XLIV Jornadas de Automática 2023

https://jautomatica.es

Predicción de gestos no-verbales usando aprendizaje profundo

Fernández-Rodicio, E.[a,*], Dondrup, C.[b], Sevilla-Salcedo, J.[a], Castro-González, A.[a], Malfaz, M.[a], Salichs, M.A.[a]

[a]*Robotics Lab, Universidad Carlos III de Madrid, Av. de la Universidad 30, Leganés, Madrid 28911, Spain; enrifern@ing.uc3m.es, javier.sevilla@uc3m.es, acgonzal@ing.uc3m.es, mmalfaz@ing.uc3m.es, salichs@ing.uc3m.es.*

[b]*The Interaction Lab, School of Mathematical and Computer Sciences, Heriot-Watt University, Campus The Avenue, Edinburgh EH14 4AS, Scotland - United Kingdom; c.dondrup@hw.ac.uk*

To cite this article: Fernández-Rodicio, E., Dondrup, C., Sevilla-Salcedo, J., Castro-González, A., Malfaz, M., Salichs, M.A. 2023. Non-verbal gesture prediction using deep learning. XLIV Jornadas de Automática, 587-592. https://doi.org/10.17979/spudc.9788497498609.587

Resumen

En años recientes, la robótica está empezando a usarse fuera de aplicaciones industriales, y los robots empiezan ya a tomar parte en tareas que requieren interactuar con personas. Para que estas interacciones resulten naturales, es necesario que el robot sea capaz de ejecutar expresiones de forma autónoma. En situaciones donde el robot está hablando, los gestos no verbales que ejecute deben apoyar el mensaje comunicativo de la componente verbal, y ambas componentes deben estar sincronizadas apropiadamente. En este trabajo presentamos un sistema de predicción de gestos no verbales para robots sociales basado en uno de los avances más significativos en años recientes en el campo del aprendizaje profundo: el modelo transformer. Esta solución será comparada con un modelo previo que combina redes recurrentes con campos aleatorios condicionales para resolver la misma tarea. Los resultados de la comparación de ambos modelos indican que, al igual que en otras tareas de procesamiento del lenguaje natural, los transformers presentan una clara mejora a la hora de resolver la tarea de predecir gestos no verbales para robots sociales.

Palabras clave: Predicción de gestos, Aprendizaje automático, Aprendizaje profundo, Gestión de expresividad, Interacción multimodal, Robótica social.

Non-verbal gesture prediction using deep learning

Abstract

In recent years, robotics is starting to expand beyond industrial applications, and robots are starting to take part in tasks that require interacting with human beings. For this interactions to be natural for the users, it is necessary that the robots are capable of performing expressions autonomously. In situations where the robot is speaking, the non-verbal gestures performed by the robot must also support the communicative message expressed by the verbal component, and both components should be properly synchronized. In this work, we present a gesture prediction system for social robots based in one of the most significant advances in the area of deep learning: the transformer model. This solution will be compared with a previous system based on a combination of recurrent neural networks and conditional random fields. The results of the comparison conducted show that, as it is the case for other tasks in the field of natural language processing, transformers present a clear improvement for the task of predicting non-verbal expressions for social robots.

Keywords: Gesture prediction, Machine learning, Deep Learning, Expressiveness Management, Multi-modal interaction, Social Robotics.

[*]Autor para correspondencia: enrifern@ing.uc3m.es

1. Introducción

Gracias a los avances tecnológicos que están teniendo lugar en el campo de la robótica en los últimos años, el uso de robots en tareas que requieren interactuar con personas está empezando a ser una realidad. Para que esta interacción sea natural, es necesario que las personas perciban al robot como un compañero de interacción apropiado (Powers et al., 2014). Una forma de conseguirlo es hacerlo parecer un ente animado (Bartneck et al., 2009), lo que se puede conseguir dándole la capacidad de moverse de forma autónoma (Gelman and Spelke, 1981) y realizar expresiones[1]. Este problema se vuelve más complejo cuando el robot tiene que llevar la iniciativa durante una interacción verbal, ya que en esta situación las expresiones deben apoyar (o al menos no interferir con) el mensaje contenido en el habla del robot. Por lo tanto, es necesario encontrar una forma de emparejar de forma adecuada las componentes verbal y no verbal de la comunicación.

Por lo general, los trabajos que buscan solucionar este problema pueden agruparse en dos categorías generales. La mayoría optan por generar la expresividad del robot de forma dinámica, lo que la dota de una gran variabilidad sin necesidad de crear los gestos manualmente. Por ejemplo, Kucherenko et al. (2019) propusieron usar un autoencoder de eliminación de ruido para generar movimientos en base a características MFCC, espectrogramas y características prosódicas del habla del robot. Yoon et al. (2019) usaron una arquitectura encoder-decoder con un mecanismo de atención para generar movimientos para robots humanoides. Ginosar et al. (2019) presentaron un trabajo donde usaban una red neuronal convolucional entrenada para generar gestos con los brazos y las manos que encajen con el estilo de gesticulación de una persona concreta. Un discriminador determina si el gesto encaja o no con el estilo buscado. Otros ejemplos de trabajos que usan arquitecturas adversariales son los trabajos de Ahuja et al. (2020), donde las expresiones se adaptan a factores como el estilo de gesticulación, la postura o la posición corporal, o de Yoon et al. (2020), donde se adaptan a la identidad del usuario. Liang et al. (2022) propusieron separar la información semántica y rítmica, y decodificar gestos para cada una de estas componentes por separado. Chang et al. (2022), extendieron la arquitectura Tacotron2 con un mecanismo de atención con una restricción de localidad buscando que aprenda el alineamiento entre gestos y frases en base a características locales. Yazdian et al. (2022) propusieron tratar el problema como una tarea de traducción, donde el mensaje verbal y los gestos no verbales son los idiomas de origen y destino, respectivamente. Para ello, usaron un autoencoder variacional con cuantificación vectorial combinado con técnicas de aprendizaje por representación.

La segunda solución (la cual sigue este trabajo) consiste en seleccionar los gestos más apropiados de entre una librería de expresiones predefinidas. Esto permite tener un control más directo sobre como el robot comunica un mensaje, lo cual es una ventaja para robots no antropomórficos o con capacidades de expresividad limitadas. En esta linea, Chiu et al. (2015) presentaron un sistema llamado Deep Conditional Neural Field, que permite el aprendizaje conjunto de una red neuronal profunda y la contingencia temporal de cadenas de segundo orden. Su sistema predice el tipo de gesto que debería acompañar las palabras en la frase. Pérez-Mayos et al. (2019) propusieron una combinación de tres estrategias: (i) conectar palabras clave con gestos simbólicos; (ii) conectar picos en el tono de voz con gestos rítmicos; y (iii) combinar ambas soluciones, sincronizando primero gestos simbólicos con palabras clabes y conectando después gestos rítmicos al resto de la frase. Kucherenko et al. (2021) presentaron el diseño de una nueva solución que, a pesar de generar expresiones desde cero, usa una red neuronal convolucional para predecir ciertas características de estos gestos (por ejemplo, su tipo o su fase).

Como se puede ver, el uso de modelos de aprendizaje automático para predecir los gestos que un robot debería ejecutar es una solución bastante extendida. Una de las áreas del aprendizaje automático que ha experimentado un mayor crecimiento en los últimos años es el *procesamiento del lenguaje natural* (NLP). Este crecimiento se debe en parte a la aparición de los modelos *transformer*, una arquitectura de aprendizaje profundo basada en el concepto de *autoatención*, esto es, conseguir que el modelo dé más importancia a ciertas partes de la entrada que a otras. Aunque han existido soluciones previas que usaban mecanismos de atención, los transformers muestran un mejor rendimiento que estos al centrarse únicamente en el mecanismo de autoatención (Vaswani et al., 2017).

Este trabajo busca evaluar la efectividad de los modelos transformer cuando se aplican a la tarea de predecir de forma automática los gestos que un robot social debería realizar en base al mensaje verbal que tiene que comunicar. Para ello, representaremos este problema como una tarea de clasificación de tokens, donde se divide el mensaje verbal del robot en tokens que son individualmente etiquetados con un tipo de gesto. En trabajos previos (Fernández-Rodicio et al., 2023) hemos presentado una arquitectura que combina redes neuronales recurrentes con campos aleatorios para generar esta secuencia de etiquetas. En este trabajo buscamos comparar el rendimiento de este modelo con el que nos pueden proporcionar algunos de los modelos transformer mas extendidos, cuando se usan en un robot real durante una interacción con una persona. El resto del manuscrito se estructurará de la siguiente manera. La Sección 2 describe los dos métodos de predicción de gestos que pretendemos comparar: la combinación de redes recurrentes y campos aleatorios condicionales y los modelos transformer. El resultado de esta comparativa se presenta en la Sección 3. Por último, en la Sección 4 presentamos las conclusiones que hemos extraído de este trabajo.

2. Predicción de gestos

En esta sección se presentan las dos soluciones al problema de predicción de gestos que queremos comparar en este

[1] En este trabajo usaremos de forma equivalente los términos "gesto" y "expresión" para referirnos a una combinación de acciones multimodales (no limitadas a movimientos corporales) que buscan conseguir un objetivo comunicativo concreto

trabajo: la arquitectura basada en campos aleatorios condicionales (CRF) (Fernández-Rodicio et al., 2023), y una nueva arquitectura basada en transformers pre-entrenados para la tarea de clasificación de tokens, y ajustados para predicción de gestos. Mientras que en el primer caso desarrollamos el modelo y lo entrenamos desde cero, en el segundo caso hemos optado por tomar algunos de los modelos más utilizados en el área de NLP. En ambos casos utilizamos el mismo dataset. Cada instancia contiene un mensaje verbal (una o múltiples frases que forman un párrafo coherente), la lista de tokens en los que este texto se va a dividir (donde cada token contiene una palabra o sub-palabra, su raíz, y su función sintáctica), y la secuencia de etiquetas que deberían asignarse a cada token. Las frases para el dataset se sacaron del corpus de diálogos de películas de la universidad de Cornell (Danescu-Niculescu-Mizil and Lee, 2011), y se tokenizaron usando la librería SpaCy[2]. Para obtener la lista completa de tipos de gestos, un anotador observó cada uno de las expresiones que nuestro robot puede usar, y anotó el mensaje comunicativo que se percibió en el gesto. A continuación se agruparon todas aquellas expresiones que comunicaban un mensaje similar, y se le asignó una etiqueta a cada grupo. Esto resultó en un lista de 21 tipos de gestos, donde cada uno puede estar conectado con una o mas expresiones (por ejemplo, se pueden tener múltiples gestos para mostrar gratitud). Estas etiquetas son: “greet”, “yes”, “no”, “other_peer”, “explain”, “self”, “question”, “neutral”, “front”, “emphatic”, “please”, “sorry”, “calm_down”, “but”, “third_person”, “come_on”, “thanks”, “iterate”, “enthusiastic” y “thinking”. Por ejemplo, “other_peer” hace referencia a gestos para señalar al usuario de forma “no agresiva” (similar a señalar con el dorso de la mano), mientras que “come_on” hace referencia a gestos para animar o incitar a la persona a hacer algo. El dataset completo contiene 2600 instancias, y se ha dividido en tres particiones: entrenamiento (60 %), validación (20 %), y test (20 %), asegurando una distribución proporcional de etiquetas. Esto quiere decir que el 60 % de las instancias donde aparece una determinada etiqueta estarán en la partición de entrenamiento.

2.1. *Método basado en Campos Aleatorios Condicionales*

La primera de las soluciones que vamos a comparar en este trabajo combina redes LSTM (Long-Short Term Memory) para codificar las entradas al modelo, y CRFs para generar la secuencia de etiquetas. Gracias al uso de redes recurrentes como encoders, el modelo es capaz de aprender las dependencias que existen entre todos los elementos de la secuencia de tokens que representa el mensaje verbal del robot. Por otra parte, los CRFs son capaces de aprender las dependencias que existen entre las etiquetas de la secuencia de salida, en base a la entrada que reciben. Como se mencionó en Fernández-Rodicio et al. (2023), optamos por dividir la tarea de predicción en dos fases: (i) predecir la intención comunicativa del mensaje verbal; y (ii) predecir los tipos de gestos a sincronizar, teniendo en cuenta la intención identificada. Dado que en nuestro análisis asignamos a cada frase una única intención, esto hace que predecir las intenciones sea más sencillo, y nos proporciona información extra que podemos usar para seleccionar gestos de forma apropiada. La lista de intenciones comunicativas se extrajo a partir de un análisis de las frases utilizadas en el dataset, buscando obtener un número de intenciones que haga que esta información sea al mismo tiempo útil para la predicción de los gestos y fácil de predecir.

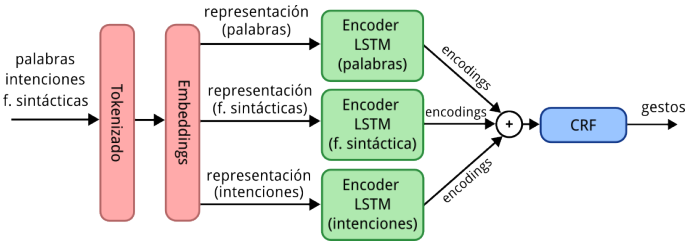

Figura 1: Arquitectura del modelo de predicción basado en RNN y CRF. En este caso se muestra el modelo de predicción de gestos. El modelo de predicción de intenciones sigue el mismo diseño, pero elimina el encoder de intenciones comunicativas.

Tabla 1: Configuración usada para el modelo basado en CRFs

Tamaño de embeddings para el modelo base	200
Tamaño de embeddings para Elmo	1024
Capa de entrada de los encoders	1224
Capa oculta de los encoders	200
Capas recurrentes en los encoders	2
Dropout	0.5
Iteraciones	19
Tasa de aprendizaje	0.001
Paciencia	25
Gradient norm	5.0
Tamaño de batch del iterador	2

El funcionamiento del sistema se divide en tres pasos: (i) pre-procesar el texto de entrada; (ii) predecir la intención comunicativa para los tokens obtenidos en el paso anterior; y (iii) predecir la secuencia de gestos en base a los tokens del paso 1 y las intenciones obtenidas en el paso 2. Los modelos de predicción para los pasos 2 y 3 han sido desarrollados usando la librería AllenNLP (Gardner et al., 2018), y siguen la misma estructura (la cual se puede ver en la Figura 1). Las entradas al modelo se convierten a representaciones densas en un espacio vectorial pasándolas a través de una capa de tokenizado y otra de embeddings. A continuación, cada tipo de entrada (palabras, categorías gramaticales y, en el caso del modelo de predicción de gestos, intenciones comunicativas) se pasa por un encoder independiente, modelado como una red LSTM bidireccional. Las salidas de los encoders se concatenan, y se pasan a través del modelo CRF para obtener la secuencia de etiquetas deseada. Ambos modelos se entrenaron de forma separada, usando la configuración que se muestra en la Tabla 1.

2.2. *Método basado en Transformers*

La nueva solución que queremos comparar con nuestro modelo original consiste en ajustar modelos transformer preentrenados para la tarea de clasificación de tokens usando el

[2]https://spacy.io/

mismo dataset desarrollado para el sistema basado en CRFs. Uno de los modelos transformer que más atención ha atraído en tareas de NLP es BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018). Este modelo consiste en un encoder bidireccional multi-capa que puede recibir como entrada un único texto, o una pareja de textos (entendiendo texto como una secuencia de frases consecutivas que forman un párrafo coherente). La principal ventaja de BERT es que es capaz de considerar el contexto a ambos lados del fragmento de texto que se esté procesando, en vez de limitarse al contexto que precede o sigue a dicho fragmento. Esto se consiguió utilizando dos objetivos de entrenamiento de forma simultánea: (i) predecir palabras ocultas tras máscaras en el texto de entrada; y (ii) predecir si dos frases de entrada aparecen seguidas la una de la otra o no en el texto original. Desde su aparición, BERT ha pasado a ser el modelo de referencia en múltiples trabajos en NLP, lo cual nos ha llevado a seleccionarlo como uno de los modelos a ajustar para la tarea de predicción de gestos.

Los otros dos modelos que hemos seleccionado son variaciones de BERT: DistilBERT y RoBERTa. Al seleccionar un modelo más ligero (DistilBERT) y uno más pesado (RoBERTa), podemos evaluar la relación entre tamaño y rendimiento. DistilBERT (Sanh et al., 2019) se desarrolló buscando reducir el tamaño de la versión base de BERT, sin variar su arquitectura. Para ello, se usó lo que se conoce como *knowledge distillation*, un proceso a través del cual se le enseña a un modelo compacto a reproducir el comportamiento de uno o varios modelos de mayor tamaño. Gracias a esto, se redujo el tamaño de BERT en un 40 % y se incrementó su velocidad en un 60 %. Por otro lado, RoBERTa (Liu et al., 2019) es un modelo que surgió tras descubrirse que la versión base de BERT había sido significativamente sub-entrenada. Para corregir esto, los autores propusieron un nuevo paradigma de entrenamiento que propone aumentar el número de información, tamaño de batch, y número de épocas de entrenamiento, eliminar el objetivo de entrenamiento de predecir si dos frases son consecutivas o no y usar frases más largas y un método dinámico para ocultar las palabras en la frase durante el entrenamiento.

Las tres versiones de estos modelos se han obtenido a través de HuggingFace (BERT base[3], DilstilBERT [4] y RoBERTa [5]). Una de las ventajas de utilizar transformers es que son capaces de trabajar directamente con el mensaje verbal del robot, sin tener que dividirlo en palabras antes, ni asignarles la información gramatical. Para el proceso de ajuste fino, fue necesario aplicar dos cambios al dataset. En primer lugar, las etiquetas se representan como cadenas de caracteres en el dataset, mientras que los modelos esperan que sean números enteros, por lo que es necesario generar un mapeado entre ambas representaciones. En segundo lugar, los modelos elegidos tokenizan el texto de entrada de forma interna para luego asignarles etiquetas. Sin embargo, estos tokens no necesariamente coincidirán con los que definimos para esa instancia del dataset, lo que lleva a un desajuste entre los tokens generados por el modelo y la lista de etiquetas correctas. Por ejemplo, si el tokenizer de BERT divide una palabra en dos tokens, pero nuestro dataset la considera como uno solo (y por lo tanto le asigna una única etiqueta de gesto), debemos asegurarnos de que a estos dos nuevos tokens se les asigna la misma etiqueta que tiene el token del dataset. La librería transformers de Huggingface que hemos usado para ajustar los modelos proporciona un método para resolver este problema. Para el ajuste usamos los mismos hiperparámetros para los tres modelos (una tasa de aprendizaje de $2 * 10^{-5}$, un decaimiento de los pesos de 0,01. y un tamaño de batch de 16). Se ajustaron los modelos durante 10 épocas de entrenamiento, y se tomó la versión para la época donde la pérdida de validación es más baja (época 3-4 en casi todos los casos). Tanto los datasets que hemos desarrollado como los modelos ya ajustados están disponibles en HuggingFace[6]

3. Evaluación

Para establecer una comparativa entre la solución basada en CRF y la nueva solución basada en transformers se tendrán en cuenta tres factores: (i) las métricas de entrenamiento; (ii) el uso de recursos (GPU y memoria) requerido; y (iii) el tiempo de inferencia. Durante el entrenamiento de los modelos usaremos las métricas de clasificación multietiqueta proporcionadas por la librería Scikit-learn (Pedregosa et al., 2011). Esta implementación permite la evaluación de coincidencias parciales en una secuencia de etiquetas (indica como de correcta es una predicción donde algunas etiquetas son correctas y otras no, en vez de considerar la predicción completamente errónea). El análisis del uso de recursos y del tiempo de inferencia busca asegurar que el modelo sea capaz de adaptarse al ritmo de las interacciones entre personas y de trabajar junto al resto de módulos de la arquitectura software del robot.

Figura 2: El robot social Mini.

La evaluación de los modelos se realizó en Mini (Salichs et al., 2020), un robot social de sobremesa diseñado para prestar ayuda a personas mayores con un leve deterioro cognitivo. Este robot se puede ver en la Figura 2. Mini cuenta con una CPU Intel i5-3550, con cuatro núcleos funcionando a 3.3 GHz, y 16 GB de RAM. Dado el alto coste computacional que supone el correr modelos de aprendizaje automático, contamos con

[3]https://huggingface.co/bert-base-cased
[4]https://huggingface.co/distilbert-base-cased
[5]https://huggingface.co/roberta-base
[6]https://huggingface.co/qfrodicio

un servidor preparado específicamente para correr este tipo de modelos. Este servidor cuenta con dos GPUs NVIDIA GeForce RTX 3090, una CPU Intel Core i9-10900K a 3.7 GHz, y 64 GB de RAM. Nuestros robots son capaces de establecer una comunicación con el servidor a través de sockets, enviando la información que se quiere pasar a los modelos (en nuestro caso el mensaje verbal que tiene que decir el robot), y recibiendo la salida generada por el modelo. La integración del sistema de predicción y sincronización de gestos en la arquitectura software de Mini se puede ver en Fernández-Rodicio et al. (2023).

Tabla 2: Resultados de las métricas obtenidas para los diferentes modelos evaluados en este trabajo. Se ha resaltado el F-score más alto.

Modelo	Precisión	Recall	F-score
CRF	0.7943	0.7761	0.7708
BERT	0.7348	0.714	0.7059
DistilBERT	0.7188	0.6975	0.6903
RoBERTa	0.7805	0.7816	**0.771**

El rendimiento de los diferentes modelos evaluados se puede observar en la Tabla 2. Para la solución basada en CRFs se evaluó el rendimiento del módulo entero (la combinación de los modelos de predicción de intenciones y de gestos). Los resultados muestran una clara diferencia entre el rendimiento que muestran el modelo basado en CRFs y RoBERTa, y el que muestran tanto BERT como DistilBERT. Entre los dos primeros, no existe mucha diferencia: RoBERTa presenta un recall más alto (0.7816 y 0.7761, respectivamente), pero una menor precisión (0.7805 y 0.7943, respectivamente), resultando en un f-score similar (0.771 y 0.7708, respectivamente). Por otra parte, podemos ver que BERT y DistilBERT presentan un rendimiento muy similar (f-scores de 0.7059 y 0.6903, respectivamente). Esto es ligeramente sorprendente, ya que la diferencia en el tamaño de los modelos (DistilBERT presenta una reducción del 40 % en el número de parámetros) podría llevar a pensar que DistilBERT presentaría un rendimiento claramente inferior.

A la hora de analizar los resultados para los tiempos de inferencia, decidimos considerar dos umbrales para evaluar la usabilidad de los diferentes modelos. La *regla de los 2 segundos* (Miller, 1968) considera que un retraso mayor de 2 segundos puede causar que el mensaje pierda parte de su significado, mientras que autores como Shiwa et al. (2008) proponen que el tiempo de respuesta más apropiado es de 1 segundo. Idealmente, buscamos que nuestro modelo sea capaz de generar inferencias en menos de 1 segundo. Sin embargo, consideramos que retrasos mayores podrían ser aceptables siempre que no superen los 2 segundos, ya que la posible pérdida de significado del mensaje afectaría negativamente a la interacción. Para realizar esta evaluación, tomamos las instancias de la partición de test de nuestro dataset (423 instancias), enviamos las frases desde el robot al servidor una a una, generamos inferencias para ellas, y devolvimos los resultados al robot. Las frases utilizadas tenían una longitud media de 43.41 caracteres (con una desviación estándar de 29.36), con longitudes que van de 7 a 229 caracteres. Para esta prueba, el robot se conectó a internet por wifi, mientras que el servidor usó una conexión ethernet. Una vez se generaron inferencias para todas las frases, se calcularon 4 medidas: (i) el tiempo medio que el modelo corriendo en el servidor necesita para generar una secuencia de etiquetas (nos referimos a él como tiempo de inferencia); (ii) el tiempo medio que pasa desde que el robot envía una frase al servidor hasta que recibe una respuesta (nos referimos a él como tiempo total); (iii) el porcentaje de uso de la GPU del servidor (entendido como el porcentaje de tiempo durante el cual al menos uno de los núcleos de la GPU está realizando operaciones); y (iv) el porcentaje de memoria de GPU utilizado. Los resultados de esta evaluación se pueden ver en la Tabla 3.

Tabla 3: Tiempos de inferencia y consumo de recursos para cada uno de los modelos evaluados. El número en cada celda representa el valor medio, más menos la desviación estándar.

	CRF	BERT	DistilBERT	RoBERTa
inferencia (s)	0,1026 ± 0,052	0,0175 ± 0,017	0,0132 ± 0,015	0,0178 ± 0,018
total (s)	0,1567 ± 0,068	0,1208 ± 0,158	0,1272 ± 0,184	0,1114 ± 0,115
GPU (%)	23,1 ± 9,116	1,6237 ± 0,751	0,7332 ± 0,421	1,7177 ± 0,61
Memoria (%)	9,1094 ± 0,171	3,1734 ± 0,004	2,4407 ± 0,004	3,4247 ± 0,004

Si comparamos el tiempo de inferencia y tiempo total para los cuatro modelos, podemos observar que todos ellos satisfacen los dos umbrales que habíamos definido (los tiempos están todos por debajo de 1 segundo). La solución basada en transformers mostró mejores resultados, ya que presentan un tiempo de inferencia 10 veces menor que el obtenido con la solución basada en CRFs. Si comparamos los resultados de los modelos basados en transformers entre si, observamos una relación inversa entre su tamaño y el tiempo de inferencia medio. DistilBERT resultó ser el más rápido (tiempo de inferencia de 0.0132 segundos), seguido de BERT (0.0175 segundos) y RoBERTa (0.0178 segundos). Sin embargo, las desviaciones estándar de estas medidas sugieren que las diferencias no son significativas. En cualquier caso, las pruebas demostraron que el tiempo de inferencia es casi despreciable frente al retardo introducido por la comunicación robot-servidor (por ejemplo, para BERT el tiempo crece de 0.0175 a 0.1208 segundos). Es importante destacar que la desviación para estos resultados es bastante alta debido a la presencia de algunas medidas donde el tiempo de comunicación se disparó (llegando incluso a superar el límite de 2 segundos). Esto se puede deber a la existencia de problemas de conexión puntuales. Aunque estos problemas solo se dieron en unos pocos ejemplos, es algo que se debe tener en cuenta. Por último, si nos centramos en el consumo de recursos, vemos que los resultados de nuevo encajan con nuestras expectativas, ya que el consumo se incrementó con el tamaño de los modelos. DistilBERT mostró los mejores resultados (0.73 % de consumo de GPU y 2.44 % de consumo de memoria), seguido por BERT (1.62 % y 3.17 %, respectivamente), RoBERTa (1.72 % y 3.42 %) y, por último, el modelo basado en CRFs (23.1 % y 9.11 %). De nuevo, vemos que la solución basada en CRFs requiere una cantidad significativamente más alta de recursos que el resto. Hay que tener en cuenta que esta solución implica utilizar dos modelos, el de predicción de intenciones y el de predicción de gestos. Esto duplica las necesidades computacionales. En cualquier caso, los resultados de la evaluación muestran que, aunque los cuatro modelos evaluados mostraron un rendimiento que los hace viables para su uso en interacciones reales, RoBERTa resultó ser el modelo que mejor rendimiento mostró

en conjunto, ya que es capaz de completar la tarea con un nivel de éxito similar al del modelo basado en CRFs (0.771 y 0.7708, respectivamente), pero con un tiempo de inferencia y consumo de recursos menor.

4. Conclusiones

En este trabajo hemos evaluado la usabilidad de los transformers a la hora de predecir qué expresiones deberían acompañar a los mensajes verbales de un robot social. Debido a su gran popularidad y al rendimiento mostrado en otras tareas, decidimos optar por tres variaciones del modelo BERT. El rendimiento de estos modelos se comparó con una solución previa, basada en una combinación de redes recurrentes y campos aleatorios condicionales. Los resultados muestran que, mientras uno de los modelos evaluado (RoBERTa) mejora los resultados presentados por el modelo basado en CRFs, los otros dos no consiguen alcanzar el mismo nivel de rendimiento. Por otra parte, todos los modelos transformer evaluados probaron ser más eficientes en cuanto a tiempo de inferencia y uso de recursos, aunque es viable utilizar las cuatro opciones en interacciones reales, siempre y cuando se tenga un hardware capaz de correr los modelos. A pesar de los buenos resultados, este trabajo presenta una limitación principal: la falta de un estudio subjetivo que evalúe como los usuarios perciben la combinación del habla y los gestos no verbales seleccionados por nuestros modelos. Esto nos daría una mejor idea de la importancia que tiene integrar este tipo de modelos en robots sociales. Esta limitación se solucionará en futuros trabajos

Agradecimientos

La investigación que ha conducido a estos resultados ha recibido financiación de los proyectos: Robots sociales para mitigar la soledad y el aislamiento en mayores (SOROLI), PID2021-123941OA-I00, financiado por la Agencia Estatal de Investigación (AEI), Ministerio de Ciencia e Innovación; y Robots sociales para reducir la brecha digital de las personas mayores (SoRoGap), TED2021-132079B-I00, financiado por la financiado por la Agencia Estatal de Investigación (AEI), Ministerio de Ciencia e Innovación. Este trabajo ha recibido el apoyo del Gobierno de la Comunidad de Madrid, bajo el Acuerdo Multianual con la UC3M ("Fostering Young Doctors Research", SMM4HRI-CM-UC3M), en el contexto del V PRICIT (Programa Regional de Investigación Científica e Innovación Tecnológica).

Referencias

Ahuja, C., Lee, D. W., Nakano, Y. I., Morency, L.-P., 2020. Style transfer for co-speech gesture animation: A multi-speaker conditional-mixture approach. In: European Conference on Computer Vision. Springer, pp. 248–265.

Bartneck, C., Kanda, T., Mubin, O., Mahmud, A., 04 2009. Does the design of a robot influence its animacy and perceived intelligence? International Journal of Social Robotics 1, 195–204.

Chang, C.-J., Zhang, S., Kapadia, M., 2022. The ivi lab entry to the genea challenge 2022–a tacotron2 based method for co-speech gesture generation with locality-constraint attention mechanism. In: Proceedings of the 2022 International Conference on Multimodal Interaction. pp. 784–789.

Chiu, C.-C., Morency, L.-P., Marsella, S., 2015. Predicting co-verbal gestures: a deep and temporal modeling approach. In: International Conference on Intelligent Virtual Agents. Springer, pp. 152–166.

Danescu-Niculescu-Mizil, C., Lee, L., 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In: Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Fernández-Rodicio, E., Castro-González, Á., Castillo, J. C., Salichs, M. Á., Onorati, T., 2023. Predicción de gestos no verbales para mejorar la interacción con un robot social. Actas de las XIII Jornadas Nacionales de Robótica y el XIV Simposio CEA de Bioingeniería.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, M., Zettlemoyer, L., 2018. Allennlp: A deep semantic natural language processing platform. arXiv preprint arXiv:1803.07640.

Gelman, R., Spelke, E., 1981. 2 the development of thoughts about animate and inanimate objects: implications for. Social cognitive development: Frontiers and possible futures 1, 43.

Ginosar, S., Bar, A., Kohavi, G., Chan, C., Owens, A., Malik, J., 2019. Learning individual styles of conversational gesture. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3497–3506.

Kucherenko, T., Hasegawa, D., Henter, G. E., Kaneko, N., Kjellström, H., 2019. Analyzing input and output representations for speech-driven gesture generation. In: Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents. pp. 97–104.

Kucherenko, T., Nagy, R., Jonell, P., Neff, M., Kjellström, H., Henter, G. E., 2021. Speech2properties2gestures: Gesture-property prediction as a tool for generating representational gestures from speech. In: Proceedings of the 21st ACM International Conference on Intelligent Virtual Agents. pp. 145–147.

Liang, Y., Feng, Q., Zhu, L., Hu, L., Pan, P., Yang, Y., 2022. Seeg: Semantic energized co-speech gesture generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10473–10482.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Miller, R. B., 1968. Response time in man-computer conversational transactions. In: Proceedings of the December 9-11, 1968, fall joint computer conference, part I. pp. 267–277.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Pérez-Mayos, L., Farrús, M., Adell, J., 2019. Part-of-speech and prosody-based approaches for robot speech and gesture synchronization. Journal of Intelligent & Robotic Systems, 1–11.

Powers, K. E., Worsham, A. L., Freeman, J. B., Wheatley, T., Heatherton, T. F., 2014. Social connection modulates perceptions of animacy. Psychological science 25 (10), 1943–1948.

Salichs, M. A., Castro-González, Á., Salichs, E., Fernández-Rodicio, E., Maroto-Gómez, M., Gamboa-Montero, J. J., Marques-Villarroya, S., Castillo, J. C., Alonso-Martín, F., Malfaz, M., 2020. Mini: a new social robot for the elderly. International Journal of Social Robotics 12, 1231–1249.

Sanh, V., Debut, L., Chaumond, J., Wolf, T., 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Shiwa, T., Kanda, T., Imai, M., Ishiguro, H., Hagita, N., 2008. How quickly should communication robots respond? In: 2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, pp. 153–160.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems 30.

Yazdian, P. J., Chen, M., Lim, A., 2022. Gesture2vec: Clustering gestures using representation learning methods for co-speech gesture generation. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 3100–3107.

Yoon, Y., Cha, B., Lee, J.-H., Jang, M., Lee, J., Kim, J., Lee, G., 2020. Speech gesture generation from the trimodal context of text, audio, and speaker identity. ACM Transactions on Graphics (TOG) 39 (6), 1–16.

Yoon, Y., Ko, W.-R., Jang, M., Lee, J., Kim, J., Lee, G., 2019. Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots. In: 2019 International Conference on Robotics and Automation (ICRA). IEEE, pp. 4303–4309.