google Charts	48
Capítulo 5. Conclusiones y trabajos futuros	51
5.1. Conclusiones	51
5.2. Trabajos futuros	52
Bibliografía	54
Anexo A: Manual de instalación.	56
A.1. Instalación de Python y sus componentes	56
A.2. Instalación de la base de datos	57
A.3. Instalación de BabelNet	57
Anexo B: Manual de usuario	59
Anexo C: Índice de ilustraciones	63
Anexo D: Índice de tablas	64

Resumen

Este proyecto trata de integrar diferentes recursos existentes para conseguir analizar emociones en la web. Para ello, se ha creado una aplicación que extrae tweets filtrados por usuario o temática, que serán analizados por los recursos seleccionados.

Una vez se han analizado los distintos recursos, se ha decidido crear una herramienta que analiza las emociones en español, que podrá ser adaptada a numerosos idiomas. Esta aplicación se va a basar en las herramientas BabelNet y WordNet-Affect.

Capítulo 1. Introducción

1.1. Introducción al proyecto

Las emociones son las reacciones que tiene el ser humano ante la percepción de un objeto, persona, lugar o suceso: alegría, tristeza, miedo... Las emociones son conocidas por todo el mundo, pero no es conocido que un mal control de ellas pueden acarrear bloqueos o incluso alguna enfermedad.

La emoción es la variación profunda del ánimo, que puede ser positiva o negativa. Según la Real Academia Española, la emoción es la alteración del ánimo intensa y pasajera, agradable o penosa, que va acompañada de cierta conmoción somática. Esto permite a la persona establecer su posición respecto a un tema o respecto al entorno que le rodea, es decir, son reacciones psicofisiológicas que representan el modo de adaptación de la persona cuando percibe un objeto, persona, lugar... Estas reacciones muestran los estados emocionales a través de la respuesta de los sistemas biológicos (expresiones faciales, voz, músculos...) que son causa de la liberación de neurotransmisores u hormonas.

Debido a la importancia de cómo reaccionan las personas respecto a los estímulos, es necesario el análisis de emociones y opiniones. Estos análisis son un área de investigación del Procesamiento del Lenguaje Natural. El objetivo de estos análisis es conocer la opinión, es decir la reacción, que tienen hacia un producto, servicio, marca... El desarrollo en este campo permite captar esta información de una manera mucho más rápida y sencilla que como se ha hecho durante toda la vida mediante las encuestas.

El uso de las redes sociales está cambiando la manera de trabajar de las empresas para cuidar su imagen, ya que la opinión de un usuario puede llegar a muchísima gente en muy poco tiempo. Al hacer efecto una crítica en tan poco tiempo, están buscando sistemas para analizar rápidamente lo que opinan los

clientes de sus marcas para extraer conclusiones y tomar decisiones para mejorar la imagen.

Para el desarrollo de estas herramientas obviamente se necesita un buen analizador de opiniones. Además del buen sistema que analice las críticas de los clientes, se necesitará un *crawler* que extraiga en tiempo real los textos a analizar, que normalmente irá buscando en las redes sociales. Una vez ya se tienen los textos y están analizados, será necesaria una herramienta de visualización de la información que ayude a entender los resultados para tomar buenas decisiones.



Ilustración 1.1. Herramientas de análisis de sentimiento

Para el análisis de estas opiniones hay distintos tipos de sistemas. Los más sencillos son los que leen la frase y buscan las palabras a las que se le ha asignado alguna positividad o emoción. Estos sistemas tan simples se están utilizando en diferentes áreas para mejorar la precisión de los sistemas. Algunas de las tareas para las que se está utilizando este tipo de sistemas son:

- Conseguir que el sistema interprete las construcciones complejas. Estas construcciones complejas se refieren a frases como las comparativas o condicionales.

- Conseguir identificar exactamente el tema que se está tratando. Ya que los sistemas se suelen equivocar y pensar que se está hablando de tu marca cuando sólo la ha nombrado sin estar opinando de ella.

1.2. Motivación

Hoy en día, la sociedad está conectada a Internet las 24 horas del día a través de diferentes dispositivos. Esto ha hecho que muchas de las grandes compañías hayan querido explotar estos datos que producen sus clientes o futuros clientes, por lo que no sólo tienen que recoger estos datos sino analizarlo y llegar a unas conclusiones.

En la actualidad cada vez hay más información en Internet, ya que las relaciones sociales y el acceso a la información han cambiado. Todo el mundo “produce” información en la red, y esta información puede ser capturada y analizada para diferentes usos.

Las computadoras tienen cada vez más capacidad para procesar la información, por lo que son capaces de analizar mucha información en poco tiempo. El problema de las computadoras es que todavía no están capacitadas para entender el lenguaje humano.

Al haber cada vez más información, las empresas e instituciones quieren saber qué es lo que opina la gente. Algunas empresas hacen este análisis manualmente, pero esto se hace prácticamente imposible con toda la información que hay en la red. Por ello, se está investigando mucho en este campo para automatizar este análisis. En la *Ilustración 1.2* se puede observar cómo está aumentando en gran cantidad la cantidad de usuarios en Internet.

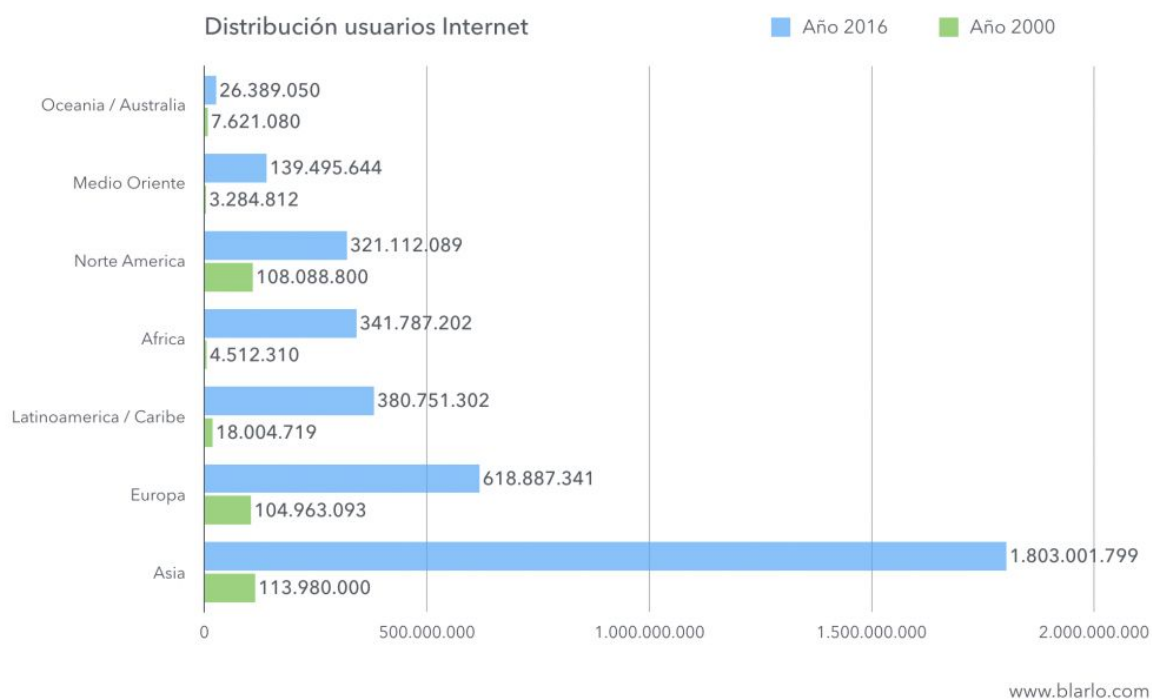
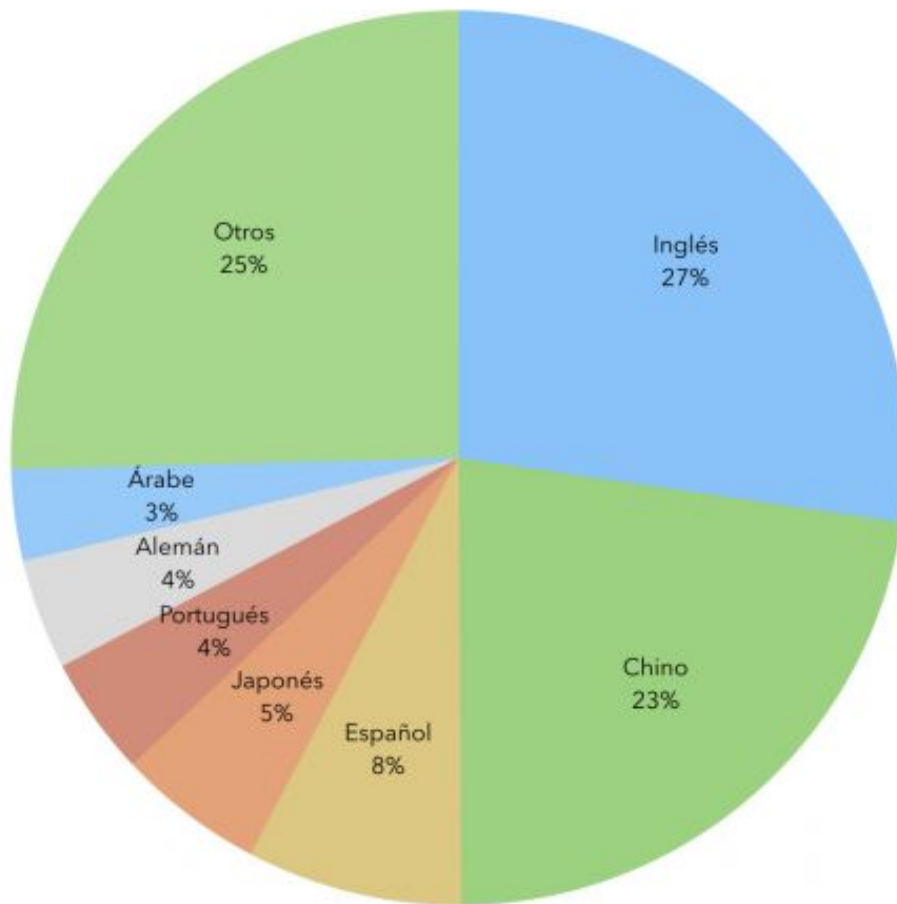


Ilustración 1.2. Distribución de usuarios en Internet

Una vez analizados diferentes recursos se ha observado que la mayoría se centran en el análisis de opiniones en inglés, por lo que se ha decidido desarrollar un sistema para el análisis en español. Los analizadores de opiniones informan de si se habla positivamente o negativamente, pero este análisis es muy escaso ya que el ser humano no reacciona sólo de manera o positiva ante los hechos, sino que le producen distintas emociones que son más complejas. Por ello para afinar más en este análisis, nos vamos a centrar en un analizador de emociones.

El español es uno de los idiomas más utilizados en la Web, concretamente el tercero, por detrás del inglés y el chino (*Ilustración 1.3*) . No tener recursos para analizar en esta lengua nos hace perder mucha información que se podría extraer de Internet.

Idiomas más utilizados en Internet

www.blarlo.com**Ilustración 1.3. Idiomas más utilizados en Internet**

Este crecimiento de información en los últimos años ha incrementado la necesidad de analizar los textos de manera automática, por lo que es un campo en el que se va a tener que investigar durante los próximos años. Esto, y el interés sobre la asignatura sobre Procesamiento del Lenguaje Natural me ha llevado a elegir un proyecto en el que se va a hacer una investigación sobre las herramientas existentes para el análisis de emociones, y a continuación hacer una aplicación para mostrar los resultados de este análisis.

1.3. Propósito del proyecto

El proyecto consiste en primer lugar en un estudio de los distintos recursos que existen tanto para el análisis de opiniones como el análisis de emociones. Una vez se analicen los recursos, se seleccionarán los que nos sirvan mejor de apoyo a la aplicación a crear. Cuando se tengan decididos los recursos, se tendrán que extraer los textos a analizar, que se hará a través de los tweets escritos por los usuarios. Estos textos deberán de ser analizados con nuestra herramienta y mostrar los resultados.

Para mostrar estos resultados de una manera más visual para el usuario, se ha decidido crear un portal web en el que el usuario pueda interactuar, solicitando sus consultas y el sistema le muestre los distintos gráficos sobre la consulta realizada.

1.4. Objetivos del proyecto

Los objetivos de este proyecto son:

1. **Seleccionar los recursos:** En primer lugar, se deben realizar un estudio sobre las diferentes APIs de análisis de opiniones y emociones. Una vez analizados los diferentes recursos se decidirán los que van a ser utilizados en el proyecto.
2. **Extraer tweets:** Se va a utilizar tanto la REST API de Twitter como la Streaming API por lo que nos permitirá extraer tanto tweets antiguos como en directo.
3. **Analizar las emociones:** Una vez se tienen los diferentes textos a analizar, se utilizará tanto el diccionario de emociones en español SEL como el análisis de emociones a través de WordNet-Affect.

4. **Desarrollar una web:** Esta web será la encargada de interactuar con el usuario, que podrá hacer las diferentes consultas y recibirá la información de ella.
5. **Análisis y conclusiones:** Se analizarán los resultados de uno y otro recurso para comprobar si da resultados similares. Si esto fuese así, nuestro sistema implementado sería satisfactorio.
6. **Redactar una memoria:** Se redactará una memoria que recoja la información del trabajo desarrollado.

1.5. Resultados esperados

Tras finalizar el proyecto, se deben obtener los siguientes resultados:

1. Investigación sobre los recursos existentes en análisis de sentimientos y emociones.
2. Extracción de un corpus basado en tweets.
3. Desarrollo de una herramienta de análisis de emociones en español.
4. Plataforma web para el uso de la herramienta.
5. Redacción de la memoria del proyecto.

1.6. Planificación

El objetivo de la planificación de un proyecto es ajustar un ámbito de trabajo, que permita realizar estimaciones razonables sobre el coste y tiempo del proyecto. Estas estimaciones se pueden ir ajustando al proyecto a medida que se va desarrollando.

1.6.1. Diagrama de Gantt

Para llevar a cabo esta planificación se va a hacer un diagrama de Gantt. El diagrama de Gantt es una herramienta para planificar y programar las tareas

durante un período de tiempo. Este diagrama permite realizar el seguimiento y control del progreso de cada una de las etapas de un proyecto, reproduciendo gráficamente las tareas, su duración y secuencia.

En la siguiente tabla (*Tabla 1.1*) se observa cuál ha sido la planificación del proyecto, con su Diagrama de Gantt (*Ilustración 1.4*).

Nombre de tarea	Duración	Comienzo	Fin
Inicio	1 día	01/02/2017	02/02/2017
Planificación	2 días	02/02/2017	04/02/2017
Estudio de APIs	30 días	04/02/2017	04/03/2017
Elección recursos sistema	15 días	04/03/2017	19/03/2017
Implementación sistema	70 días	19/03/2017	29/05/2017
Diseño interfaz	3 días	29/05/2017	01/06/2017
Estudio recursos interfaz	15 días	01/06/2017	15/06/2017
Implementación interfaz	25 días	15/06/2017	10/07/2017
Fase de pruebas	10 días	10/07/2017	20/06/2017
Redactar la memoria	10 días	20/06/2017	30/06/2017

Tabla 1.1. Planificación del proyecto



Ilustración 1.4. Diagrama de Gantt

1.6.2. Estimación de costes

Cualquier proyecto software que se desarrolle implica unos costes previos, durante su implementación y unos costes a posterior, el mantenimiento. Para realizar el cálculo de estos costes habrá que tener en cuenta los gastos mensuales y la duración del proyecto.

1.6.2.1. Costes hardware

Los costes de hardware hacen referencia a los costes de la parte física del sistema informático. Las especificaciones del equipo usado durante el desarrollo son las siguientes, aunque no es necesario que sean las mismas para su ejecución.

- Ordenador portátil: Lenovo Z50..... 550€
- Procesador: Intel Core i5-4210U (1.7GHz)
- Memoria RAM / HDD: 8GB / 1 TB
- Tarjeta gráfica: NVIDIA GeForce 840M

Se debe de amortizar este coste durante el desarrollo. Se estima una vida útil de cuatro años. Esto supone un coste de 137,50€ al año. El proyecto tiene una duración aproximada de seis meses, por lo que el coste sería de **68.75€**.

1.6.2.2. Costes software

Los costes software hacen referencia a los costes derivados a las licencias de los programas, que son los siguientes:

- Ubuntu 16.04 0€
- Java SE Development Kit 7 (64 bits) 0€
- NetBeans IDE 8.2 0€
- Base de datos MongoDB 0€
- Navegadores web 0€
- Google Drive 0€
- GitLab 0€
- Índices de BabelNet 0€
- Licencia consultas Babelfy 0€
- Licencia desarrollador de Twitter 0€

Como el coste de las licencias ha sido de **0€**, no habría ninguna cantidad que amortizar.

1.6.2.3. Costes de personal

Los costes de personal son los derivados del salario del personal asignados al proyecto. Aunque el proyecto va a ser realizado por una sólo persona tendrá asignados varios roles, y dependiendo del rol hará unas tareas u otras.

- **Analista:** encargado del diseño del sistema y obtención de los algoritmos. También encargado de analizar las posibles utilidades para mejorar la eficacia.
- **Programador:** encargado de la implementación diseñada previamente.

A continuación, se muestra la distribución del gasto en personal que va a tener el proyecto y el número de meses que se va a necesitar al personal (*Tabla 1.2*).

	Sueldo mensual	Nº de meses	Total
Analista	1422€	6	8532€
Programador	782€	6	4692€

Tabla 1.2. Costes de personal

Los sueldos han sido extraídos de indeed.es

El coste total de personal es de **13.224€**. Estos gastos hay que amortizarlos durante el transcurso del proyecto, por lo tanto hay que repartir el gasto entre los diferentes meses. El coste al mes será de 2.204€.

1.6.2.4. Otros costes

Otros de los costes derivados del proyecto que no han sido contemplados anteriormente es la conexión a Internet. Una conexión de 50Mb de Movistar tiene un coste de 14.90€/mes, por los seis meses del proyecto; el coste total será de **89,40€**.

1.6.2.5. Coste Total

El coste total del proyecto será la suma de los costes hardware, costes software, costes del personal y otros costes. Se ha decidido obtener un beneficio del 10%. En la *Tabla 1.3* se muestra el concepto y su gasto correspondiente.

Costes hardware	68,75€
Costes software	0€
Costes de personal	13.224€
Otros costes	89,40€
Costes totales	13.382,15€
Beneficio del 10%	1.338,22
TOTAL	14.720,37€

Tabla 1.3. Costes del proyecto

1.7. Estructura del proyecto

A continuación se describe la estructura de la memoria perteneciente a este proyecto. La memoria consta de cinco capítulos:

- El capítulo 1, perteneciente a la introducción, se ha explicado la motivación y objetivos del trabajo fin de grado que hay que seguir. Finalmente se concreta la planificación y los costes del proyecto.
- El segundo capítulo está dedicado a estudiar los diferentes recursos existentes tanto en el análisis de opiniones como en el análisis de emociones. Se va a estudiar de qué trata el recurso para decidir el que tenga unas características que mejor se adapte a nuestro proyecto.

- En el tercer capítulo se va a describir qué recursos de los estudiados en el capítulo anterior son los elegidos para utilizar en este proyecto. Además se explicará brevemente qué función va a tener cada recurso en el proyecto.
- El cuarto capítulo es el dedicado al proceso del desarrollo del proyecto. En primer lugar, se indicará una descripción y los objetivos del proyecto. En segundo lugar, se elegirá una metodología de Ingeniería del Software y las diferentes historias de usuario. Finalmente se comentará la solución adoptada y las diferentes tecnologías elegidas para poder desarrollar la aplicación.
- En el último capítulo se expondrán las conclusiones que nos ha aportado realizar este trabajo fin de grado y las posibles líneas de trabajo y mejoras a nuestro proyecto.
- En el apartado de bibliografía se va a indicar los diferentes materiales consultados para la realización de este trabajo.
- Finalmente se pueden encontrar cuatro anexos. El primer anexo consiste en cómo instalar el sistema; se indica los distintos componentes de Python a instalar, la base de datos y la configuración de los índices de BabelNet. En el segundo anexo se puede observar un breve manual para interactuar con el sitio web. Los dos últimos anexos son dos índices, el primero es el índice de ilustraciones y el último el índice de tablas.

Capítulo 2. Estudio y análisis de las recursos

En este capítulo se van a analizar los distintos recursos existentes para el análisis de opiniones y emociones. Se hará una pequeña introducción de cada recurso y se citarán algunas de sus características.

2.1 WordNet-Affect

WordNet-Affect es una extensión de *WordNet Domains*, un recurso léxico creado por WordNet que aumenta las el dominio de la etiquetas, que incluye un subespacio de *synsets* que correlaciona las palabras afectivas con conceptos afectivos. Es similar al método de las etiquetas afectivas (*a-labels*), es decir asigna un número de *synsets* de Wordnet a una o más *a-labels*. WordNet-Affect asigna un conjunto de *a-labels* adicionales organizados jerárquicamente, que representa los conceptos afectivos con la etiqueta **emotion**. Es decir, WordNet Domains es un árbol que representa las distintas etiquetas asignadas a los términos de WordNet. Dentro de esta herramienta se ha añadido otra parte del árbol que pertenece a WordNet-Affect. Esta estructura jerárquica ha sido construida basándose en las relaciones de hiperónimos de WordNet. El nodo de WordNet-Affect está dividido en cuatro hijos: **positive**, **negative**, **ambiguous** y **neutral**:

- **Positive:** caracterizado por presencia de señas positivas (divertido, entusiasmo).
- **Negative:** caracterizado por señales negativas (enfado, tristeza).
- **Ambiguous:** represente un estado afectivo u otro dependiendo de la semántica del contexto (sorprendido).
- **Neutral:** estados mentales que no indican nada positivo ni negativo.

2.2. SentiWordNet

SentiWordNet es un recurso léxico ideado para la clasificación de sentimientos. SentiWordNet es el resultado de la anotación automática de todos los *synsets* de *WordNet* acordando las nociones de “positividad”, “negatividad” y

“neutralidad”. Cada *synset* está asociado con tres puntuaciones numéricas: *Pos(s)*, *Neg(s)* y *Obj(s)*, que indican cómo de positivo, negativo y objetivo (p.e. neutral) es el término. Los diferentes significados de cada término puede tener diferentes propiedades de opinión. Cada una de las tres puntuaciones tiene un intervalo de [0.0 ,1.0], y su suma es 1.0 para cada *synset*. Esto significa que un *synset* no puede tener cero en las tres categorías, lo que implica que todos los términos tienen que tener las tres propiedades de opinión en cierto grado.

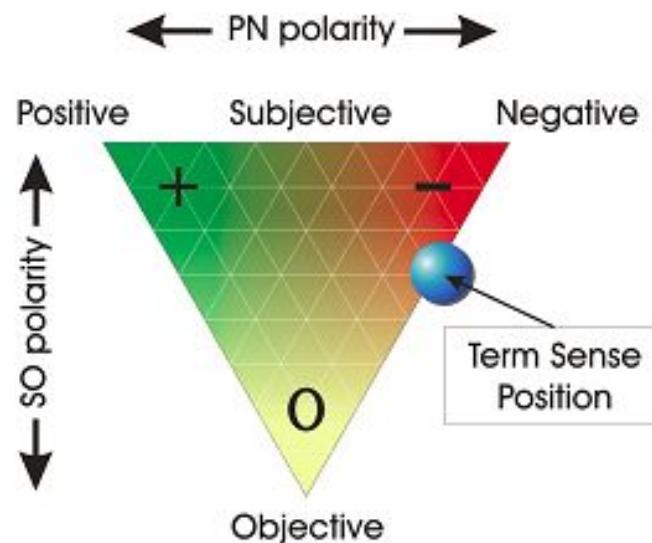


Ilustración 2.1. Triángulo SentiWordNet

2.3. SentimentStrength

SentimentStrength es un proyecto desarrollado por Hitesh Parmar. Se trata de un clasificador de sentimientos no supervisado, que trabaja con oraciones en inglés. Actualmente al no usar aprendizaje supervisado el clasificador se basa en un diccionario de palabras positivas y negativas que asigna a cada término una puntuación entre -1 y +1.

2.4. SentiStrength

SentiStrength es un lexicón de 2.310 palabras en inglés que contienen sentimientos, y que han sido extraídas del *Linguistic Inquiry and Word Count* (LIWC) y de *General Inquirer*. Para cada palabra SentiStrength muestra una puntuación para los sentimientos positivos de 1 a 5 y una puntuación para los negativos de -1 a -5. Cada palabra o raíz del diccionario obtiene una puntuación entre esos rangos. Inicialmente, se basó en un corpus compuesto por 2600 palabras extraídas de la red social *MySpace*. Este corpus fue analizado manualmente por personas que fueron asignando las puntuaciones a cada término.

El lexicón es usada de una manera muy simple. Cuando SentiStrength lee un texto, lo separa en palabras y extrae su puntuación. La puntuación total es la puntuación positiva menos la negativa.

2.5. SenticNet

SenticNet es una herramienta que lleva a cabo un análisis de sentimiento a nivel de concepto, esto es, tiene en cuenta la semántica y la lingüística no únicamente la frecuencia de ocurrencia de palabras.

SenticNet provee un conjunto de semánticas, *sentics*, y polaridad asociados con 50000 conceptos del lenguaje natural. En particular, las semánticas son conceptos que son relacionados semánticamente con la entrada, *sentics* tienen categorizados las emociones expresados en cuatro términos (amabilidad, atención, sensibilidad y aptitud) y la polaridad que es un número entre -1 y +1.

2.6. SO-CAL

SO-CAL (Semantic Orientation Calculator) es una aplicación para el cálculo de opiniones en los textos. Ha sido diseñado para el dominio de las valoraciones online de los productos.

La creación del diccionario se ha hecho manualmente. El diccionario contiene 2252 adjetivos, 1142 nombres, 903 verbos y 745 adverbios. A cada tipo de palabra se le ha asignado una puntuación, a mano, entre 5 y -5 por un experto. Los diccionarios han sido revisados a continuación por un comité de tres personas para minimizar la subjetividad del ranking SO, que es la puntuación asignada a cada palabra por SO-CAL.

La manera de trabajar con SO-CAL es ir sumando las puntuaciones obtenidas en los diccionarios de nombres, adjetivos, adverbios y verbos; mientras que con los intensificadores se multiplica la puntuación obtenida de las palabras a las que afecta por el porcentaje indicado. Por ejemplo, si la palabra *bien* tiene un valor de 3, entonces *realmente muy bueno* tendría un valor de $(3 * [1 + 0.25]) * (1 + 0.15) = 4.3$. Esta puntuación viene dado por:

- realmente: intensificador → multiplica por 1.25
- muy: intensificador → multiplica por 1.15
- bueno: adjetivo → suma un valor de 3

2.7. SEL

SEL es un diccionario en español en el que a cada palabra se le asigna una de las seis emociones básicas. Las seis emociones básicas son: alegría, enojo, miedo, tristeza, sorpresa y repulsión. Para crear este diccionario las palabras fueron valoradas por múltiples evaluadores. Una vez evaluadas las palabras se crearon unas estimaciones que representaban el factor de probabilidad de uso afectivo (FPA).

Para la creación de este diccionario en primer lugar se utilizó WordNet-Affect, traduciendo los términos que pertenecían a este recurso. La traducción se realizó a través del traductor de Google, *English-Spanish Interpreter Pro* y *Babylon*. Para garantizar la calidad de la traducción de estas palabras, se realizó una revisión manual de cada una de las 3591 palabras y verificando si existía alguna relación con las emociones básicas.

Capítulo 3. Selección de recursos y propuesta de aplicación

Una vez estudiados los diferentes recursos mencionados en el apartado 2, se han observado dos necesidades principalmente: la escasez de recursos de análisis de emociones y la escasez de recursos en español. Esto ha llevado a decidir crear una herramienta que sea capaz de analizar las emociones de textos en español. Por lo que los recursos citados anteriormente que van a ser utilizados son WordNet-Affect, que permite analizar textos en inglés, y SEL, que permite analizar textos en castellano.

Para poder analizar los textos originalmente en castellano, habría que hacer una traducción al inglés, ya que es el idioma utilizado por WordNet-Affect. Esta traducción se va a sustituir gracias a Babelfy, que es un desambiguador léxico. Además de hacer la función de desambiguador léxico, Babelfy nos permitirá obtener los códigos con los que trabaja WordNet interiormente, por lo que ya estaremos preparados para utilizar WordNet-Affect.

Para que los dos sistemas muestren los resultados en el mismo formato, hay que relacionar las emociones de WordNet-Affect con las seis emociones de Ekman que son las utilizadas por el recurso SEL. Para hacer esta comparación habrá que mapear las distintas emociones que se obtienen a través de WordNet a las emociones de Ekman, tarea que habrá que hacer manualmente ya que es una tarea compleja de automatizar.

3.1 Emociones de Ekman

Paul Ekman es un científico y psicólogo que dedicó su vida a ayudar a las personas con trastornos mentales. Comenzó una investigación que consistía en el desarrollo de maneras sistemáticas para medir el lenguaje corporal. En este estudio pudo identificar las diferentes expresiones faciales creadas por los músculos de la cara. Para la realización de este experimento, Paul Ekman viajó a Papúa Nueva

Guinea para el estudio de las expresiones faciales en una tribu. Allí pudo detectar que las expresiones faciales son interculturales, por lo que en todos los humanos utilizan las mismas expresiones faciales con el mismo significado. Las seis emociones son: alegría, enojo, miedo, repulsión, sorpresa y tristeza.

Emoción	Movimientos musculares
 Alegría	• Estiramiento hacia arriba de las comisuras de la boca. • Contracción de los músculos orbitales alrededor de los ojos.
 Enojo	• Reducción de las cejas. • Apretamiento y estrechamiento de los labios. • Apretamiento de los párpados inferiores con menos frecuencia.
 Miedo	• Levantamiento de las cejas. • Cejas más juntas. • Tensión en párpados inferiores. • Extensión horizontal sobre los labios. • Boca ligeramente abierta.

 Repulsión	• Reducción de las cejas. • Curvatura del labio superior. • Arruga en la nariz.
 Sorpresa	• Levantamiento de cejas (puede causar arrugas en la frente) • Apertura grande de los ojos. • Deja caer la mandíbula.
 Tristeza	• Párpados caídos. • Reducción de las esquinas de la boca. • Labios fruncidos. • Ojos bajos.

Tabla 3.1. Emociones de Ekman

Capítulo 4. Desarrollo del proyecto

4.1. Descripción del problema

En la actualidad ha crecido exponencialmente la información en Internet. Cualquier persona produce información en la red a través de un portal web, redes sociales... Este crecimiento ha hecho que haya mucha información sin analizar por lo que es necesario crear herramientas que recolecten esta información, la analicen y muestren un panel de herramientas que facilite al usuario la toma de decisiones a partir de esta información. Por ello, en este proyecto, se va a desarrollar una herramienta para el análisis de emociones de tweets en español.

4.2. Objetivos del sistema

El objetivo principal del sistema es crear una herramienta que permita consultar un usuario o tema y muestre cómo ha ido variando la emoción respecto al tiempo.

Este objetivo se puede descomponer en:

- Búsqueda de la respuesta. El usuario introduce una consulta y se debe de buscar la información necesaria.
- Analizar la respuesta. Analizar las emociones de la consulta realizada por el usuario.
- Presentar la respuesta. Mostrar la información lo más clara posible.

4.3. Metodología

En primer lugar habrá que decidir si elegir entre una metodología tradicional o ágil. Las dos metodologías tienen un objetivo común que es planificar el tiempo y el coste que va a suponer dicho proyecto, aunque lo hacen de maneras diferentes.

Por un lado, la **metodología tradicional** planifica el proyecto con pocas entregas (normalmente una única entrega), por lo que el feedback que se genera

con el cliente es tardío y se puede perder un gran tiempo en cosas que no quiere el cliente.

Por otro lado, la **metodología ágil** toma el control de proyecto. Plantea un proyecto con “muchos proyectos” es decir, se van creando fases de 2 a 4 semanas en la que se van mostrando al cliente el incremento del producto. El cliente va observando el producto y planificando las siguientes entregas con las nuevas expectativas.

Por ello se va a elegir una metodología ágil, ya que se va a conseguir haciendo pequeñas entregas de nuestro proyecto que permita interactuar a los clientes con estos y poder ir observando qué nuevas funcionalidades se necesitan. Los principios en los que se deriva la metodología ágil son:

- La principal prioridad es satisfacer al cliente con las diferentes entregas.
- Se acepta que los requisitos cambien, incluso en etapas de desarrollo tardías.
- Se entrega software funcional en el período de tiempo más corto posible.
- Los responsables del negocio y los desarrolladores trabajan juntos de manera cotidiana.
- El método más eficiente y efectivo de comunicación es la conversación cara a cara.
- El software funcionando es la principal medida de progreso.
- Promueve un desarrollo sostenido.
- La atención continua mejora la agilidad.

Se ha elegido una metodología de **desarrollo iterativo e incremental**, en la que cada entrega se considera como un “miniproyecto”. La ventaja de esta metodología de desarrollo es que el cliente puede observar cómo se va desarrollando el producto y empezar a utilizarlo en las versiones previas sin tener que esperar a la entrega final.

El desarrollo iterativo e incremental planifica diversos bloques temporales llamados iteraciones. En todas las iteraciones se repite un proceso similar, ya que todas las iteraciones tienen el mismo objetivo que es entregar al cliente un producto con el que interactuar.

Esta metodología fue creada para solucionar las debilidades del modelo tradicional de cascada. La idea principal es ir desarrollando el proyecto de manera incremental, permitiendo al desarrollador ir aprendiendo del desarrollo de las anteriores entregas. La clave es empezar con unos requerimientos del sistema e ir realizando pequeños cambios en el sistema. El objetivo es ir creando un producto con el que el usuario pueda interactuar.

En ciertos casos, el código puede representar la mayor fuente de documentación del sistema. El análisis de una iteración se basa en la retroalimentación del usuario en la que se puede observar qué funciones son las necesarias de manera más inmediata.

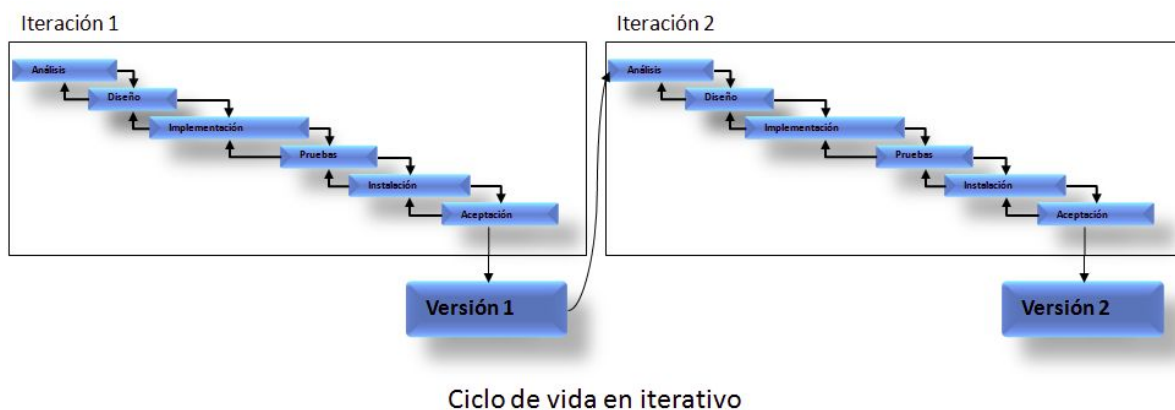


Ilustración 4.1. Ciclo de vida en iterativo

Los beneficios de este tipo de desarrollo son los siguientes:

- Se puede gestionar las expectativas del cliente, ya que el cliente en la fase inicial no sabe exactamente lo que necesita, esto lo va viendo a medida que

se desarrolla el proyecto. Además, permite al equipo saber que es lo que el cliente espera del producto.

- El cliente obtiene resultados importantes y usables desde las primeras iteraciones.
- Permite conocer el progreso real del proyecto, lo que facilita a la hora de gestionar la complejidad del proyecto y saber si se van a cumplir los plazos de entrega.

Las desventajas de este tipo de desarrollo son:

- Se requiere que el cliente tenga una disponibilidad alta, ya que debe de ir interactuando durante el proceso de desarrollo del software, detallando los nuevos requisitos en cada análisis y revisando el producto final de cada iteración.
- En cada iteración se debe de entregar como resultado los requisitos terminados, ya que cada iteración debe ser el resultado de un proyecto útil para el cliente.
- Cada iteración debe de aportar valor, no se pueden entregar iteraciones en la que no mejore el producto y siga igual que en la iteración anterior.

4.4. Historias de usuario

En el desarrollo ágil, para obtener los requisitos del sistema se utilizan las historias de usuario. Se trata de un instrumento que emplea la información utilizada por los usuarios para obtener los requisitos de la aplicación.

Las historias de usuario deben estar detalladas lo suficiente como para poder empezar a trabajar, aunque se vaya modificando y añadiendo detalles durante la fase del desarrollo. Los atributos que forman las historias de usuario son: ID, título, prioridad y descripción. Además, se deben añadir las dependencias si una historia de usuario depende de otra.

ID (dep.)	Título	Prioridad	Descripción
1	Extracción tweets	Alta	Como usuario quiero que se puedan extraer los tweets a analizar
2 (1)	Análisis de las emociones	Alta	Como usuario quiero un sistema que analice los textos y extraiga la emoción
3 (2)	Clasificación emociones	Alta	Como usuario quiero clasificar las distintas emociones para poder mostrarlas
4	Visualizar gráficos	Baja	Como usuario quiero poder visualizar distintos gráficos para observar cómo han ido evolucionando las emociones
5	Visualizar mensajes	Media	Como usuario quiero poder ver qué emoción se le ha asignado a cada mensaje
6 (4,5)	Interfaz	Media	Como usuario quiero tener una interfaz para poder interactuar con el sistema

Tabla 4.1. Historias de usuario

4.5. Propuesta de solución

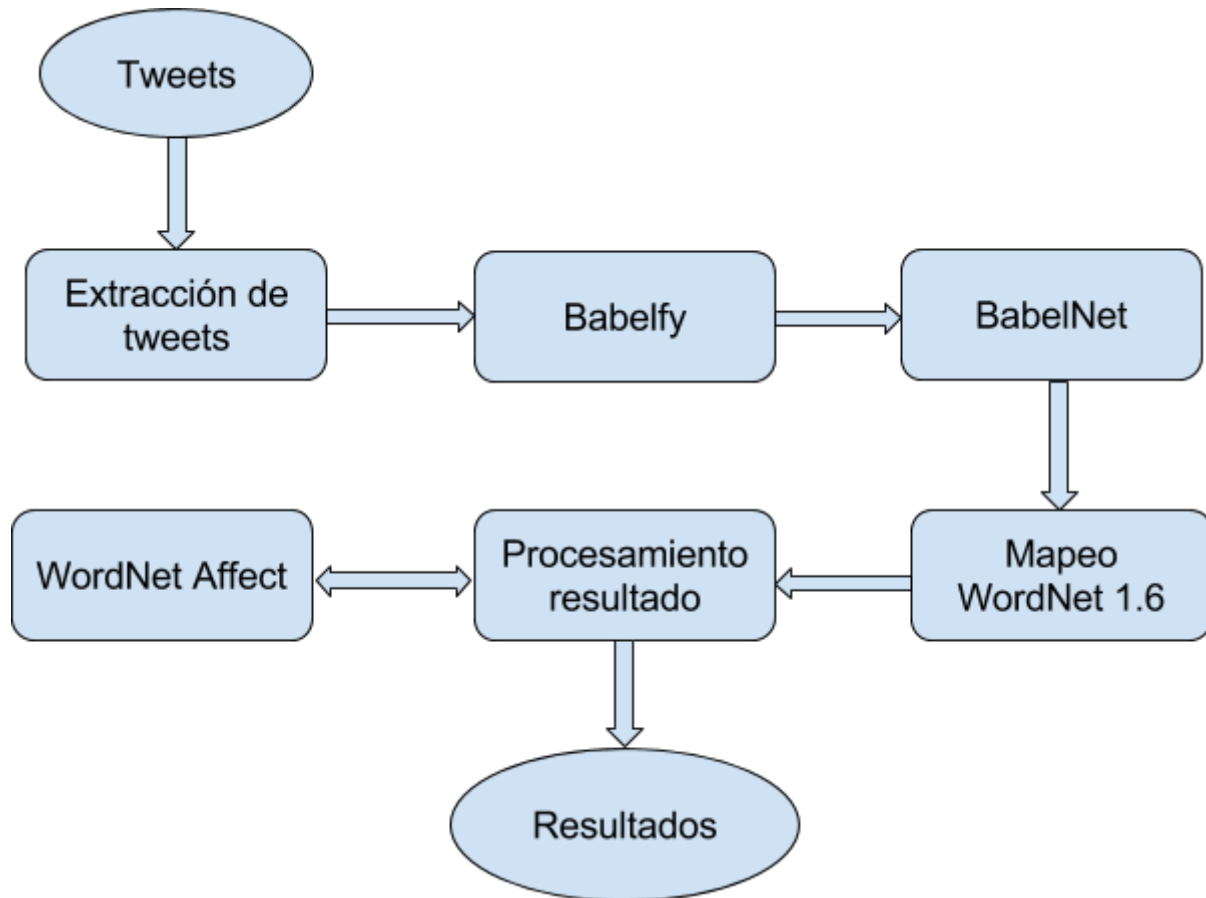


Ilustración 4.2. Esquema de aplicación

El funcionamiento de la aplicación puede sintetizarse en los siguientes puntos:

1. Para empezar se tienen que extraer los tweets a analizar. La minería de tweets puede ser o con tweets en directo, en la que mientras se van extrayendo los tweets se van analizando; o tweets antiguos, en los que en primer lugar se llevará a cabo un proceso de extracción y a continuación se analizará la emoción.
2. Una vez obtenido el texto de un tweet se conecta a la API de Babelfy. Babelfy es un desambiguador léxico multilingüe. Esta librería recibe texto en cualquier idioma, en este caso en español y devolverá la lista de palabras con significado y desambiguadas a través de su código en BabelNet.

3. Una vez se tienen todas las palabras a tener en cuenta, se utiliza WordNet que es una gran enciclopedia multilingüe. Entre todas las funciones que tiene, la que se va a usar en esta aplicación es traducir los códigos de las palabras de BabelNet a la de WordNet 3.0.
4. El siguiente paso es convertir todos los offsets de WordNet 3.0 a la versión WordNet 1.6 que es la utilizada por WordNet-Affect.
5. Una vez se tiene todo a punto para utilizar WordNet-Affect habrá que ir llamando a esta API palabra por palabra para que nos devuelva la emoción asociada a cada una de ellas. Las diferentes emociones se irán almacenando y se decidirá cuál es la emoción más representativa para dicho texto.

4.6. Descripción de la solución

4.6.1. Iteración 1

Iteración 1: “Como usuario quiero que se puedan extraer los tweets a analizar”

Para la extracción de los tweets se han utilizado dos APIs: para la extracción de tweets offline se ha utilizado la REST API mientras que para extraer tweets en tiempo real se ha utilizado la Streaming API.

4.6.1.1. Extracción tweets offline

Para la extracción de tweets offline de un usuario, se ha utilizado la API de twitter *Tweepy*. Esta librería ayuda a la autenticación y utilización de los diferentes métodos que proporcionan las propias APIs nativas de Twitter. Las funciones para la extracción de tweets están situados en el script *MinningTweets.py*

Se ha creado el método *get_tweets* que devuelve una lista de tweets, la idea de esta función es que se puedan recolectar los tweets antiguos en ese mismo

momento y servir de apoyo a otras APIs. Este método se apoya en el *get_tweets_id*, que extrae los tweets a partir del ID indicado. El último método es *get_tweets_date* que devuelve la lista de tweets del usuario igual que el citado anteriormente pero extrae los mensajes entre las dos fechas indicadas.

Estas funciones son simples y similares. En primer lugar hacen una autenticación con Twitter, mediante las credenciales que nos ha asignado al registrar la aplicación. Una vez se establece la conexión, comienza a extraer tweets del usuario.

Para la extracción de estos tweets habrá que pasarle el usuario a analizar y el intervalo de la fecha de los mensajes que se quiera recibir o el número de tweets que se quiere recolectar.

El problema de esta extracción es que la API de Twitter sólo permite la extracción de los 3200 tweets más recientes por cada usuario.

4.6.1.2. Extracción tweets online

Para el uso de la Streaming API, se ha utilizado un breve script *stream2mongo.py* que es el encargado de ir extrayendo a través de Tweepy los tweets en directo sobre la consulta que se le ha indicado. Estos tweets que se van extrayendo en tiempo real, se van almacenando en una base de datos.

La base de datos elegida es MongoDB que es una base de datos NoSQL con la flexibilidad que nos da esto, ya que cada entrada en la base de datos pueden tener atributos diferentes. Además MongoDB trabaja con documentos, que se corresponden a los diccionarios de Python.

La colección de documentos de Mongo se limpia al iniciar la nueva consulta. Una vez deja esta colección vacía empieza a insertar los tweets que vayan encontrado. Se podrán acceder a estos tweets a través de la base de datos. Estos

tweets han sido almacenados en una colección dentro de la base de datos llamada “tweets”.

Para usar este método sólo habrá que indicarle la consulta a realizar e irá recolectando en directo los tweets que coincidan con la consulta.

4.6.2. Iteración 2

Iteración 2: “Como usuario quiero un sistema que analice los textos y extraiga la emoción”

Para el análisis de textos se han utilizado dos recursos diferentes: SEL que es un lexicón en español y BabelNet junto a WordNet-Affect que son dos herramientas que trabajan en inglés.

4.6.2.1. SEL

Se ha implementado una clase llamada SEL, que como su propio nombre indica analiza textos a través del *Spanish Emotion Lexicon* (SEL) que es un diccionario de emociones creado por Grigori Sidorov. Este diccionario contiene 2036 palabras asociadas a un Factor de Probabilidad Afectiva (FPA) respecto a una de las seis emociones: alegría, repulsión, enojo, miedo, sorpresa y tristeza. Dicho diccionario ha sido creado a partir de *WordNet Affect* pero a continuación ha sido revisado manualmente por diferentes anotadores.

El SEL es un lexicón creado por Sidorov, es decir es un fichero de texto que asocia cada palabra con una emoción y su FPA. Como solo nos da este documento de texto, habrá que crear un pequeño sistema que vaya buscando la palabra y almacene las distintas emociones que tiene el texto junto a su PFA. La estructura del documento es la siguiente:

Palabra PFA Categoría

abundancia 0.83 Alegría

acabalar 0.396 Alegría

acallar 0.198 Alegría
acatar 0.198 Alegría
acción 0.397 Alegría
aceptable 0.594 Alegría
aceptación 0.696 Alegría
acicate 0.429 Alegría
aclamación 0.799 Alegría
aclamar 0.799 Alegría
acogedor 0.83 Alegría
...

Al inicializar el objeto SEL, se crean las diferentes herramientas que van a ser utilizadas para el análisis del texto como son los diferentes tokenizadores: *TreebankWordTokenizer* y *PunktTokenizer*. *PunktTokenizer* es el encargado de tokenizar las distintas oraciones, mientras que *Treebank* trabaja con esta oración y la tokeniza en palabras. Al inicializar el objeto también se crea el diccionario de palabras del SEL extraído del fichero *SEL.txt*. Este diccionario asigna a cada palabra una tupla con la categoría que pertenece y el FPA.

Una vez inicializado el objeto SEL, es el momento de analizar los textos. Se llama a la función *analysis_text(self, text)*. A este método hay que pasarle el texto a analizar y devolverá un string con uno de las seis emociones citadas anteriormente o *None* si no ha conseguido detectar ninguna. Para seleccionar una u otra emoción, va sumando los diferentes FPA asignados a la palabra que encuentra, esta suma se agrupa por emociones. La emoción que obtenga un mayor FPA es la que devolverá este método.

4.6.2.2. BabelNet

Se ha creado una clase llamada *BabelNet* que será la encargada de gestionar tanto la parte de BabelNet como la parte de Babelfy.

Lo primero que hace esta clase es almacenar WordNet-Affect, el mapeado de WordNet 3.0 a WordNet 1.6 y el fichero para convertir las emociones de WordNet-Affect a las seis emociones de Ekman utilizado por SEL.

Una vez se quiere analizar el texto se debe de llamar a la función *analysis_babel* pasándole el texto a analizar. En primer lugar accede mediante consultas HTTP a Babelfy para que desambigüe el texto. Esto nos devolverá la lista de identificadores representativos en BabelNet y se almacenarán en la variable *data*. A continuación, se convierten los identificadores de BabelNet a los identificadores de WordNet en la versión 3.0 para ello, al haber solicitado los índices a BabelNet para poder trabajar de manera offline y estos sólo poder acceder en Java, se ha tenido que crear una comunicación entre estos ficheros Java y nuestro programa en Python a través de un Socket. Esta conexión está implementada en el fichero *connectSocket.py*.

Una vez tiene una lista de offsets de BabelNet 3.0 es el momento de transformarlos a 1.6. Al mapearlo a la versión 1.6 nos devuelve el código del offset de esta versión y el factor de confianza, que se almacenará en la variable *offset16*. Una vez se tienen los offsets en la versión adecuada, es el momento de trabajar WordNet-Affect. WordNet-Affect irá recibiendo el código de cada palabra y devolviendo la emoción correspondiente a dicha palabra. Las emociones que se vayan devolviendo se irán almacenando en la variable *emotions*. Para decidir qué emoción es la más representativa. además de la emoción también se almacenará el factor de confianza nombrado anteriormente. Finalmente se elegirá la emoción con mayor factor de confianza y la devolverá, ya que en nuestro caso sólo nos interesa tener una emoción por cada texto.

4.6.3. Iteración 3

Iteración 3: “Como usuario quiero clasificar las distintas emociones para poder mostrarlas”

Una vez se tienen los dos sistemas implementados, se ha decidido unificar los resultados de ambos sistemas. Al estar unificados, se podrá observar si extraen las mismas emociones ambos sistemas.

Para asociar los resultados de ambos recursos se ha decidido convertir las emociones que nos da WordNet Affect ya que es un sistema de granularidad más fina a las seis emociones de Ekman que son las utilizadas por el SEL. Para ello, se ha hecho una relación manual de las diferentes emociones de WordNet Affect con la que más se asocia del SEL. Las diferentes emociones que nos puede dar WordNet Affect son:

```

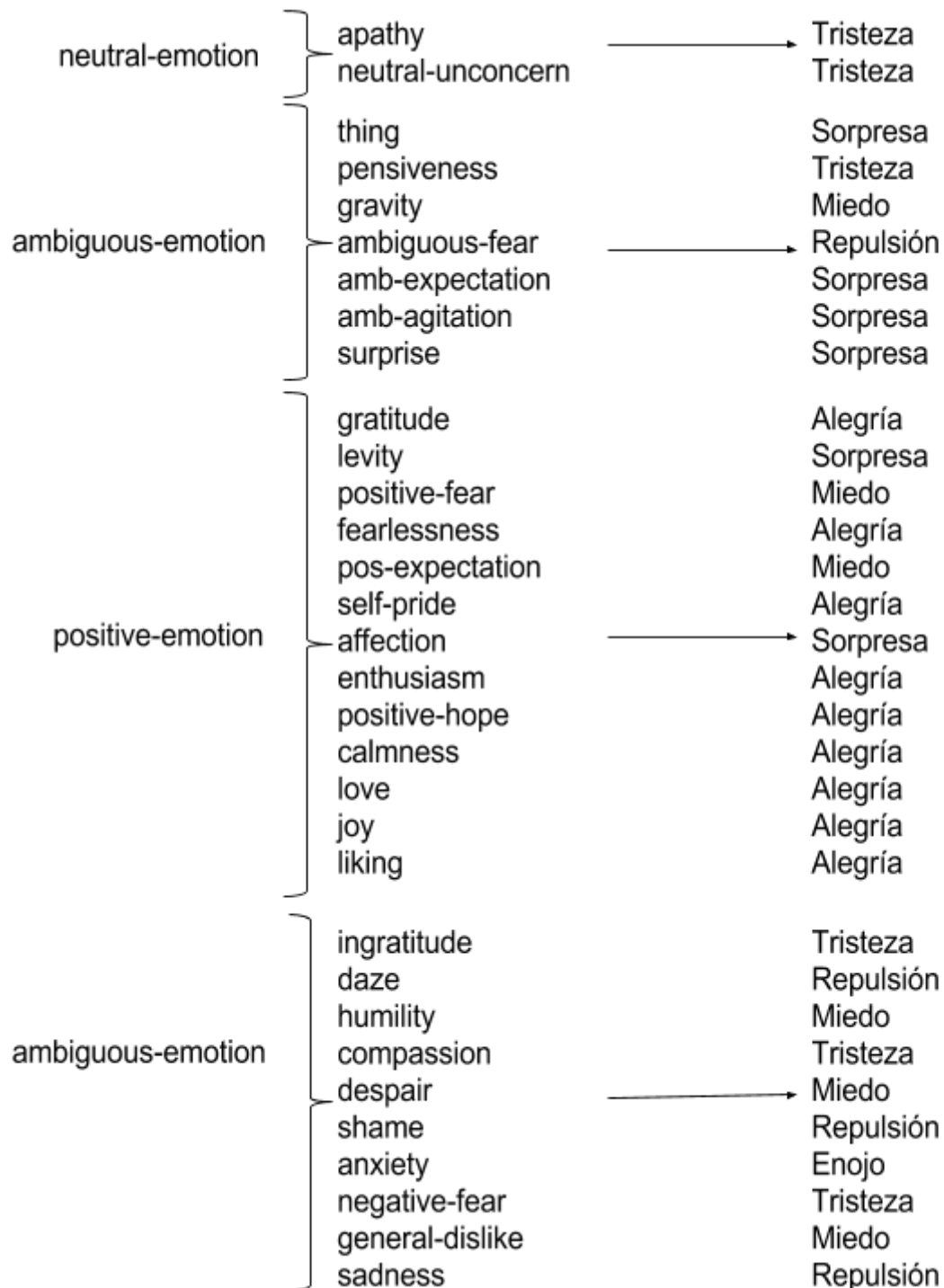
└physical-state
└behaviour
└trait
└sensation
└situation
|   └emotion-eliciting-situation
└signal
|   └edonic-signal
root
|   └cognitive-state
|   └cognitive-affective-state
└mental-state
    |   └mood
    └affective-state
        |   └emotionlessness
        |   └apathy
        |   |   └neutral-languor
        |   └neutral-emotion
        |   |   └neutral-unconcern
        |   |   |   └distance
        |   |   └indifference

```

				↳withdrawal
			└thing	
			└pensiveness	
			└gravity	
				↳earnestness
			└ambiguous-fear	
				↳reverence
				└ambiguous-hope
			└ambiguous-expectation	
				↳fever
				↳buck-fever
			└ambiguous-emotion	
				└unrest
				└tumult
			└ambiguous-agitation	
				└electricity
				↳stir
				↳sensation
			↳surprise	
				└surprise
				└stupefaction
			↳astonishment	
				↳wonder
				↳awe

.....

Ya que hay muchas emociones distintas, se han decidido elegir los elementos más cercanos de la raíz y asociarle una de las emociones. Aunque esta traducción puede tener muchas versiones, se podrían asociar todas las emociones con la de Ekman o no relacionarlas con las emociones principales y sóloamente traducirlas al español. La traducción decidida son:



4.6.4. Iteración 4

Iteración 4: “Como usuario quiero poder visualizar distintos gráficos para observar cómo han ido evolucionando las emociones”

Para el uso de gráficos se va a utilizar una herramienta externa como es Google Charts, en el que se le pasarán los datos apropiados y Google será la encargada de crear las imágenes de los diferentes gráficos. Esta herramienta se explicará con más detalle en el *Apartado 4.7.9*.

4.6.5. Iteración 5

Iteración 5: “Como usuario quiero poder ver qué emoción se le ha asignado a cada mensaje”

Para conseguir cumplir esta iteración, a cada tweet habrá que almacenar también la emoción a la que corresponde y almacenarlo en nuestra base de datos. Hacer esto, permitirá poder mostrar en una lista cada tweet y la emoción asignada.

4.6.6. Iteración 6

Iteración 6: “Como usuario quiero tener una interfaz para poder interactuar con el sistema”

Para la gestión de datos y vistas de la web, como para la implantación del servidor en local, se ha utilizado el framework Flask. Flask es un framework de Python que ayuda a la gestión de rutas de nuestra web.

En el script *frontend.py* es el que se encarga de esta tarea. Dependiendo de la ruta accedida se necesita una serie de datos u otros. Estos datos serán solicitados y tratados para adaptarlo a la estructura necesaria para que se pueda mostrar en nuestra página.

En esta web se ha utilizado Bootstrap para hacer una web responsive y mejorar el diseño de esta. También se ha utilizado Google Charts con la que se harán las diferentes gráficas que queremos mostrar.

Una de las ventajas de utilizar Flask es que dentro de los ficheros HTML se pueden realizar bucles y recorrer nuestras estructuras de datos de una manera muy simple al igual que se realiza en Python.

En nuestra aplicación se van a tener dos páginas. La primera constará de un formulario para realizar el tipo de consulta que se quiere hacer. Si es analizar los tweets de una cuenta o analizar algún tema en directo. Si se quiere hacer la primera consulta, se deberá de indicar el nombre de la cuenta del usuario sin el '@' y la fecha de los tweets. Por ejemplo si se quiere analizar la cuenta de la Universidad de Jaén (@ujaen), en el formulario se deberá de introducir "ujaen".

Analizador de emociones

Formulario de búsqueda de tweets para el analizador de emociones. El formulario incluye un campo de texto para la búsqueda, dos campos de fecha para el rango de tiempo, y dos botones de radio para seleccionar el tipo de consulta.

Buscar

ujaen

Desde

01/05/2017

Hasta

15/06/2017

☒ Analizar cuenta ☐ En directo

Consultar

Ilustración 4.3. Esquema Web

Al realizar la consulta, el sistema extraerá los tweets correspondientes y analizará su emoción. Una vez analizada la emoción, procesará la información para

adaptarla a los gráficos. En la misma página del formulario, nos añadirá los gráficos correspondientes y todos los tweets con emoción.

Por el contrario, si se quiere analizar algún tema en directo, se debe de seleccionar en los botones de radio “En directo” y la consulta deseada. En esta situación no es necesario introducir ninguna fecha ya que se monitorizaran los tweets en tiempo real.

Analizador de emociones

Formulario de búsqueda de tweets por emoción. El formulario contiene los siguientes elementos:

- Un campo de texto con el placeholder "Buscar" y el contenido "#TrabajoFinDeGrado".
- Un campo de texto con el placeholder "Desde".
- Un campo de texto con el placeholder "Hasta".
- Una fila de botones de radio: "Analizar cuenta" (desseleccionado) y "En directo" (seleccionado).
- Un botón "Consultar" situado debajo de los botones de radio.

Ilustración 4.4. Esquema Web 2

Al elegir el análisis en directo nos redireccionará a la página /streaming en la que irán apareciendo los tweets en los que se detecten emoción sobre la consulta realizada. Esta página se irá actualizando automáticamente ya que su función es ir añadiendo los mensajes con emoción que se vayan encontrando.

4.7. Tecnologías

4.7.1. Python

Python ha sido elegido como el lenguaje de programación utilizado casi en la totalidad del proyecto, usando la versión 3 de Python. Es un lenguaje de programación interpretado basado en scripts. Este lenguaje está preparado para cualquier tarea, desde aplicaciones Windows a servidores de red. Es un lenguaje de programación que soporta la orientación a objetos y la programación imperativa. En los últimos años se ha hecho muy popular por:

- Tener una cantidad de librerías que contienen datos y funciones incorporados al propio lenguaje para realizar tareas sin tener que implementarlas desde cero.
- Su gran calidad de sintaxis. Un bloque de código se crea a través de indentaciones que obliga al desarrollador a garantizar una legibilidad notoria.
- Es un lenguaje interpretado. Esto quiere decir que no se debe compilar el código antes de su ejecución. Aunque en ciertos casos, cuando se ejecuta por primera vez, se producen unos bytecodes que sirven para acelerar la compilación implícita que realiza el intérprete cuando se ejecuta el código.
- Python es un lenguaje interactivo. Dispone de línea de comandos en la que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible.

4.7.2. Java

Este lenguaje de programación ha sido utilizado para transformar los identificador de BabelNet a los de WordNet, comunicándose con la parte de Python a través de un socket. Java es un lenguaje de programación desarrollado por James Gosling en 1995. Este lenguaje de programación se basa en C y C++ aunque tiene menos utilidades de bajo nivel. Es un lenguaje de programación concurrente y orientado a objetos, aunque no se considera de los puros, ya que todos los tipos son clases, excepto los tipos fundamentales de variables como son los números enteros

o los número reales. Como lenguaje orientado a objetos se apoya en la encapsulación, polimorfismo, etc.

Este lenguaje dispone de un amplio conjunto de bibliotecas en las que el programador puede apoyarse y no tener que desarrollar todas las herramientas por sí solo.

Es un lenguaje simple, por lo que posee una curva de aprendizaje muy rápida.

4.7.3. Flask

Flask ha sido usado para controlar la parte del servidor de una manera sencilla permitiendo programar en Python y no en los lenguajes típicos de la web. Flask es un framework muy simple escrito en Python para la creación de aplicaciones web.

Flask incluye un servidor web de desarrollo sin necesidad de instalar algo complementario como Apache. Además, Flask soporta el uso de cookies seguras y se centra en el motor Jinja2.

Al no tener configuraciones complejas, se convierte en el candidato ideal para aplicaciones ágiles.

4.7.4. BabelNet

Babelnet se ha utilizado junto a Babelfy (*Apartado 4.7.5*) para desambiguar y traducir el texto. Posteriormente, se utiliza para relacionar los identificadores de BabelNet con WordNet-Affect. BabelNet es un proyecto financiado por el Profesor Roberto Navigli de la Sapienza University de Roma. El proyecto ha recibido financiación de la Unión Europea. Su objetivo principal es crear un recurso léxico para permitir la comprensión de textos multilingües.

BabelNet ha sido creado gracias a la integración de recursos como WordNet, Wikipedia, Wikidata, Wiktionary, OmegaWiki y GeoNames. Actualmente cubre 14 millones de conceptos y nombres de entidades en 271 idiomas. Entre estos idiomas incluye todos los idiomas europeos, muchos asiáticos y el latín. Cada synset contiene 5.5 sinónimos de media. Respecto al tema multimedia, posee 11 millones de imágenes asociadas a los Babel synsets.

Con BabelNet se pueden obtener resultados multilingües en cualquier idioma. BabelNet almacena los synsets, sus identificadores, los diferentes sentidos, definiciones, imágenes y categoría gramatical.

4.7.5. Babelfy

Babelfy es un algoritmo de desambiguación de texto escrito en cualquier lenguaje. La principal tarea de Babelfy es ser un desambiguador léxico multilingüe. Babelfy está basado en la red semántica de BabelNet.

4.7.6. MongoDB

MongoDB ha sido utilizado para almacenar los diferentes tweets que son extraídos en las consultas online. MongoDB es una base de datos NoSQL orientada a documentos. MongoDB en lugar de guardar los datos en tablas relacionales lo hace en estructuras de datos basadas en documentos similar a JSON (en mongoDB se llama BSON).

El desarrollo de MongoDB empezó en octubre de 2007 por la compañía 10gen cuando estaban desarrollando una plataforma Paas similar a Google App Engine. En 2009 fue lanzado como producto independiente y bajo la licencia AGPL.

MongoDB es una base de datos ágil que permite a los esquemas cambiar rápidamente. No necesita definir un esquema anteriormente, pero sí se debe decidir

si necesitamos normalizar o aproximar los datos, para que mejore el rendimiento de nuestra aplicación.

Otra de las ventajas de MongoDB es que está disponible para varios sistemas operativos como Windows, Linux y OS X. Además de los diferentes sistemas operativos, también existen frameworks para poder ser utilizado en diferentes lenguajes de programación como Java, C++, Python... En este proyecto se va a utilizar Python, por lo que el framework a utilizar será PyMongo.

4.7.7. JavaScript

Esta tecnología ha sido usada en el proyecto para poder crear funciones que mejoren la interfaz cliente/servidor. JavaScript es un lenguaje de programación interpretado, orientado a objetos, imperativo y dinámico. Es un lenguaje independiente de la plataforma. El programa creado puede ser ejecutado tanto en Linux, Windows, etc.

Muchos confunden JavaScript con Java, pero son lenguajes distintos. JavaScript tiene la ventaja de poder ser incorporado a cualquier página y ser utilizado sin la necesidad de instalar otro programa. Se puede decir que JavaScript es un lenguaje basado en prototipos mientras que Java es un lenguaje más orientado a objetos. Normalmente, suele ser utilizado en el lado del cliente implementado como parte del navegador web, aunque también existe una versión de JavaScript para el servidor. Todos los navegadores actuales interpretan el código JavaScript que permite interactuar con una página que provee a JavaScript una implementación del DOM.

Actualmente es utilizado para enviar y recibir información del servidor con el apoyo de otras tecnologías como AJAX. JavaScript se interpreta en el cliente mientras que va descargando las sentencias junto al código HTML.

4.7.8. Bootstrap

Bootstrap ha permitido crear páginas web *responsive* que se adapten a diferentes dispositivos. Es un framework para el front-end de código abierto para aplicaciones web. Esta herramienta contiene plantillas de HTML y CSS, como extensiones de JavaScript, la cual nos permite crear una web de manera rápida y sencilla ya que no es necesario crear todos estos ficheros. Otra de las ventajas de este recurso es que es compatible por la mayoría de los navegadores ya que se ha estandarizado.

Bootstrap ha sido desarrollado por Twitter, encabezado por Mark Otto y Jacob Thornton. La primera idea era desarrollar este sistema para el uso interno, aunque fue tan bueno que se decidió compartirlo bajo Open Source. En la actualidad, la comunidad de Bootstrap es de las más numerosas en GitHub.

4.7.9 Google Charts

Google Chart es un servicio web interactivo que crea gráficas a partir de los datos obtenidos del usuario. El servidor de Google crea una imagen dinámica de tipo PNG de la gráfica con los parámetros especificados por el usuario a través de la consulta HTTP. Este recurso soporta una gran variedad de gráficos como líneas de tiempo, gráficas de barras, histogramas...

Google Chart necesita obtener los datos con un formato determinado para poder capturarlo, este formato será diferente dependiendo del tipo de gráfico a mostrar. Estos datos pueden provenir de distintos sitios:

- Desde el propio código de la página html.
- A través de servicios que soporten el recurso como Google Spreadsheets y Salesforce.

Cada tipo de gráfico tiene muchas opciones para poder ser personalizado añadiendo título, colores, grosores de línea.

En primer lugar se va a utilizar el “Donut Chart” un gráfico que nos permite visualizar de manera muy clara el porcentaje de tweets con una emoción u otra. Para utilizar este método sólo necesita la emoción y el número total de tweets a los que le pertenece.

My Daily Activities

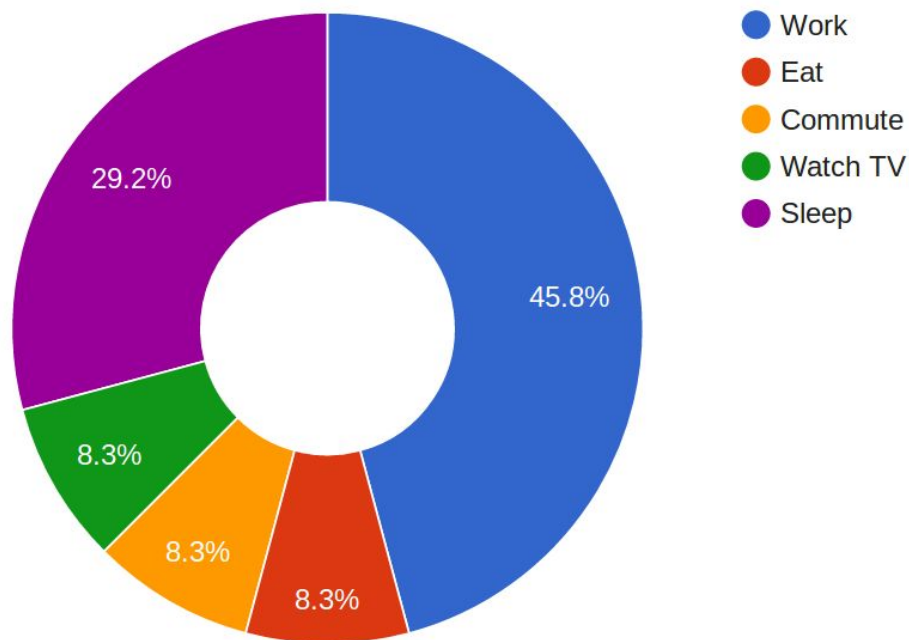


Ilustración 4.5. Gráfico Donut

Por otro lado se va a utilizar el gráfico conocido como “Timelines” que permite visualizar cómo ha ido variando la emoción en el tiempo. Se ha elegido la opción en la que muestra la gráfica en una sola fila. Para este método habrá que indicarle el inicio y fin del período y la emoción.



Ilustración 4.6. Gráfico Timeline

Capítulo 5. Conclusiones y trabajos futuros

En este capítulo se describe las conclusiones extraídas una vez se ha terminado el proyecto y las futuras mejoras que se podrían incluir.

5.1. Conclusiones

La elección de este trabajo fin de grado fue llevado a cabo al haber tenido una breve introducción a este campo en la asignatura de Procesamiento de Lenguaje Natural que se da en 4º curso, ya que esta asignatura la he cursado mientras realizaba el proyecto.

Al obtener información sobre este campo, observé que es un campo en el que queda mucho por investigar y que va a ser muy necesitado en un presente inmediato, ya que tanto los gobiernos, empresas o instituciones quieren saber qué está pasando en Internet y qué opina la gente de ello. Esto nos hizo pensar tanto a mis tutoras como a mi en desarrollar una herramienta que fuese capaz de ir extrayendo textos y analizando su emoción en nuestro idioma, el castellano.

Este proyecto me ha permitido entender que es el Procesamiento de Lenguaje Natural, y más concretamente el campo de análisis de opiniones y emociones. Además, me ha hecho progresar en otros campos como son:

- Extracción de textos a través de las APIs de Twitter.
- Aprender un lenguaje de programación que era totalmente desconocido para mi como era Python.
- Conocer numerosos recursos para análisis de opiniones, análisis de emociones, desambiguadores léxicos, etc.
- Mejorar sobre el desarrollo web, especialmente sobre Flask y Bootstrap.

Este trabajo ha sido muy útil ya que me ha permitido crear una herramienta para el análisis de emociones en español, algo que no se había hecho hasta el momento. Esta herramienta no sólo es capaz de satisfacer la necesidad de un

analizador de emociones en castellano, sino que se puede adaptar para numerosos idiomas por lo que permitiría tener un analizador de emociones multilingüe. Hasta ahora, estos analizadores se han tenido que especializar en un idioma para poder conseguir buenos resultados.

5.2. Trabajos futuros

El análisis de emociones en español es una rama que no ha sido suficientemente desarrollada, ya que como se ha citado en el proyecto solo se conoce el lexicon SEL. Este proyecto podrá continuar explotándose. Por ejemplo, se podrán validar los resultados proporcionados por el clasificador del proyecto mediante la anotación manual de los textos por expertos con el fin de comprobar la precisión y cobertura del sistema. Una de las ideas que se me ocurren que se podrían estudiar son:

- Realizar la conversión de las emociones de WordNet Affect a las seis emociones básicas entre diferentes personas, ya que al hacer este mapeado solo una persona puede ser erróneo ya que no es fácil relacionar estas emociones y cada persona tendrá diferentes puntos de vista.
- Hacer un mapeado de todas las emociones existentes en WordNet, ya que en la actualidad se ha hecho buscando los elementos superiores del árbol. WordNet tiene una granularidad muy fina por lo que podría mejorar los resultados notablemente ya que esta traducción sería mucho más exacta.
- En nuestro sistema se ha trabajado con textos breves como son los tweets, de máximo 140 caracteres. Sería interesante probar resultados con textos grandes ya que existirán numerosas emociones en estos textos y el sistema deberá de elegir correctamente cuál es la que más representa.
- Otra aplicación sería traducir al español las emociones de WordNet Affect y no hacer esta relación con las emociones de Ekman. Se obtendrían muchas más emociones diferentes pero más precisas.
- Al haber utilizado como recursos BabelNet y Babelify, se podría internacionalizar el proyecto. Es decir, se podría trabajar con numerosos

idiomas, ya que estos recursos lo soportan. Se conseguiría un sistema que además de en español, fuese capaz de analizar las emociones en textos en inglés, alemán, italiano, etc.

- Trabajar con multietiqueta. Esto quiere decir, que en vez de devolver sólo una emoción por texto se podría devolver las diferentes emociones que representen ese texto.
- BabelNet es una API que trabaja con datos enlazados. Se podría estudiar alguna aplicación en la que enriqueciese nuestro sistema utilizando estos datos extra que nos da BabelNet.
- Realizar un análisis sintáctico. Al hacer un análisis sintáctico podremos detectar las relaciones de palabras y poder detectar elementos como los intensificadores o negadores que acompañan a la parte de la oración que nos muestra emoción ya que esto podría cambiar el sentido de la oración completamente.
- Analizar emoticonos. Si se va a utilizar para analizar mensajes de redes sociales como es su función en la actualidad, será necesario detectar los emoticonos ya que son muy utilizados y es una de las maneras que utiliza la gente para mostrar su estado de ánimo.

Bibliografía

Baccianella, S., Esuli, A., & Sebastiani, F. (2010, May). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *LREC* (Vol. 10, pp. 2200-2204).

Boldrini, E., Balahur, A., Martínez-Barco, P., & Montoyo, A. (2012). Using EmotiBlog to annotate and analyse subjectivity in the new textual genres. *Data Mining and Knowledge Discovery*, 25(3), 603-634.

Buechel, S., & Hahn, U. (2017). EMOBANK: Studying the Impact of Annotation Perspective and Representation Format on Dimensional Emotion Analysis. *EACL 2017*, 578.

Díaz Rangel, I., Sidorov, G., & Suárez Guerra, S. (2014). Creación y evaluación de un diccionario marcado con emociones y ponderado para el español. *Onomazein*, 1(29).

Navigli, R., & Ponzetto, S. P. (2010, July). BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 216-225). Association for Computational Linguistics.

Navigli, R., & Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193, 217-250.

Redondo, J., Fraga, I., Padrón, I., & Comesaña, M. (2007). The Spanish adaptation of ANEW (affective norms for English words). *Behavior research methods*, 39(3), 600-605.

Strapparava, C., Valitutti, A., & Stock, O. (2006, May). The affective weight of lexicon. In *Proceedings of the fifth international conference on language resources and evaluation* (pp. 423-426).

Strapparava, C., & Mihalcea, R. (2014). Affect detection in texts.

Strapparava, C., & Valitutti, A. (2004, May). WordNet Affect: an Affective Extension of WordNet. In *LREC* (Vol. 4, pp. 1083-1086).

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2), 267-307.

Thelwall, M., Buckley, K., Paltoglou, G. Cai, D., & Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12), 2544–2558.

Thelwall, M. (2013). Heart and soul: Sentiment strength detection in the social web with sentistrength. *Proceedings of the CyberEmotions*, 5, 1-14.

¿Qué es un diagrama de Gantt y para qué sirve? de obs-edu.com

Anna Samiotou. BabelNet's contribution to the fall of Babel Tower de Taus.net.

Desarrollo iterativo e incremental de proyectos ágiles de proyectosagiles.org

El manifiesto ágil - Scrum Manager BoK de scrummanager.net

La importancia de los idiomas en las páginas web de las empresas de Daniel Puzzi: blarlo.com

Paul Ekman: Las 6 Emociones Básicas de serperuano.com

SentimentStrength: <https://github.com/hitesh915/sentimentstrength>

Anexo A: Manual de instalación.

En este manual, se explica el proceso que hay que llevar a cabo para ejecutar el sistema. Se indicarán los pasos a seguir.

A.1. Instalación de Python y sus componentes

Para este proyecto se ha utilizado la versión de Python 3. En primer lugar habrá que entrar en la web de Python (<https://www.python.org/downloads/>) y descargar la versión correspondiente. En nuestro caso se ha utilizado la versión Python 3.5.2.

Para facilitar la instalación de módulos de Python se va a utilizar la herramienta pip 3. Para ello habrá que utilizar los siguientes comandos:

```
sudo apt-get install python3-setuptools  
sudo easy_install3
```

Una vez se tiene instalado pip, se podrán instalar la mayoría de módulos mediante la orden *pip install nombrepaquete*.

Para este proyecto habrá que instalar otros recursos externos como *tweepy* y *flask*. Para ello se utilizarán los siguientes comandos:

```
sudo pip3 install tweepy  
sudo pip3 install Flask
```

Una vez instalados los paquetes, será necesario instalar la base de datos (Anexo A.2) y la instalación de los índices de BabelNet (Anexo A.3). Cuando se hayan completado estas acciones, se podrá iniciar el servidor con la orden:

```
python3 frontend.py
```

Mientras se está ejecutando el servidor, deberá de estar ejecutándose el socket de Java llamado BabelSocket.

A.2. Instalación de la base de datos

La instalación de MongoDB, que es la base de datos elegida, es sencilla. Para ello se utiliza la siguiente orden:

```
sudo apt-get install -y mongodb-org
```

Una vez se tiene instalado el recurso para poder utilizar la base de datos, es el momento de lanzarla a andar. Para ello habrá que utilizar los siguientes comandos:

```
sudo systemctl start mongod
mongod
```

A.3. Instalación de BabelNet

Para la instalación de BabelNet en primer lugar habrá que registrarse en <http://babelnet.org/login> . Una vez se tiene la cuenta habrá que solicitar los índices para consultarlos de manera offline. Para ello habrá que rellenar un formulario indicando para qué van a ser utilizados y otros datos del proyecto.

Una vez realizado este primer paso, si nuestra solicitud ha sido aceptada, recibiremos los índices para descargar.

Una vez se obtienen los índices, habrá que configurar nuestro proyecto JAVA para que consulte nuestros índices y no realice la consulta a los servidores de BabelNet. Para ello habrá que:

- Añadir los ficheros JAR extraídos a nuestro proyecto. Estos ficheros JAR pueden ser descargados de <http://babelnet.org/guide>.

- En el fichero *config/babelnet.var.properties* modificar la ruta de WordNet por donde esté alojada en el equipo sin el sufijo de la versión. Por ejemplo, si está en la carpeta */opt/WordNet-3.0*, nuestro fichero debería de contener:

jlt.wordnetPrefix=/opt/WordNet

- Añadir al fichero *config/babelnet.var.properties* la ruta de los índices para consultar en offline que han sido solicitados a BabelNet.

babelnet.dir=/home/your_user/BabelNet-3.7

Anexo B: Manual de usuario

En primer lugar, habrá que visitar el portal web. Su página inicial es la que se mostrará en la Ilustración B.1.

The screenshot shows a web browser window with the address bar displaying 'localhost:8000'. The main content area has a teal background. At the top center is a dark grey circle containing a white Twitter bird logo. Below this, the text 'MONITOR DE OPINIONES' is written in white, bold, uppercase letters. Underneath the title is a small white star icon and the text 'Analiza las emociones de tu red social preferida - Twitter'. Below this section is a white box with the heading 'PRECÚNTAME' and a small star icon. Inside this box is a search form with four input fields: 'Buscar', 'Desde', 'Hasta', and a radio button group with options 'Analizar cuenta' (selected) and 'En directo'. A green 'Consultar' button is at the bottom of the form.

Ilustración B.1. Monitor de opiniones

Se puede observar cómo hay un formulario para realizar la consulta. El primer campo se introduce la búsqueda a realizar, si se va a analizar a un usuario se deberá de introducir el nombre de dicho usuario sin @. Por ejemplo, si se quiere consultar la cuenta @jordievole en el formulario habrá que introducir *jordievole*. Además, si se decide consultar una cuenta de usuario, habrá que introducir el rango de fecha a analizar. La fecha desde la que se empieza a buscar será obligatoria introducirla; si se decide dejar en blanco la fecha final, por defecto buscará hasta ese mismo día. En caso de que se decida hacer una consulta “En directo”, se deshabilitará los dos campos fecha, ya que se recolectará mensajes escritos en ese momento. En la búsqueda “En directo” se realizará la consulta tal cual sea introducida al sistema.

Si se solicita información offline, se seguirá estando en la página principal, pero aparecerá nueva información. Por un lado, aparecerán distintos gráficos (Ilustración B.2) que representan cómo han ido variando las emociones y los mensajes en los que se ha detectado emoción (Ilustración B.3).

GRÁFICOS

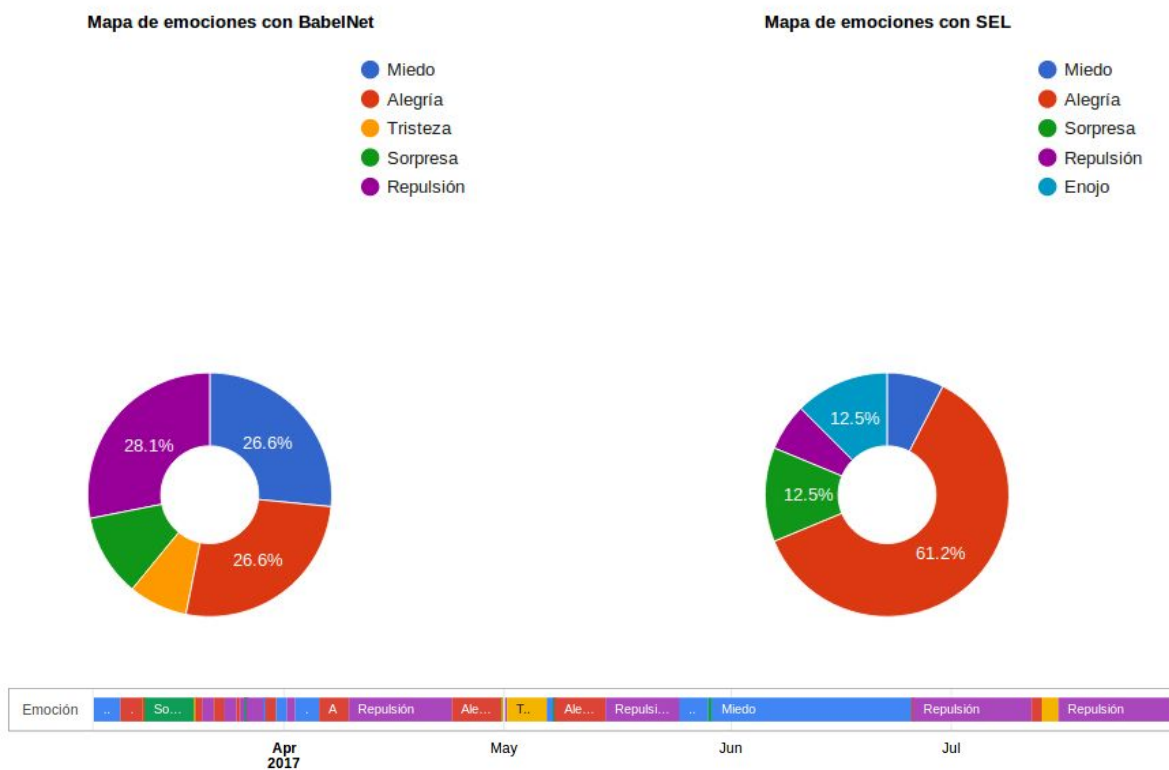


Ilustración B.2. Gráficos en monitor de opiniones



Ilustración B.3. Mensajes en monitor de opiniones

Si se decidiese hacer una consulta online, redireccionará al usuario a la página correspondiente que se irá actualizando y mostrando los mensajes con emoción sobre la consulta realizada (Ilustración B.4).

CRISTIANO RONALDO



Alegría

Máxima expectación: Cristiano Ronaldo, ante el juez | Noticias de hoy El futbolista declara acusado por la Fiscali...
<https://t.co/C0AzuLCxGM>



Repulsión

RT @saitoel: –Don Cristiano Ronaldo, ¿tiene algo que declarar? –Me tienen envidia porque soy bueno, rico y guapo.



Repulsión

Hoy empieza el linchamiento de Cristiano Ronaldo. En este país sabemos cómo acabará : si culpable; ladrón , si inoc...
<https://t.co/sxpcROuJcB>



Alegría

RT @FootballLeaksSp: Tras la declaració de Cristiano Ronaldo, si la jueza estima que hay indicios de delito suficientes, Ronaldo será acusa...



Alegría

FOOTBALL LEAKS - La defensa de Cristiano Ronaldo tiene depositadas buena parte de sus esperanzas en la división... <https://t.co/2zxUxqXHND>



Alegría

FOOTBALL LEAKS - La defensa de Cristiano Ronaldo tiene depositadas buena parte de sus esperanzas en la división... <https://t.co/FnAnWCFUVP>

Ilustración B.4. Mensajes en consulta online

Anexo C: Índice de ilustraciones

Ilustración 1.1. Herramientas de análisis de sentimiento.....	8
Ilustración 1.2. Distribución de usuarios en Internet.....	10
Ilustración 1.3. Idiomas más utilizados en Internet.....	11
Ilustración 1.4. Diagrama de Gantt.....	14
Ilustración 2.1. Triángulo SentiWordNet	20
Ilustración 4.1. Ciclo de vida en iterativo.....	29
Ilustración 4.2. Esquema de aplicación	32
Ilustración 4.3. Esquema Web.....	42
Ilustración 4.4. Esquema Web 2.....	43
Ilustración 4.5. Gráfico Donut.....	49
Ilustración 4.6. Gráfico Timeline.....	49
Ilustración B.1. Monitor de opiniones.....	58
Ilustración B.2. Gráficos en monitor de opiniones.....	59
Ilustración B.3. Mensajes en monitor de opiniones.....	60
Ilustración B.4. Mensajes en consulta online.....	61

Anexo D: Índice de tablas

Tabla 1.1. Planificación del proyecto.....	14
Tabla 1.2. Costes de personal	16
Tabla 1.3. Costes del proyecto	17
Tabla 3.1. Emociones de Ekman	25
Tabla 4.1. Historias de usuario.....	31