UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf
Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Creación de un modelo predictivo para clasificar las consultas ciudadanas sobre el transporte público

Trabajo Fin de Máster
Máster Universitario en Gestión de la Información

Autor: Eduardo Gallardo Pardo
Tutora: Antonia Ferrer Sapena
Tutores: José M. Calabuig Rodríguez
Lluís M. García Raffi

2018-2019

Resumen

Supondría una mejora para los ciudadanos la creación de un modelo matemático predictivo, que proporcionara la inteligencia necesaria para el diseño de un programa *chatbot*. Dicho robot ofrecería un servicio ininterrumpido a la ciudadanía a través de la mensajería instantánea, optimizando tiempos de consulta y costes para la empresa.

Para empezar se han recopilado las consultas ciudadanas enviadas a la Empresa Municipal de Transportes en València (España). Se trata de gestiones de viajeros dentro del ámbito de la movilidad y el transporte público. En general, los viajeros consultan cómo ir de un sitio a otro en la ciudad, tiempo hasta la salida del próximo autobús, incidencias con las tarjetas de transporte, reclamaciones, objetos perdidos, etcétera.

Con ese propósito, se ha analizado el estado del arte en el procesamiento del lenguaje natural y se han prospectado librerías de código libre actuales, basadas en *machine learning* y redes neuronales. En cumplimiento de las leyes españolas se ha procedido a anonimizar los datos. Posteriormente se ha continuado con el análisis y tratamiento de estos.

Finalmente, ha sido creado un modelo que predice y extrae la información relevante de cada consulta, permitiendo su clasificación e integración con otros sistemas. La evaluación del modelo otorga una precisión del 95,22%, teniendo en cuenta la estacionalidad y tamaño de la muestra recogida.

Palabras clave: Transporte público, mensajería instantánea, procesamiento de lenguaje natural, red neuronal, *machine learning*, *chatbot*.

Abstract

It would be an improvement for citizens to create a predictive mathematical model, which would provide the intelligence necessary for the design of a chatbot program. This robot would offer an uninterrupted service to citizens through instant messaging, optimizing consultation times and costs for the company.

To begin with, the citizen queries sent to the company *Empresa Municipal de Transportes in Valencia* (Spain) have been gathered. These are passenger management within the scope of mobility and public transport. In general, passengers consult how to go from one place to another in the city, time until the departure of the next bus, incidents with transport cards, complaints, lost objects and so on. In compliance with Spanish law, the data has been anonymised. Subsequently, the analysis and treatment of these data has been continued.

To this end, the state of the art in natural language processing has been analysed and current open source libraries based on machine learning and neural networks have been prospected.

Finally, a model has been created that predicts and extracts the relevant information from each query, allowing its classification and integration with other systems. The evaluation of the model gives an accuracy of 95.22%, taking into account the seasonality and size of the sample collected.

Keywords: Public transport, instant messaging, natural language processing, neural network, *machine learning*, chatbot.

Tabla de contenidos

Índice de figuras

Índice de tablas

1. INTRODUCCIÓN

1.1 Introducción

En primer lugar, hablamos de los dispositivos móviles. En lo social el uso de esta tecnología no discrimina a ningún colectivo y el poder de comunicación, participación y eco que otorga al individuo es muy relevante. Esto se demuestra en el análisis según Castells:

> A partir de los años noventa se produjo otra revolución de las comunicaciones en todo el mundo: la explosión de las comunicaciones inalámbricas, con mayor capacidad de conectividad y ancho de banda en las sucesivas generaciones de teléfonos móviles. Ha sido la tecnología de más rápida difusión en la historia de las comunicaciones. En 1991 había casi 16 millones de contratos de teléfonos inalámbricos en el mundo. En julio de 2008 se habían superado los 3.400 millones de contratos, casi un 52% de la población mundial. Utilizando un factor multiplicador conservador (los bebés no usan móviles, al menos todavía, y en los países pobres familias y aldeas comparten un único teléfono), podemos calcular sin temor a equivocarnos que más del 60% de la población mundial tenía acceso a las comunicaciones inalámbricas en 2008, aunque esta cifra está muy limitada por los ingresos. Efectivamente, estudios realizados en China, América Latina y África han demostrado que los pobres dan una alta prioridad a sus necesidades de comunicación y utilizan una parte importante de su escaso presupuesto para satisfacerlas. En los países desarrollados, la tasa de penetración de los contratos de telefonía móvil varía entre el 82,4% de Estados Unidos y el 102% de Italia o España, y está llegando al punto de saturación (Pag. 98) (Castells y Hernández 2009).

La obra de Manuel Castells fue escrita en el año 2009. Ya entonces los datos de penetración de la tecnología eran importantes. Diez años después sin tener que investigar demasiado en *Internet,* podemos ver que el crecimiento ha sido exponencial. Muchos más *smartphones*, mucha más Red donde no llegaba y muchas más redes sociales.

Fue en el 2000 cuando se produjo el boom tecnológico, dando pie a la creación de *Facebook* en el año 2004, le siguieron *Youtube*, *Twitter*, *Instagram*, *Linkedin* y otras muchas en lo que también conocemos como *web 2.0*. Se produjo el nacimiento de una nueva *Internet* con una clara componente social.

Hoy en día no solo hablamos de comunicación de las personas con las personas, sino que comenzamos a hablar de comunicación de máquinas con máquinas y de web semántica, *open data* y *open government*. Se habla también de pequeños dispositivos, más pequeños que los *smartphones*, de *Internet of things*, que vuelcan sus mediciones a la Red de forma continua y son capturadas en grandes bases de datos. Y con esos nuevos datos y los que ya se venían guardando bastante tiempo atrás, cantidades ingentes de datos, *big data*, podemos clasificar mejor los datos, podemos optimizar mejor los costes, podemos predecir mejor el futuro, podemos crear inteligencia artificial y podemos crear más robots alojados en la nube. De todo eso que los profesionales del mundo de la información hablan hoy en día, se han adquirido amplios conocimientos en el Máster Oficial Universitario de Gestión de la Información cursado, en adelante *MUGI*, y desencadenante del presente trabajo de fin de máster, en adelante *TFM*.

Según la *Organización de Naciones Unidas* la población mundial en el año 2020 será de 7.800 millones de personas y según *Cisco Systems* tendremos 5.500 millones de dispositivos móviles inteligentes, lo que supondrá un 70% de la población mundial (Prieto, *Cromwell* y *Bashkaran* 2016).

Según distintos estudios realizados, no hay lugar a dudas, aquello que nos parecía a todos algo difícil de alcanzar se ha sobrepasado y se confirma en el siguiente titular: "La red social *Whatsapp* cumplió diez años en febrero de 2019 con 1.500 millones de usuarios en todo el mundo de los cuales 25 millones son españoles *enganchados*" (*RTVE* y *EFE* 2019). Es la aplicación de moda, adalid del éxito de las aplicaciones de mensajería instantánea. Soslayar la utilización de la palabra *enganchados* en el titular anterior, dando a entender que su uso descontrolado conlleva adicción. No en vano, en España, este "uso masivo", ha significado desde hace un tiempo, la inclusión de los términos wasap(ear) y guasap(ear) como adaptaciones válidas en la *Real Academia de la lengua Española* (*RAE*).

Cuesta imaginarse la gran cantidad de información que esos "25 millones de españoles *enganchados"* pueden generar. Si en un día por ejemplo una persona envía de media 25 frases en total tendríamos 625 millones de frases en un día. El almacenamiento de toda esta información por parte de las empresas podría ser de utilidad para su análisis, pero, para empezar, la empresa debería tener, por un lado, una importante cantidad de almacenamiento y por otro salta la cuestión de qué puede hacer con los datos. Sabemos que nuestros datos producen dinero de diversas formas y sabemos que están avanzando muy rápidamente las disciplinas que trabajan con muchos datos para explotar y extraer beneficios económicos, es parte de lo que se denomina *big data*.

Los programas de mensajería instantánea, en general, tienen cuestiones controvertidas con la opinión pública. Esto se puede apreciar en lo referente a la *privacidad y seguridad* de las aplicaciones. Ha crecido la desconfianza en el sentido de la inviolabilidad de las comunicaciones por un tercero, de forma que los datos pudieran ser explotados sin el consentimiento de los autores de una conversación, o incluso peor aún que pudieran ser cedidos los datos personales a otras empresas. Para intentar soliviantar esto, en concreto, *WhatsApp* ha intentado mejorar recientemente la seguridad del envío, garantizando la comunicación extremo a extremo ante un tercero. Aunque expertos en la materia han dictaminado que esa seguridad es insuficiente justamente en la parte de sus servidores ya que no certifican ni el cifrado ni el borrado de los datos como sí lo hace otra empresa de mensajería instantánea competidora: *Telegram*.

Es importante comentar que los programas de mensajería instantánea cubren el espectro de comunicación *interpersonal* y el de comunicación de masas o *unidireccional* (tipo red social). Un ejemplo del tipo de comunicación unidireccional es cuando publicamos los *estados*, imágenes, fotos o textos que definen nuestra situación o estado de ánimo en un momento determinado.

En *Whatsapp*, la incorporación del doble *check* azul suscitó cierta polémica al principio. Esta funcionalidad informa al remitente que el mensaje ha sido leído por el destinatario. Al principio se consideró como algo obligatorio, pero más adelante, tras una nueva actualización, volvieron a hacerlo opcional atendiendo a las reclamaciones de las masas. En la Tabla 1, se muestra la evolución histórica de la aplicación en estos diez años desde su creación, no exenta de polémicas propiciadas por el gran número de usuarios.

Año	*Mes*	*Evento*
2009	Agosto	Lanzamiento *Iphone*
2009	Diciembre	Compartir fotos y vídeos
2010	Junio	Compartir ubicación
2010	Octubre	Lanzamiento *Android*
2011	Febrero	Chats de grupo
2011	Octubre	1.000 millones mensajes/día
2013	Agosto	Mensajes de voz
2014	Abril	500 millones de usuarios
2014	Octubre	Adquirido por *Facebook*
2014	Noviembre	Confirmación lectura mensajes
2015	Enero	*Whatsapp web*
2016	Abril	Cifrado extremo a extremo
2016	Noviembre	Videollamadas
2017	Febrero	Estados
2018	Enero	1.500 millones usuarios/mes
2018	Enero	*Whatsapp Business*
2018	Julio	Llamadas grupales
2018	Octubre	*Stickers*
2019	Febrero	10 años

Tabla 1. Evolución Whatsapp. Fuente: https://blog.whatsapp.com

En la *Empresa Municipal de Transportes de València,* en adelante *EMT,* es muy importante la comunicación con los viajeros. Sin ella sería imposible conseguir la excelencia y la mejora continua en el transporte público. Dentro de *EMT,* es la *Oficina de Atención al Cliente*, en adelante *OAC,* la encargada de articular gran parte de la comunicación con la ciudadanía. Para poder realizar esta labor es condición *sine qua non*, adaptarse a los nuevos canales y formas de comunicación que aparecen en la sociedad. Es aquí donde aparece una problemática relacionada con la forma de comunicarse de cada generación, es decir, vayámonos a los extremos, la gente mayor suele utilizar los medios de comunicación tradicionales: Teléfono y correo ordinario, sin embargo, la gente joven actualmente utiliza la mensajería instantánea.

Dejando de lado la comunicación telefónica y la comunicación por redes sociales, en estos momentos podemos decir que a la *OAC* que llegan dos tipos de comunicaciones escritas interpersonales:

1. Se reciben las denominadas *RQSCF*'s, que es el acrónimo de Reclamaciones, Quejas, Consultas, Felicitaciones y Sugerencias. Estas se reciben la mayoría por correo electrónico y unas pocas por correo ordinario. *EMT* ha adoptado la hoja oficial de reclamaciones de la Generalitat Valenciana. A modo de ejemplo, para poder apreciar de qué volúmenes de consultas estamos hablando se ha creado la Tabla 2.

Tipo	**Total Gestiones**
Reclamación	47
Queja	533

Consulta	257
Felicitación	11
Sugerencia	32

Tabla 2. RQSCFs EMT mayo 2019. Fuente: EMT

2. Desde hace aproximadamente cuatro años, se introdujeron otros medios de comunicación en *EMT*. *WhatsApp* y *Telegram* se incorporaron a través del número de teléfono publicado en la web oficial *www.emtvalencia.es*. Esta forma de comunicación se está imponiendo a las demás, incrementándose poco a poco el número de consultas a través de este medio.

Realmente existen diferencias entre las dos formas de comunicación escritas que se reciben en la *OAC*. Las consultas que se reciben por mensajería instantánea son aquellas que requieren de una contestación casi inmediata de *EMT* porque su razón de ser es que intentan ahorrarse toda la burocracia de la otra vía. En cambio, el formato *email* tiene otros tiempos, con un ciclo de vida más largo, siguiendo un flujo de trabajo en el que intervienen otras áreas y departamentos dentro de *EMT* como por ejemplo el área Técnica (Autobuses), Operaciones (Conductores), Planificación (líneas, rutas) etcétera.

Tal y como hemos empezado también concluiremos la introducción con una cita de Manuel Castells “Hoy en día cualquier mensaje que desea libertad y autonomía no pasa por un partido o un periódico, dicho mensaje se conecta con otras mentes conectadas en las redes sociales, iniciándose un movimiento. Los actores son colectivos, sin papeles, sin jerarquía, sin líderes”.

1.2 Contexto y Motivación

Huelga decir que el transporte público es un servicio básico en cualquier ciudad. Las personas tienen necesidades de desplazamiento para desarrollar sus actividades y los medios de transporte público (metro, taxi, bicicleta, autobús) se complementan para poder satisfacerlas. Hoy en día, además, debe existir una mejor gestión del espacio público en las ciudades y respetarnos mutuamente los peatones, usuarios particulares de coche, transporte público y los nuevos medios de transporte como son el patinete y la motocicleta eléctricos.

Sabemos que el número de viajeros, en un espacio de tiempo dado, es un indicador importante en la marcha de la economía de un país y en ese sentido la crisis económica del 2009 provocó una bajada importante de toda actividad. Paulatinamente se ha ido recuperando el número de viajeros y desplazamientos. En estos momentos, *EMT de València* tiene un papel importante en esa recuperación ya que transportó 96,1 millones de personas durante el año 2018 según la información publicada (EMTValència 2019).

Un poco de historia, en el año 1875 el *Ayuntamiento de València* aprobó el proyecto para la implantación de dos líneas[1] de tranvías una de *València* al *Grao* y otra interior que enlazaba el Puente del Real con la calle de las Barcas, reseñar que los tranvías por

[1] Recorrido aprobado por el Ayuntamiento, normalmente circular, por el que circulan una serie de autobuses en una determinada frecuencia.

aquel entonces eran de tracción animal. En 1885 se fundó la *Sociedad Valenciana de Tranvías* con la concesión por parte del *Ayuntamiento* de dos líneas más, en 1886 entró en funcionamiento el tranvía a vapor, en años posteriores se constituyen otras compañías de tranvías cubriendo líneas del norte por ejemplo uniendo València con la Pobla de Farnals. En 1898 comienza la electrificación de las primeras líneas de tranvía tras la compra de las compañías de tranvías por una francesa, en 1917 la compañía francesa pasó a denominarse *Compañía de Tranvías y Ferrocarriles de València* (*CTFV*), aunque no sin dificultades derivadas de la I Guerra Mundial, las labores de electrificación de los tranvías prosiguieron y el *Ayuntamiento de València* concedió varias líneas más. En 1927 el *Ayuntamiento* concedió el establecimiento de un servicio de autobuses dentro del casco urbano y para paliar esta competencia el *CTFV* constituyó la *Valenciana Autobuses Sociedad Anónima* (*VASA*). En 1963 se celebró la constitución la *Sociedad Anónima Laboral de Transportes Urbanos de València* (*SALTUV*) haciéndose cargo del transporte urbano en València. En 1986 el *Ayuntamiento de València* adquirió la totalidad de las acciones de *SALTUV* pasando a denominarse *Empresa Municipal de Transportes de València* (*Wikipedia* 2019) (Busvalencia.com 2014).

Actualmente el autor del presente trabajo es empleado activo en *EMT* lo que ha permitido contrastar de forma más precisa la información que en el presente trabajo se expone. *EMT* da servicio a los ciudadanos de València con cerca de 500 autobuses, alrededor de 1.200 conductores y 57 líneas de las cuales 12 son líneas nocturnas y una es el servicio especial puerta a puerta para personas discapacitadas. Además dispone de tres *OAC's*, situadas en puntos céntricos y estratégicos donde se realizan actividades de atención y gestión a la ciudadanía.

El presente trabajo trata de poner las bases, procedimientos y desarrollos para poder crear un modelo predictivo como antesala para el posterior diseño de un *robot* que pueda llevar a cabo ciertas tareas en la parte *receptora* de la comunicación (Atención al Cliente de *EMT)*.

Para poder crear el modelo existe toda una disciplina científica que estudia esta cuestión, se llama procesamiento del lenguaje natural, en sus siglas en inglés *NLP*. Dicha disciplina es la conjunción de varias ciencias tales como Gestión y Sistemas de Información, Documentación, Inteligencia Artificial y Psicología Cognitiva. *NLP* trata de otorgar a las máquinas la capacidad de leer y comprender el lenguaje humano y donde, por ejemplo, detectar patrones de comunicación es una parte. Más adelante profundizaremos en estos conceptos y otros por los que ha transcurrido el trabajo.

Expuesto todo lo anterior, estamos en condiciones de resumir, a alto nivel, los puntos más importantes de la motivación del proyecto. Los conceptos expuestos hasta ahora serían:

- Evolución de la empresa hacia soluciones inteligentes y modernas que permitan la optimización de los recursos y la energía utilizada, siempre de una forma sostenible.

- Aprovechamiento de disponer de los datos provenientes de las consultas ciudadanas que se realizan diariamente a *EMT.*

- Puesta en práctica de las competencias y conocimientos en gestión de la información aprendidos en el máster *MUGI*.

- Mejora en el servicio ofrecido a los ciudadanos al ser ofrecido por un robot inteligente alojado en la nube y disponible en las *apps* de mensajería instantánea.

- Utilización de las tecnologías actuales: *Smartphones* y aplicaciones (*Apps*) de uso generalizado de comunicación instantánea. En nuestro caso *Whatsapp* y *Telegram*.

- Aprovechamiento de la condición de empleado de *EMT* como responsable de proyectos en soluciones tecnológicas.

- Disposición de herramientas y lenguajes de código libre futuristas y específicos en el ámbito de la inteligencia artificial.

1.3 Objetivos

El objetivo principal del trabajo consiste en crear un modelo predictivo para clasificar automáticamente las consultas ciudadanas que llegan a *EMT*, cuya temática es el transporte público. La automatización consiste en predecir y extraer la información relevante para poder dar una respuesta también automatizada e integrada con otros sistemas de información. La consecución del objetivo supondría poder disponer de la inteligencia necesaria para poder diseñar un *chatbot*. Una vez el *chatbot* estuviera en producción supondría una ayuda en el desempeño de las labores de atención al cliente, beneficiándose todos los ciudadanos de València que quieran utilizar el transporte público y necesiten realizar consultas y gestiones.

Para ello, se dispone de una serie de consultas ciudadanas, en formato texto, recopiladas y posteriormente clasificadas manualmente. La clasificación manual no requiere de un conocimiento exhaustivo de la temática y los tipos de consultas inicialmente son un número acotado. Mediante el uso de las tecnologías y técnicas actuales de gestión de la información se desea desarrollar un modelo de datos matemático que prediga cuales son las palabras clave de una frase.

El modelo pretende construirse con una combinación de técnicas basadas en redes neuronales y modelos estadísticos que le otorguen rapidez por un lado y precisión por otro respectivamente. El alcance del trabajo es llegar a crear dicho modelo y evaluarlo como base para la creación posterior de un *chatbot*, aunque su diseño e implementación no esté en el alcance del trabajo.

Como objetivos complementarios tenemos:

a) Por un lado, establecer los pasos a seguir y puntos para tener en consideración en un proyecto de estas características.

b) Por otro lado, hacer pública toda la información posible relativa a las consultas ciudadanas. Siempre respetando la normativa vigente sobre todo en lo que a protección de datos se refiere, para promover la investigación, la transparencia y retroalimentar aquella información que procede de la ciudadanía que esté disponible para la ciudadanía. Se tienen en cuenta las directrices expuestas en el artículo

"Acceso a los datos públicos y su reutilización: *Open Data* y *Open Government*" (Ferrer-Sapena, Peset, Aleixandre-Benavent 2011).

c) Por último, servir de base de conocimiento y consideraciones para la posible creación de un pliego para un concurso público de diseño de un *chatbot* de atención al cliente en el sector del transporte público.

2. ESTADO DEL ARTE

A continuación, y de forma introductoria se intenta plasmar en este capítulo las diferentes temáticas estudiadas para reunir los conocimientos necesarios. Básicamente dividimos este capítulo en dos grandes partes: La parte legal y ética y la parte técnica y tecnológica.

2.1 Marco legal

Cuando hablamos de datos, uno de los primeros pasos en cualquier nuevo proyecto es verificar el cumplimiento de las leyes y que no se vulnera ningún derecho en ninguno de los ámbitos local, nacional o internacional. Parece que se complica más cuando el proyecto está relacionado con las tecnologías de la información e *Internet*. De este último sabemos que se ha convertido en una realidad omnipresente tanto en nuestra vida personal como colectiva. Una gran parte de nuestra actividad profesional, económica y privada se desarrolla en la Red y adquiere una importancia fundamental tanto para la comunicación humana como para el desarrollo de nuestra vida en sociedad.

Se aprecia que la sociedad de la información está cambiando a un ritmo vertiginoso y que la normativa legal intenta adaptarse lo más rápido posible pero claramente a un ritmo menor. Esto es consecuencia de que por un lado, las novedades tecnológicas han de consolidarse en la sociedad presentando su definición y mostrando poco a poco los problemas derivados de su utilización.

Es cuando el avance de las tecnologías se asienta, adoptadas por la sociedad como nueva forma de comunicarse e interrelacionarse cuando surgen nuevos tipos de delitos que no son sino una transformación de los antiguos delitos. Lo que antes era la estafa de “la estampita” ahora es la estafa electrónica (un usuario virtual, que no existe). Lo que antes era difamación por medio escrito papel o boca a boca ahora es difamación por medios electrónicos que tienen una visibilidad y repercusión mucho mayor, si es a través de las redes sociales.

Todo ello hay que legislarlo, creando o adaptando nuevos Reglamentos, Directivas, Leyes Orgánicas, Leyes, Reales Decretos, Órdenes, Normas Técnicas o, por qué no también, reformando la Constitución Española.

A continuación, se enumeran, sin la intención de ser una lista exhaustiva, algunas leyes que están adaptándose a los tiempos actuales y que están relacionadas con las tecnologías de la información y con la materia del trabajo. La información, en general, se ha obtenido del portal de administración electrónica del Gobierno de España. (Gobierno de España n.d.)

- **La protección de datos**: Regulada por la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y Garantía de los Derechos Digitales y el Reglamento Unión Europea 2016/679 del Parlamento Europeo. Regula lo que respecta al tratamiento de datos personales y a la libre circulación de esos datos. Ha tenido mucha repercusión por el cambio que implica en las Administraciones públicas, empresas públicas y empresas privadas ya que supone una gran inversión de recursos para poder aplicarla correctamente. Además, en muchas ocasiones valga la metáfora de que esta ley “choca de

frente" o por lo menos, "se lo pone muy difícil" al Real Decreto de Interoperabilidad. Aunque sabemos que el rango de la Ley Orgánica es mayor que el rango de un *RD* así que, en caso de duda, hay que obedecer la primera.

- **La propiedad intelectual**: Regulada por el Real Decreto Legislativo 1/1996, de 12 de abril, de la Propiedad Intelectual protegiendo los derechos de autor, morales y patrimoniales, las copias privadas fundamentadas en el derecho a la intimidad y especial hincapié en la protección jurídica del software como una creación artística o científica expresada en cualquier medio y/o licenciamiento.

- **La transparencia política y el buen gobierno**: Regulada por la Ley 19/2013, de 9 de diciembre, de Transparencia, acceso a la información pública, datos abiertos, y el buen gobierno de aplicación a los partidos políticos, organizaciones sindicales y empresas. Se toman medidas sancionadoras sobre los gobernantes para que los ciudadanos cuenten con servidores públicos que ajusten sus actuaciones a los principios de eficacia, austeridad, imparcialidad y sobre todo responsabilidad.

- **La intermediación de datos**: Regulado por el Real Decreto 4/2010, de 8 de enero, por el que se regula el Esquema Nacional de Interoperabilidad en el ámbito de la administración electrónica, el cual establece una serie de normas técnicas de interoperabilidad que son de obligado cumplimiento por las AA.PP. y que desarrollan aspectos concretos de la interoperabilidad entre las AA.PP. y con los ciudadanos.

- **La accesibilidad**: Regulada por el Real Decreto 1494/2007, de 12 de noviembre, por el que se aprueba el Reglamento sobre las condiciones básicas para el acceso a las personas con discapacidad a las tecnologías, productos y servicios relacionados con la sociedad de la información y medios de comunicación social, de aplicación en sitios *web* y aplicaciones para *smartphones* (*Apps*).

- **Las telecomunicaciones**: Regulada por la Ley 9/2014, de 9 de mayo, General de Telecomunicaciones, por el que se regulan las comunicaciones electrónicas, derechos y obligaciones de operadores y usuarios.

- **El procedimiento administrativo común de las Administraciones Públicas**: Regulado por las Leyes 39 y 40/2015 que contemplan que la tramitación electrónica debe constituir la actuación habitual de las Administraciones Públicas para servir mejor a los principios de eficacia, eficiencia, el ahorro de costes, a las obligaciones de transparencia y a las garantías de los ciudadanos.

- **Esquema Nacional de seguridad**: Regulado por el Real Decreto 3/2010, de 8 de enero, trata de garantizar la seguridad de los sistemas, los datos, las comunicaciones y servicios electrónicos que permita a los ciudadanos y a las Administraciones públicas el ejercicio de derechos y el cumplimiento de deberes a través de estos medios.

- **Sociedad de la información y telecomunicaciones**: Regulada por la Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico, trata de amparar a los consumidores regulando a los prestadores de servicios por Internet.

La normativa que afecta a los derechos que podrían ser vulnerados en el caso de las consultas ciudadanas mediante mensajería instantánea es la que figura a continuación.

Para su elaboración nos hemos basado en una publicación de la Nueva Revista Española del Derecho del Trabajo sobre el uso del *Whatsapp* en las relaciones laborales (Garrido 2014):

a) **Derecho al secreto de las comunicaciones**: protegiendo la intimidad de las personas de forma que los mensajes enviados al destinatario, pero aún no leídos por este, deben entenderse como protegidos por el derecho al secreto de las comunicaciones. Para nuestro trabajo se interpreta que no hay un tercero que intercepta los mensajes entre ciudadano y *EMT*, ya que los *chats* han sido recopilados o interceptados por un interlocutor de las partes en la comunicación.

b) **Derecho a la intimidad**: garantizando el derecho al honor, a la intimidad y la propia imagen, es necesario el consentimiento de la persona para acceder a cierta información calificada como información perteneciente a la esfera íntima. En nuestro trabajo se interpreta que el ciudadano envía de *motu propio* una consulta a la *EMT* dando su consentimiento.

c) **Derecho a la protección de datos**: derecho a ser informado de quien posee sus datos personales, a qué uso se están sometiendo y a oponerse, en su caso, a una posesión ilegítima o uso ilícito, en definitiva, es la potestad de control sobre el uso de los datos propios con el propósito de impedir un tráfico lesivo para la dignidad de la persona afectada. Por último, se procede a definir qué es el dato personal de forma global:

   El dato personal se define como *cualquier información concerniente a personas físicas identificadas o identificables*, lo que abarca a información de cualquier tipo, vídeos, fotos, audios y textos.

En el presente trabajo se produce una recopilación de las consultas ciudadanas, del orden de 1.500 mensajes y en cada uno de ellos originalmente aparece un número de teléfono. Este número se considera un dato personal según la definición anterior, revisamos y analizamos el Reglamento General de Protección de Datos Personales, en adelante *RGPD.* (Agencia Estatal Boletín Oficial del Estado 2016)

Por un lado, indica el *RGPD*, que hay que analizar la satisfacción del interés legítimo perseguido en el objetivo del tratamiento de la información. En nuestro caso la información recopilada trata sobre el transporte público y por ahí se podría justificar el tratamiento de las consultas ciudadanas, obteniendo como resultado un beneficio para sociedad fruto de la investigación científica de interés público general.

Por otro lado, no disponemos de consentimiento de las personas que realizan las consultas para realizar este tratamiento. Razón por la que es necesario realizar un proceso de disociación que desvincule el dato con la identidad de la persona y protegiendo así los datos personales. Aparece el concepto de *seudonimización* de manera tal que no puedan atribuirse a un interesado sin utilizar información adicional y teniendo en cuenta factores objetivos de tiempo y recursos según los avances tecnológicos para descifrar y poder identificar a la persona. Normalmente se utiliza la *seudonimización* cuando después del tratamiento es necesario volver a identificar a la persona. En nuestro caso no es necesario realizar ese proceso de vuelta atrás por lo que la técnica a utilizar que se adapta mejor a nuestro caso es la *anonimización*.

La Agencia Española de Protección de datos, en adelante *AEPD,* resalta, en cuanto a la selección de las técnicas de *anonimización*, la utilidad de los algoritmos de cifrado para este tipo de procesos. *AEPD* destaca que los algoritmos de ***hash*** son la fórmula para garantizar la confidencialidad del dato por tratarse de una operación en un solo sentido, es decir, partiendo de un dato podemos generar siempre la misma huella digital, pero partiendo de una huella digital nunca podremos obtener el dato original (Aced, Heras y Alberto 2015).

Para el presente trabajo se debe realizar un proceso de *anonimización* teniendo en cuenta los principios para cumplir la protección de datos desde el diseño (*AEPD* 2016):

- **Principio proactivo**: La protección de la privacidad es el primer objetivo de la *anonimización* y debe realizarse de forma proactiva y no reactiva. En primer lugar, hay que elaborar un análisis de riesgos. Se aboga por aplicar el principio de privacidad por defecto y que el diseño de la *anonimización* garantice la confidencialidad de los interesados.

- **Principio de privacidad objetiva**: Existe un riesgo residual de *reidentificación* que debe ser asumido por el responsable de tratamiento.

- **Principios de plena funcionalidad**: Desde el inicio del diseño del sistema de *anonimización* se tendrá en cuenta la utilidad final de los datos *anonimizados*

- **Principio de privacidad en el ciclo de vida de la información**: Las medidas que garantizan la privacidad de los interesados son aplicables durante el ciclo completo de la vida de la información.

- **Principio de información y formación**: Para garantizar la privacidad de los interesados en la formación e información que se facilite al personal involucrado en el proceso de *anonimización*.

A continuación, se resumen las fases que recomienda *AEPD* en todo proceso de *anonimización*:

- **Identificación y categorización de activos implicados en el proceso de anonimización**: Identificar los elementos implicados en el proceso de *anonimización*, datos personales, grado de sensibilidad de la información, *hardware* utilizado, etc.

- **Constitución del equipo de trabajo**: Distribución de roles y responsabilidades, equipo multidisciplinar.

- **Identificación de los riesgos**: Catalogación teniendo en cuenta los riesgos conocidos, los riesgos potenciales y los riesgos no conocidos.

- **Valoración de los riesgos existentes**: Atendiendo al conjunto de activos y al catálogo de riesgos existentes se realizará una categorización de cada uno de los riesgos de *reidentificación* que hubieran sido detectados.

- **Salvaguardas**: Para cada uno de los riesgos identificados se deben proponer varias salvaguardas. Cada una de ellas tiene un coste asociado.

- **Cuantificar el impacto**: Consistente en medir el impacto de materialización de un riesgo (daños materiales, pérdida de confianza, indemnizaciones, etc.).

- **Informe de riesgos**: Informe de los puntos anteriores de carácter ejecutivo.

- **Determinación del umbral de riesgos aceptable**: El responsable del tratamiento a propuesta del equipo de evaluación de riesgos, del equipo de *anonimización* y del equipo de seguridad de la información serán en última instancia quienes decidan sobre los riesgos aceptables resultantes del proceso de *anonimización*.

- **Gestión de los riesgos asumibles**: Para cada uno de los riesgos que hubieran sido determinados como asumibles se establecerán medidas encaminadas a atenuar el posible impacto para la privacidad de las personas que pudieran ser *reidentificadas*.

- **Informe Final**: Las posibles medidas que se establezcan serán conocidas por todas las personas implicadas en los procesos de *anonimización*.

- **Revisión de riesgos**: El análisis de riesgos debe realizarse de forma periódica a lo largo del ciclo de vida de la información y siempre que se produzcan cambios en la fuente de datos.

Por último, se resumen los consejos para la selección de las técnicas de *anonimización*:

- **Algoritmos de *hash***: Son muy útiles estos algoritmos en estas cuestiones, sin embargo, un algoritmo *hash* por sí sólo no es suficiente para hacer irreversible la *anonimización* en el caso de que haya que utilizarlo con pequeñas cadenas de texto. Pequeñas cadenas de texto denominados *microdatos* como por ejemplo los códigos postales que son fácilmente *reidentificables* con un programa informático que genere cifras consecutivas y sus correspondientes huellas digitales. Para estos *microdatos* hay que utilizar el algoritmo *HMAC* que puede ser utilizado en combinación con algoritmos *hash* como por ejemplo el *MD5* (*Ius Mentis law and technology explained* 2005).

  La función *MD5* es un algoritmo criptográfico que toma una entrada de longitud arbitraria y produce un resumen de mensajes de 128 bits de longitud. El resumen también se denomina a veces *hash* o "huella dactilar" de la entrada. *MD5* se utiliza en muchas situaciones en las que es necesario procesar y/o comparar rápidamente un mensaje potencialmente largo. La aplicación más común es la creación y verificación de firmas digitales. *MD5* fue diseñado por el conocido criptógrafo *Ronald Rivest* en 1991. En 2004, se encontraron algunos defectos graves en el *MD5*. Todavía no se han determinado todas las implicaciones de estos defectos.

- **Algoritmos de cifrado**: O algoritmo de cifrado homomórfico, permite realizar operaciones con datos cifrados de tal manera que el resultado de las operaciones es el mismo que si las operaciones se hubieran realizado con los datos sin cifrar.

- **Sello de tiempo**: Cabe la posibilidad de añadir el *timestamp* en el proceso de *anonimización* con el fin de garantizar la fecha y la hora en la que la *anonimización* ha sido realizada.

- **Capas de anonimización:** Consiste en añadir procesos de *anonimización* anidados.

- **Perturbación de datos:** Variación y supresión de datos que evita que las cifras resultantes faciliten información sobre casos específicos. Permutación de registros, *microagregación*, intercambio aleatorio de datos, redondeo, ruido aleatorio, distorsiones de datos, etc.

- **Reducción de datos:** Se reduce el número de datos originales sin alterar los mismos, reduciendo variables y /o reduciendo registros.

2.1.1 Jurisprudencia

A modo de ejemplo, por la vía administrativa, se comenta una resolución de la *AEPD*, del 26 de marzo de 2018. Que siendo su directora Mar España Martí resuelve que el *Instituto de Salud Pública y Laboral de Navarra* infringió lo dispuesto en los artículos 4.2 y 10 de la *LOPD* tipificadas como infracción grave, hay que tener en cuenta que todavía no estaba en marcha la nueva *LOPD* basada en *RGPD*, ya que surgió poco después en mayo 2018.

En el procedimiento quedó acreditado que el citado Instituto creó, a principios de febrero de 2017, un grupo de *Whatsapp* con los números de teléfonos móviles de 30 funcionarios en prácticas. Se creó el grupo para comunicarles las citaciones para las revisiones médicas que debían realizar en fase de prácticas previas para acceder a los puestos de trabajo. Resultó que expusieron los datos de carácter personal entre los propios miembros del grupo sin autorización de los participantes. Los datos personales que quedaron al descubierto fueron los números de teléfono, fotos de perfil y nombre de usuarios utilizados por estas 30 personas (Aranzadi Instituciones 2019).

2.2 Ética y moral

Como continuación al apartado anterior, el que se inicia ahora pretende profundizar un poco más, tratando de reflejar los efectos, fines y posibles daños colaterales que un proyecto de este tipo puede infligir en la sociedad. La idea central es la siguiente: “los robots están sustituyendo y van a sustituir cada vez más a las personas en determinadas tareas”, o dicho de otra forma, están desapareciendo y van a desaparecer puestos de trabajo. Aparte de la repercusión macroeconómica que tiene en un país, el presente trabajo quiere poner el foco en el hecho de que la brecha entre las personas que tienen menos recursos y los que tienen más, se va a hacer más grande. Y todo ello podría desembocar en otros problemas mayores.

De hecho, según un informe realizado por profesores de la universidad de *Oxford* (*Bruccoleri* et al 2018), el 47% del empleo total está en situación de alto riesgo ya que muchas de sus ocupaciones serán susceptibles de ser automatizadas en una o dos décadas. En una primera fase están en riesgo la mayoría de los trabajadores del sector transporte y de la logística, así como administrativos y en general todos los empleos relacionados con la oficina y también todos los vinculados con los procesos de fabricación y producción.

En una segunda fase estarán en riesgos todos los puestos de trabajo del sector servicios, ventas y ocupaciones de la construcción. En general todos esos puestos de trabajo darán paso a otros nuevos como ya pasó en la anterior revolución industrial

donde los herreros pasaron a ser mecánicos, pero sí que es cierto que aquellos puestos de trabajo que no corren tanto riesgo son aquellos que requieren aplicar inteligencia, creatividad o un alto nivel de complejidad o destreza. Lo ratifica Andrés *Oppenheimer* en su libro "¡Sálvese quien pueda!" donde dice que los populismos han ocultado la revolución de los robots; es más fácil culpar a un mexicano de la falta de empleo (*Chiappe* 2019).

En general, la percepción es que esta cuestión tiene una carga negativa en la motivación de las personas que trabajan en la evolución de los sistemas de inteligencia artificial. Las empresas tenderán a invertir en más robots porque a la larga abaratarán costes. Debemos velar por el cumplimiento de un código ético en la finalidad de los programas. En el mundo de la informática son famosos los códigos éticos de las asociaciones *ACM*[2] y del *IEEE*[2], el decálogo de este último se detalla a continuación (IEEE and UNED n.d.):

1. Aceptar la responsabilidad en la toma de decisiones de ingeniería consecuentes con la seguridad, salud, y bienestar de las personas, y revelar rápidamente los factores que pudieran poner en peligro a las personas o al entorno.

2. Evitar conflictos de intereses reales o percibidos siempre que sea posible y revelarlos a las partes afectadas cuando existan.

3. Ser honestos y realistas en las reclamaciones declaradas o estimadas basadas en datos disponibles.

4. Rechazar los sobornos en todas sus formas.

5. Mejorar la comprensión de la tecnología, su aplicación apropiada y sus consecuencias potenciales.

6. Mantener y mejorar nuestra competencia técnica y emprender tareas tecnológicas para otros sólo si están cualificadas por la experimentación o la experiencia, o después de revelar completamente las limitaciones pertinentes.

7. Observar, aceptar y ofrecer críticas honestas de los trabajos técnicos, reconocer y corregir errores, y acreditar apropiadamente la contribución de otros.

8. Tratar justamente a todas las personas, sin distinción de factores como la raza, la religión, el sexo, la discapacidad, la edad o su país de origen.

9. Evitar injurias a otros, su propiedad, reputación o empleo, mediante acciones falsas o maliciosas.

10. Asistir a colegas y compañeros de trabajo en su desarrollo profesional, y darles soporte en el seguimiento de este código ético.

Enlazando con el código ético en las empresas, cada vez más tenemos la obligación moral de que los sistemas que creamos sean sostenibles y que no influyan negativamente en ningún ámbito local, nacional o internacional. Según Juan Vicente Oltra (2017) "las Naciones Unidas jugaron un papel clave colocando la sostenibilidad en la agenda de las grandes empresas y organizaciones públicas", se trata del *Global Compact* (United Nations n.d.), los principios por los que aboga son:

- Las empresas deben apoyar y respetar la protección de los *Derechos Humanos* reconocidos internacionalmente.

- Asegurarse de que no son cómplices en la vulneración de los *Derechos Humanos*.

- Las empresas deben defender la libertad de asociación y el reconocimiento efectivo del derecho a la negociación colectiva.

- La eliminación de todas las formas de trabajo forzoso y obligatorio.

- La abolición efectiva del trabajo infantil.

- La eliminación de la discriminación en materia de empleo y ocupación.

- Las empresas deben apoyar un enfoque preventivo frente a los problemas medioambientales.

- Emprender iniciativas para promover una mayor responsabilidad medioambiental.

- Fomentar el desarrollo y la difusión de tecnologías respetuosas con el medio ambiente.

- Las empresas deben trabajar contra la corrupción en todas sus formas, incluidas la extorsión y el soborno.

Lo deseable sería que todas las empresas adquirieran y asumieran el denominado ***Balance Social***. Es decir, no sólo realizar el ejercicio anual de balance económico de la empresa, el cual es el principal, sino también ir alimentando un sistema de información empresarial que refleje el estado de las relaciones de la empresa con la sociedad, su entorno y sus grupos internos y externos, con el fin de proporcionar una información transparente de todas las aportaciones que preste a la sociedad y al medio ambiente.

2.2.1 Problemática en *EMT*

En *EMT*, se ha contado, desde un primer momento, con el beneplácito de Dirección-Gerencia para acometer el presente trabajo de fin de máster. También, se ha podido comprobar desde un primer momento que, una vez conocida la finalidad del proyecto, éste alteraba el clima de serenidad del personal contratado para realizar las labores de gestión y atención al cliente con el programa de mensajería instantánea.

Hay que decir que se recopilaron en varias sesiones los *chats* directamente del programa para después poder comenzar el análisis (más adelante se explicará esta cuestión). Aunque realmente la finalidad del proyecto no era directamente la creación de un robot, ya se intuía que tarde o temprano, apoyado por los avances en el presente trabajo, llegaría un *chatbot* que viniera a sustituir el puesto de trabajo de alguna persona. De ahí que totalmente justificada la desconfianza generada por tal situación.

En *EMT*, existe un convenio de empresa y trabajadores por el cual se intentan proteger ciertos derechos de los trabajadores, también existen comisiones de igualdad, promociones de la salud, política de gestión ambiental y eficiencia energética, se cumple la norma *UNE-EN 13816* y *UNE-EN ISO 9001*, seguridad y salud *OHSAS 18001:2007.*

Pero hasta donde se ha podido conocer, *no* existe en la empresa un código ético *ni* balance social, por lo menos documentado, más allá de la gestión ambiental y eficiencia energética, que recoja la problemática social planteada en este trabajo.

Somos conscientes de la controversia actual, aunque añeja, de los puestos de trabajo cuyas tareas pueden ser automatizadas por las máquinas. La tendencia está clara pues hemos visto que la población mundial crece, el número de máquinas y *smartphones* crece y el número de tareas automatizadas y automatizables crece. Por lo tanto, existe un desequilibrio evidente entre aumento de personas y disminución de puestos de trabajo.

2.3 Procesamiento del lenguaje natural: *Chatbots*

Actualmente no es trivial elaborar un programa basado en inteligencia artificial que sea capaz de dar una respuesta eficiente y satisfactoria a cualquier situación conversacional que le plantee un ser humano en donde intervienen innumerables condicionantes (temática, vocabulario, sintaxis, …). Sin embargo, la situación se torna más accesible cuando la temática de las conversaciones resulta acotada como es el caso de un *chat* de consultas sobre un servicio concreto, en nuestro caso el del transporte público.

Aunque no es la meta del trabajo diseñar y programar un *chatbot* en producción, sí se tiene por objetivo poner las bases para su creación. Nuestro modelo predictivo clasificará la temática de la conversación o del mensaje, paso previo imprescindible para después poder emitir una respuesta acorde.

Conviene en primer lugar definir que es un *chatbot*, según *Techopedia* un *chatbot* es un programa de inteligencia artificial (*IA*) que simula una conversación humana interactiva utilizando frases clave de usuario pre-calculadas y señales auditivas o basadas en texto. Los *chatbots* se utilizan con frecuencia para sistemas básicos de servicio al cliente y *marketing* que frecuentan los centros de redes sociales y los clientes de mensajería instantánea. También se incluyen a menudo en los sistemas operativos como asistentes virtuales inteligentes (*Techopedia Inc* 2019).

Un *chatbot* es también conocido como una *entidad de conversación artificial*, robot de *chat*, *TalkBot*, Agente de conversación, Asistente personal por voz, *chatterbot* o *chatterbox*. Los primeros *bots* nacieron como un juego, luego fueron muy comunes en los *chats* de *IRC*, hasta ir extendiéndose poco a poco, algunos de los más famosos se pueden observar en la Tabla 3:

Chatbot	Año	Autor/es	Comentarios
Eliza	1966	*Joseph Weizenbaum, MIT*	Fue uno de los primeros en procesar el lenguaje natural
SHRDLU	1970	Terry Winograd, MIT	Incluía una memoria básica que contenía el contexto de la conversación.
Parry	1972	*Kenneth Colby, Stanford University*	Intentaba simular a una persona con esquizofrenia.
Racter	1984	*William Chamberlain, Thomas Etter*	Es similar a Eliza, es decir puedes conversar con él hasta el aburrimiento, aunque *Racter* no está del todo cuerdo y puede hacer que la conversación sea más divertida.

Jabberwacky	1997	*Versatility.com*	Almacena todo lo que todo el mundo ha dicho, y encuentra la cosa más apropiada para decir usando técnicas de concordancia de patrones contextuales.
SmarterChild	2001	Peter Levitan, *ActiveBuddy*	Utilizado ampliamente por *AOL*, *Microsoft*, *Yahoo!*

Tabla 3. Relación de primeros chatbots. Fuente: Elaboración propia

No hay que olvidarse de los asistentes personales por voz, estos pueden considerarse dentro de la misma familia de los *chatbots* donde la única diferencia es el uso de la voz o no. Aunque resulta evidente la comodidad de poder liberar las manos y mediante la voz poder dar instrucciones para la realización de ciertas tareas como por ejemplo activar alarmas despertadoras, llamar telefónicamente a amigos, convocar una reunión, realizar ciertas consultas a Internet, o simplemente poder interactuar y hablar con alguien o algo (este último caso de una forma bastante limitada por el momento), no esperemos tener profundas conversaciones con nuestro asistente personal de voz. Hay que decir sin embargo que los grandes expertos en inteligencia artificial vaticinan grandes avances en el aprendizaje del lenguaje, todo ello debido a sendos avances en las técnicas de reconocimiento de voz y el poder de predicción de las redes neuronales. Dada la importancia este último, se tratará en capítulo aparte.

Tenemos varios ejemplos de asistentes personales que mucha gente va conociendo y que también poco a poco la gente va utilizando y adquiriendo destreza en su manejo. Hay que señalar que algunos vienen de la mano de los sistemas operativos más utilizados como es el caso de *Cortana* (*Microsoft*), *Siri* (*Apple*) y otros son módulos aparte o pueden ser incorporados a posteriori en otros sistemas como por ejemplo *Alexa* (*Amazon*), *Now* (*Google*) y *Bixby* (*Samsung*). Vamos a mostrar con más detalle que nos pueden ofrecer estos asistentes por voz, en la Tabla 4 se muestra una clasificación y comparación de los asistentes de voz actuales señalando las funciones que incorporan.

	Cortana	***Siri***	***Now Google***	***Alexa***	***Bixby***
Apps	Sí	Sí	Sí	Sí (*FireTV*)	*Android* apps
Predicción tiempo	Sí	Sí	Sí	Sí	Sí
Calendario	Sí	Sí	Sí	Sí	Sí
Alarmas	Sí	Sí	Sí	Sí	Sí
Modo escritura	Sí	No	Sí	No	Sí
Recordatorios	Sí	No	No	Sí	Sí
Acceso a funciones con apps	Sí	Sí	Limitado	Limitado	Sí
Llamadas	Sí	Sí	Sí	Sí	Sí
Enviar mensajes y correos	Sí	Sí	Sí	Sí	Sí
Música	Sí	Sí	Sí	Sí	Sí
Reconocer música	Sí	Sí	Sí	No	No

Búsquedas en Internet	Bing	*Bing Wolfrang Alpha*	*Google*	Personalizable (*Google* por defecto)	*Google*
Sentido del humor	Sí	Sí	No	Sí	Sí

Tabla 4. Comparativa de asistentes personales. Fuente: Estefanía Oliver DigitalTrends

Actualmente el reconocimiento y síntesis de voz tiene un 99% de precisión. Esto se ha conseguido gracias a un arduo trabajo de los desarrolladores y expertos matemático-estadísticos. Cuando se realiza un reconocimiento de voz desde la web, de modo breve estos serían los pasos que se siguen: El sonido captado es agrupado en fonemas. Cada lenguaje tiene una colección de fonemas; por ejemplo, el inglés tiene 44 fonemas. A partir de esta colección de fonemas se convierte a texto respetando rigurosamente su orden y se transmite al servidor y es ahí cuando empieza todo el trabajo específico de análisis con sofisticados programas de tratamiento de audio apoyado con consultas a bases de datos que tienen en cuenta el contexto y los patrones habituales. Puede que haya sonidos de palabras que no se han captado bien pero que, gracias al resto de la frase, al contexto y a los patrones habituales, se les acaban dando significado. Finalmente se transmite el resultado. Todas estas operaciones se realizan prácticamente en tiempo real.

Inicialmente se conseguían precisiones del 95% utilizando los modelos estadísticos basados en acústica y léxica después llegó el modelo oculto de *Markov* que optimizaba considerablemente todos los anteriores sobre todo con la voz continua. Pero ahora, gracias al *Deep Learning*, se ha alcanzado una precisión del 99% (GeitGey 2016) en el reconocimiento de voz. Ese 4% más es un salto muy importante. Es el caso del algoritmo *Connectionist Temporal Classification* (Graves et al. 2006). En la Ilustración 1 se muestra el esquema mencionado.

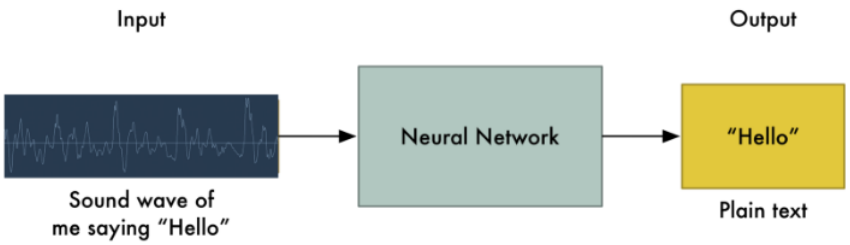


Ilustración 1. Reconocimiento de voz con redes neuronales. Fuente: Geitgey, Adam

2.3.1 Disciplinas que estudian la creación de *chatbots*

Son aquellas que permiten su desarrollo y evolución, donde el objetivo fundamental de cualquier *chatbot* es poder interpretar el motivo o intención de la conversación, en inglés *intent*, entender las respuestas de un ser humano y en base a ellas decidir qué debe responder o qué acción/es debe tomar a continuación.

Dicho de otra forma, consiste en recabar información relevante del interlocutor ya sea una persona u otra máquina para poder llevar al éxito una conversación. Una conversación se puede catalogar como exitosa cuando la persona obtiene

satisfactoriamente la información que pretendía o simplemente que el dialogo sea agradable y se disfrute de la experiencia.

Para conseguir esto, tenemos una de las ramas de las ciencias de la computación más amplia, la inteligencia artificial. No podemos sino citar en este trabajo al padre de esta ciencia, *Alan Turing*, que en 1950 ya se planteaba la pregunta: "¿Pueden las máquinas pensar?". De la inteligencia artificial cuelgan segregadas otras materias importantes.

En la Ilustración 2 se plasman las disciplinas y su relación de interdependencia en el mundo de la inteligencia artificial que se han visto involucradas en este trabajo, dicha ilustración es fruto de la interpretación construida a tenor del conocimiento adquirido en las materias, el razonamiento y el intento de dar una visión global. Hay que señalar que evidentemente no se incluyen otras ramas pertenecientes a la inteligencia artificial, por no ser objeto de alcance del proyecto.

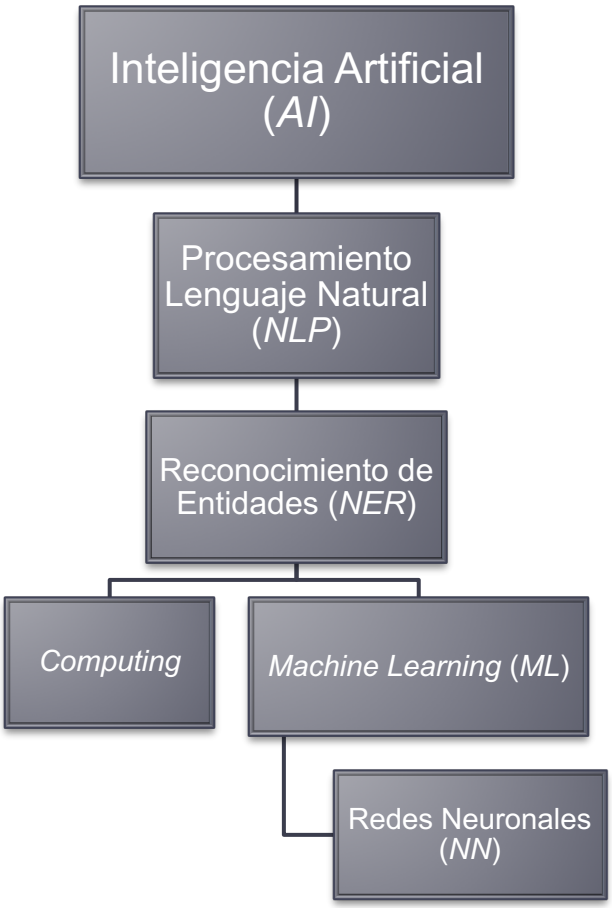


Ilustración 2. Ramas científicas involucradas. Fuente: Elaboración propia

2.3.2 Creación de los *chatbots*

Sirva la Ilustración 2 para explicar los dos caminos o clasificaciones iniciales que tenemos para crear un *chatbot*. Por un lado, tenemos la opción clásica pero un poco menos eficiente, que en este trabajo se ha denominado *Computing*. Y, por otro lado, tenemos las opciones basadas en técnicas de *Machine Learning (ML)*, siendo estas más eficientes en general.

- ***Computing***: Su principal cualidad es programar en forma de árbol todas las opciones que el interlocutor tiene para conversar sobre un determinado tema o una determinada gestión con el objetivo de que el programa pueda satisfacer la necesidad de su interlocutor. La parte negativa es que el programa sólo es capaz de gestionar un espectro limitado de preguntas o palabras clave inicialmente seleccionado y donde la eficacia del *chatbot* decae mucho cuando la intención, usando esas mismas palabras clave, es otra diferente.

  Existen productos en el mercado especializados en esta técnica para crear *chatbots*, por ejemplo, *chatfuel* (Chatfuel 2015) que, aunque no dejan de ser muy buenas herramientas y muy utilizadas por muchas otras empresas, requieren mucho trabajo de programación para hacerlas inteligentes y en general son un poco menos eficientes que aquellos elaborados con *machine learning*. Muchas veces esta carencia es suplida con la combinación de otras técnicas que proporcionan una forma más sencilla para dotarlas de inteligencia.

- ***Machine learning***: La idea principal es que utilizan técnicas de aprendizaje automático para comprender las expresiones del lenguaje natural, hacerlas coincidir con las intenciones y extraer datos estructurados (en sus siglas en inglés *NER*, *Named Entity Recognition*).

  Tratan de construir el diálogo guiado, *workflow*, basadas en las entidades capturadas en la respuesta anterior. Los problemas aquí son que las herramientas suelen ser muy engorrosas para introducir todas las formas de decir algo. No se tiene en cuenta el contexto de la frase. Una vez que se ha detectado la intención de la conversación, si luego es errónea va a ser muy difícil rectificar y el diálogo entero será erróneo.

  También existen en el mercado productos y herramientas para crear *chatbots* inteligentes basada en *machine learning – patterns*, es el caso de *Dialogflow* (*Dialogflow* n.d.) adquirido recientemente por *Google* y que se puede utilizar libremente.
  Se han realizado pruebas con dicha herramienta y se han podido comprobar dos dificultades a reseñar:

  1. La primera a nivel funcional, si no existe el patrón dado de alta manualmente, el sistema no es capaz de adivinar el *intent*, en concreto se hizo la pregunta *¿Cómo llegar a la calle Valencia por favor?* Y al no estar creado como *training phrase* no se pudo cursar la petición, lo cual supondría una comunicación fallida. Esto nos da pie a pensar que hay que introducir y entrenar manualmente muchos tipos de frases distintos para un mismo *intent* y aun así puede haber problemas. Como parte positiva es posible activar en el *DialogFlow* un corrector ortográfico, aunque supone más tiempo de procesado, de manera que si el usuario introduce "*I want aples*" el sistema lo corregiría a "*I want apples*".

  2. También es necesario comentar a nivel de integración técnica con otros sistemas que *DialogFlow* en su versión 2, con el ánimo de hacerlo más seguro, requiere de amplios conocimientos del *SDK* de *Google*. Esto es: *Google Application Credentials*, gestionar el *API* de *Google* para permitir el acceso por *token* y después gestionar el esquema de *Request-Response* en formato ***json***. Se supone que con el tiempo se hará más ergonómico computacionalmente, pero por el momento, según la experiencia en las

pruebas realizadas para este proyecto, no lo es. Por último, hay que comentar que se puede encontrar un análisis muy detallado en el trabajo fin de máster de Arnau Campos tutelado por el profesor Diego Álvarez de la Escuela Técnica Superior de Informática (Campos 2018).

2.3.3 Futuro de los *chatbots*

Se considera relevante la predicción realizada por la prestigiosa compañía consultora americana *Gartner, Inc* miembro de *S&P 500*, este último considerado como el índice más representativo de la situación real del mercado. *Gartner, Inc* en su estudio *Gartner Predicts 25 Percent of Digital Workers Will Use Virtual Employee Assistants Daily by 2021* (*Omale* 2019) señala que el uso de *chatbots* en el lugar de trabajo está creciendo y para el año 2021 el 25 por ciento de los trabajadores digitales utilizarán un asistente virtual de empleados diariamente, frente al 2 por ciento actual. Debido a la democratización de la inteligencia artificial y el desarrollo de interfaces de usuario conversacionales precisos e inteligentes. Los bancos y las compañías de seguros están mostrando un gran interés en pilotar este tipo de proyectos, también ha crecido el interés en las oficinas de atención al cliente y consultas de información. Algunos ejemplos son *Alexa for Business* de *Amazon*, que ayuda a los empleados a delegar tareas como la programación de reuniones y operaciones logísticas, y *MIKA* de *Nokia*, que ayuda a los ingenieros a encontrar respuestas cuando realizan tareas complejas o diagnostican problemas.

Gartner anuncia que, para el año 2023, el 25 por ciento de las interacciones de los empleados con las aplicaciones se realizarán a través de la voz, frente a menos del 3 por ciento en 2019. Aunque la mayoría de los *chatbots* estando basados en texto, los servicios de voz a texto y de texto a voz habilitados por la inteligencia artificial están mejorando rápidamente. Como resultado, el despliegue de soluciones basadas en la voz crecerá. “Creemos que la popularidad de los altavoces conectados en el hogar, como el *Amazon Echo*, *Apple HomePod* y *Google Home*, aumentará la presión sobre las empresas para que habiliten dispositivos similares en el lugar de trabajo", dijo *Van Baker*, vicepresidente de *Gartner*.

Gartner anuncia que el gasto de los consumidores y las empresas en altavoces asistentes personales superará los 3.500 millones de dólares en 2021. Un ejemplo reciente de la integración de los *chatbots* en las empresas es la asociación de *Amazon* con *Marriott*. El operador de hoteles utiliza *Echo de Alexa* para ayudar con los procedimientos de pago y la gestión de los servicios de las habitaciones. En el sector de la salud, el diagnóstico remoto y las aplicaciones para el cuidado de personas mayores serán habilitados por altavoces con *chatbots*, algunos ya están siendo pilotados como por ejemplo el uso de la tecnología de voz para documentar los datos de los pacientes en las historias clínicas electrónicas. Las interfaces de voz liberan a los trabajadores digitales de tener que utilizar el ratón y el teclado al interactuar con las aplicaciones empresariales. Esta libertad puede beneficiar enormemente a los trabajadores de primera línea.

2.3.4 Selección de algunos *chatbots* textuales interesantes

A continuación, se muestra una relación de *chatbots* que han sido testeados, se incluyen algunas reseñas. (Ver Tabla 5).

Chatbots textuales
Lecker.de (“*Rezepte Suchen per WhatsApp - so Geht’s! | LECKER”* n.d.): Es un *chatbot* que es una revista de cocina alemana para *Facebook Messenger* y *Whatsapp*, el *chatbot* responde a solicitudes que giran en torno a todo tipo de platos. El usuario recibe de inmediato ideas de cocina. Además regularmente o casi todos los días el *chatbot* envía también inspiraciones en forma de diferentes recetas.
Elecciones.chat (“Elecciones Chat - El *Chatbot* y *Voicebot* Comparador de Programas Políticos Para Elecciones Generales 2019. Vox, PP, PSOE, Podemos, Ciudadanos” n.d.): Es un *chatbot* comparador de los programas electorales de los partidos políticos. Se ha utilizado con *Partido Popular*, *Partido Socialista Obrero Español*, *Ciudadanos*, *Unidas Podemos* y *VOX* como candidatos a la presidencia del Gobierno en las pasadas elecciones generales del 28 de abril de 2019 en España. Desarrollado por la empresa *chatbot chocolate* ha sido testeado en *Whatsapp,* en este caso es interesante señalar que también está desarrollado para *voicebot* para *Alexa* de *Amazon*.
U-Report (“*U-Report - U-Report Available on Facebook Messenger!”* n.d.): Es un *chatbot* de *Unicef* para *Facebook Messenger* cuyo objetivo es recabar información de los jóvenes alrededor del mundo acerca de los asuntos que ellas y ellos consideran más trascendentes. El objetivo es amplificar sus voces y crear un impacto en las políticas globales. Realiza encuestas, recopila información y la clasifica. Es un *chatbot* para una buena causa y muy interesante.
Billy Seguros (*Chatbot* Chocolate n.d.): Se trata de un *chatbot* comparador de seguros de coches y motos en España. Aquí se ha encontrado un problema grave y es que como para poder comparar seguros tiene que solicitar datos personales y cederlos a terceros para calcular las pólizas, éste pide el consentimiento de la persona para dicho tratamiento que actualmente pueden ser consideradas unas condiciones leoninas y no actualizadas al *RGPD de mayo 2018*. Deberían actualizar su marco legal lo antes posible. Está desarrollado por *Chatbot* Chocolate y se ha probado en *Telegram*.
Mitsuku (*Pandorabots* n.d.): Es uno de los mejores *chatbots* conversacionales testeados, se aproxima mucho a charlar con un ser humano. Está desarrollado por *Pandorabots* en un excelente trabajo que en los años 2018, 2017, 2016 y 2015 fue galardonado con el primer premio *Loebner Prize Turing Test*, pasando el test de *Alan Turing. Loebner Prize* (*Worswick* 2018) es un concurso anual celebrado en Inglaterra donde participan *chatbots* no conectados a *Internet* y donde tienen que pasar el *test de Turing.* Este consiste en parecer lo más humano posible, que no quiere decir lo más inteligente posible, es decir si le preguntamos al *chatbot* cúal es la población de Madeira y nos contesta; ni idea! Eso sería un comportamiento humano. Por poner un ejemplo del nivel del concurso *SIRI* hubiera quedado en la posición 14ª. Actualmente *Mitsuku* es utilizado por el *New York Times*, *Wall Street Journal*, *BBC*, *The Guardian.* En el ámbito de este trabajo se ha podido probar su versión hecha para *Facebook Messenger*.

Tabla 5. Selección chatbots testeados. Fuente: Elaboración propia

2.4 Modelos predictivos

2.4.1 Introducción

Como se ha comentado anteriormente los modelos predictivos aportan una evolución importante al procesamiento del lenguaje natural y *machine learning*. Son más eficientes que los modelos basados en reglas ya que los modelos predictivos conversan prediciendo la siguiente frase dada la frase o frases anteriores en una conversación. La fortaleza de estos modelos reside en que pueden ser entrenados de principio a fin y por lo tanto requieren de muchas menos reglas a mano.

Para entender cómo funcionan los modelos predictivos pongamos un ejemplo (*GeitGey* 2019): Podemos extraer el significado de las reseñas de los restaurantes entrenando a un clasificador para predecir una clasificación por estrellas (de 1 a 5) basada únicamente en el texto de la reseña (ver esquema de funcionamiento en la Ilustración 3). Si el clasificador puede leer cualquier crítica de un restaurante y asignar de forma fiable una calificación de estrellas que refleje con precisión la forma positiva o negativa en que la persona describió el restaurante, eso demuestra que podemos extraer significado del texto, pero la desventaja de usar un clasificador de texto es que normalmente no tenemos idea de por qué clasifica cada trozo de texto de la manera en que lo hace: el clasificador es una caja negra.

No saber cómo interpretar el funcionamiento interno de los modelos ha sido durante mucho tiempo un reto en el aprendizaje automático. A veces tener un modelo que funciona no es suficiente, necesitamos conocer cómo funciona. Para el caso que nos ocupa podemos descargar libremente millones de reseñas de restaurantes y entrenar un modelo clasificador para predecir cuántas estrellas otorgar al restaurante. Según *Geitgey* esto es como magia porque cogemos una gran cantidad de datos, creamos un clasificador de texto que parece entender el lenguaje natural de las personas y que a su vez es capaz de clasificar las reseñas, pero realmente es una caja negra.

Esto plantea la cuestión de si debemos confiar plenamente en esa caja negra sin saber cómo funciona. Por ejemplo, tal vez los usuarios que aman un restaurante tiendan a escribir reseñas cortas como "Me encanta este lugar", pero los usuarios que odian absolutamente a un restaurante escribirán página tras página de quejas porque están muy enfadados. Entonces se nos plantea la nueva cuestión de cómo sabremos que nuestro modelo de clasificación es realmente el resultado de entender las palabras de la revisión y no simplemente de clasificar las revisiones en función de su extensión La respuesta es que no lo sabemos. Es muy posible que nuestro clasificador esté tomando un atajo y no esté aprendiendo nada útil. Para ganar confianza en el modelo, necesitamos ser capaces de entender por qué hizo una predicción.

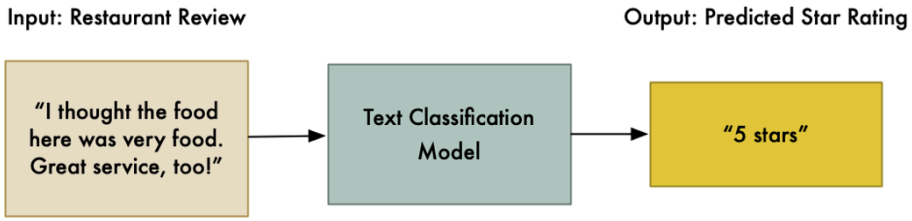


Ilustración 3. Esquema funcionamiento modelo predictivo. Fuente: Adam Geitgey

Para algunos tipos de modelos de aprendizaje de máquinas, como la regresión lineal, podemos ver el modelo en sí mismo y comprender fácilmente por qué se le ocurrió una predicción. Por ejemplo, si tuviéramos un modelo que predijera el precio de una casa sólo en base al tamaño de la casa, en primer lugar, el algoritmo separaría las reseñas en *tokens*, cada *token* en general viene a ser una palabra. Luego para cada *token* buscaría su significado. Esta división se puede apreciar en la Ilustración 4.

Ilustración 4. Reseña ejemplo separada en Tokens. Fuente: Adam Geitgey

Cuando el clasificador fue entrenado por primera vez se le asignaron un conjunto de números llamado vector de palabras. Para cada palabra que aparecía en los datos de entrenamiento al menos una vez, estos números codificaron el significado/semántica de cada palabra como un punto en el espacio imaginario de 100 dimensiones. Aunque el número de dimensiones depende del algoritmo; por ejemplo, en *Tensorflow* es de 512 dimensiones. La idea es que las palabras que representan semánticamente conceptos similares tendrán conjuntos de números similares, y al contrario palabras que tienen un significado contrario tendrán un conjunto de números muy diferente.

Para la frase en cuestión en la Ilustración 5 podemos observar su vector de representación.

Word	One hundred numbers that encode the "meaning" of each word →																
i	-0.010	0.118	-0.026	-0.097	0.097	0.078	-0.148	0.064	0.086	-0.056	0.029	0.084	-0.124	0.141	-0.139	0.072	0.01
didn	-0.024	0.029	-0.015	-0.025	-0.017	0.034	-0.082	0.062	-0.020	0.033	0.008	0.001	0.021	-0.022	-0.060	-0.038	-0.00
'	-0.041	-0.178	0.081	0.112	0.101	0.044	-0.114	0.186	0.150	0.049	-0.019	0.125	-0.108	0.156	-0.254	0.025	-0.06
t	0.017	0.049	-0.051	-0.055	0.013	0.052	-0.066	0.027	0.009	-0.025	0.018	0.078	-0.062	0.032	-0.046	0.092	-0.03
love	0.051	0.315	-0.035	-0.191	0.243	0.181	0.026	-0.286	0.201	0.320	-0.242	-0.137	0.189	-0.049	0.242	-0.156	0.05
this	0.068	-0.051	-0.022	0.054	0.142	0.112	0.016	-0.046	0.158	0.054	-0.039	0.155	-0.112	0.177	-0.092	0.134	-0.06
place	-0.016	0.017	-0.032	-0.052	0.022	0.065	-0.034	-0.043	-0.010	0.051	-0.002	-0.002	0.034	-0.002	0.060	0.014	-0.00
:	-0.060	0.003	-0.034	-0.042	-0.059	0.021	-0.118	0.106	-0.059	0.044	0.043	-0.013	0.041	-0.037	-0.085	-0.051	-0.01
(	-0.061	-0.062	0.028	0.000	0.036	-0.019	-0.065	0.100	0.062	-0.056	0.028	0.066	-0.100	0.100	-0.205	0.025	-0.04

Ilustración 5. Vector de representación de cada token. Fuente: Adam Geitgey

A continuación, el algoritmo promediará las columnas verticales de números que representan cada palabra para crear una representación de 100 números del significado de toda la frase llamada *vector de documento*, este vector 100nario se puede observar en la Ilustración 6.

Sentence	One hundred numbers that encode the "meaning" of the sentence →												
I didn ' t love this place : (	-0.008	0.027	-0.012	-0.033	0.064	0.063	-0.065	0.019	0.064	0.046	-0.019	0.040	-0.025

Ilustración 6. Vector de documento (frase). Fuente: Adam Geitgey

Llegados a este punto tenemos un vector de documento (reseña entrante) representado por 100 números del cual queremos predecir el número de estrellas que le corresponden. También tenemos en el modelo una gran cantidad de reseñas convertidas igualmente a vectores, aquí la diferencia es que en estas sabemos la valoración en estrellas de cada una de ellas. Todos los vectores conforman el espacio vectorial. El modelo obtiene las reseñas más similares semánticamente a la reseña

entrante, mediante la distancia euclídea en el espacio vectorial, y calcula la probabilidad de tener 0,1, 2, 3, o 4 estrellas. En nuestra reseña ejemplo tenemos que hay un 74% de probabilidades de la reseña tenga dos estrellas, 16% de que sean 3, 10% de que sea 1 y 0% de que sea 4 o 0 estrellas.

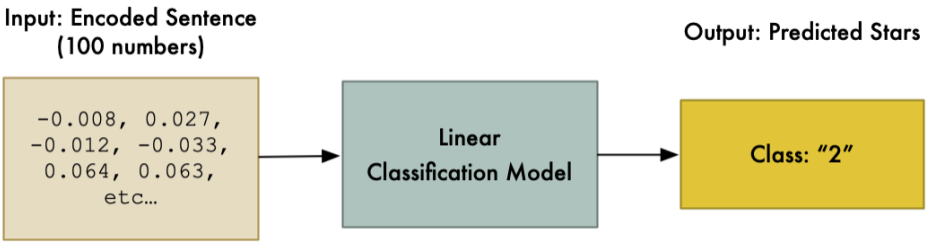


Ilustración 7. Clasificación lineal reseña. Fuente: Adam Geitgey

La cuestión es por qué son similares semánticamente algunas reseñas representadas por vectores numéricos en un espacio multidimensional. Para intentar deducirlo *Adam Geitgey* realizó un experimento, mediante una herramienta (denominada *LIME*) que creaba miles de variaciones en la frase ***I didn't love this place*** ☹***.*** Después aplicaba el modelo obteniendo la predicción de estrellas de cada una de ellas. Acto seguido el modelo se retroalimentaba con dicha información como datos de entrenamiento. Dicho modelo *LIME* tuvo como resultado qué palabras tienen más peso sobre otras a la hora de otorgar o quitar estrellas, o dicho de otra forma cuánto positiva o negativamente influyen las palabras en una reseña. Podemos observar en la Ilustración 8 como se han coloreado con fondo azul aquellas palabras que influyen positivamente y con fondo verde aquellas palabras que influyen negativamente en la valoración final de estrellas independientemente de la posición de estas en la frase.

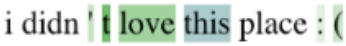


Ilustración 8. Representación con LIME. Fuente: Adam GeitGey

2.4.2 Algoritmos de aprendizaje

En el presente trabajo se requiere realizar un aprendizaje de patrones para poder ofrecer una predicción. Sin pretender ser una lista exhaustiva, en este apartado se enumeran y explican brevemente los modos de aprendizaje y los tipos de algoritmos más utilizados.

- **Aprendizaje supervisado**: En este tipo de aprendizaje, la máquina se enseña con el ejemplo. Cuando disponemos de un conjunto de patrones de entrenamiento para los que conocemos perfectamente la salida deseada de la red y cuyo objetivo es minimizar el error cometido entre la salida de la red y la salida deseada. Este tipo de aprendizaje es muy utilizado para resolver problemas de clasificación y predicción.

- **Aprendizaje no supervisado***:* Cuando no conocemos las salidas deseadas de la red y es la red por sí misma la que buscará su comportamiento más adecuado atendiendo a cierto criterio y encontrará estructuras o prototipos en el conjunto de patrones de entrenamiento. En este tipo los algoritmos intentan organizar los datos de alguna forma para describir su estructura. A medida que evalúa más datos, su capacidad para tomar decisiones sobre los mismos mejora gradualmente y se vuelve más refinada.

- **Aprendizaje por refuerzo**: Está basado en un proceso de ensayo y error que busca maximizar el valor esperado de una función criterio conocida como una señal de refuerzo. Aprende de experiencias pasadas y comienza a adaptar su enfoque en respuesta a la situación para logar el mejor resultado posible.

Tenemos varios tipos de algoritmos de aprendizaje. Se resumen brevemente algunos de ellos:

1. **Regresión** (Victor Roman 2019b): Son algoritmos de aprendizaje supervisado y estiman las relaciones entre las variables, enfocándose en predecir una variable dependiente. En el *algoritmo de Regresión lineal*, se crea una línea arbitraria lo más próxima posible a todos los puntos (x, y)**,** la línea se va recalculando con cada iteración. El objetivo es minimizar la distancia (error) de todos los puntos con respecto a la línea de regresión. Funciona muy bien para pronosticar.

2. **Basados en Reglas**: Son los algoritmos de clasificación más sencillos que existen. Por ejemplo, el *ZeroR*, como su nombre indica, tiene cero capacidades de predicción, ya que sólo tiene en cuenta la salida deseada y no tiene en cuenta el resto de las variables independientes. Se utilizan como base de medición para otros algoritmos de predicción.

3. **Bayesianos** (Victor Roman 2019a)***:*** Basados en el *Teorema de Bayes.* Este algoritmo funciona muy bien cuando se asume que las variables son independientes, pero tiene problemas para predecir en el mundo real.

4. **De agrupación** (Benja Lara 2013)***:*** Se utilizan en aprendizaje no supervisado y sirven para categorizar datos no etiquetados. Por ejemplo, *SimpleKMeans* trata de agrupar en k grupos instancias similares.

5. **Árboles de decisión*:*** Son del tipo aprendizaje supervisado. Uno de los más representativos es el *RandomForest* (*Will Koehrsen* 2017) que está formado por árboles de decisión que mediante preguntas van acotando cada vez más la predicción. Con estos algoritmos se obtienen predicciones muy precisas y son muy rápidos.

6. **Redes neuronales** (*Assaad Moawad* 2018)***:*** Se puede definir como una caja negra con dos métodos. El primero se llama *entrenar* y coge las entradas y las salidas deseadas, las procesa y actualiza su estado interno en consecuencia, de forma que la salida calcula se acerque lo máximo posible a la salida deseada. El segundo se llama *predecir* que toma la entrada y genera, utilizando el estado interno de la red, el resultado. Las redes neuronales se explican en profundidad más adelante.

7. **Redes neuronales recurrentes (RNNs)** (*Chris Nicholson* n.d.)**:** Las redes recurrentes son un tipo de red neuronal artificial diseñada para reconocer patrones en secuencias de datos, tales como texto, genomas, escritura a mano, la palabra hablada o datos numéricos de series de tiempo que emanan de sensores, mercados de valores y agencias gubernamentales. Estos algoritmos tienen en cuenta el tiempo y la secuencia, tienen una dimensión temporal. Las redes recurrentes tienen dos fuentes de entrada, el presente y el pasado reciente, que se combinan para determinar cómo responden a los nuevos datos. Se trata de encontrar correlaciones entre eventos separados por muchos

momentos, y estas correlaciones se denominan "dependencias de largo plazo". Las investigaciones demuestran que son una de las redes neuronales más poderosas y útiles, junto con los mecanismos de atención y las redes de memoria. Las *RNNs* son aplicables incluso a las imágenes, que pueden descomponerse en una serie de parches y ser tratadas como una secuencia.

8. **Deep learning** (Rubén López 2014): Basado en redes neuronales, pero con diferencias que lo hacen más eficiente. Realmente es la combinación de algunas técnicas complejas que ya existían. Como, por ejemplo, comienza con entrenamiento no supervisado, información no etiquetada, para después continuar con el entrenamiento supervisado, información etiquetada. Además, se intenta empezar a iterar con unos pesos de partida *útiles*, en vez de con unos pesos al azar.

2.4.3 Redes Neuronales

Muchos modelos de procesamiento del lenguaje natural actuales están basados en redes neuronales. A continuación, se procede a resumir los conceptos básicos de las redes neuronales a partir de los apuntes aportados por los profesores J.M. Calabuig, Lluís M. García Raffi y E.A. Sánchez Pérez (Redes neuronales y algoritmos genéticos*)*.

Las neuronas artificiales son capaces de aprender a reconocer patrones. La neurona artificial más sencilla se llama *perceptrón* simple y es una función tal que dado un vector de entradas le asocia una única salida binaria, para diseñar un *perceptrón* simple seguimos los siguientes pasos (ver Ilustración 9):

1. Partimos de que sabemos el valor de salida $\boldsymbol{z}$ para un determinado vector de entrada $\boldsymbol{x} = (x_1, x_2, x_3, \dots, x_n)$.
2. Fijamos los pesos iniciales $w_1, w_2, w_3, \dots, w_n$ y calculamos la salida $\boldsymbol{y}$ de la siguiente manera:

$$\boldsymbol{y} = \begin{cases} 0, & w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n \leq 0 \\ 1, & w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n > 0 \end{cases}$$

3. Si la salida $\boldsymbol{y}$ coincide con el valor real $\boldsymbol{z}$ ya lo tenemos, pero si $\boldsymbol{y} \neq \boldsymbol{z}$ entonces tenemos que modificar los pesos.
4. La modificación de los pesos se realiza mediante el método del descenso del gradiente.
5. Definimos una nueva familia de pesos:

$$\tilde{\boldsymbol{w}} = \boldsymbol{w} + \eta(z - y)\boldsymbol{x}$$

Siendo η una constante positiva llamada tasa de aprendizaje.

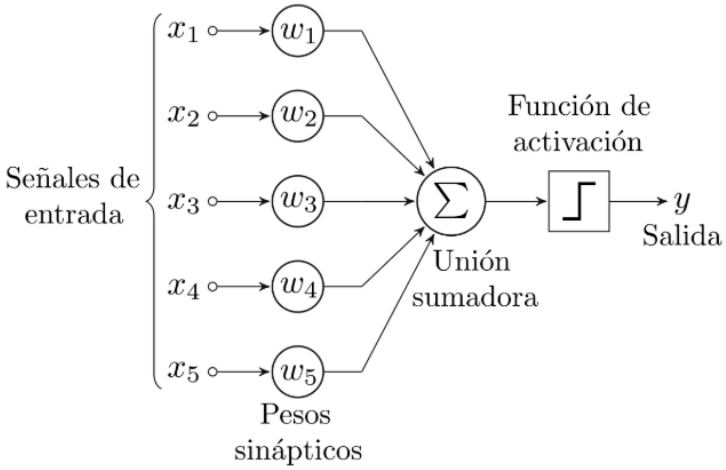


Ilustración 9. Perceptrón 5 unidades. Fuente: Alejandro Cartas

Existen otras funciones de activación o algoritmos de aprendizaje que recalculan los pesos cada vez que no hay acierto, dando lugar a otros tipos de neuronas: *sigmoide*, *adalina* y *tangente hiperbólica*. Las neuronas se conectan entre sí dando lugar a las redes neuronales. Las principales propiedades de las redes neuronales se resumen con el nombre ***PACA*** porque:

- **Predicen**: dada una serie temporal de datos, estiman el valor del dato en un tiempo posterior.
- **Aproximan** funciones de varias variables.
- **Clasifican patrones**: asignan una entrada a una clase de entre un conjunto previamente establecido. Tienen por imagen variables booleanas cuando clasifican.
- **Agrupan**: Un algoritmo de agrupamiento explora similitudes entre patrones. Tiene por imagen un grafo, con relaciones de vecindad y proximidad.

Las redes neuronales *monocapa* tenían limitaciones. Como, por ejemplo, la incapacidad para resolver la función lógica *XOR*. Es por ello, por lo que muchos investigadores decidieran abandonar el estudio de redes neuronales. Pero más tarde esto se resolvió con la introducción de capas ocultas intermedias dando lugar a las redes neuronales *multicapa*.

Un problema conocido en este tipo de redes es el entrenamiento excesivo que puede conducir a una mala generalización, el modelo acaba aprendiendo los detalles y olvidando la información relevante del caso general. Lo ideal es parar el entrenamiento justo antes de que se complete el descenso de gradiente, aunque no resulta trivial.

Por otro lado, existen técnicas para optimizar una red neuronal realizando un podado. Esto se realiza eliminando aquellas conexiones que no aportan cambios a la salida de la red.

Generalmente el conjunto de datos de entrenamiento se divide en tres grupos, donde un 70% es para el entrenamiento y el 30% restante se divide a su vez en 15% para validación. Este 15% sirve para detener el proceso de aprendizaje cuando el error de

los datos de validación empieza a crecer y el proceso de aprendizaje se detiene. Finalmente, el otro 15% restante son para test que se utilizan para ver cómo generaliza la red, también llamado *performance*.

Se han testeado algunas librerías actuales de código libre, que utilizan redes neuronales para incrementar su capacidad y velocidad de procesamiento.

TensorFlow (*Google* n.d.)

Librería de código abierto para el desarrollo de modelos de *machine learning*. Está desarrollada por *Google* y está basada en redes neuronales. Esto significa que puede detectar y descifrar correlaciones análogas al aprendizaje y razonamiento usados por los humanos. Fue liberada en el año 2015 y es la librería más popular para desarrollar aplicaciones de inteligencia artificial. Las incorporaciones (*embeddings)* son el mapeo o transformación de las palabras a vectores (de números reales). En *TensorFlow* los *embeddings* tienen 512 dimensiones. Los clasificadores y las redes neuronales en general trabajan sobre estos vectores de números reales y se entrenan mejor en vectores densos donde todos los valores contribuyen a definir un objeto. El uso común de los *embeddings* es calcular la similitud entre las palabras encontrando los vecinos más cercanos.

Con el objetivo de realizar una prueba primero se ha estado analizando un modelo entrenado en lengua inglesa y publicado por *Google* denominado *Universal Sentence Encoder* (*Google* n.d.). El modelo codifica el texto en vectores de alta dimensión que pueden utilizarse para la clasificación de textos, la similitud semántica, la agrupación y otras tareas de lenguaje natural. Su esquema de funcionamiento se puede observar en la Ilustración 10.

Por último, hay que reseñar que existe una funcionalidad de visualización muy interesante proporcionada también por *Google*. Dicha herramienta se ha utilizado en el presente trabajo, se llama *Tensorboard*.

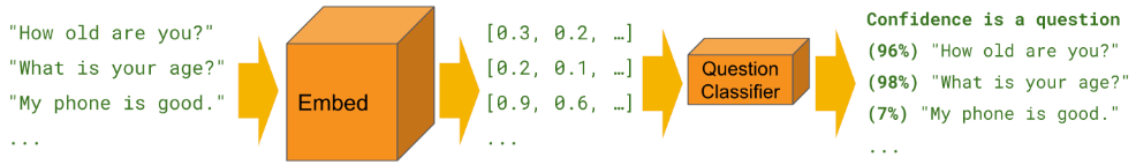


Ilustración 10. Esquema del Universal Sentence Encoder. Fuente: Web Tensorflow

A continuación, se explica una de las pruebas realizadas con esta librería. Consiste en calcular la similitud semántica, mediante la distancia del coseno, de una frase denominada **pivote** o **patrón** (en azul) con respecto a otras frases. Todas ellas obtenidas de las consultas ciudadanas. En la Tabla 6 se pueden observar los resultados.

Frase	Similitud (%)
Buenos días autobús cojo desde la estación del norte a la estación de autobuses?	**100,0**
Cómo puedo ir a la plaza del Ayuntamiento?	**84,7**
De zapadores a nuevo centro q bus puedo cojer?	**83,6**
Quiero ir al Hospital Peset que está cerca de Patraix	**82,6**

¿Qué ocurrió, no funcionan los autobuses?	**82,5**
Me gustaría que me dijeras como llegar a la calle Zapadores	**82,0**
Yo tiro petardos en Benetusser	**81,5**
Habrá huelga mañana?	**81,2**
No me lo había preguntado nunca pero es difícil de contestar	**81,1**
Q bus tengo q cojer de emilio baro a nuevo centro?	**81,0**
Mi móvil se está quedando sin batería	**80,7**
ir a la calle Valencia en bus	**80,2**
Vengo de hacer gimasia	**79,9**
Si estoy en Benicalap como puedo ir a calle Cavanilles?	**79,2**
Puedes coger la línea 73 hasta Pérez Galdós con la línea 89 que te deja en Peris y Valero-Zapadores. Saludos!	**78,7**
Te gustaría tomar un café conmigo?	**76,4**
Sabes qué hora es?	**76,1**
Tengo bastante hambre	**75,8**
Buenos días, recuerda que nuestro horario de Atención al Cliente por WhatsApp es de Lunes a Viernes (laborables) de 8 a 21	**75,2**
¿Qué hora es?	**74,5**
De Emilio Baro 11 a ciudad fallera?	**67,4**

Tabla 6. Resultado similitud semántica. Fuente elaboración propia

- La carga del modelo pre-entrenado es muy pesada ya que ocupa 1 GB que hay que cargar en memoria del sistema. Con lo cual, dependiendo de la velocidad de la red la primera ejecución le puede costar bastante tiempo, por otro lado, se evidencia que el modelo está entrenado en lengua inglesa porque el resultado no clasifica bien para todas las frases de prueba, como se puede observar en la tabla hay frases que no están bien clasificadas.

Gensim (*Řehůřek* 2019)

Radim Řehůřek es el creador de una librería de código libre, hecha en *Python,* para crear modelos de procesamiento del lenguaje natural usando para ello técnicas matemático-estadísticas. La librería es fruto de una tesis doctoral publicada en el año 2011 *dissertation Scalability of Semantic Analysis in Natural Language Processing. Gensim* está diseñado para manejar grandes colecciones de texto utilizando *streaming* de datos y algoritmos en línea incrementales, lo que lo diferencia de la mayoría de los otros paquetes de *software* de aprendizaje automático que se dirigen únicamente al procesamiento en memoria.

Gensim ha sido citado por cerca de 1.400 aplicaciones académicas y comerciales. Se han implementado algoritmos como *Word2Vec*, *FastText*, *Latent Semantic Analysis*,

Latent Dirichlet Allocation, etcétera basados en descubrir los patrones de coocurrencias dentro de un corpus de documentos preparados para entrenar.

Características:

- Independencia de la memoria, no siendo necesario que todo el corpus de formación resida en la memoria *RAM* en un momento dado (puede procesar corpus de gran tamaño a escala web).
- Compartir memoria - los modelos entrenados pueden persistir en el disco y ser cargados de nuevo a través de *mmap*. Múltiples procesos pueden compartir los mismos datos, reduciendo el espacio de RAM.
- Implementaciones eficientes para varios algoritmos de espacio vectorial populares, incluyendo *Word2Vec*, Doc2Vec, *FastText*, *TF-IDF*, *Latent Semantic Analysis* (*LSI/LSA*), *Latent Dirichlet Allocation* (*LDA*) o *Random Projection*.
- Módulos de integración de E/S y lectores de varios formatos de datos populares.
- Consultas rápidas de similitud para documentos en su representación semántica.

Los principales objetivos de diseño detrás de *Gensim* son:

- Interfaces sencillas y baja curva de aprendizaje de *API* para desarrolladores. Bueno para la creación de prototipos.
- Independencia de la memoria con respecto al tamaño del corpus de entrada; todos los pasos intermedios y algoritmos funcionan de forma secuencial, accediendo a un documento a la vez.

Se ha analizado esta librería con resultados muy interesantes. A continuación, se explica cómo se utilizó un modelo con vectores *word2vec* para *Gensim* entrenado para lengua española (Almeida y Aritz 2018). Donde el modelo lo crearon utilizando una ventana de +/- 5 palabras, descartando las palabras con menos de 5 instancias y creando un vector de 400 dimensiones para cada palabra.

El texto utilizado para crear las incorporaciones ha sido recuperado de noticias, *Wikipedia*, el *BOE* español, *web crawling* y fuentes literarias abiertas. El texto utilizado tiene un total de 3.257.329.900 palabras y 18.852.481.207 caracteres. De las múltiples opciones y algoritmos que nos ofrece *Gensim,* se realizó una prueba de similitud buscando la palabra ***BUS*** en el modelo y que nos devolviese el *topten* de palabras similares semánticamente (Ver Tabla 7).

Palabra	**Similitud (%)**
autobús	**79,8**
ómnibus	**75,1**
tranvía	**72,9**
trolebús	**71,6**
autocar	**70,7**
tren	**69,9**
minibús	**68,5**
autobus	**67,9**
monorraíl	**67,1**
taxi	**66,9**

Tabla 7. Resultados similitud palabra bus en modelo. Fuente elaboración propia

- La primera vez que se carga el modelo pre-entrenado tarda alrededor de 25 minutos debido a que es muy voluminoso. Resulta un tiempo considerable. Después los resultados demuestran que el modelo y la herramienta funcionan correctamente ya que los datos son coherentes.

GloVe (Pennington, Socher, and Manning 2019)

Desde la Universidad de *Stanford* (California) *Jeffrey Pennington*, *Richard Socher*, y *Christopher D. Manning* en el año 2014 desarrollaron GloVe que es un algoritmo de aprendizaje no supervisado enfocado en obtener representaciones vectoriales de palabras.

El entrenamiento se realiza sobre estadísticas globales agregadas de coocurrencias de palabras de un corpus, y las representaciones resultantes muestran interesantes subestructuras lineales en el espacio vectorial. El modelo *GloVe* se entrena en las entradas que no son cero, podemos decir que se realiza una poda, de una matriz global de coocurrencia de palabras. Para rellenar esta matriz se requiere un solo paso por todo el corpus para recopilar las estadísticas. Para los corpus grandes, este pase puede ser costoso desde el punto de vista informático, pero es un costo inicial único. Las iteraciones de entrenamiento subsiguientes son mucho más rápidas porque el número de entradas de matriz que no son cero es normalmente mucho menor que el número total de palabras en el corpus.

Por último, hay que comentar que se ha intentado probar esta librería, pero sin éxito porque no se ha podido instalar la librería de código libre.

SpaCy (*Honnibal* y *Montani* 2019)

Los alemanes *Mathew Honnibal* e *Ines Montani* crearon esta librería de código abierto para procesamiento del lenguaje natural escrita en *Python*. Actualmente ofrece modelos estadísticos de redes neuronales *convolucionales* en idiomas inglés, español, portugués, francés e italiano. Está optimizado para el uso en entornos de producción con muy buena precisión y con altas velocidades de procesamiento. Se demostró en 2015 cuando investigadores independientes de *Emory University* y *Yahoo*! *Labs* demostraron que *spaCy* ofrecía el analizador sintáctico más rápido del mundo y que su precisión estaba dentro del 1% de la mejor disponible (*Choi* et al., 2015).

SpaCy v2.0, publicado en 2017, es más preciso que cualquiera de los sistemas evaluados por *Choi et al*. En marzo del 2019 se lanzó la versión 2.1 que se centra en el rendimiento y correcciones de errores.

Características de esta versión:

- *Tokenización* no destructiva.
- Reconocimiento de entidades con nombre.
- Soporte para más de 49 idiomas.

- 16 modelos estadísticos para 9 idiomas.
- Vectores de palabras pre-entrenados.
- Velocidad de última generación.
- Fácil integración de aprendizaje profundo.
- Etiquetado de parte de la voz.
- Análisis de dependencias etiquetadas.
- Segmentación de oraciones basada en la sintaxis.
- Visualizadores incorporados para sintaxis y *NER*.
- Cómodo mapeo de cadena a guion.
- Exportación a arreglos de datos numéricos.
- Serialización binaria eficiente.
- Fácil empaquetado y despliegue del modelo.
- Precisión robusta y rigurosamente evaluada.

Los creadores también han invertido bastante en publicitar su trabajo y expandir su uso todo lo posible. También han creado su propia empresa *Explosion AI* y un producto comercial llamado *Prodigy.* Sus creadores es frecuente encontrarlos en *Twitter*, *Stack Overflow*, *Github* y otros *blogs* importantes especializados.

A continuación, se explica un ejemplo de visualización del análisis sintáctico de la consulta ciudadana “Hasta la calle de la Paz, hay bus directo?***”.*** El resultado de la prueba se puede observar en la Ilustración 11. La librería *SpaCy* clasifica sintácticamente de forma correcta todas las palabras, además establece las dependencias de las palabras dentro de la oración. Por último y no menos importante *SpaCy* ofrece las herramientas de visualización necesarias para poder trabajar. Por ejemplo:

Hay -> AUX, es un verbo auxiliar.
Bus -> NOUM, es un nombre y además es el objeto directo de la oración.
Directo-> ADJ, es un adjetivo y además acompaña al nombre bus.

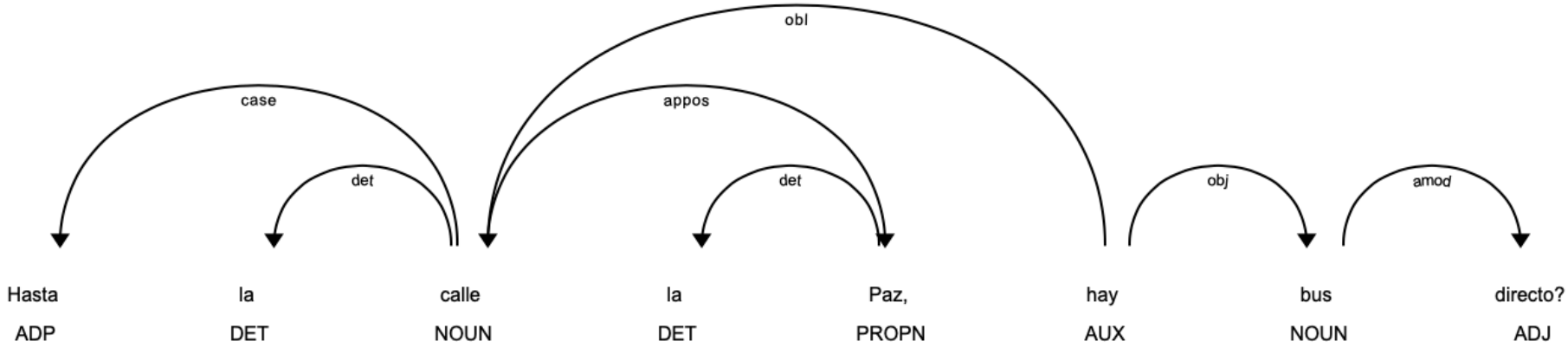


Ilustración 11. Ejemplo Visualizador sintáctico SpaCy. Fuente elaboración propia

Se puede decir que el equilibrio que tiene esta librería entre precisión, facilidad de manejo, documentación *API*, velocidad y amplitud de funciones de que dispone para el procesamiento del lenguaje natural es muy bueno.

2.4.4 Glosario de conceptos *NLP*

A continuación, se explican brevemente algunas técnicas de procesamiento del lenguaje natural, modelos y conceptos importantes que se han analizado en el presente trabajo con el objetivo de adquirir conocimientos y poder elegir las más idóneas para clasificar las consultas ciudadanas.

- ***Corpus* (lingüístico)**: Es una colección de documentos digitales de ejemplos reales de uso de la lengua. Debe ser lo suficientemente grande y representativo en cuanto a formas y expresiones para que el modelo pueda obtener sus frutos.

- **Tokenización*:*** Proviene de *token*, convertir un texto en palabras individuales. Es una de las primeras tareas a la hora de procesar el lenguaje natural para que las máquinas entiendan el significado de las frases y puedan comunicarse.

- ***TF-IDF***: Es una medida numérica que expresa cuán relevante es una palabra para un documento en una colección. Esta medida se utiliza mucho como factor de ponderación en la recuperación de la información y minería de texto. El valor *tf-idf* aumenta proporcionalmente según el número de veces que una palabra aparece en el documento, pero es compensada por la frecuencia de la palabra en la colección de documentos.

  El pesaje *tf-idf* es bueno para detectar palabras menos comunes y que tienen mucho peso para identificar una frase. Se utiliza sobre todo en los motores de búsqueda para medir la relevancia de un documento dada una consulta de usuario, estableciendo así un *ranking* u ordenación de estos. Para probar su funcionamiento hemos cogido una colección de consultas ciudadanas reales de entre las recopiladas de la *EMT* y hemos aplicado el algoritmo *tf-idf*, para luego examinar una frase concreta, extrayendo interesantes conclusiones.

  - Consulta examinada: **Buenos días.q autobús cojo desde la estación del norte a la estación de autobuses?**

  - Resultados y pesos obtenidos tras aplicar algoritmo *tf-idf* en la colección de consultas ciudadanas *(*Ver Tabla 8).

Palabra	Peso (%)
buenos	**16**
días	**16**
autobús	**25**
cojo	**34**
desde	**19**
estación	**65**
del	**19**
norte	**39**
autobuses	**30**

Tabla 8. Resultados y pesos tf-idf. Fuente elaboración propia

Observamos que la palabra *estación* tiene un peso de *65.* Es la palabra que más peso tiene y eso quiere decir que es la más significativa. Aunque en esta frase aparezca dos veces es una palabra muy relevante globalmente en la colección de frases en las que se ha aplicado el algoritmo. Esto es debido a que no es muy común.

- **Bigrams, Trigrams, Ngrams**: Son grupos de dos, tres o *n* palabras respectivamente que frecuentemente van unidas. Suelen ser utilizados comúnmente como base para el simple análisis estadístico de texto, así como también como base para importantes modelos predictivos. Probamos su funcionamiento aplicando el algoritmo a una gran colección de consultas ciudadanas a la *EMT* y observamos

interesantes resultados. A continuación, se muestran una selección de ejemplos representativos:

- Consulta ciudadana 1: **Cual es el próximo 99 por la parada de Hipercor**

  ⇒ Resultado tras aplicar algoritmo *bigram*:
  [***'cual_es'***, ***'el_próximo'***, 'por', 'la', 'parada', 'de', 'hipercor']

- Consulta ciudadana 2: **¿Buenas noches, para ir a CE monteolivete que autobus he de coger desde nuevo centro?**

  ⇒ Resultado tras aplicar algoritmo *bigram*:
  ['buenas', 'noches', ***'para_ir'***, 'ce', 'monteolivete', 'que', 'autobus', 'he', 'de', 'coger', 'desde', ***'nuevo_centro'***]

- Consulta ciudadana 3: **Buenos días, vivimos en la calle Castellón queremos coger un Bus para ir al Museo San Pío V.Gracias.**

  ⇒ Resultado tras aplicar algoritmo *bigram*:
  ['buenos_días', 'vivimos', 'en', 'la', 'calle', 'castellón', 'queremos', 'coger', ***'un_bus'***, ***'para_ir'***, 'al', 'museo', 'san', 'pío', 'gracias']

- Consulta ciudadana 4: **Cómo puedo llegar a Ruzafa en bus? Por la calle Cádiz, Cuba, Literato Azorín..**

  ⇒ Resultado tras aplicar algoritmo *trigram*:
  [***'cómo_puedo_llegar'***, 'ruzafa', 'en', 'bus', 'por', 'la', 'calle', 'cádiz', 'cuba', 'literato', 'azorín']

Gracias a esta técnica podemos observar algunos patrones habituales de los ciudadanos para expresarse dentro de una misma temática.

- **Stopwords**: Es una técnica utilizada para depurar y limpiar aquellas palabras que no tienen significado o que son muy comunes, por ejemplo, los artículos *a, del, el, la, los, las, un, unos*, etcétera. Evidentemente cada idioma tiene sus propias palabras denominadas *stopwords* y además se pueden añadir palabras propias para que se eliminen. En nuestro caso se han eliminado las palabras de cortesía como son los saludos iniciales y las despedidas: *hola, adiós, buenos días, feliz día, saludos*, etcétera. Normalmente se utiliza como técnica previa para homogeneizar los datos de partida y de entrada al modelo.

- **Lematización**: Es una técnica que consiste en convertir una forma flexionada a su raíz, obteniendo una representante de cada palabra con el objetivo de que sea más sencillo comparar y gestionarla a posteriori. Por ejemplo, si tenemos la palabra *puedes* será convertida a su raíz, *poder.* Normalmente se utiliza como técnica previa para homogeneizar los datos de partida y de entrada al modelo.

- ***LDA*** (*Latent Dirichlet Allocation):* es un modelo que presupone que cada documento es una mezcla de un pequeño número de categorías, denominados tópicos. Cada tópico está representado por un número de palabras clave con su peso, teóricamente mirando las palabras que conforman un tópico podríamos saber de qué trata esa agrupación. Se realiza una aproximación con *1.654* frases de conversaciones por mensajería instantánea entre ciudadanos y *EMT* València configurando el algoritmo para que agrupe toda la colección dentro de cinco tópicos, cada uno de ellos definido por diez palabras clave en una proporción o peso determinado. En la Tabla 9 podemos observar está composición de palabras con su determinado peso para cada una de las 5 clasificaciones.

Tópico	Composición
0	0.047*"hora" + 0.039*"ser" + 0.039*"tener" + 0.029*"volver" + 0.026*"contactar" + 0.026*"lunes" + 0.026*"horario" + 0.025*"viernes" + 0.025*"atención" + 0.025*"tarjeta"
1	0.035*"ser" + 0.026*"saber" + 0.021*"número" + 0.020*"poder" + 0.015*"preguntar" + 0.015*"haber" + 0.015*"emt" + 0.013*"necesitar" + 0.013*"aun" + 0.012*"tarjeta"
2	0.063*"bus" + 0.038*"poder" + 0.030*"parar" + 0.030*"llegar" + 0.028*"recargar" + 0.028*"haber" + 0.022*"querer" + 0.016*"bono" + 0.015*"tarjeta" + 0.015*"hacer"
3	0.062*"línea" + 0.049*"coger" + 0.043*"poder" + 0.035*"saludar" + 0.033*"gracia" + 0.029*"tarde" + 0.022*"ir" + 0.022*"metro" + 0.015*"parar" + 0.014*"dejar"
4	0.039*"tener" + 0.039*"parar" + 0.036*"callar" + 0.034*"ser" + 0.033*"minuto" + 0.030*"hacer" + 0.022*"salir" + 0.022*"pasar" + 0.020*"bueno" + 0.019*"consultar"

Tabla 9. Tópicos y su composición. Fuente elaboración propia

  - **Interpretación**: El tópico 0 está formado por las palabras *hora, ser, tener, volver, contactar, lunes, horario, viernes, atención y tarjeta* con unos pesos determinados. Esto no lo hace el algoritmo, pero viendo las agrupaciones de palabras, podemos clasificar los tópicos: Tenemos que el tópico 0 que se podría clasificar como *HORARIO/CONTACTO* en atención al cliente, el tópico 1 que podríamos clasificarlo como *INFORMACION* de *EMT* al cliente, el tópico 2 estaría relacionado con la *RECARGA* de bonos, el tópico 3 del *CÁLCULO DE RUTAS* y el tópico 4 en *ESTIMACIÓN DE LLEGADA*.

  - Es un algoritmo interesante del tipo aprendizaje sin supervisión, pero resulta bastante compleja su interpretación. No queda muy clara la clasificación automática de los tópicos según las palabras que lo componen porque existen palabras que aparecen en casi todos los tópicos. En el ejemplo de las consultas a *EMT* es necesario realizar un gran esfuerzo intelectual para poder realizar una clasificación más o menos coherente, ya que existe cierta transversalidad en algunas palabras clave. De nuevo nos encontramos con un obstáculo para los robots actuales ya que el algoritmo requeriría tener una mayor inteligencia para poner nombre a las agrupaciones, esta labor se puede observar en la Ilustración 12.

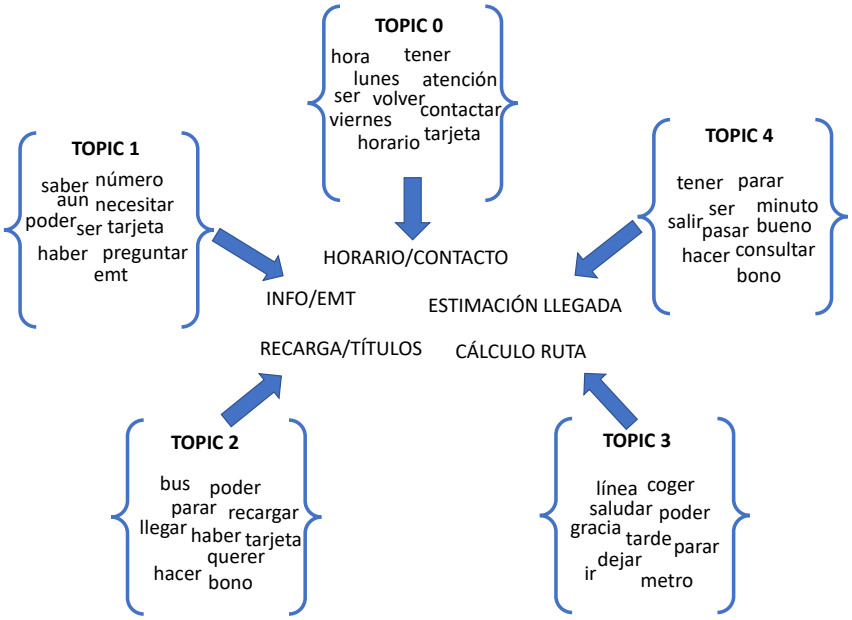


Ilustración 12. Algoritmo LDA, caso EMT. Fuente: Elaboración propia

- ***NER*** (***Named Entity Recognition)***: En el presente trabajo hemos visto dos tipos de *NER*. En este caso nos referimos a aquel *NER* que no utiliza técnicas basadas en reglas y patrones (véase el caso de *DialogFlow)*, sino que utiliza modelos matemático-estadísticos y redes neuronales. Traducimos *NER* como el reconocimiento de entidades nombradas, realmente busca localizar, clasificar y extraer la información del texto a partir de unas categorías predefinidas. Estas categorías o entidades pueden ser genéricas como nombres de ubicación geopolítica, organizaciones, tiempo, cantidades económicas, etcétera o pueden ser entidades particulares definidos particularmente para resolver un problema concreto. Estos modelos toman decisiones mediante predicciones, las predicciones están basadas en los ejemplos que el modelo ha visto durante el entrenamiento.

  Para entrenar un modelo en primer lugar se necesitan datos de entrenamiento, palabras o conjunto de palabras y etiquetas que se desea que el modelo vaticine. El proceso de entrenamiento y actualización de un modelo es sencillo para el ejemplo del caso de *EMT* que nos atañe disponemos de las consultas ciudadanas donde debemos en primer lugar etiquetar manualmente las entidades que queremos que sean estimadas, por ejemplo, etiquetamos el *bigram* “PARA IR” como “RUTAS”. Por otro lado, sabemos las respuestas correctas para cada entidad de cada consulta y podemos comparar la respuesta del modelo con la respuesta correcta real.

  Se realizan una serie de iteraciones o pasadas donde el modelo analiza el contexto de la frase, los patrones habituales y devuelve una predicción sobre cada entidad de cada frase. Además, el modelo compara la predicción y el resultado correcto y calcula la diferencia entre los dos, originando un error. Por último, el modelo trata de reducir el error mediante una función de activación de la red neuronal que finalmente actualiza los pesos de dicha red y en definitiva la forma de predecir del modelo. El proceso se puede apreciar claramente en la Ilustración 13 obtenida de la documentación de *SpaCy*.

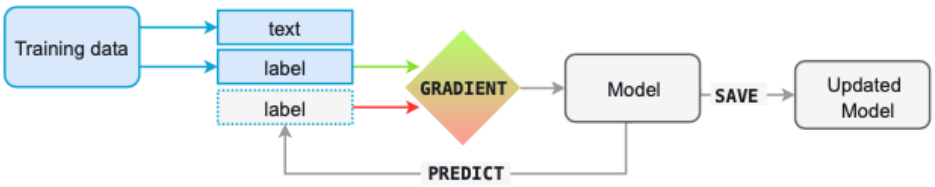


Ilustración 13. Esquema NER. Fuente: Web Spacy

Cuando entrenamos un modelo, no sólo queremos que memorice nuestros ejemplos, queremos que se nos ocurra una teoría que pueda generalizarse a través de los ejemplos. Por ejemplo, si tenemos la frase real "*¿Cómo podría hacer para ir a la Quirón en Blasco Ibañez, desde la calle olta?*" queremos que aprenda que ***Quirón en Blasco Ibáñez*** es el *destino* y no es el *origen*, lo mismo ocurre para la ***calle olta*** queremos que el modelo aprenda que es el *origen* y no el *destino*. Es por eso por lo que debe tener en cuenta el contexto de la frase y su semántica.

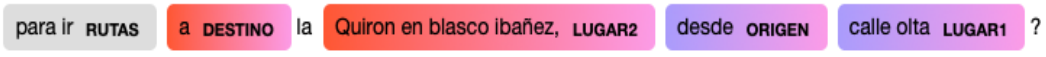
Los datos de entrenamiento, el muestreo, debe ser siempre lo más representativo posible. Un modelo entrenado en *Wikipedia*, donde las frases en primera persona son extremadamente minoritarias, probablemente se funcionará mal prediciendo en *Twitter*. Del mismo modo, un modelo entrenado en novelas románticas probablemente se funcionará mal en el texto legal.

Esto también significa que para saber cómo se está funcionando el modelo, si está aprendiendo las cosas correctas, no sólo se necesitan datos de entrenamiento, sino que también se necesitarán datos de evaluación, porque si sólo se prueba el modelo con los datos sobre los que fue entrenado, no se tendrá una idea de lo bien que se está generalizando.

Aun teniendo en cuenta estas pautas en la predicción se pueden producir falsos positivos y falsos negativos que tienen una repercusión en el coste del negocio porque suponen una comunicación emisor-receptor *fallida* en el caso de los *chatbots*. A continuación, vemos un ejemplo de predicción extrayendo las entidades relevantes (*NER)*:

Consulta ciudadana: **para ir a la Quirón en blasco ibañez, desde calle olta?**

- Resultado tras aplicar modelo:



- Se puede observar en el ejemplo que con la técnica *NER* basada en modelos matemático-estadísticos clasificamos y además extraemos la información relevante de las consultas ciudadanas. Se puede decir que nos encontramos con una aproximación válida para cumplir el objetivo.

- ***WORD2VEC*** (*Mikolov* et al. 2003): Este modelo se usa para obtener las representaciones vectoriales de las palabras. Utiliza *embeddings*, término anglosajón que podría traducirse como incorporaciones en el espacio *n* dimensional, donde el tamaño de *n* depende de cuán grande sea el diccionario de palabras a representar en un modelo concreto. Es un método eficiente (para procesar los vectores de 1.600 millones de palabras, tarda menos de un día) para aprender representaciones vectoriales distribuidas de alta calidad que capturan un gran

número de relaciones de palabras sintácticas y semánticas precisas. *Word2vec* aprende la similitud de los significados de las palabras a partir de información simple. Aprende la representación de palabras a partir de frases. La idea central se basa en el supuesto de que el significado de una palabra se ve afectado por las palabras que la rodean.

Esta idea persigue la hipótesis distribucional (*Zellig S. Harris* n.d.) que es una hipótesis estadística que dice que palabras en el mismo contexto tienden a tener significados similares. El modelo se centra, en una palabra, denominada *palabra central*, y las palabras alrededor de la palabra central, denominados *palabras de contexto*, luego existe una ventana de tamaño C que determina el número de palabras de contexto que se van a considerar en el modelo. Si consideramos como ejemplo la frase "***El bonito gato salta sobre el cansado perro***" donde la palabra gato es la palabra central y la ventana C fuera de tamaño igual a 0 no tendríamos en cuenta ninguna palabra de contexto, si C=1 tendríamos que las palabras de contexto serían *bonito* y *salta*, y si C=2 tendríamos que las palabras de contexto serían *El*, *bonito*, *salta* y *sobre*. *Word2vec* está compuesto a su vez por dos modelos basados en redes neuronales *Skip-gram* y *CBoW.* A grandes rasgos el primero predice todas las palabras de contexto dada una palabra central y el segundo todo lo contrario dadas unas palabras de contexto el modelo predice cual es la palabra central (*Preferred Networks Inc.* y *Preferred Infrastructure Inc*. 2015).

Allison Parrish (*Allison Parrish* 2018) con algunos ejemplos ayuda a entender cómo funcionan los vectores de palabras. O cómo describir una palabra a partir de su similitud con otras palabras que se le acerquen en un espacio vectorial definido. Ella lo explica con un ejemplo muy esclarecedor que se plasma a continuación. Partimos de un conjunto de datos de animales identificando para cada uno de ellos dos cualidades: La primera lo bonito que es ese animal en una escala de 0 a 100 y la segunda qué tamaño tiene también en una escala de 0 a 100. (ver Tabla 10).

	Belleza	**Tamaño**
Gatito	95	15
Hamster	80	8
Tarántula	8	3
Cachorro	90	20
Cocodrilo	5	40
Delfín	60	45
Oso panda	75	45
Langosta	2	15
Capibaras	70	30
Elefante	65	95
Mosquito	1	1
Carpa dorada	25	2
Caballo	50	50
Pollo	25	15

Tabla 10. Datos de ejemplo word2vec. Fuente Allison Parrish

Estos valores nos dan todo lo que necesitamos para determinar qué animales son similares (al menos, similares en las propiedades que hemos incluido en los datos). Tratamos de responder a la siguiente pregunta: *¿Qué animal se parece más a un capibara?* Se podría repasar los valores uno por uno y hacer los cálculos para hacer esa evaluación, pero visualizar los datos como puntos en un espacio bidimensional hace que encontrar la respuesta sea muy intuitivo (Ver Ilustración 14).

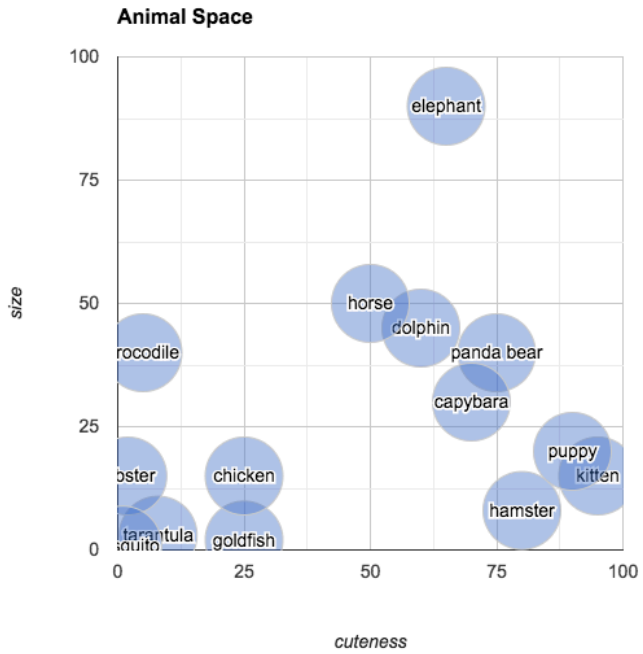


Ilustración 14. Representación vectorial animales. Fuente: Allison Parrish

La Ilustración 14 nos muestra que el animal más cercano al capibara es el *oso panda* (de nuevo, en términos de su tamaño subjetivo y su belleza). Una forma de calcular cuán separados están dos puntos es encontrar su distancia *euclídea*. Para los puntos en dos dimensiones, la distancia *euclídea* entre *capibara* (70, 30) y *oso panda* (74, 40) es 11,2 que es menos que la distancia *euclídea* entre *tarántula* y *elefante* 104,0. Modelar animales de esta manera tiene otras propiedades interesantes. Por ejemplo, puede elegir un punto arbitrario en el espacio animal y luego encontrar el animal más cercano a ese punto. Si imaginamos un animal de belleza 25 y tamaño 30, podemos fácilmente mirar el espacio vectorial para encontrar el animal que más se ajusta a esa descripción: el *pollo*.

Razonando visualmente, también se pueden responder a preguntas como: *¿Qué hay a medio camino entre un pollo y un elefante?* Simplemente dibujamos una línea desde *elefante* hasta *pollo*, marque el punto medio y encontramos el animal más cercano. Según la gráfica, a medio camino entre un *elefante* y un *pollo* es un *caballo.* También nos podemos preguntar: *¿cuál es la diferencia entre un hámster y una tarántula?* Según nuestra gráfica, son unas setenta y cinco unidades de belleza y unas pocas unidades de tamaño. La relación de "diferencia" es interesante, porque nos permite razonar sobre relaciones análogas.

- Un conjunto de vectores que forman parte del mismo conjunto de datos a menudo se denomina espacio vectorial. El espacio vectorial de los animales en

esta sección tiene dos dimensiones, lo que significa que cada vector en el espacio tiene dos números asociados con él (es decir, dos columnas en la hoja de cálculo). El hecho de que este espacio tenga dos dimensiones simplemente hace que sea fácil visualizar el espacio dibujando una gráfica en 2D. Pero la mayoría de los espacios vectoriales con los que trabajará tendrán más de dos dimensiones, a veces muchos cientos. En esos casos, es más difícil visualizar el "espacio", aunque las matemáticas funcionan casi de la misma manera.

A continuación, se explica una prueba típica y muy interesante realizada con los vectores de números. Se trata de comprobar que se cumple la ecuación matemática:

$$\boldsymbol{posible\_Rey} = \boldsymbol{hombre} - (\boldsymbol{mujer} + \boldsymbol{Reina})$$

Mediante la búsqueda de los vecinos más cercanos. Para ello nos basamos en un corpus en lengua española extraído de *Wikipedia*, dicho corpus contiene 20.000 vectores únicos de 50 dimensiones cada uno. El resultado muestra los 10 primeros *wordvectors* ordenados por proximidad a la ecuación $\boldsymbol{posible\_Rey}$, curiosamente aparece en séptima posición la palabra ***indra*** como palabra semánticamente parecida a ***rey*** (Ver Tabla 11).

Posición	Palabra
1ª	faraón
2ª	sultán
3ª	faraón
4ª	pileser
5ª	ZAMURO
6ª	MAYORDOMO
7ª	indra
8ª	SULTÁN
9ª	REY
10ª	EMIR

Tabla 11. Similitud semántica fórmula. Fuente elaboración propia

Por último, se realiza otra prueba relacionada directamente con el objetivo de la investigación. Podría ser válido como clasificador si se dispone de un modelo entrenado con espacio vectorial con las consultas ciudadanas que sea representativo y lo suficientemente voluminoso. Se trata de fijar manualmente unas consultas patrones que se corresponderían con los tipos clasificados manualmente y por otro lado, utilizar la función distancia del coseno para medir y comparar cada una de las consultas nuevas que entraran en el sistema. En teoría, solamente sería necesario crear el espacio vectorial de base y comparar las nuevas consultas que entraran en el sistema calculando el coseno. A continuación, se muestra un ejemplo con consultas reales del tipo *CALCULO DE RUTAS* donde se pone en práctica la teoría anterior (Ver Tabla 12).

Tipo	Consulta real	Similitud

Patrón	Me gustaría ir a la Calle València	-
Consulta real 1	Buenos días, como podria hacer para ir a la Quiron en blasco ibañez, desde calle olta?	95,1
Consulta real 2	Buenos días, a que horas pasa el 99 parada 1752 Fausto Elio (impar) Gracias, tengo que llegar a la primera parada del trayecto	95,0
Consulta real 3	Buenos días, Tengo que ir a la avenida Constitucion desde Primado Reig n12	95,9
Consulta real 4	Don Quijote y Sancho Panza son leyendas de nuestra literatura	89,4

Tabla 12. Similitud vectorial usando Word2Vec. Fuente: elaboración propia

- Se podría concluir que las 3 primeras consultas son similares y por lo tanto podríamos llegar a clasificarlas de ese tipo. Sin embargo, se nota una ligera distancia mayor entre la consulta patrón y la consulta real 4, aunque no es suficiente para determinar que nos encontramos con otro tipo distinto del *CALCULO DE RUTAS*.

2.5 Metodología

Se realiza una búsqueda en la Red de procedimientos, estándares o metodologías capaces de definir la secuencia de pasos o fases para tener en cuenta en un tipo de proyecto como el actual. Evidentemente no es un proyecto de desarrollo *software* clásico: en cascada, incremental, evolutiva, prototipos, *Scrum*, Ágil, etcétera. Es un proyecto centrado en los datos, es decir, cómo encontrarlos, cómo extraerlos, cómo tratarlos, cómo analizarlos, y por último y más importante cómo crear inteligencia a partir de ellos.

Ha llamado la atención un enfoque llamado *Tubería de Datos*, en inglés *Data Pipeline,* metodología creada por la comunidad *School of Data* ("*Methodology | School of Data - Evidence Is Power"* 2019). La tubería de datos es fruto de un trabajo que está en constante evolución, dicha comunidad va experimentando y ajustando las fases para reflejar los pasos básicos que están presentes en todo tipo de proyectos basados en datos. Los pasos consolidados hoy en día en el *Data Pipeline* son:

1. **Definir**: Comentan que consiste en definir el problema a resolver para hacernos una idea de los datos que vamos a necesitar.

2. **Buscar**: Básicamente comentan que consiste en buscar la información necesaria para resolver el problema definido en la fase anterior.

3. **Recoger**: Básicamente trata sobre cómo extraer los datos de donde están ubicados, hasta el lugar donde se va a realizar el análisis. Algunas habilidades utilizadas para ello es el *web scraping*[2], o simplemente descargando los

[2] *Web scraping* es una técnica utilizada mediante programas de *software* para extraer información de sitios *web*.

conjuntos de datos de los sitios del gobierno o utilizando sus portales de datos especializados.

4. **Verificar**: Una vez se tienen los datos, pero eso no significa que sean los datos que necesitamos, tenemos que comprobar si los detalles son válidos, la metodología de recogida, si sabemos quién organizó el conjunto de datos y si es una fuente creíble.

5. **Limpiar**: A menudo los datos que obtenemos y validamos son confusos. Filas duplicadas, nombres de columnas que no coinciden con los registros, valores que contienen caracteres que dificultan el procesamiento de un ordenador, etc. En este paso, necesitamos habilidades y herramientas que nos ayuden a obtener los datos en un formato legible por máquina, para poder analizarlos.

6. **Analizar*:*** Es aquí donde obtenemos información sobre el problema que definimos al principio. Se van a usar habilidades matemáticas y estadísticas para extraer conocimiento. Podemos usar visualizaciones para obtener información de diferentes variables, podemos usar paquetes de lenguajes de programación, como *Python* o *R*.

7. **Presentación**: Se trata de visualizar el análisis de datos realizado en la fase anterior de forma que se entienda perfectamente el resultado del análisis siendo esta fase rigurosa.

Esta metodología realmente tiene cierta similitud con un proceso *ETL*, siglas inglesas de *Extract Transform and Load* (*Zhao* 2017). *ETL* se ha venido utilizando como un protocolo de integración de datos entre sistemas. En nuestro caso sería un proceso *ETL* acompañado de la creación del modelo matemático.

Podemos adoptarla para el presente trabajo, aunque de forma genérica, porque como veremos más adelante, en nuestro caso vamos a encontrar algunas particularidades.

3. IMPLEMENTACIÓN

Llegados a esta fase, se ha hecho una provisión de conocimientos suficientes en los ámbitos legales, éticos y tecnológicos y estamos en disposición de poder diseñar una propuesta de implementación para la creación de un modelo predictivo.

A continuación, se describen los problemas encontrados, las decisiones tomadas, el esfuerzo realizado y los resultados obtenidos.

3.1 Acotar datos

Los datos que se necesitaban para crear el modelo matemático se localizaban en los programas de mensajería. Por lo tanto, el primer obstáculo ha sido decidir y acotar en qué aplicación de mensajería instantánea enfocar los esfuerzos. Se ha decidido trabajar con *WhatsApp* debido a que es donde reside la mayor cantidad de consultas ciudadanas, se ha desechado *Telegram*.

Hay que comentar que los datos se han recogido en una horquilla de tiempo, en concreto, la primera conversación se obtuvo el 7 de febrero y la última conversación el 13 de abril de 2019, conteniendo los periodos de Fallas y Semana Santa. Desde el primer momento, hemos sido conscientes de que la muestra de datos recogida no es una representación completa de las consultas que se reciben globalmente en *EMT*, algo que se antoja difícil de conseguir por intervenir numerosos factores.

Del mismo modo, ha sido necesario enfocar los esfuerzos seleccionando el formato de los datos a capturar. En nuestro caso se decide tratar solamente datos en formato texto, a priori el más sencillo de los existentes. *WhatsApp* dispone de otros formatos y los usuarios, cada vez más, utilizan otros formatos para comunicarse. Nos referimos a fotos, audio, vídeo y *emojis*. A continuación, se resumen los puntos interesantes en este apartado:

- Recogemos datos de *WhatsApp*, desechamos *Telegram*.
- Acotamos espacio temporal de recogida de datos y por lo tanto el tamaño de la muestra. Desde el 7 de febrero hasta 13 de abril de 2019.
- Recogemos datos en formato texto, desechamos fotos, audio, vídeo y *emojis*.

3.2 Capturar los datos

Previamente, se ha informado y solicitado permiso a la delegada de Protección de Datos (*DPD)* y al director del área de Desarrollo a la sazón Director de Sistemas de Información de *EMT*.

Dado el volumen de datos a recoger, no ha resultado trivial realizarlo por dos motivos: El primer obstáculo ha sido que *WhatsApp* no permite descargar los datos masivamente. La aplicación sólo dispone de una opción, además únicamente disponible en la versión móvil, para descargar el historial de conversaciones seleccionando un determinado *chat*. Con lo cual, sería muy tedioso ir descargando uno a uno y almacenando los *chats* en algún repositorio con el móvil.

El segundo obstáculo ha sido que para descargar los datos necesitábamos interrumpir, por un lapso de tiempo diario, el servicio de consultas por mensajería instantánea en *EMT*. Lógicamente, se ha hecho eligiendo una hora valle y en algunos días a la semana, esto es: desde las 15:00h hasta las 15:30h para reducir todo lo posible el impacto de la interrupción del servicio a los ciudadanos y también para alterar lo menor posible el trabajo de los/las profesionales de atención al cliente.

Después de analizar la situación y para atenuar más el impacto de ambos problemas se ha optado por adquirir una extensión de *Google Chrome* denominada *Backup WhatsApp Chats*(fattynoparents, n.d.), cuyo precio ha sido *2,99 €,* el funcionamiento es sencillo puesto que la *OAC* utiliza la versión *web* de *WhatsApp* en un computador de sobremesa enlazado éste con el teléfono. Por lo tanto, se ha instalado y se ha comenzado a realizar la descarga de datos, seleccionando los *chats* uno a uno, pero siendo este proceso más eficaz y rápido gracias al programa.

El formato seleccionado finalmente ha sido *csv* por ser el más extendido en el uso por las librerías testeadas. En la Ilustración 15 se puede observar una captura de pantalla del programa. A continuación, se listan los puntos importantes de esta fase:

- Autorizada recolección de datos por *DPD* y Director Desarrollo de *EMT*.
- Organización de las descargas y adquisición de la extensión de *Google Chrome* llamada *Backup WhatsApp Chats* para optimizar más la descarga de los *chats*.
- Se decide utilizar el formato *CSV*.

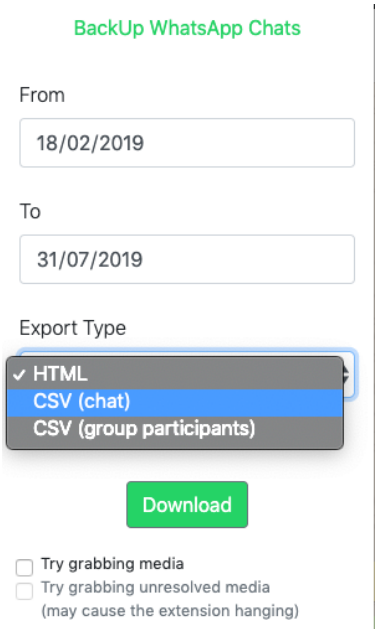


Ilustración 15. Captura de pantalla App Backup Whatsapp Chats

3.3 Anonimizar los datos

Se han recopilado más de 300 conversaciones de *WhatsApp* en sendos ficheros con formato *CSV*. Se han aplicado las pautas y recomendaciones de la *AEPD* procediendo

a anonimizar los datos personales, en nuestro caso disociamos el número de teléfono como dato que potencialmente puede asociar unívocamente una conversación con una persona. Pese a ser un trabajo de investigación, el número telefónico se ha considerado un dato personal y ha sido necesario realizar este proceso para evitar cesiones involuntarias.

Se ha aplicado la función ***hash***. Sería muy costoso computacionalmente volver a rescatar dicho número, no tiene vuelta a atrás. Se han destruido los ficheros originales para únicamente realizar el procesamiento del lenguaje natural con los ficheros anonimizados, realmente conservar los números de teléfono no tenían ningún interés para trabajo. En la Tabla 13 se muestran unos ejemplos de *chats* anonimizados.

Fecha	Hora	Mensaje
2019-02-17	09:48	Hola buenos días. Un teléfono de contacto de objetos perdidos tenéis? He extraviado la cartera, por saber si la habrían llevado allí
2019-03-24	15:01	Hola desde la calle San Pancracio 29 a la calle Maestro gozalbo N 23 q autobús tengo q coger? Tengo q estar a las 14 h
2019-03-25	07:52	Hola. Me puedes decir horarios de la línea 26 de ida y de vuelta a partir de las 17:00? Gracias.
2019-04-03	11:12	Cuanto le queda al bus número 19 para llegar a la parada 722? Muchas gracias 😘

Tabla 13. Ejemplos frases. Fuente: elaboración propia

3.4 Preparar datos

Se ha realizado un pre-análisis de los datos para familiarizar con ellos y a la vez extraer algunas conclusiones.

Estamos hablando de alrededor de 1.960 frases de ciudadanos y respuestas de personal de *EMT*. Se ha realizado la lectura de las 300 conversaciones con el objetivo de realizar una clasificación inicial. Teniendo en cuenta la idiosincrasia de las consultas se ha concluido con los 6 tipos de consultas:

- **Rutas**: El ciudadano quiere saber cómo llegar de un lugar origen a un lugar destino en València ciudad.
- **Estimación**: El ciudadano quiere saber cuándo llegará el bus a una determinada parada y de una determinada línea.
- **Recarga online**: Cuando ha habido un fallo en el proceso de recarga online de bonos a través de la web o app de EMT y el usuario se pone en contacto para preguntar qué ha pasado.
- **Objetos perdidos**: Cuando alguien ha perdido un objeto y pregunta en OAC si lo tienen allí.
- **Información**: Cuando el ciudadano quiere otro tipo de información que no se corresponde con ninguna de las anteriores.

- **Reclamación**: Cuando el ciudadano presenta una queja por este medio.

La mayoría de las consultas son del tipo cálculo de rutas, seguido de la estimación de llegada, problemas con la recarga online y el resto son los otros tipos de consultas. Se ha profundizado más en el análisis realizando un conteo de palabras que más se utilizan por cada tipo de clasificación, el resultado se puede observar en la Tabla 14.

Palabra	Tipo	Conteo
bus	Rutas	158
parada	Estimación	138
calle	Rutas	119
ir	Rutas	111
coger	Rutas	88
puedo	Recarga online	77
línea	Estimación	74
pasa	Estimación	73
bus	Estimación	54
tarjeta	Recarga online	53

Tabla 14. Conteo de palabras por tipo. Fuente: elaboración propia

Es de especial interés fijarse en la palabra ***bus***, esta aparece en dos ocasiones, pero en dos tipos de consulta distinta *Rutas* y *Estimación*. Nos encontramos con una problemática difícil de resolver porque para poder clasificar estas consultas correctamente y de forma automática vamos a necesitar ver el resto de las palabras que suelen acompañar a la palabra ***bus*** para ver las diferencias. Es decir, vamos a tener que ver el contexto de la frase, como veremos más adelante requiere de un modelo estadístico y de buena velocidad de procesamiento.

Se han filtrado los contenidos de los todos los chats, dejando sólo las frases pertenecientes a los ciudadanos, recordemos que inicialmente también estaban incluidas las respuestas de *EMT*. Los puntos más importantes de este paso son los siguientes:

- Preanálisis de datos y batida para clasificar por tipos manualmente las 300 conversaciones.
- Análisis de la descomposición de valores singulares como posible estrategia de solución.
- Filtrado *chats* para dejar sólo la consulta del ciudadano, desechando la respuesta de *EMT*.

3.5 Crear modelo

Se han obtenido 1.550 consultas ciudadanas para entrenar el modelo. En cuanto a la estrategia a seguir, en una primera instancia, se ha sondeado la posibilidad de utilizar

la *descomposición de valores singulares* (Calabuig, Garcia Raffi y Sánchez-Perez 2015). Aunque finalmente se ha optado por la librería *SpaCy* (en lenguaje *Python)* ya que ha resultado idónea porque dispone de un modelo *NER* y un *API* muy cómoda para trabajar.

Con *SpaCy* se ha entrenado un modelo de procesamiento del lenguaje natural indicándole las entidades que queremos que prediga. Teóricamente de 1.550 frases se habría que entrenar 1.085 que es el 70%, después evaluar con 232, para luego hacer el test con 233.

Ha resultado bastante tedioso entrenar el modelo porque hay que seguir un formato muy específico de *SpaCy*. Se han podido llegar a formatear 323 frases de usuario, intentando seleccionar las más representativas. A continuación, se muestran algunos ejemplos de frases de entrada al modelo, donde aparece la consulta literal del ciudadano, seguido de las entidades, que son palabras o conjunto de palabras, marcadas manualmente con posición de inicio y posición de fin y por último la frase con el formato para que sea entrenado. *SpaCy* sigue la codificación *BILUO* para el caso de entidades que son un conjunto de palabras (*Begin In Last Unit Out*) (Mathew Honnibal e Ines Montani n.d.):

Consulta 1*: Necesito ir a la Calle del Impresor Monfort número 8 y estoy en la calle Luis Bolinches número 20*

> Entidades: *Necesito ir* -> (0, 11, 'RUTAS'), *a* -> (12, 13, 'DESTINO'), *Calle del Impresor Monfort número 8* -> (27, 52, 'LUGAR2'), *estoy en* -> (55, 63, 'ORIGEN'), *Luis Bolinches número 20* -> (73, 97, 'LUGAR1')
>
> SpaCy: *('Necesito ir a la Calle del Impresor Monfort número 8 y estoy en la calle Luis Bolinches número 20\n', {'entities': [(0, 11, 'RUTAS'), (12, 13, 'DESTINO'), (27, 52, 'LUGAR2'), (55, 63, 'ORIGEN'), (73, 97, 'LUGAR1')]})*

Consulta 2: *Como puedo ir desde pont de fusta a c democracia o cerca de ahí?*

> Entidades: ir -> (11, 13, 'RUTAS'), desde -> (14, 19, 'ORIGEN'), pont de fusta > (20, 33, 'LUGAR1'), a -> (34, 35, 'DESTINO'), c democracia -> (36, 48, 'LUGAR2')
>
> SpaCy: *('Como puedo ir desde pont de fusta a c democracia o cerca de ahí?\n', {'entities': [(11, 13, 'RUTAS'), (14, 19, 'ORIGEN'), (20, 33, 'LUGAR1'), (34, 35, 'DESTINO'), (36, 48, 'LUGAR2')]})*

Consulta 3: Buenas días.Ayer hize una recarga de bonobús de 10 viajes y aún no me llegado la recarga.el número de tarjeta mobilis es 271503002099. El número de pedido es 2080944.gracias

> Entidades: *recarga* -> (26, 33, 'VENTAONLINE'), *bonobús* -> (37, 44, 'BONO'), *no me ha llegado* -> (64, 77, 'INCIDENCIAS'), *271503002099* -> (121, 133, 'TARJETA')
>
> SpaCy: *('Buenas días.Ayer hize una recarga de bonobús de 10 viajes y aún no me llegado la recarga.el número de tarjeta mobilis es 271503002099. El número de pedido es 2080944.gracias\n', {'entities': [(26, 33, 'VENTAONLINE'), (37, 44, 'BONO'), (64, 77, 'INCIDENCIAS'), (121, 133, 'TARJETA')]})*

Consulta 4: Buenos días, Me podrían informar a que hora pasa en la mañana el primer bus de la ruta 92 los sábados en la parada de joan verdeguer pare porta

> Entidades: *a que hora pasa* -> (33, 48, 'ESTIMACION'), *92* ->(87, 89, 'LINEA'), *joan verdeguer* -> (118, 132, 'PARADA')

SpaCy: *('Buenos días, Me podrían informar a que hora pasa en la mañana el primer bus de la ruta 92 los sábados en la parada de joan verdeguer pare porta\n', {'entities': [(33, 48, 'ESTIMACION'), (87, 89, 'LINEA'), (118, 132, 'PARADA')]})*

Consulta 5: En cuanto tiempo pasa el 72 por la parada 509

Entidades: *pasa* -> (17, 21, 'ESTIMACION'), *72* -> (25, 27, 'LINEA'), *509* -> (42, 45, 'PARADA')

SpaCy: *('En cuanto tiempo pasa el 72 por la parada 509?\n', {'entities': [(17, 21, 'ESTIMACION'), (25, 27, 'LINEA'), (42, 45, 'PARADA')]})*

Ha sido importante ajustar el número de iteraciones que actualizan los pesos del modelo por correr riesgo de sobre-aprendizaje. Se probaron diversas combinaciones siendo el número de 200 el más idóneo. Se ha podido comprobar el descenso del gradiente (Ver Ilustración 16) hasta su estabilización.

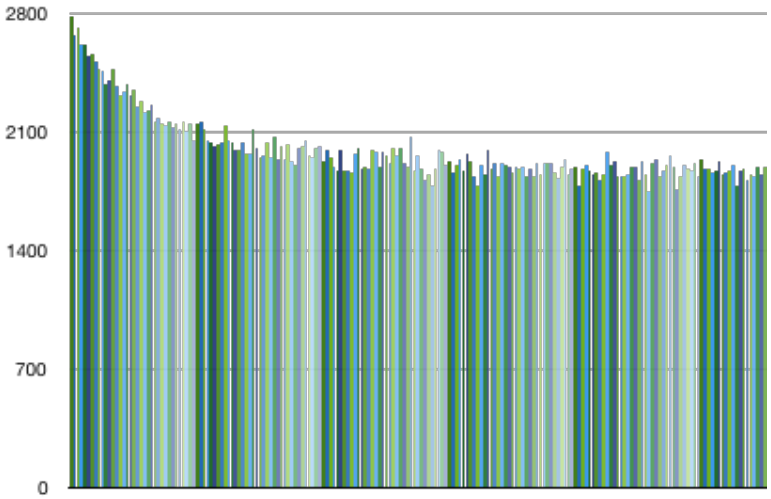


Ilustración 16. Descenso de gradiente. Fuente: elaboración propia

3.6 Evaluar modelo

Para evaluar el modelo se ha realizado un análisis, para ver cómo predice el modelo, sobre 57 nuevas consultas que no participaron en el entrenamiento del modelo. No se ha podido incrementar ese número.

Se ha encontrado en la *Red* documentación(Renuka Joshi n.d.) sobre cómo evaluar un modelo predictivo, aunque parte de la base de que las anotaciones manuales son correctas. El proceso no es nada trivial. Para cada consulta ciudadana se han comparado las entidades clasificadas manualmente y las entidades pronosticadas por el modelo. Es decir, comparamos lo que queríamos que hiciera con lo que ha hecho el modelo. Para explicarlo empezamos definiendo los 4 casos que nos podemos encontrar:

1. **Verdadero-Positivo**: Cuando la entidad pronosticada por el modelo coincide totalmente con la entidad clasificada manualmente. Pronosticado correctamente.

2. **Verdadero-Negativo**: Cuando el modelo no encuentra una entidad donde tampoco se ha clasificado manualmente ninguna entidad. Este caso a priori no hay forma de comprobarlo.

3. **Falso-Positivo**: Cuando la clasificación manual tiene marcada que no existe ninguna entidad, pero el modelo sí ha pronosticado una con unas determinadas posiciones de inicio y final.

4. **Falso-Negativo**: Cuando la clasificación manual tiene marcada que existe una entidad en determinadas posiciones de inicio y final pero el modelo no la ha acertado y por lo tanto no ha sido pronosticada.

Añadiría un quinto caso para cuando el modelo predice correctamente pero por algún error en la anotación (etiquetado) de la entidad manualmente no se ha encontrado, se definiría tal como sigue:

5. **Verdadero (Fallo Anotación)-Positivo*:*** Existe un error en la anotación manual dentro de una consulta, necesario para el aprendizaje supervisado del modelo. Debido a ese error al compararlo con la predicción del modelo no se encuentra, aún así el modelo puede predecir correctamente porque ya lo había aprendido de otras consultas.

En la Tabla 15, se puede observar la matriz de confusión con el número de verdaderos positivos, falsos positivos y los falsos negativos. No aparecen los verdaderos negativos puesto que partimos de que las consultas son correctas.

Entidad	Verdaderos positivos	Falsos positivos	Falsos negativos
AFIRMACION	2	0	0
AGRADECIMIENTO	1	0	0
ALTERNATIVA	2	0	0
ATRIBUTO	3	0	0
BONO	1	0	0
CONTACTO	1	0	0
DESTINO	17	0	2
ESTIMACION	6	0	0
INCIDENCIAS	3	0	0
INFORMACION	5	0	1
ITINERARIO	5	0	0
LINEA	14	0	0
LUGAR	9	0	0
LUGAR1	20	1	0
LUGAR2	19	1	1
OBJETOS	1	0	0
ORIGEN	20	2	0
OTRAC	4	0	0
PARADA	4	0	1
QUE	3	0	0
QUEJA	3	1	0
RECLAMACION	2	0	0
RUTAS	13	0	0
SMILE	1	0	0
TARJETA	2	0	0
TEMP	2	0	1
VENTAONLINE	1	0	0

Tabla 15. Matriz de confusión. Fuente: Elaboración propia

Para evaluar el modelo correctamente e intentar mejorar la ratio de falsos positivos y de falsos negativos necesitamos definir unos parámetros. Se explican a continuación:

1. ***Accuracy***: Es la medida de rendimiento más intuitiva y es simplemente una relación entre la observación correctamente pronosticada y las observaciones totales. Es necesario revisar otros parámetros para evaluar el rendimiento del modelo.

$$Accuracy = \frac{Verdaderos\ Positivos + Verdaderos\ Negativos}{V.Positivos + V.Negativos + F.Positivos + F.Negativos}$$

2. ***Precision:*** Es la relación entre las observaciones positivas correctamente pronosticadas y el total de observaciones positivas pronosticadas.

$$Precision = \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos + Falsos\ Positivos}$$

3. ***Recall (Sensibilidad)*** - Es la relación entre las observaciones positivas correctamente pronosticadas y todas las observaciones en la clase real.

$$Recall = \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos + Falsos\ Negativos}$$

4. ***F1-Score***: Es el promedio ponderado de *Precision* y *Recall*. Por lo tanto, esta puntuación tiene en cuenta tanto los falsos positivos como los falsos negativos. Intuitivamente no es tan fácil de entender como la precisión, pero F1 suele ser más útil que la precisión.

$$F1\ Score = \frac{2 * (Recall * Precision)}{Recall + Precision}$$

En la Tabla 16, observamos el resultado de la evaluación. Se aplican las fórmulas de *Accuracy*, *Precision*, *Recall* y *F1-Score* con los valores de la matriz de confusión. Realizando la media del parámetro *Accuracy* de todas las entidades podemos decir que el modelo creado globalmente ha obtenido una precisión del ***95,22%***.

Entidad	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
AFIRMACION	100	100	100	100
AGRADECIMIENTO	100	100	100	100
ALTERNATIVA	100	100	100	100
ATRIBUTO	100	100	100	100
BONO	100	100	100	100
CONTACTO	100	100	100	100
DESTINO	89,47	100	89,47	94,44
ESTIMACION	100	100	100	100
INCIDENCIAS	100	100	100	100
INFORMACION	83,33	100	83,33	90,90
ITINERARIO	100	100	100	100
LINEA	100	100	100	100

LUGAR	100	100	100	100
LUGAR1	95,23	95,23	100	97,56
LUGAR2	90,47	95	95	95
OBJETOS	100	100	100	100
ORIGEN	90,90	90,90	100	95,23
OTRAC	100	100	100	100
PARADA	80	100	80	88,88
QUE	100	100	100	100
QUEJA	75	75	100	85,71
RECLAMACION	100	100	100	100
RUTAS	100	100	100	100
SMILE	100	100	100	100
TARJETA	100	100	100	100
TEMP	66,66	100	66,66	80
VENTAONLINE	100	100	100	100
MEDIA	**95,22**	**98,37**	**96,83**	**97,59**

Tabla 16. Evaluación modelo. Fuente: Elaboración propia

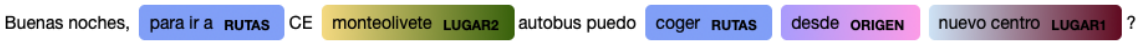
Se ha realizado un test con una batería de consultas. Resulta más visual. Es una selección de consultas reales ciudadanas junto con su predicción de entidades encontradas. Se pueden observar que algunas entidades no están pronosticadas correctamente:

Consulta 1: *Buenos días. Deseo ir desde la Patacona al Hospital Arnau de Vilanova*

Entidades: *[('Deseo ir', 'RUTAS'), ('desde', 'ORIGEN'), ('la Patacona', 'LUGAR1'), ('al', 'DESTINO'), ('Hospital Arnau de Vilanova', 'LUGAR2')]*

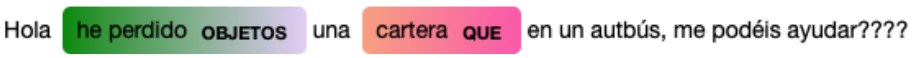
Consulta 2: *Buenas noches, para ir a CE monteolivete autobus puedo coger desde nuevo centro?*

Entidades: *[('para ir a', 'RUTAS'), ('monteolivete', 'LUGAR2'), ('coger', 'RUTAS'), ('desde', 'ORIGEN'), ('nuevo centro', 'LUGAR1')]*



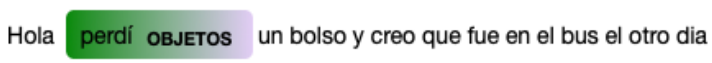
Consulta 3: *Hola he perdido una cartera en un autbús, me podéis ayudar????*

Entidades: *[('he perdido', 'OBJETOS'), ('cartera', 'QUE')]*



Consulta 4: *Hola perdí un bolso y creo que fue en el bus el otro dia*

Entidades: *[('perdí', 'OBJETOS')]*

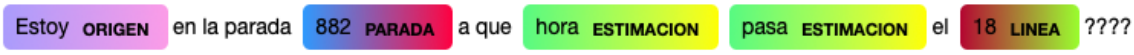


Consulta 5: *Hola me gustaría ir desde calle Valencia hasta avenida pais valenciano*

Entidades: *[('ir', 'RUTAS'), ('desde', 'ORIGEN'), ('calle Valencia', 'LUGAR1'), ('hasta', 'DESTINO'), ('avenida pais valenciano', 'LUGAR2')]*

Consulta 6: *Estoy en la parada 882 a que hora pasa el 18?*

Entidades: *[('Estoy', 'ORIGEN'), ('882', 'PARADA'), ('hora', 'ESTIMACION'), ('pasa', 'ESTIMACION'), ('18', 'LINEA')]*

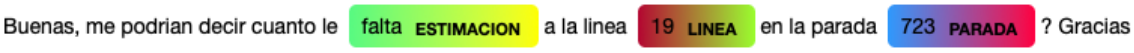


Consulta 7: *Gracias, hice una recarga online ayer y no me funciona en el bus*

Entidades: *[('recarga online', 'VENTAONLINE')]*

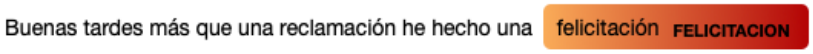
Consulta 8: *Buenas, me podrian decir cuanto le falta a la linea 19 en la parada 723? Gracias*

Entidades: *[('falta', 'ESTIMACION'), ('19', 'LINEA'), ('723', 'PARADA')]*



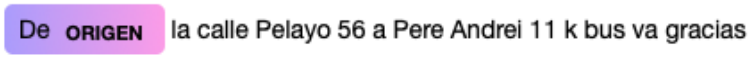
Consulta 9: *Buenas tardes más que una reclamación he hecho una felicitación por el formulario para el conductor que salvo del robo para que quede constancia un saludo*

Entidades: *[('felicitación', 'FELICITACION')]*



Consulta 10: *De la calle Pelayo 56 a Pere Andrei 11 k bus va gracias*

Entidades: *[('De', 'ORIGEN')]*



Consulta 11: *De peris y valero a la calle leones 73 k bus va gracias*

Entidades: *[]*

Consulta 12: *Buenas tardes, me pueden informar de si hay previstos en las próximas horas paros del servicio de autobuses? Gracias*

Entidades: *[]*

Consulta 13: *Que autobus pasa por emilio baro*

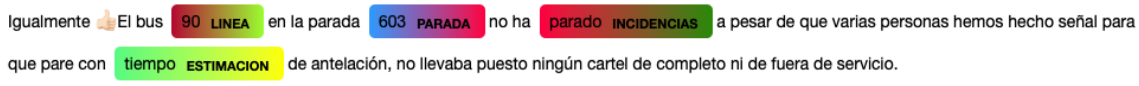
Entidades: *[('pasa', 'ESTIMACION'), ('emilio baro', 'LUGAR')]*

Consulta 14: *Hola, puedo ir a Benimamet con la EMT?????*

Entidades: *[('ir', 'RUTAS'), ('a', 'DESTINO'), ('Benimamet', 'LUGAR2')]*

Consulta 15: *Igualmente 👍El bus 90 en la parada 603 no ha parado a pesar de que varias personas hemos hecho señal para que pare con tiempo de antelación, no llevaba puesto ningún cartel de completo ni de fuera de servicio.*

Entidades: *[('90', 'LINEA'), ('603', 'PARADA'), ('parado', 'INCIDENCIAS'), ('tiempo', 'ESTIMACION')]*

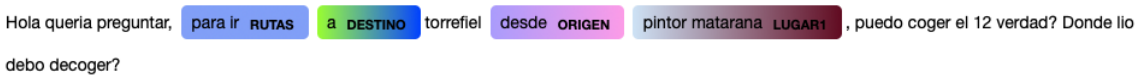


Consulta 16: *Hola, como puedo ir de guillem de castro a la altura del ivam, hasta zona patraix donde virgen de la cabeza*

Entidades: *[('ir', 'RUTAS'), ('hasta', 'DESTINO'), ('zona patraix', 'LUGAR2')]*

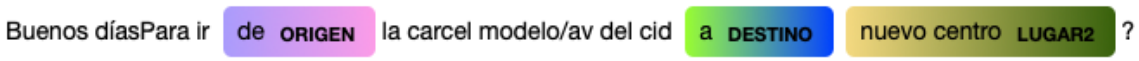
Consulta 17: *Hola queria preguntar, para ir a torrefiel desde pintor matarana, puedo coger el 12 verdad? Donde lio debo decoger?*

Entidades: *[('para ir', 'RUTAS'), ('a', 'DESTINO'), ('desde', 'ORIGEN'), ('pintor matarana', 'LUGAR1')]*



Consulta 18: *Buenos díasPara ir de la carcel modelo/av del cid a nuevo centro?*

Entidades: *[('de', 'ORIGEN'), ('a', 'DESTINO'), ('nuevo centro', 'LUGAR2')]*

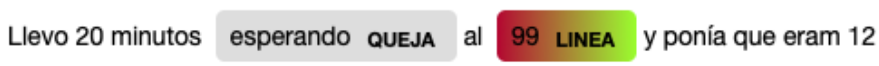


Consulta 19: *Para ir desde nuevo cento a av. La Malvarrosa que linea tengo que coger?*

Entidades: *[('Para ir', 'RUTAS'), ('desde', 'ORIGEN'), ('nuevo cento', 'LUGAR1'), ('a', 'DESTINO'), ('av.', 'LUGAR2')]*

Consulta 20: *Llevo 20 minutos esperando al 99 y ponía que eram 12*

Entidades: *[('esperando', 'QUEJA'), ('99', 'LINEA')]*



3.7 Visualizar modelo

El modelo creado contiene una representación vectorial de cada una de las palabras (*embeddings*), formando un espacio vectorial intrínseco. Resultaría interesante poder visualizar ese espacio vectorial, donde cada vector que representa a cada palabra tuviera una posición en el espacio. Y donde se pudieran observar asociaciones de palabras relacionadas semánticamente. Lo anterior se consigue gracias a la herramienta *Tensorboard* (comentada en el capítulo 2). Con esa aplicación se han implementado sendas visualizaciones que se comentan a continuación.

Visualización palabra incidencias: En esta visualización (3D) nos centramos en la palabra *incidencias*. Se colorean los vectores más próximos, calculados mediante la distancia del coseno. Se puede observar que las palabras *fallas, desviadas, cortes, molestias* y *paros* se visualizan cerca de la palabra *incidencias*. Esas palabras tienen similitud semántica (palabras vecinas) o se suelen utilizar juntas en las consultas ciudadanas. En la Ilustración 17 se puede observar la visualización con la herramienta *Tensorboard* y en la Tabla 17 se observan la distancia entre vectores.

Palabra	***Distancia coseno***
fallas	0,385
cortes	0,418
faltan	0,508
paradas	0,547
desviadas	0,554
molestias	0,565
paros	0,581
muchísimas	0,605
podrían	0,606

Tabla 17. Vecinos palabra "incidencias". Fuente: Elaboración propia

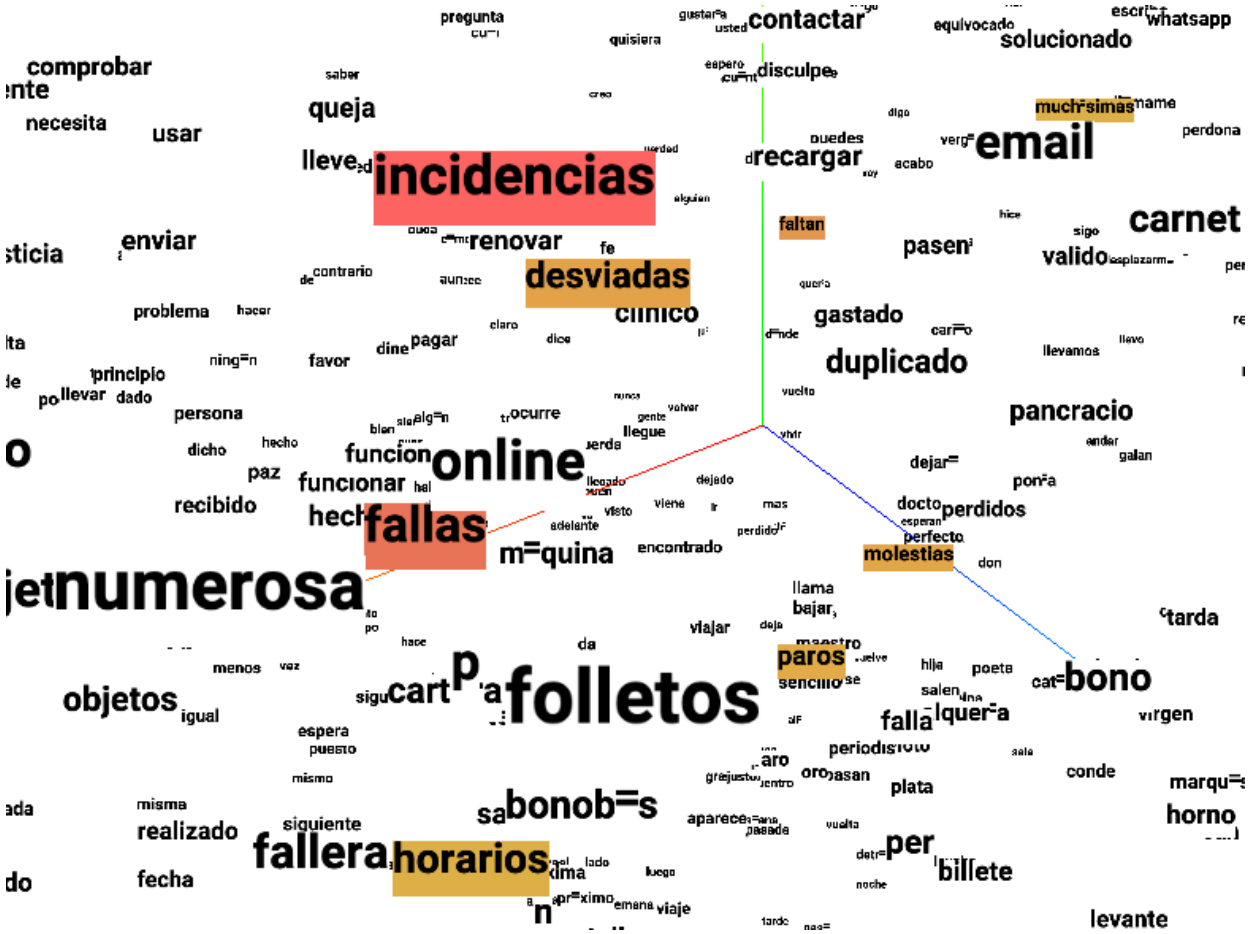


Ilustración 17. Representación vectorial "incidencias". Fuente: Elaboración propia

Visualización palabra *"billete":* En esta visualización 3D ponemos el foco en la palabra *billete*. Para medir la proximidad semántica está vez utilizamos la distancia euclídea entre vectores. En este caso podemos observar que las palabras *tarjeta, ida, carnet, móvil, bono, pase*, etcétera son las más próximas. Se puede entender que son sinónimos de soportes para viajar o en el caso de *ida* sería la palabra compuesta *billete de ida*. (Ver Tabla 18).

Palabra	*Distancia euclídea*
tarjeta	0,984
ida	0,988
carnet	0,992
móvil	1,054
bono	1,058
pase	1,070
línea	1,085
recargo	1,096
bonobús	1,107

Tabla 18. Vecinos palabra "billete". Fuente: Elaboración propia

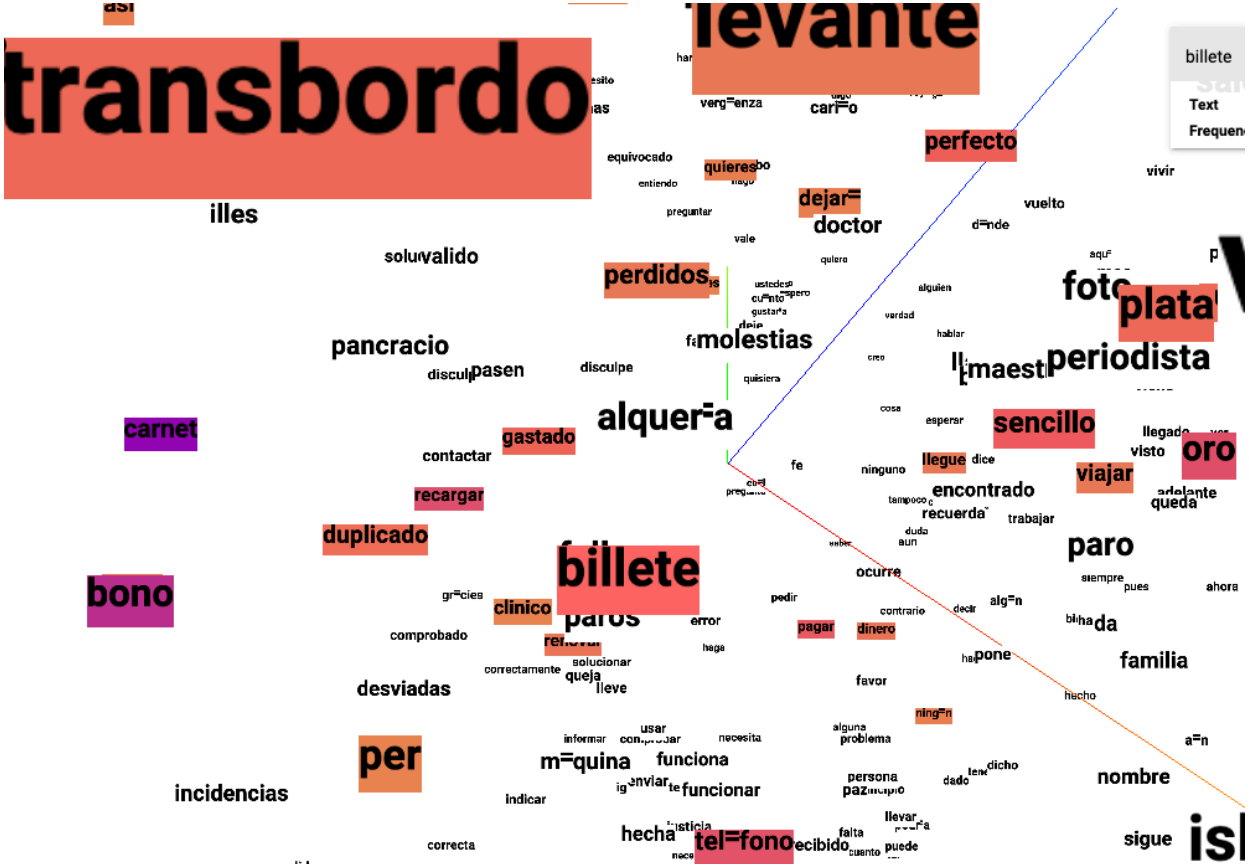


Ilustración 18. Representación vectorial "billete". Fuente: Elaboración propia

Visualización palabra *"parada":* Fotograma en 3D**.** Ponemos el foco en la palabra *paradas*. En este caso mediante el cálculo de la distancia del coseno podemos observar las palabras más próximas semánticamente hablando. *Autobuses, parada, trayecto, buses, líneas*, etcétera. (Ver Tabla 19).

Palabra	***Distancia coseno***
autobuses	0,302
parada	0,332
trayecto	0,334
buses	0,339
líneas	0,359
avenidas	0,366
autobús	0,407
autobus	0,423
estación	0,446

Tabla 19. Vecinos palabra "parada". Fuente: Elaboración propia

Ilustración 19. Representación vectorial "paradas". Fuente: Elaboración propia

Visualización palabra *"pasa"*: Fotograma en 3D. Nos centramos en la palabra *pasa*. A través de la distancia del coseno vemos que aparecen sinónimos o palabras que se utilizan comúnmente en las consultas del tipo estimación de llegada. (Ver Tabla 20).

Palabra	*Distancia coseno*
llega	0,270
va	0,287
pasando	0,412
ahí	0,442
vuelve	0,445
viene	0,466
lleva	0,480
deja	0,486

Tabla 20. Vecinos palabra "pasa". Fuente: Elaboración propia

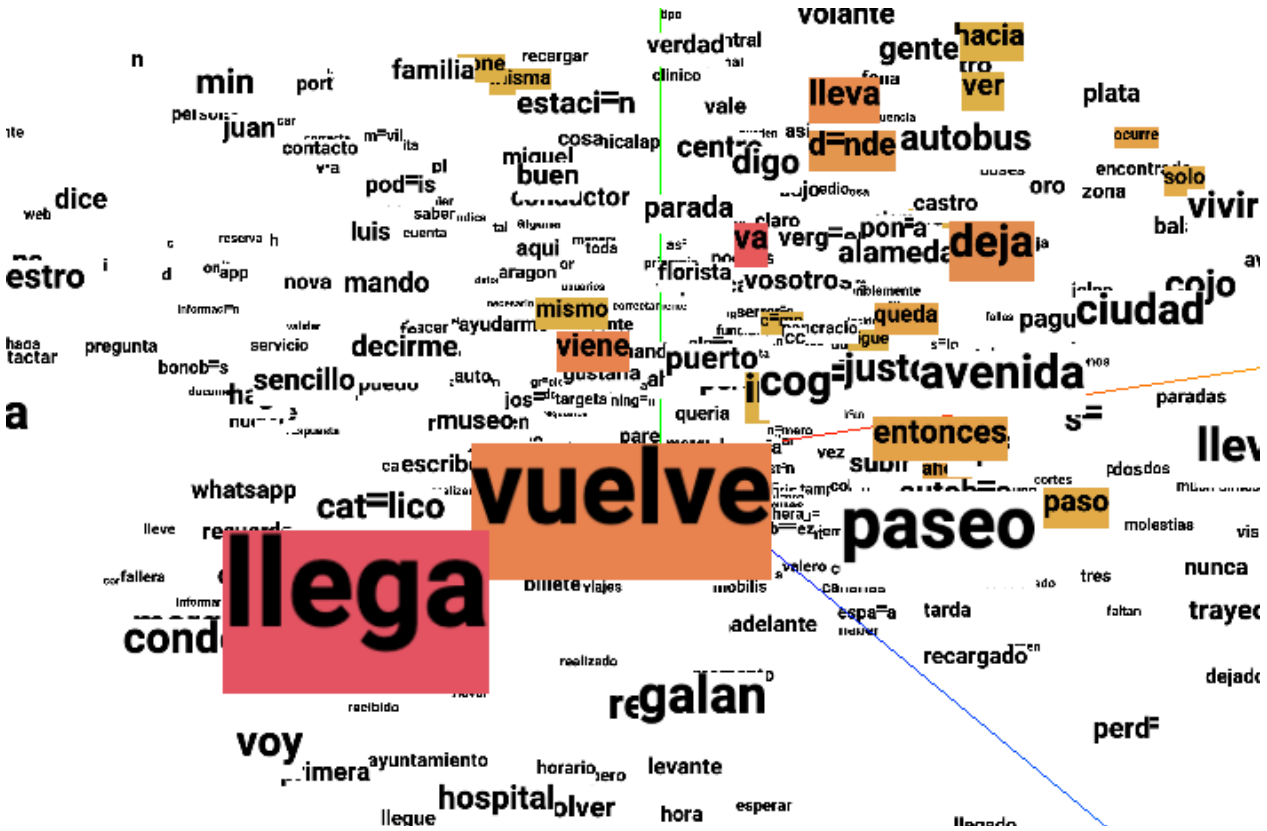


Ilustración 20.Representación vectorial "pasa". Fuente: Elaboración propia

Por último, se muestran dos visualizaciones representando una secuencia de imágenes en 2D. Están realizadas mediante la técnica llamada **incrustación de vecinos estocásticos distribuidos *(T-SNE)***. Es un algoritmo de aprendizaje no supervisado cuyo creador es *Laurens van der Maaten* y *Geoffrey Hinton* en el año 2008. Consiste en una técnica para la visualización de conjuntos de datos de alta dimensión y su único objetivo es la exploración de datos y la visualización. En términos muy simples *T-SNE* da una sensación o intuición de cómo se organizan los datos en un espacio de alta dimensión (Manish Pathack n.d.). Además *Laurens van der Maaten* muestra muchos ejemplos de uso en el áreas de investigación como el clima, ciberseguridad, bioinformática, investigación del cáncer, los vídeos se pueden ver en (*GoogleTechTalks* 2013)

El algoritmo tiene como objetivo reducir la complejidad de los números con muchas dimensiones, consiguiendo que sea comprensible por los seres humanos. Justamente trata de reducir el número de dimensiones comparando las similitudes (similitudes semánticas) de los puntos de datos. En nuestro caso, el efecto visual es que en cada iteración las palabras se agrupan por similitud semántica. Por ejemplo, los nombres de las calles se van agrupando.

A partir del modelo de datos predictivo creado que contiene los vectores de las palabras dentro del contexto de todas las consultas ciudadanas, se realiza una aproximación de funcionamiento del algoritmo mediante la aplicación *TensorBoard*. Ponemos el foco en la palabra *ayuntamiento* y obtenemos las 100 primeras palabras más próximas según la distancia del coseno. Después aplicamos el algoritmo *T-SNE* que se visualiza en *TensorBoard*. A lo largo de 3.000 iteraciones se aprecia un movimiento o recomposición de las palabras que se van uniendo por similitud.

En la Ilustración 21 se observa la situación inicial, primeras iteraciones del algoritmo, donde existe una cierta distancia entre los puntos. A partir de aquí ya se aprecia un movimiento de acercamiento entre ellos.

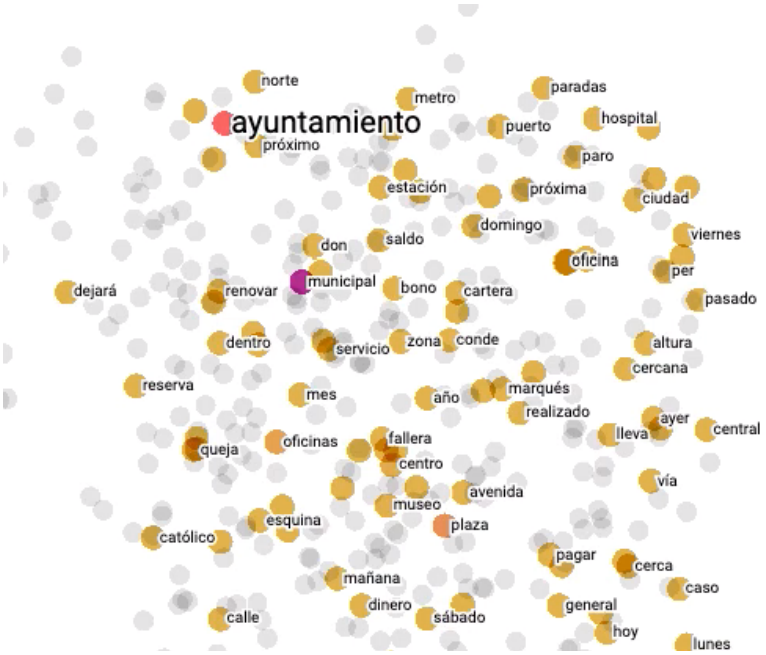


Ilustración 21. T-SNE secuencia 1. Fuente: Elaboración propia

En la Ilustración 22 ya se observa más cercanía o agrupación entre los puntos. Especial atención al punto “ayuntamiento” y “municipal” que en la ilustración anterior (21) tenían cierta distancia, pero en la ilustración actual (22) ya se encuentran alineadas, como el

fotograma es en 2D se visualizan solapados. Significa que la palabra “ayuntamiento” y la palabra “municipal” aparen juntas en muchas frases de usuario.

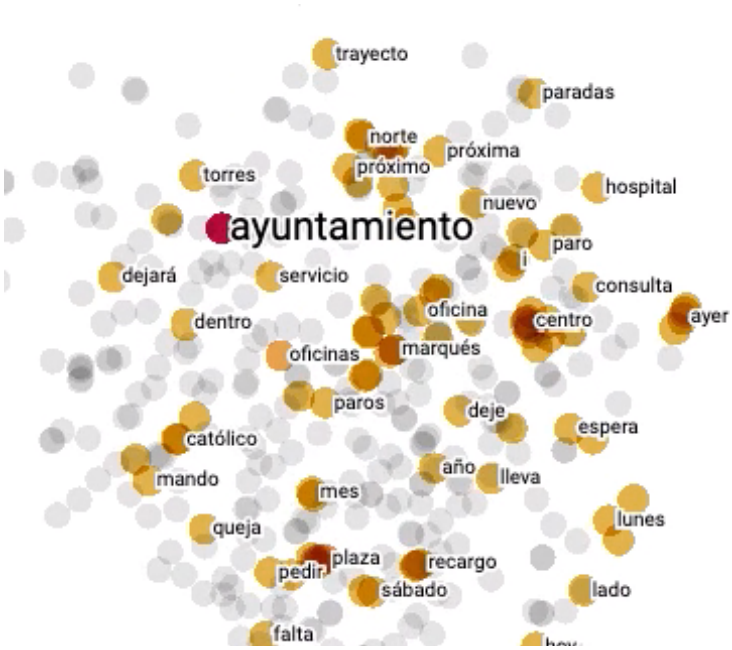


Ilustración 22.T-SNE secuencia 2. Fuente: Elaboración propia

4. ESQUEMA TÉCNICO

En el presente capítulo se explica la infraestructura técnica, repositorios de datos y los programas elaborados en cada fase de la implementación, que han sido necesarios para la creación del modelo.

4.1 Infraestructura técnica

En el trabajo todos los programas específicos se han elaborado con el lenguaje de programación *Python* versión 3. La extensión de los ficheros es *py.* Se han utilizado algunas librerías de licencia libre entre las cuales algunas son propias de la librería estándar de *Python* (*Python Software Foundation* 2014) y otras han sido creadas por autores. Las librerías se enumeran a continuación:

- **Os:** Sirve para administrar fácilmente las rutas relativas y completas de ficheros locales.

- **Locale:** Con él se configuran parámetros locales del equipo donde se va a ejecutar el programa. Por ejemplo como representamos los decimales numéricos.

- **Csv:** Permite la cómoda gestión de ficheros csv, tanto lectura como escritura.

- **Nltk:** Es una plataforma para el procesamiento del lenguaje natural. Proporciona interfaces fáciles de usar a más de 50 corpus y dispone de un conjunto de librerías de procesamiento de texto para clasificación, *tokenización*, *stemming*, *tagging*, *parseo* y razonamiento semántico además de un foro de discusión activo. Creado por *Dan Garrette*, *Peter Ljunglöf, Joel Nothman* et al.

- **NumPy:** Es el paquete fundamental para la computación científica con *Python*. *NumPy* también puede ser utilizado como un eficiente contenedor multidimensional de datos genéricos. Se pueden definir tipos de datos arbitrarios. Esto permite que *NumPy* se integre sin problemas y rápidamente con una amplia variedad de bases de datos. Creado por *Guido van Rossum, Jim Fulton, Jim Hugunin* et al.

- **SpaCy:** Librería de uso industrial para el procesamiento del lenguaje natural. Creado por *Matthew Honnibal*, *Ines Montani* y *Justin DuJardin* et al. Sus funcionalidades se han descrito ampliamente en capítulos anteriores.

- **Plac:** Sirve para pasar argumentos en línea de comandos fácilmente. Creado por *Michele Simionato.*

- **Random:** Este módulo implementa generadores de números pseudo-aleatorios para varias distribuciones.

- **Pathlib:** Este módulo ofrece clases que representan rutas de sistemas de archivos con semántica apropiada para diferentes sistemas operativos.

- **Sklearn:** Es un conjunto de módulos para el aprendizaje automático y la minería de datos. Creado por *Joris Van den Bossche, Loïc Estève, Thomas J. Fan* et al.

- **Pandas:** Proporciona estructuras de datos de alto rendimiento fáciles de usar y herramientas de análisis de datos para el lenguaje de programación *Python*. Creado por *Wes McKinney*, *Jeff Reback*, *Joris Van den Bossche et al.*

- **Math:** Este módulo proporciona acceso a las funciones matemáticas definidas por el estándar C.

- **Tqdm:** Es un medidor de progreso rápido y extensible para los bucles iterativos.

- **Tensorflow:** Es un marco de aprendizaje de máquina de código abierto para todo el mundo. Autor: *Google Inc*.

- **Telegram:** Es una librería que ayuda a construir *chatbots*. Autor: *Alexander Akhmetov*.

- **Geocoder:** Es una librería de geocodificación simple y consistente. Autor: *Denis Carriere*.

4.2 Programas específicos

En la Ilustración 23 se pueden observar los programas *Python* que han sido necesarios crear en cada una de las fases que componen el proyecto. Se realiza una breve descripción de cada programa a continuación:

1. **Anonimización*:*** Los *chats* originales (en formato *csv*) se han dejado en un repositorio temporal. El programa anonimiza el número de teléfono móvil que aparece en los *chats,* se hace lo propio con el nombre del fichero y se colocan los nuevos ficheros en un nuevo repositorio. Para ello se utiliza la función *hash* de *Python*.

2. **Filtrado y adecuación*:*** Sendos programas que se encargan de realizar filtros: desechar las cabeceras; quedarse con las consultas ciudadanas; desechar respuestas de *EMT* y sobre todo mapear a formato de entrada del modelo *NER* de *SpaCy*.

3. **Creación del modelo:** Con el algoritmo de creación de modelos predictivos de *SpaCy* se realizan 300 iteraciones para conseguir el descenso de gradiente y estabilizarlo. Nos basamos en el modelo de base ***es_core_news_md*** entrenado con palabras en español que contiene los vectores de palabras. Este programa genera el modelo.

4. **Evaluación del modelo:** Apoyados en la librería *sklearn* realizamos las comparaciones de lo real con lo predicho por el modelo generando una matriz de confusión.

5. **Visualización:** El programa genera un modelo *Tensorflow* a partir del modelo predictivo generado en *SpaCy*. El modelo *Tensorflow* se visualiza con la herramienta *Tensorboard*.

6. **Demostración:** Programa que crea un *chatbot* en *Telegram* enlazado con el modelo predictivo creado.

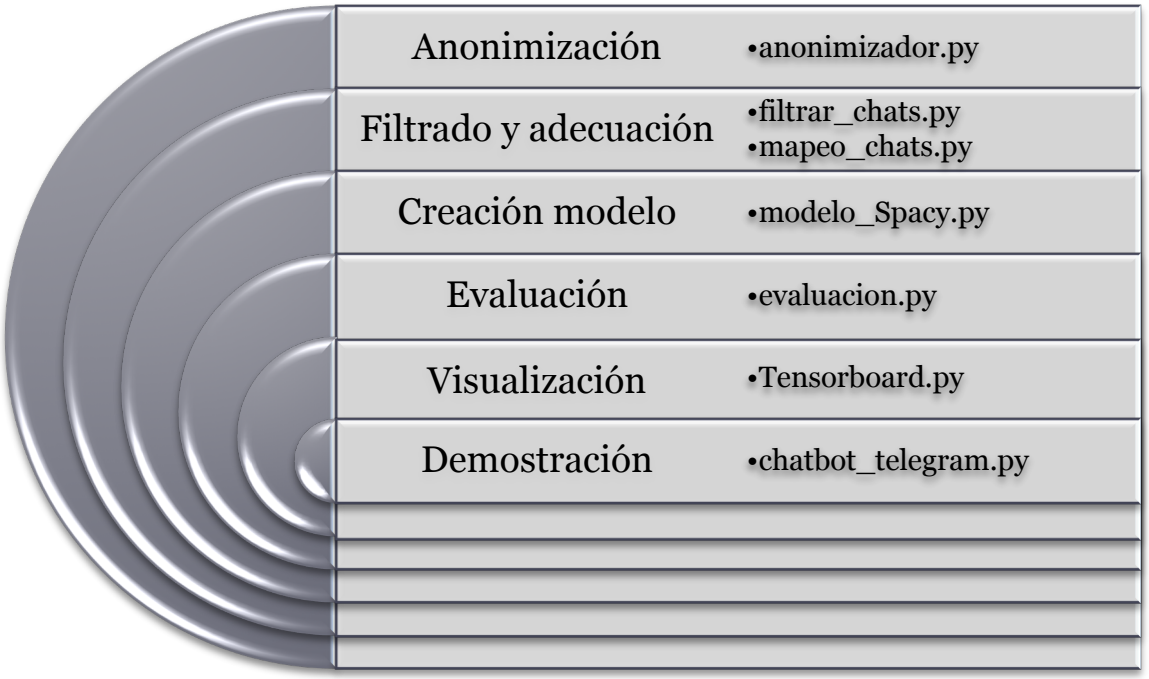


Ilustración 23. Fases y programas creados. Fuente: Elaboración propia

4.3 Repositorio proyecto: *Github*

En el presente trabajo se utiliza la capa gratuita del repositorio *Github* para almacenar todos los fuentes creados.

Github es una plataforma de desarrollo colaborativo con control de versiones idónea para ser el repositorio público del proyecto. En él se ha dejado tanto algunos datos anonimizados y filtrados reales como los programas utilizados para la creación del modelo.

El autor del presente trabajo queda a disposición de aclaraciones a través de esta vía.

La *URL* del repositorio es la siguiente: https://github.com/edgalpar/TFM

4.4 Alojamiento *chatbot: Amazon*

Una parte importante por comentar en un proyecto de este tipo es dónde alojar el robot. Se entiende que un robot debe ser un servicio siempre accesible para el público pues es una de sus mejores cualidades a explotar. Para ello debemos apoyarnos en un sistema informático robusto que ejecute el programa *chatbot* de manera continua,

escalable[3] y que esté preparado para cualquier contingencia o corte del servicio. Evidentemente debe estar publicado en *internet* dentro de la red y el protocolo de alguna plataforma de mensajería instantánea que lo permita.

Para las necesidades y tecnologías utilizadas en el presente trabajo hoy en día existen muchos proveedores de servicios en la nube entre los que destacan las americanas *Amazon Web Services (AWS*), *Microsoft Azzure*, *Google Cloud* e *IBM*.

En el presente trabajo se utiliza a modo de prueba la capa gratuita de *AWS* ("Capa Gratuita de *AWS* | *Cloud Computing* Gratis |*AWS*" n.d.), que permite aplicar la denominada informática sin servidor (*Serverless*), es decir, permite crear y ejecutar aplicaciones y servicios abstrayéndonos de la parte de los servidores (*hardware*). De esta forma nos centramos en la lógica de negocio, en el programa.

Para poder experimentar el funcionamiento del modelo predictivo en un *chatbot* se ha seleccionado el servicio *Elastic Compute Cloud* de *AWS*, en adelante *EC2.* Con *EC2* creamos y ponemos en funcionamiento un servidor *Ubuntu* 18.04 con recursos *t2.micro* (1 CPU virtual y 1 GB de memoria). Para poder crear el *chatbot* primero adecuamos el servidor realizando las siguientes configuraciones:

1. Establecer el par de claves *AWS.*
2. Conectarse al servidor *Ubuntu* mediante protocolo *ssh*[4] y *ftp*[5]*.*
3. Actualizar de *Ubuntu.*
4. Instalación de *Python 3.*
5. Instalación librerías *Spacy*, *Telegram*, etc
6. Subir modelo predictivo al servidor (*ftp*).
7. Crear programa y dejarlo corriendo.

Hay que comentar que para que la demo funcionase en el servidor dentro de la capa gratuita (servidor *EC2 t2.micro* 1 *CPU*, 1 GB de *RAM*) ha sido necesario optimizar el consumo de *RAM* ya que el modelo ocupa una gran cantidad memoria. Fuera de la capa gratuita hay varias modalidades de facturación en *AWS* donde la más común es bajo demanda. El precio por hora para hacer funcionar el *chatbot* de prueba en una instancia tipo t2.micro sería de 0,0116$ por hora más el 21% de I.V.A. Redondeando estaríamos hablando de 120€ impuestos incluidos.

[3] Es un anglicismo que describe la capacidad de un negocio o sistema de crecer en magnitud. *Wikipedia*
[4] Protocolo para conexión segura a un servidor remoto. Puerto *TCP* 22. *Wikipedia*.
[5] Protocolo de transferencia de archivos entre sistemas remotos. Puerto *TCP* 21. *Wikipedia*.

5. DEMO CHATBOT TELEGRAM

El modelo predictivo ha sido creado como un paquete independiente y portable[6] en un lenguaje ampliamente extendido actualmente como es *Python*. Por lo tanto, en estos momentos es un elemento perfectamente integrable con otros componentes desarrollados en el mismo lenguaje.

A modo de ejemplo integramos el modelo con el *API* proporcionado por *Telegram* ("*Telegram Bot API*" n.d.) y el *planificador de rutas* de *EMT* para la construcción de un *chatbot* que ayude a los ciudadanos para ir de un sitio a otro dentro de la ciudad de València.

El *chatbot*, denominado *Jhonny5_bot,* tiene un alcance limitado, con unas funcionalidades acotadas. Únicamente se pretende demostrar la potencialidad del modelo creado y su integración con otros sistemas de información. El funcionamiento de la *demo* se explica a continuación:

Planificador de Rutas: El viajero pregunta al *chatbot* desde dónde y hasta dónde quiere ir dentro de la ciudad de València. El modelo analiza la consulta entrante y acto seguido predice la información de origen y destino. Acto seguido si se ha obtenido el lugar de origen (desde) y el lugar destino (hasta) el *chatbot* traduce los lugares en coordenadas latitud-longitud e integra con el *geoportal* de *EMT* para planificar la ruta detallada para realizar el viaje. Finalmente el *chatbot* contesta al viajero con un enlace con la información requerida.

En la Ilustración 24 vemos una captura de pantalla con la conversación entre una persona y el *chatbot* y en la Ilustración 25 vemos una captura de pantalla del *planificador de EMT*.

[6] Se trata de un *software* que puede ejecutarse en cualquier plataforma, reutilizándose. *Wikipedia.*

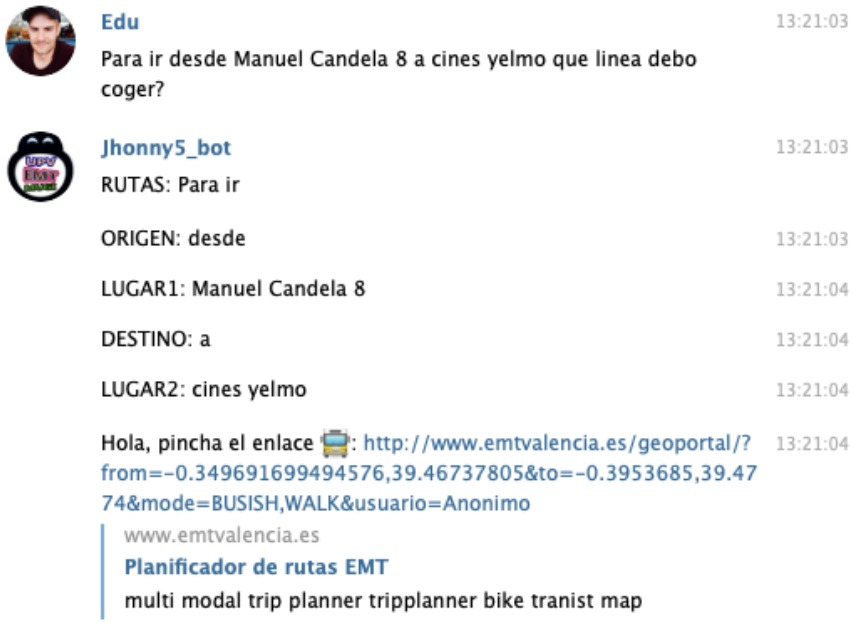


Ilustración 24. Bot Telegram con modelo predictivo 1. Planificador de rutas

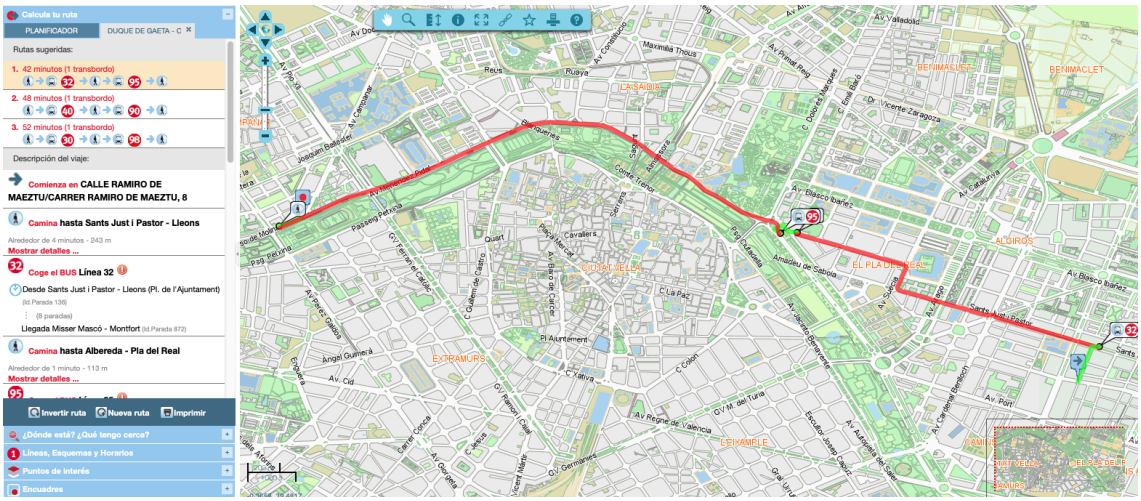


Ilustración 25. Planificador de rutas de EMT. Fuente: EMT

Estimación de llegada: El viajero pregunta cuánto tiempo falta para que llegue su autobús en una determinada parada. Se puede observar en la Ilustración 26 cómo el robot responde inmediatamente las palabras clave. En este caso faltaría integrarlo con el servicio de estimación de llegada de *EMT*.

Ilustración 26. Bot Telegram con modelo predictivo 2. Estimación de llegada

Por último, para experimentar y testear el *chatbot* de pruebas se ha dispuesto de 750 horas de ejecución gratuita al mes en el *hosting* de *AWS* durante el primer año. Los detalles de configuración se han descrito con más detalle en el capítulo anterior. Como se ha comentado *Johnny* 5 está disponible temporalmente en *Telegram* teniendo en cuenta la fecha de arranque de este a 25 de agosto de 2019.

El enlace del robot es el siguiente: ***https://t.me/Johnny5_chatbot***

6. CONCLUSIONES

En este capítulo final nos disponemos a hablar de lo hablado. En el inicio surge la idea de construir un robot que sea capaz de responder automáticamente las dudas de los viajeros y viajeras de València. Justamente para que el robot resulte de ayuda a la ciudadanía se requiere dotarle de inteligencia artificial. La inteligencia artificial se obtiene aprendiendo patrones y prediciendo a partir del histórico de datos existentes además de procesando el lenguaje natural.

Se obtienen los datos de las consultas ciudadanas de la *EMT*, en concreto de la Oficina de Atención al Cliente. Se realiza un estudio del marco legal del proyecto y se procede a anonimizar los datos para poder gestionarlos asegurándonos el cumplimiento de las leyes.

Se realiza un análisis del estado del arte para poder crear un modelo predictivo eficiente. Se crea el modelo predictivo a partir de herramientas actuales que aúnan algoritmos de *machine learning* y redes neuronales. El canal de comunicación, en nuestro caso es la mensajería instantánea, esto añade cierta complejidad y particularidades en la construcción del modelo.

Sobre la precisión del modelo creado se realizan las siguientes apreciaciones:

1. La inteligencia del modelo es elevada (95,22%) dentro de la muestra de consultas recogidas.

2. Si se evalúa el modelo con consultas no entrenadas, la precisión decae.

3. Es necesario acrecentar el esfuerzo en recoger datos de consultas ciudadanas lo más representativas posible.

4. Sería muy costoso crear un modelo que se retroalimentara autónomamente realizando un aprendizaje no supervisado. Esto es debido al trabajo manual que hay que realizar para formatear correctamente las entradas del modelo.

Sobre la integración del modelo con otros sistemas de información requeriría la realización de sendos proyectos de integración, se sugieren los siguientes:

1. ***Planificador rutas:*** Integración con el sistema de planificador de rutas de *EMT*.

2. ***Estimación de llegadas:*** Integración con el servicio web de estimación de llegadas de *EMT*.

Por último, se publica toda la información concerniente al presente trabajo para favorecer futuras investigaciones.

7. BIBLIOGRAFÍA

20Minutos.es. 2019. “Un Informático chino crea un *bot* que responde a su novia 24 horas al día,” 2019. https://www.20minutos.es/noticia/3670406/0/informatico-chino-crea-bot-responde-novia-24-horas-dia/.

Aced, Emilio, Rosario Heras y Carlos Alberto. 2015. Código de Buenas Prácticas En Protección de Datos Para Proyectos Big Data (*AEPD*). https://www.aepd.es/media/guias/guia-codigo-de-buenas-practicas-proyectos-de-big-data.pdf.

AEPD. 2016. “Orientaciones y Garantías En Los Procedimientos de Anonimización de Datos Personales.” https://www.aepd.es/media/guias/guia-orientaciones-procedimientos-anonimizacion.pdf.

Agencia Estatal Boletín Oficial del Estado. 2016. “BOE.Es - Documento DOUE-L-2016-80807.” 2016. https://www.boe.es/buscar/doc.php?id=DOUE-L-2016-80807.

Allison Parrish. 2018. “*Understanding Word Vectors: A Tutorial for Reading and Writing Electronic Text. (Python 2.7) Code Examples Released under CC0 Other Text Released under CC BY 4.0. GitHub.*” Consultado 2 mayo 2019. https://gist.github.com/aparrish/2f562e3737544cf29aaf1af30362f469.

Almeida, Aitor y Bilbao Aritz. 2018. “*Morelab - Word2vec Models for the Spanish Language*.” 2018. https://morelab.deusto.es/datasets/info/word2vec-models-for-the-spanish-language/.

Aranzadi Instituciones. 2019. “Procedimiento Administrativo Nº AP/00050/2017.”

Assaad Moawad. 2018. “*Neural Networks and Backpropagation Explained in a Simple Way*.” https://medium.com/datathings/neural-networks-and-backpropagation-explained-in-a-simple-way-f540a3611f5e.

Benja Lara. 2013. “Algoritmo *SimpleKmeans by* Benja Lara *on Prezi*.” 2013. https://prezi.com/ho_muzm2w5vq/algoritmo-simplekmeans/.

Blog de *WhatsApp*. n.d. Consultado 1 mayo 2019. https://blog.whatsapp.com/?lang=es.

Bruccoleri, Fernando. 2018. “El 47% de los empleos actuales desaparecerán en los próximos 25 años | El *Huffington Post*.” *Huffingtong Post*, 2018. https://www.huffingtonpost.es/fernando-bruccoleri/el-47-de-los-empleos-actuales-desapareceran-en-los-proximos-25-anos_a_23503110/.

Busvalencia.com. 2014. “Transporte Urbano de La Ciudad de València, Un Recorrido Por La Historia Hasta La Actualidad.” 2014. http://www.busvalencia.com/.

Calabuig, Jose Manuel, Lluis Miquel Garcia Raffi y Enrique Alfonso Sánchez-Perez. 2015. “Álgebra Lineal y Descomposición En Valores Singulares.” *Modelling in Science Education and Learning* 8 (2): 133. https://doi.org/10.4995/msel.2015.4010.

Campos, Arnau. 2018. “Asistente Virtual En Telegram Para Acceder a La Información Económica Municipal Del Ajuntament de València.”

https://riunet.upv.es/bitstream/handle/10251/111622/Campos - Asistente virtual en Telegram para acceder a la información económica municipal del Ajun....pdf?sequence=1&isAllowed=y.

Capa Gratuita de *AWS* | *Cloud Computing Gratis* |*AWS*. n.d. Consultado 17 agosto 2019. https://aws.amazon.com/es/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc.

Castells, Manuel y María Hernández. 2009. *Comunicación y Poder*. Alianza. https://escuelacriticavaldiviana.files.wordpress.com/2012/06/castells-manuel-comunicacion-y-poder.pdf.

Chatbot Chocolate. n.d. "Tu *Bot* de Seguros. Encuentra El Seguro Más Barato Vía *Chatbot* - *Billy*." Consultado 6 abril 2019. https://billyseguros.com/.

Chatfuel. 2015. "*About Chatfuel.*" 2015. https://chatfuel.com/about-us.html.

Chiappe, Doménico. 2019. "Los Robots No Podrán Gobernarnos Mejor | Las Provincias." *Las Provincias*, 2019. https://www.lasprovincias.es/sociedad/oppenheimer-robots-no-podran-gobernarnos-mejor-20190414220842-ntrc.html.

Chris Nicholson. n.d. "*A Beginner's Guide to LSTMs and Recurrent Neural Networks* | *Skymind*." Consultado 9 mayo 2019. https://skymind.ai/wiki/lstm.

"*Dialogflow*." n.d. Consultado 15 marzo 2019. https://dialogflow.com/.

Ding Jiangbo. 2018. "*Wireless AI for Networks That Understand You.*" https://www-file.huawei.com/-/media/corporate/pdf/publications/communicate/85/05-en.pdf.

Elecciones *Chat* - El *Chatbot* y *Voicebot* Comparador de Programas Políticos Para Elecciones Generales 2019. Vox, PP, PSOE, Podemos, Ciudadanos." n.d. Consultado 19 mayo 2019. https://elecciones.chat/.

EMTValència. 2019. "La EMT ha registrado 96,1 millones de viajeros 2018, 600.000 más que en 2017 - EMT València." *EMT Info*. http://emtvalencia.info/es/2019/01/la-emt-ha-registrado-961-millones-de-viajeros-en-2018-600-000-mas-que-en-2017/.

fattynoparents. n.d. "*Backup WhatsApp Chats.*" https://chrome.google.com/webstore/detail/backup-whatsapp-chats/gmbicfpadlmgkfhfepknbmemfhahelll.

Ferrer-Sapena, Antonia, Fernanda Peset y Rafael Aleixandre-Benavent. n.d. "Acceso a los datos públicos y su reutilización: *Open Data* y *Open Government*." Consultado 6 junio 2019. https://doi.org/10.3145/epi.2011.may.03.

Garrido, María Elisa Cuadros. 2014. "El Uso Del Whatsapp En Las Relaciones Laborales."

GeitGey, Adam. 2016. "*Machine Learning Is Fun Part 6: How to Do Speech Recognition with Deep Learning.*" 2016. https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a.

GeitGey, Adam. 2019. "*Natural Language Processing Is Fun Part 3: Explaining Model*

P*redictions.*" 2019. https://medium.com/@ageitgey/natural-language-processing-is-fun-part-3-explaining-model-predictions-486d8616813c.

Gobierno de España. n.d. "PAe - Leyes y Normativa Básicas En Administración Electrónica." Consultado 18 mayo 2019. https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_LegNacional/pae_NORMATIVA_ESTATAL_Adm_Elect_basica.html#.XN-UU1MzbCR.

Google. n.d. "*Embeddings | TensorFlow.*" Consultado 1 marzo 2019. https://www.tensorflow.org/guide/embedding.

Google. n.d. "*TensorFlow Hub.*" Consultado 1 marzo 2019. https://tfhub.dev/google/universal-sentence-encoder/2.

GoogleTechTalks. 2013. "*Visualizing data using T-SNE - YouTube.*" 2013. https://www.youtube.com/watch?v=RJVL80Gg3lA.

Graves, Alex, Santiago Fernández, Faustino Gomez y *Jürgen Schmidhuber*. 2006. "*Connectionist Temporal Classification.*" *In Proceedings of the 23rd International Conference on Machine Learning - ICML '06*, 369–76. New York, New York, USA: ACM Press. https://doi.org/10.1145/1143844.1143891.

Honnibal, Mathew e *Ines Montani.* 2019. "*Install SpaCy · SpaCy Usage Documentation.*" 2019. https://spacy.io/usage.

IEEE y UNED. n.d. "Código Ético Del IEEE." Consultado 24 mayo 2019. http://www.ieec.uned.es/investigacion/ieee_dieec/co_etico_ieee.htm.

Ius Mentis law and technology explained. 2005. "T*he MD5 Cryptographic Hash Function (in Technology > Hashfunctions @ Iusmentis.Com).*" 2005. https://www.iusmentis.com/technology/hashfunctions/md5/.

Manish Pathack. n.d. "I*ntroduction to T-SNE (Article) - DataCamp.*" Consultado 18 julio 2019. https://www.datacamp.com/community/tutorials/introduction-t-sne.

Mathew Honnibal e *Ines Montani*. n.d. "*Annotation Specifications · SpaCy API Documentation.*" Consultado 14 julio 2019. https://spacy.io/api/annotation#biluo.

Methodology | School of Data - Evidence Is Power. 2019. https://schoolofdata.org/methodology/.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2003. "10.1162/153244303322533223." *CrossRef Listing of Deleted DOIs* 1 (January). https://doi.org/10.1162/153244303322533223.

Omale, Gloria. 2019. "*Gartner Predicts 25 Percent of Digital Workers Will Use Virtual Employee Assistants Daily by 2021.*" *Gartner Inc*. https://www.gartner.com/en/newsroom/press-releases/2019-01-09-gartner-predicts-25-percent-of-digital-workers-will-u.

Pandorabots. n.d. "*Mitsuku*." Consultado 11 abril 2019. https://www.pandorabots.com/mitsuku/.

Pennington, Jeffrey, Richard Socher y *Christopher D. Manning*. 2019. "*GloVe: Global*

Vectors for Word Representation." 2019. https://nlp.stanford.edu/projects/glove/.

Preferred Networks Inc. y Preferred Infrastructure Inc. 2015. "*Word2vec.*" 2015. https://docs.chainer.org/en/stable/examples/word2vec.html.

Prieto, Raquel, *Carter Cromwell* y *Suresh Bashkaran*. 2016. "*70 Percent of Global Population Will Be Mobile Users | The Network*" 2016. https://newsroom.cisco.com/press-release-content?articleId=1741352.

Python Software Foundation. 2014. "T*he Python Standard Library — Python v3.2.6 Documentation.*" 2014. https://docs.python.org/3.2/library/.

Řehůřek, Radim. 2019. "*Gensim: Topic Modelling for Humans.*" 2019. https://radimrehurek.com/gensim/.

R*enuka Joshi*. n.d. "*Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog.*" Consultado 13 julio 2019. https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/.

Rezepte Suchen per WhatsApp - so Geht's! | *LECKER.* n.d. Consultado 27 mayo 2019. https://www.lecker.de/rezepte-suchen-whatsapp-so-gehts-68932.html.

RTVE y EFE. 2019. "Tecnología: *WhatsApp* cumple diez años con más de 1.500 millones de usuarios en todo el mundo - RTVE.Es." http://www.rtve.es/noticias/20190224/whatsapp-cumple-diez-anos-mas-1500-millones-usuarios-todo-mundo/1889660.shtml.

Rubén López. 2014. "¿Qué es y cómo funciona *'Deep Learning'*? | Rubén López." 2014. https://rubenlopezg.wordpress.com/2014/05/07/que-es-y-como-funciona-deep-learning/.

Techopedia Inc. 2019. "*What Is a Chatbot? - Definition from Techopedia.* 2019. https://www.techopedia.com/definition/16366/chatterbot.

Telegram Bot API. n.d. Accessed August 13, 2019. https://core.telegram.org/bots/api.

U-Report - U-Report Available on Facebook Messenger! n.d. Consultado 27 mayo 2019. https://ureport.in/story/254/.

United Nations. n.d. "*The Ten Principles | UN Global Compact.*" Consultado 11 julio 2019. https://www.unglobalcompact.org/what-is-gc/mission/principles.

Victor Roman. 2019. "Algoritmos *Naive Bayes*: Fudamentos e Implementación." 2019. https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fudamentos-e-implementación-4bcb24b307f.

Victor Roman. 2019. "*Machine Learning* Supervisado: Fundamentos de La Regresión Lineal." 2019. https://medium.com/datos-y-ciencia/machine-learning-supervisado-fundamentos-de-la-regresión-lineal-bbcb07fe7fd.

Wikipedia. 2019. "Empresa Municipal de Transportes de València." https://es.wikipedia.org/wiki/Empresa_Municipal_de_Transportes_de_Valencia.

Will Koehrsen. 2017. "*Random Forest Simple Explanation - Will Koehrsen - Medium*." 2017. https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d.

Worswick, Steve. 2018. "*Mitsuku Wins Loebner Prize 2018! – Pandorabots-Blog – Medium.*" 2018. https://medium.com/pandorabots-blog/mitsuku-wins-loebner-prize-2018-3e8d98c5f2a7.

Zellig S. Harris. n.d. "*Distributional Hypothesis - ACL Wiki.*" Consultado 3 junio 2019. https://aclweb.org/aclwiki/Distributional_Hypothesis.

Zhao, Shirley. 2017. "*What is ETL? (Extract, Transform, Load) | Experian.*" 2017. https://www.edq.com/blog/what-is-etl-extract-transform-load/.

8. AGRADECIMIENTOS

Manuel Ángel López Fuentes
Mª Carmen Álvarez Guaita
Vicente Buendía Ramón

Antonia Ferrer Sapena
María Fernanda Peset Mancebo
Juan Vicente Oltra Gutiérrez
Diego Álvarez Sánchez
Germán Moltó Martínez
Marta Fernández Diego
Y resto de profesores
Álvaro Durá de Lamo
Liliana Bayona Castañeda
Fabio Santos Lobao
Luiza Pretrosyan
Victoria Bulavina
y resto de compañeros del máster

José Manuel Calabuig Rodríguez
Lluís Miquel García Raffi

Y sobre todo a mi familia por su apoyo incondicional

Ilustración 27. ¡Ojalá vuelva el Ártico! Autor: Álvaro Gallardo (8 años)