

Traducción instantáneas en llamadas telefónicas

Informe inicial

UAB Ingeniería informática

Raúl Aquilué – 1500773

En este informe se documentará una introducción del proyecto de TFG explicando la motivación de este, los objetivos que se quieren conseguir y la metodología que se usará a largo del trabajo. Además, se hará una breve explicación del estado actual de proyectos similares y del enfoque que se quiere obtener con este proyecto.

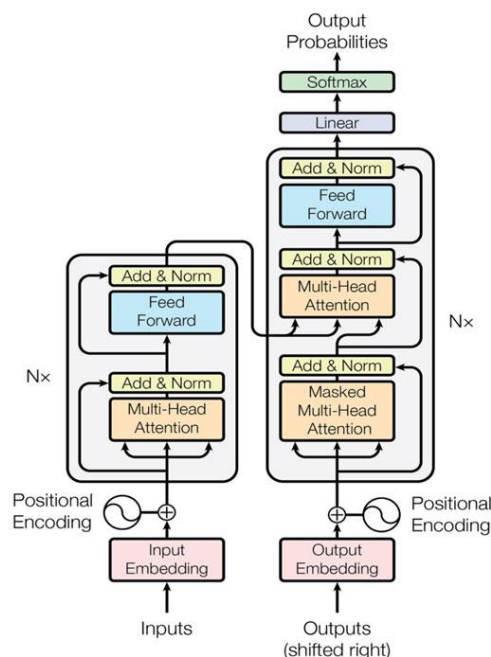
Objetivos del proyecto

El proyecto pretende desarrollar un programa que permita la traducción de una llamada telefónica entre 2 personas centrándose principalmente en el apartado de la traducción.

Este objetivo surge de la curiosidad de aprender sobre los actuales algoritmos de traducción y las posibilidades que pueden llegar a ofrecer, además de averiguar qué tan eficaces son y poder llegar a realizar implementaciones comprendiendo dichos algoritmos.

La idea del proyecto será realizar un algoritmo de traducción llamado Transformers con la finalidad de obtener una traducción entre 2 idiomas mediante una entrada de texto y obteniendo la traducción en texto. También existe la posibilidad de ampliar el proyecto añadiendo a futuro un reconocimiento y síntesis de voz para llegar a aplicarlo a una situación real.

La decisión de realizar un algoritmo transformer viene dada por el hecho de que es uno de los algoritmos más utilizados en la resolución de este tipo de problemas en la actualidad, desde la publicación del paper “attention is all you need”.



El problema constará principalmente de lograr entender el algoritmo Transformers, aprender a aplicarlo a problemas de traducción de texto, generar dicho programa y entrenarlo para la tarea descrita pudiendo medir los resultados obtenidos para lograr ver su viabilidad.

Se dejarán como añadidos adicionales la posibilidad de cambiar la entrada o salida de datos del algoritmo para poder cambiarlos de datos en texto a una grabación de voz para, de ser así, contar con todas las bases para realizar una implementación en una llamada telefónica.

State of the art

Hasta antes de la publicación del paper dando a conocer el algoritmo de Transformers, los algoritmos más utilizados eran los de RNN (Recurrent Neural Network).

Estos, a diferencia de los Transformers, tenían varias fallas entre la que nos encontramos, principalmente, la “pérdida de memoria”; esta falla consistía en la incapacidad del algoritmo de poder lograr acordarse de toda la información de una frase a medida que avanzaba, dando así situaciones en las que, al solo centrarnos en un punto concreto de la frase a estudiar, no lográbamos encontrar relaciones entre palabras muy lejanas entre sí.

Además, el hecho de que el algoritmo tuviese que leer una frase de forma secuencial y ordenada, impedía facilitar el uso de estos algoritmos de forma paralela y requerían de un gran uso de memoria. Todas estas dificultades fueron mejoradas y aliviadas con el uso de Transformers.

Estos algoritmos pretendían dar un valor de atención a las distintas palabras de una frase y relacionarlas con todas las demás. De esta forma pretendía analizar la frase al completo, encontrando relaciones entre todas las palabras y no dando prioridad solo a las últimas.

De esta forma permitía la más fácil paralelización del algoritmo y un menor uso de memoria al no tener que memorizar todo el conjunto de palabras de una frase sino los valores que relacionaban a una palabra concreta con las demás.

Actualmente estos algoritmos son usados en tecnologías de traducción como Google translate y otras tecnologías que, al permitir el entrenamiento con grandes bases de datos permitían realizar acciones como las de GPT-3 de predicción de texto o interacciones mediante texto que no pertenecen a un solo ámbito concreto como terminar frases o crear texto nuevo para finalidades concretas.

Metodología a seguir

Para este proyecto pretendo realizar uso de una metodología agile que me permita iterar sobre el proyecto, permitiendo así el ir desarrollando el algoritmo e ir mejorándolo a medida que obtengo resultados y voy testeando las iteraciones para asegurar un correcto flujo del proyecto.

Considero que esta metodología es la correcta a seguir pues me permitiría obtener resultados desde un punto bastante inicial, así como obtener métricas y datos que comparar con futuras iteraciones para lograr ver un avance en el proyecto.

Además, con esta metodología pretendo centrarme en diferentes puntos del proyecto a medida que se desarrolla para ver cómo afectan al estado del programa y aprender cómo se relacionan las partes entre sí, sin tener que tener un proyecto finalizado al completo.

Además de seguir la metodología agile para la creación del proyecto, se irán realizando una documentación del avance del proyecto para permitir documentar los cambios que surjan, así como las nuevas adiciones o los conocimientos obtenidos de las actuales.

Planificación

Este trabajo constará de diferentes puntos de entrega a lo largo de su transcurso, por eso he decidido emplear una planificación semanal que pretende lograr un objetivo final en los distintos puntos de entrega.

La planificación, aunque no necesariamente rigurosa, pretende obtener resultados cada semana y poder reevaluarse a medida que avance el proyecto si se considera necesario.

Esta sería la siguiente:

Objetivo	Semana/fecha
<ul style="list-style-type: none">·Crear un documento con la información preliminar del problema·Consultar distintas fuentes que puedan ser de ayuda en la resolución del trabajo.·Detallar la propuesta del TFG y los objetivos a conseguir.·Elegir una metodología a seguir.·Crear la planificación del proyecto.·Redactar el informe inicial.	1 (6-marzo)
<ul style="list-style-type: none">·Crear una plataforma de GitHub para guardar toda la información del desarrollo del proyecto.·Leer el paper "Attention is all you need"·Buscar diferentes ejemplos de modelos de transformar a ejecutar·Comparar los distintos ejemplos	2 (13-marzo)
<ul style="list-style-type: none">·Decidir una implementación del algoritmo a seguir·Ver cómo afectan los parámetros al rendimiento y resultados del programa·Buscar una plataforma donde correr el código·Probar la ejecución del código en la plataforma elegida·Entender las partes que conformar el transformar	3 (20-marzo)
<ul style="list-style-type: none">·Comprender las distintas partes del código a ejecutar·Ver como se relaciona el código con el modelo de Transformers del paper "attention is all you need"·Empezar a analizar el código y su funcionamiento·Empezar a comprender los distintos parámetros del modelo·Buscar como analizar y cuantificar los resultados del modelo	4 (27-marzo)
<ul style="list-style-type: none">·Realizar una primera implementación estable del código·Realizar una investigación sobre los data sets a utilizar en el entrenamiento del algoritmo.·Informarse de cómo organizar los datos de entrada y como obtener mejores resultados con estos.·Calcular los recursos necesarios para la ejecución final del programa·Calcular un aproximado del tiempo de ejecución del entrenamiento en la plataforma elegida.·Mirar de tener la implementación del código funcional en la plataforma elegida.	5 (3-abril)
<ul style="list-style-type: none">·Realización y entrega del informe Progreso (I)·Evaluación del estado actual del proyecto·Ver hasta donde se ha llegado·Considerar cambios a realizar	6 (10-abril)

·Avanzar y mejorar en la implementación en código ·Comprender las distintas partes del código por separado ·Compararlas con otras implementaciones del algoritmo de Transformers ·Documentar el código	7 (17-abril)
·Terminar la documentación ·Realizar pruebas de funcionamiento y análisis ·Mirar de obtener resultados de una “ejecución final”	8 (24-abril)
·Obtener métricas para cuantificar los resultados obtenidos ·Comparar resultados con otros proyectos ·Evaluar la posibilidad de implementar un modelo síntesis y reconocimiento de voz	9 (1-mayo)
·Documentar la información del proyecto ·Tener un modelo terminado y entrenado ·Obtener resultados del entrenamiento	10 (8-mayo)
·Crear documentación final ·Tener esquema del informe Progreso(II)	11 (15-mayo)
·Entrega del informe Progreso (II)	12 (22-mayo)
Realizar informe final	13 (28-mayo)
Realizar informe final	14 (5-junio)
Entrega informe final	15 (12-junio)
Realizar presentación	16 (19-junio)
Entrega presentación	17 (26-junio)

Bibliografía

- (1) LANGUAGE TRANSLATION WITH NN.TRANSFORMER AND TORCHTEXT
https://pytorch.org/tutorials/beginner/translation_transformer.html
- (2) torchtext repository
<https://github.com/pytorch/text>
- (3) The Annotated Transformer
<https://nlp.seas.harvard.edu/2018/04/03/attention.html#positional-encoding>
- (4) SpaCy english dataset
<https://spacy.io/models/en>
- (5) Attention is all you need paper
<https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
- (6) Pytorch Transformers for Machine Translation
<https://www.youtube.com/watch?v=M6adRGJe5cQ&list=WL&index=37&t=364s>
- (7) How Google Translate Works - The Machine Learning Algorithm Explained!
<https://www.youtube.com/watch?v=AlpXjFwVdIE&list=WL&index=35&t=164s>
- (8) Las REDES NEURONALES ahora prestan ATENCIÓN! - TRANSFORMERS ¿Cómo funcionan?
<https://www.youtube.com/watch?v=aL-EmKuB078&list=WL&index=31>
- (9) how speech recognition works in under 4 minutes.
<https://www.youtube.com/watch?v=iNbOOgXjnzE&list=WL&index=30>
- (10) How Does Speech Recognition Work? Learn about Speech to Text, Voice Recognition and Speech Synthesis
<https://www.youtube.com/watch?v=6altVgTOf9s&list=WL&index=29&t=313s>
- (11) A Basic Introduction to Speech Recognition (Hidden Markov Model & Neural Networks)
https://www.youtube.com/watch?v=U0XtE4_QLXI&list=WL&index=28
- (12) Generative Model-Based Text-to-Speech Synthesis
<https://www.youtube.com/watch?v=nsrSrYtKkT8&list=WL&index=26&t=322s>
- (13) State of the Art in Natural Language Processing (NLP)
https://www.youtube.com/watch?v=uz_eYqutEG4&list=WL&index=16
- (14) Transformers, explained: Understand the model behind GPT, BERT, and T5
<https://www.youtube.com/watch?v=SZorAJ4I-sA&list=WL&index=15>