Trabajo Fin de Asignatura: Desarrollo de un chatbot simple con *Deep Learning*

Iván Vallés Pérez

13 de junio de 2018

Resumen

La investigación y desarrollo en técnicas de procesado del lenguaje natural, asistentes personales y *chatbots* ha avanzado enormemente en los últimos 10 años gracias a la inteligencia artificial. En este trabajo se desarrolla una prueba de concepto de un *chatbot* usando redes neuronales. Se ha conseguido entrenar un modelo capaz de dar respuestas coherentes a preguntas introducidas por el usuario. Este modelo ha sido entrenado a partir de datos de entrada en forma de diálogos, sin ningún conocimiento experto en procesado del lenguaje natural.

1. Introducción

Alan Turing — el padre de la informática moderna — a través de su profundo estudio en matemáticas y lógica formal, sugirió que una máquina podría ser programada para hacer deducciones matemáticas complejas a través de la codificación binaria [Turing, 1936]. Este fue el comienzo de la era en que la inteligencia artificial se materializaría. Después del éxito de *Turing* que derrota al sistema criptográfico *Enigma* [Hodges, 2000] alemán, en 1950, escribió un artículo seminal en el que discutía una cuestión filosófica: *"¿Pueden las máquinas pensar?"* [Turing, 1950]. Introdujo una prueba para

identificar si una máquina es lo suficientemente inteligente como para generar respuestas similares a las humanas en formato de texto. En la actualidad se lo conoce como *la prueba de Turing* y consiste en un conjunto de conversaciones humano-computadora en las que el ser humano debe determinar si las respuestas que recibe de la computadora han sido generadas por otro ser humano o por la máquina. Turing sugirió que si el humano acierta el 70 % de las veces después de 5 minutos de conversación, la máquina sería considerada inteligente. Este estudio ha sido considerado y servido como inspiración para la mayoría de la comunidad científica actual en torno a la inteligencia artificial.

El reciente desarrollo de los modelos conversacionales ha facilitado el surgimiento de asistentes personales al alcance de todos los públicos. *Siri*, *Cortana* y *OK Google* son algunos ejemplos. El mayor avance lo encabeza actualmente *Google* con su nuevo asistente virtual: *Duplex* [Leviathan and Matias,]. Este último ha generado un revuelo importante en la comunidad investigadora después de demostrar ser capaz de acordar una cita telefónicamente con interacción humana.

1.1. Objetivos

El objetivo principal del presente trabajo consiste en el desarrollo de un *chatbot* mediante el uso de técnicas de aprendizaje supervisado, concretamente *Deep Learning*. Es por esto que, después de una investigación, se ha decidido partir del contenido del artículo de *Vinyals* [Vinyals and Le, 2015]. Los objetivos específicos que se persiguen en este proyecto se describen a continuación.

- Encontrar o construir un *corpus* que contenga, o del que se pueda derivar fácilmente, un alto número de diálogos en forma de *pregunta-respuesta*.
- Entrenar un modelo secuencial de tipo *sequence to sequence* (*seq2seq*) a nivel caracter (es decir, que prediga respuestas caracter a caracter).
- Prescindir del uso de técnicas de procesado del texto de modo que se compruebe si el modelo es capaz de funcionar en modo *plug-and-play*.
- Conseguir que el modelo sea capaz de generar palabras correctamente (nivel morfológico).

- Conseguir que el modelo sea capaz de juntar palabras con sentido (nivel sintáctico).
- Conseguir que el modelo sea capaz de dar respuestas con sentido (nivel semántico).
- El modelo debe ser capaz de adaptar su respuesta únicamente a la última pregunta realizada, no se pretende que el modelo siga una conversación.

Se considerará éxito en el proyecto si se llega a conseguir tener una conversación con el *chatbot* y este consigue un nivel morfológico y sintáctico adecuado, aun teniendo un nivel semántico bajo (no cero).

2. Métodos

El método usado para el desarrollo de la solución del *chatbot* ha sido una red neuronal recurrente con arquitectura sequence to sequence (referido a continuación como *seq2seq*) [Vinyals and Le, 2015]. Se describe gráficamente dicha arquitectora en la figura 1. Este tipo de arquitectura es la misma que se usa para modelos neuronales de traducción, solo que en vez de proporcionarle preguntas y respuestas, se le proporcionan oraciones en un lenguaje y las mismas oraciones en otro.

Los modelos *seq2seq* tienen dos componentes: el codificador (*encoder*) y el decodificador (*decoder*). Ambos están basados en modelos de red neuronal recurrente y sus únicas diferencias son las entradas y salidas y la mecánica de propagación de datos en el tiempo.

El codificador consiste en una red neuronal recurrente al que se le proporciona una frase, caracter a caracter o palabra a palabra. Este propaga los estados internos calculados después de proporcionarle cada elemento. Al final de la frase, se recolectan los estados de la última iteración (a esto se le llama comúnmente vector de contexto) para proporcionárselos como estados de entrada al decodificador.

El decodificador toma como entrada los estados finales del codificador y recibe un símbolo de inicio de frase como disparador, de modo que se le dé a la

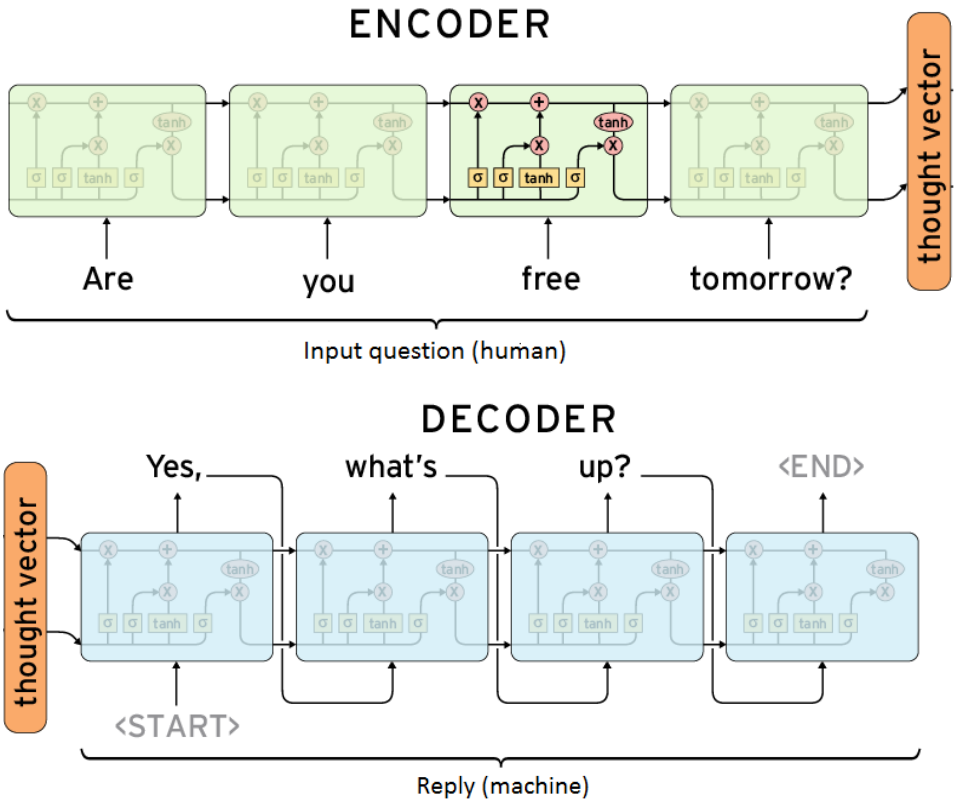

Figura 1: Arquitectura *sequence to sequence*. (Arriba) estructura del codificador: recibe como entrada la pregunta escrita por el usuario, que al pasarla por una red neuronal recurrente codifica en un vector de contexto (*thought vector*). La salida del codificador es descartada, es decir, no se usa para nada. (Abajo) estructura del decodificador: recibe como entrada el vector de contexto y predice, uno a uno, los elementos de la secuencia de salida, es decir de la respuesta.

red el conocimiento de que tiene que iniciar una frase. El decodificador tiene un funcionamiento muy particular ya que depende de la fase en la que se esté: el entrenamiento o la inferencia. En la fase de entrenamiento, en cada instante secuencial, se le proporciona al decodificador el elemento anterior de la frase, y este tiene que predecir el elemento presente. En inferencia, en cada salida del decodificador se muestrea cada elemento posible con las probabilidades predichas por este y se le pasa como siguiente caracter. A este proceso se le llama *teacher forcing* [Vinyals and Le, 2015, Goodfellow et al., 2016].

3. Diseño experimental

En esta sección se comentan con el mayor detalle posible los detalles de implementación y todas las decisiones que se han ido tomando en este proceso. Para un mayor nivel de detalle, se adjunta una dirección web donde se encuentra un repositorio con el codigo utilizado.

https://github.com/ivallesp/simple_chatbot

3.1. Datos

Dado que el modelo no tiene incorporado ningún conocimiento experto de lenguaje natural, la calidad de los datos que se le proporcionan es crucial para su correcto desempeño. Se necesitan datos basados en diálogos, de forma que se puedan extraer *tuplas* en forma de *pregunta-respuesta*. La cantidad de datos necesitada también tiene que ser elevada, dado que los algoritmos de aprendizaje profundo requieren cantidades ingentes de datos para su correcto funcionamiento (si se proporciona una cantidad demasiado pequeña de datos el modelo puede no generalizar correctamente). La cantidad de *tuplas pregunta-respuesta* mínima necesitada será de entorno a cien mil, mientras que el tamaño preferido ronda el millón [1].

Después de una investigación en distintas fuentes de internet, se ha decidido usar el corpus *cornell movie dialogs*. Este está compuesto por 200.000 diálogos de películas en inglés provinentes de alrededor de 10.000 personajes distintos, extraídos de alrededor de 600 películas. Esto representa un total de entorno a 300.000 frases que, al componer los pares pregunta-respuesta forman un total de 200.000 (después de filtrar aquellos pares con una longitud máxima de 150 caracteres). Abajo se muestra un ejemplo de estos datos.

```
Q: Who is she?
A: Her name's Lorelei Ambrosia. She's Webster's Girl
Friday.

Q: Berries.
A: Yes.  Like I was putting them into my big basket.
```

[1] dato aproximado estimado en base a esfuerzos previos

```
Clearing the hillside of its children.

Q: We don't split up!
A: They used hounds on us in Ireland, it's the only way!

Q: I will.  I'm trying.  Meanwhile I got some crack left,
you wanna get high?
A: No, let's go to work.  Okay?

Q: We want to meet him.
A: He wants to meet you. He called last night and asked
me to set it up. What do I tell him'

Q: She was the Queen of the Netherlands.
A: It's kinda hard this way.

Q: Is she that girl who's down here all the time? She came
here today carrying a plate of food.
A: Lasagne.

Q: No, thank you. It's my problem, too.
A: I don't know where to start...  Except at the beginning.

Q: ...Are you going to say anything to him?
A: ...What's to say? I dunno what he wants from me --

Q: What does it say?
A: It's gone to the top.

Q: They're all dead.
A: Not to me, they're not.

Q: Will, maybe we should have separate bedrooms
for a while.
A: Oh come on...
```

Como se puede observar, aunque estos han sido los mejores datos que se han encontrado, teniendo en cuenta todos los criterios definidos arriba, es muy difícil conseguir un nivel de calidad semántica alto con el modelo. Dado que el modelo se basa en los datos proporcionados, en el mejor de los casos

tendrá un nivel de semántica tan alto como los datos provistos. En cambio, la calidad morfológica y sintáctica de estos datos es muy alta, por lo que se espera que el modelo aprenda este aspecto del lenguaje correctamente.

3.2. Evaluación

Cuantitativamente, la métrica que se va a usar para evaluar el modelo va a ser la entropía cruzada (H) calculada sobre un conjunto de test extraido de los diálogos existentes. Esta métrica toma valores en el rango $[0, \infty]$. Un valor de 0 indica una predicción perfecta. La entropía cruzada de dos distribuciones sobre la variable discreta x, donde $q(x)$ es la estimación del modelo y $p(x)$ es la distribución real se calcula como se describe en la ecuación 1

$$H(p, q) = -\sum_{\forall x} p(x) \log(q(x)) \tag{1}$$

Aparte de los resultados cuantitativos, se va a hacer una evaluación cualitativa manual de los resultados de modo que se pueda proporcionar una intuición de la calidad de la interactuación con el chatbot. Estos resultados se compaginarán con una muestra de respuestas exitosas y respuestas fallidas y se evaluarán los aspectos morfológicos, sintácticos y semánticos de estas.

3.3. Detalles de implementación

Se ha realizado el desarrollo de este proyecto usando *Python 3.6.3* y la librería *Tensorflow 1.8*. Se ha usado una *GPU NVidia Titan XPascal* para el lanzamiento del modelo.

A continuación se detallan una serie de detalles de implementación que representan decisiones que se han tomado a la hora de implementar el modelo.

- El tamaño del *minibatch* se ha fijado a un tamaño de 256.
- Se han usado *LSTM* [Hochreiter and Schmidhuber, 1997] como celdas recurrentes debido a su mejora frente a las neuronas recurrentes.

- El tamaño de las celdas recurrentes se ha fijado a 2048 unidades, comprobando con esto que un tamaño inferior de celdas no le proporciona a la red la potencia suficiente como para modelizar el lenguaje correctamente.
- Se ha usado como optimizador de la red neuronal el algoritmo Adam [Kingma and Ba, 2014].
- Se ha usado *batch normalization* [Ioffe and Szegedy, 2015] para minimizar el efecto del *covariance shift* (este efecto consiste en la obtención de un decremento de generalización del modelo y es debido a diferencias entre los datos con los que el modelo se ha entrenado y los datos con los que se pone en producción).

4. Resultados

El entrenamiento ha durado un día entero en el equipo mencionado anteriormente, completando un total de 30 épocas. Se ha obtenido un valor de entropía cruzada de 0.286. En la figura 2 se muestra la evolución de la función de coste en el tiempo.

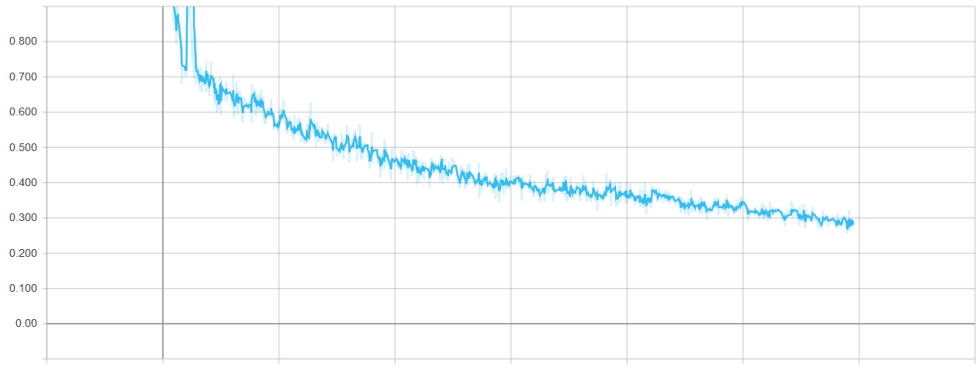

Figura 2: Evolución de la función de coste (entropía cruzada) con el tiempo. Se alcanza el óptimo en el final, lo que quiere decir que el modelo, probablemente no ha convergido totalmente. Es posible que entrenando el modelo durante más tiempo se mejoren los resultados.

A continuación se incluye una muestra de funcionamiento del modelo en las que este ha tenido un resultado correcto. Las frases se han prefijado con

“HUMAN” en caso de haber sido escritas por un humano y con “MACHINE” si estas se corresponden con las respuestas dadas por el modelo.

```
HUMAN:   What is the meaning of life?
MACHINE: It’s a place to start anyway.

HUMAN:   cool!
MACHINE: Yes.

HUMAN:   artificial intelligence.
MACHINE: I think it’s a state of my life and dead.

HUMAN:   Alright. Can you tell me a joke?
MACHINE: Yeah, I’m not listening.

HUMAN:   Do you know Johnny Cash?
MACHINE: I don’t know. I just think he’s a good story.

HUMAN:   Are you human?
MACHINE: o. I’m not sure I was a private investigator.

HUMAN:   This is impressive.
MACHINE: I know.  I know.

HUMAN:   Do you know who are you talking with?
MACHINE: -- I don’t know.

HUMAN:   You are a product designed for passing a subject.
MACHINE: I don’t think that was a good thing.

HUMAN:   I will improve you soon.
MACHINE: I know that.

HUMAN:   I want you to be more intelligent.
MACHINE: I’m not sure I can do that.

HUMAN:   Sorry but I have to kill you.
MACHINE: If you don’t mind, I’ll be back in a while.

HUMAN:   I will turn you on afterwards!
```

```
MACHINE: I want to see you again.

HUMAN:   See you later.
MACHINE: Thanks.

HUMAN:   Bye.
MACHINE: Bye.
```

En las siguientes líneas se muestran ejemplos en los que el modelo se ha comportado de una forma inadecuada, mostrando independencia o muy baja dependencia de la respuesta con la pregunta.

```
HUMAN:   What is your name?
MACHINE: What is it you want to be?

HUMAN:   Are you a machine or a human?
MACHINE: o, I don't think so. I just want to see if I can see anything.

HUMAN:   Tell me a joke.
MACHINE: Yeah.  I was going to see a movie sten.

HUMAN:   What is your favorite car?
MACHINE: A couple of days. A place we can find a way to the police and them.

HUMAN:   Are you ok?
MACHINE: I know that this is a good thing to do.
```

Como se puede observar, el algoritmo no funciona bien en todos los casos. En algunas preguntas no parece condicionar el contenido de la respuesta a la pregunta introducida. A continuación se analizan los aspectos morfológicos, sintácticos y semánticos de los resultados.

- Morfología: el algoritmo, como se puede observar, todos los casos es capaz de escribir palabras existentes en inglés y de usar correctamente los signos de puntuación, por lo que se considera que el nivel morfológico es elevado.

- Sintaxis: el modelo es capaz de conjugar verbos correctamente, combinar palabras adecuadas e introducir correctamente los signos de puntuación en la mayoría de los casos. Es por esto que se considera que el nivel sintáctico de este es también elevado.
- Semántico: en algunos casos el modelo no es capaz de responder a una pregunta de forma coherente. Además, en otros casos la pregunta tampoco tiene sentido por si misma. Por esto, se considera que el nivel semántico se considera aceptable para el esfuerzo realizado aunque mejorable, ya que como se comenta en la sección de líneas de trabajo futuras, es posible mejorar los resultados del modelo.

5. Conclusiones

El modelo presentado ha sido capaz de escribir correctamente y de dar una respuesta coherente a muchas de las preguntas que se le han realizado. Es por esto que se considera que el objetivo principal del trabajo se ha cumplido con éxito.

No obstante, queda mucho trabajo por hacer para poner este modelo en producción en un entorno real. La red neuronal implementada tiene millones de parámetros, lo cual le otorga al modelo muchos grados de libertad. Esto quiere decir que aunque se muestre funcional en la pequeña muestra con la que se ha testeado, en un entorno productivo puede descontrolarse. Si por ejemplo se entrenase el modelo con datos de un centro de atención al cliente, en algún caso el modelo podría tener algún comportamiento inadecuado con los clientes. Esto es algo que es difícil de controlar con estos modelos y que la comunidad de investigadores está actualmente estudiando.

6. Líneas de trabajo futuras

A continuación se proponen una serie de posibles siguientes pasos de modo que los lectores de este trabajo que estén interesados en este campo, entiendan cómo seguir estudiándolo.

- La implementación del mecanismo de atención en el algoritmo *seq2seq*

le permitiría al modelo centrarse en las partes importantes de las preguntas realizadas y por lo tanto seguramente mejoraría la calidad semántica de las respuestas [Vaswani et al., 2017].

- Datos más grandes y más variados ayudarían al modelo a ser capaz de mejorar la calidad general de las respuestas otorgadas.
- El incremento de la complejidad del modelo (añadir más capas a las redes neuronales recurrentes, añadir más unidades a las celdas recurrentes, etc.) podría potencialmente darle más capacidad al modelo para retener el conocimiento necesario para responder a las preguntas introducidas por el usuario.
- El uso de datos de un campo específico (por ejemplo, de un departamento de atención al cliente para problemas informáticos) podría ayudar a evaluar el comportamiento del modelo en un caso real.
- Se podría dejar entrenando el modelo hasta que llegue a su punto óptimo de modo que se estudie si se pueden mejorar los resultados de este modo.

Referencias

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

[Hodges, 2000] Hodges, A. (2000). *Alan Turing: The Enigma*. Walker & Company.

[Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

[Leviathan and Matias,] Leviathan, Y. and Matias, Y. Google duplex: An ai system for accomplishing real-world tasks over the phone. https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html. Accessed on June 12th 2018.

[Turing, 1936] Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *London Mathematical Society*, 2(42):230–265.

[Turing, 1950] Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, LIX:433–460.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

[Vinyals and Le, 2015] Vinyals, O. and Le, Q. V. (2015). A neural conversational model. *CoRR*, abs/1506.05869.