PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

ANÁLISIS DE LA SINTAXIS APRENDIDA POR BETO, UN MODELO DE LENGUAJE EN ESPAÑOL BASADO EN TRANSFORMERS

ALEJANDRO QUIÑONES

Tesis para optar al grado de
Magíster en Ciencias de la Ingeniería

Profesor Supervisor:
ÁLVARO SOTO

Santiago de Chile, Mayo 2021

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

ANÁLISIS DE LA SINTAXIS APRENDIDA POR BETO, UN MODELO DE LENGUAJE EN ESPAÑOL BASADO EN TRANSFORMERS

ALEJANDRO QUIÑONES

Miembros del Comité:

ÁLVARO SOTO

DENIS PARRA

JORGE PEREZ

FRANCO PEDRESCHI

Tesis para optar al grado de
Magíster en Ciencias de la Ingeniería

Santiago de Chile, Mayo 2021

A mi familia y amigos

RECONOCIMIENTOS

Quiero agradecer a mi profesor guía, Álvaro Soto, por inspirar mi interés en la inteligencia artificial y orientarme durante mi investigación, a los miembros del IA Lab por su ayuda y compañerismo, a mi familia por su amor y respaldo, a María José por la amistad, a Camila por acompañarme en el proceso, y al café por su apoyo constante.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ABSTRACT

Progress in the interpretability and comprehension of models like BERT have helped greatly in the development of more secure and comprehensible tools. Nevertheless, many of the explanations regarding the model's behaviour have been centered around its linguistic capabilities, thus making its results only applicable to models based in English.

In this thesis, we study the syntactic capabilities of BETO, a Spanish version of BERT, developing the comprehension over the model. We show that BETO is capable of recognizing syntax, even more so than BERT, through specific heads of the model. We then study the degree, limitations and structure of this knowledge. We find the model's activations display patterns that are similar when similar relations are processed. Also, the main cause for the model to fail to perceive a syntactic relation is that the relation is presented in an unusual order. Furthermore, we indicate that the model has part of the syntactic context that it failed to recognize, suggesting a disagreement in the syntax tree's structure with respect the original annotations. Additionally, the lack of BETO's syntactic knowledge could result in a lower performance during question-answering. Finally, we show that heads of the model with low syntactic knowledge achieve high syntax recognition when they work together, pointing towards some distributed knowledge in the model.

Keywords: BERT, BETO, syntax, NLP, XAI.

RESUMEN

Avances en la interpretabilidad y comprensión de modelos como BERT han sido de utilidad para el desarrollo de mejores herramientas, más seguras y comprensibles. Sin embargo, muchas explicaciones del funcionamiento del modelo son en base a capacidades lingüísticas aprendidas, significando que los resultados son solo aplicables para los modelos basados en el inglés.

En esta tesis se estudian las capacidades sintácticas de BETO, la versión de BERT en español, desarrollando la comprensión del modelo. Se muestra que BETO posee capacidades sintácticas, incluso mayores que las de BERT, presentes en distintas *heads* del modelo. Además, se realizan estudios con respecto a las competencias, limitaciones y estructura de este conocimiento. Se encuentra que las activaciones del modelo se producen en patrones similares cuando se procesan relaciones parecidas. Se indica que la principal causa para que el modelo falle en reconocer relaciones sintácticas es cuando éstas se estructuran de manera poco común. Se muestra que el modelo posee parte del contexto sintáctico que falla en reconocer, sugiriendo un desacuerdo en la formación del árbol sintáctico con respecto a las anotaciones originales. También, la falta de conocimiento sintáctico del modelo podría significar una reducción en su rendimiento al evaluarlo en responder preguntas. Por último, se demuestra que *heads* con bajo conocimiento sintáctico logran un alto reconocimiento de la sintaxis cuando trabajan en conjunto, indicando la presencia de un conocimiento distribuido.

Keywords: BERT, BETO, sintaxis, NLP, XAI.

1. INTRODUCCIÓN

1.1. Contexto

El área de procesamiento de lenguaje natural (NLP) es ahora más importante que nunca. Permite el procesamiento artificial de ideas complejas, y tiene utilidad en una gran cantidad de fuentes de información, desde redes sociales a artículos y libros.

En esta categoría de estudio destacan los modelos de lenguaje pre-entrenados, por su buen rendimiento. Estos son modelos que fueron entrenados previamente en una gran cantidad de datos, para procesar un texto y transformarlo a un *embedding* (o representación vectorial) que tenga el contexto interiorizado. Así, estos *embeddings* pueden ser utilizados posteriormente en distintas tareas. Los más conocidos son los *Transformer Networks* BERT (Devlin, Chang, Lee, & Toutanova, 2018) y GPT (1 (Radford, Narasimhan, Salimans, & Suutskever, 2018)), 2 (Radford et al., 2019) y 3 (Brown et al., 2020)). Ambos modelos usan distintos bloques del modelo *Transformer* (modelo basado en el concepto de la atención), pero varían en su funcionalidad. En particular, BERT tiene la capacidad de agregar contexto a cada palabra del *input* a partir de todo el *input*, a diferencia de los modelos GPT, donde solo se les permite observar la parte del *input* anterior a la procesada.

BERT entregó resultados extraordinarios; Al momento de su publicación, logró obtener rendimientos estado-del-arte en 11 distintas tareas de NLP, como SQuAD (v1.1 (Rajpurkar et al., 2016) y v2.0 (Rajpurkar, Jia, & Liang, 2018)) y MNLI (Williams, Nangia, & Bowman, 2018).

Actualmente, BERT es usado en el proceso de Búsqueda de Google (Nayak, 2019), mejorando los resultados en una de cada diez búsquedas gracias a la inclusión del contexto, como se ve en la Figura 1.1.

Una limitación de BERT es que su pre-entrenamiento es exclusivamente con texto en inglés, siendo incapaz de comprender cualquier *input* en otro idioma. Para solucionar esto, los autores de BERT presentaron mBERT: una versión de BERT pre-entrenado con

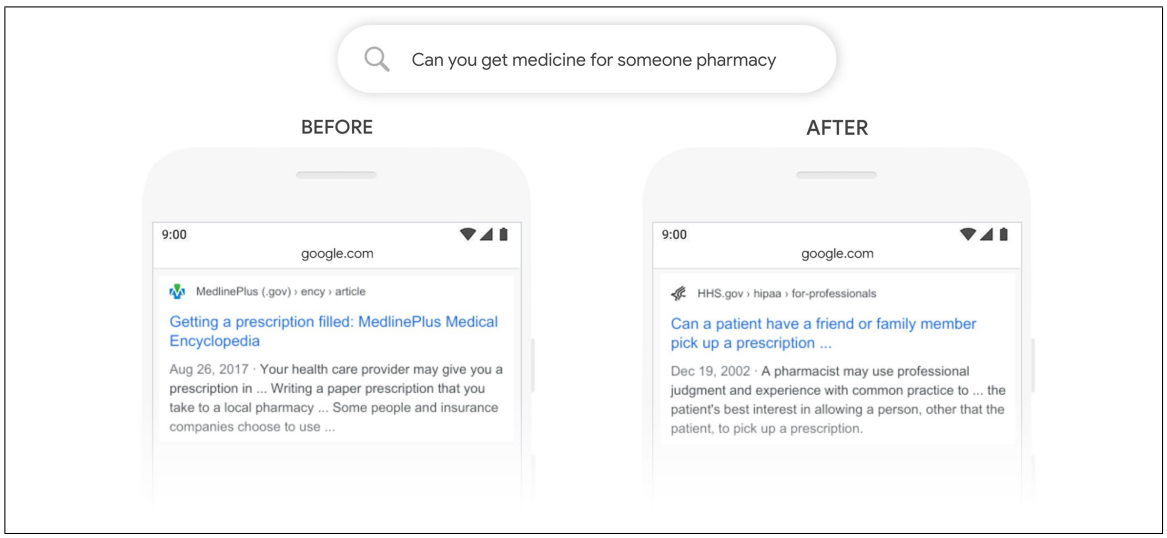


Figura 1.1. Comparación de resultados de Búsquedas en Google antes y después de agregar a BERT. La inclusión del contexto en la búsqueda permite la comprensión de *"for someone"*, donde previamente esta parte era ignorada. Fuente: Nayak, 2019.

la concatenación de Wikipedia en 104 lenguajes (Devlin et al., 2018). Sin embargo, han surgido versiones del modelo pre-entrenados en distintos lenguajes en específico, como por ejemplo camemBERT (Martin et al., 2020) para el francés, BERTje (de Vries et al., 2019) para el holandés y BETO (Cañete et al., 2020) para el español. Estos modelos naturalmente suelen tener mejor rendimiento que mBERT en múltiples tareas.

Por otro lado, otra área de estudio destacada en la inteligencia artificial es la explicabilidad de estos modelos, que tiene una gran importancia por múltiples razones. Se sabe que inherente a la data de pre-entrenamiento de BERT se encuentra un sesgo de género (Bhardwaj, Majumder, & Poria, 2020) que el modelo lamentablemente hereda. Además, organizaciones como la Unión Europea demandan que se pueda dar una explicación a las decisiones que nos afectan a los humanos, por nuestro *"derecho a una explicación"* (Parliament and Council of the European Union, 2016), independiente de si la decisión la tomó un humano o una máquina. Ambos casos sugieren que darle interpretabilidad a modelos como BERT es un trabajo necesario de realizar.

Distintos estudios intentan ayudar en la interpretabilidad del modelo. Hoover et al., 2019 presentan ExBERT: una herramienta de visualización para el funcionamiento de BERT, mientras que Hao et al., 2020 proponen un algoritmo para explicar la interacción de la información procesada. Por último, también se han realizado estudios en las capacidades lingüísticas de BERT. Mientras que Ettinger, 2019 muestra que BERT tiene algún grado de conocimiento semántico, Clark et al., 2019 demuestran que ciertas partes del modelo son capaces de reconocer relaciones sintácticas. Analizar las capacidades lingüísticas permiten el desarrollo de la comprensión del modelo, la que puede ser utilizada para construir nuevas estrategias de interpretabilidad.

1.2. Intención de la investigación

Avances en la interpretabilidad de modelos como BERT permite el desarrollo de herramientas en NLP con menos falencias y consecuencias negativas. Tener un entendimiento de los procesos internos de estos modelos permite comprender las razones detrás de sus errores y desarrollar soluciones a estos. Sin embargo, estudios lingüísticos como los de sintaxis limitan la comprensión desarrollada al modelo entrenado en inglés, y sus variaciones en otros idiomas no reciben los mismos avances.

En esta tesis se realizará una investigación sobre las capacidades sintácticas de BETO. Se aplicarán metodologías presentes en la literatura para determinar el reconocimiento del modelo sobre la sintaxis, y se comparará con BERT. Además, se profundizará en el estudio, viendo las competencias de las habilidades, junto a sus limitaciones y la estructura de su formación.

1.3. Estructura de la tesis

En el Capítulo 2 se entregará una revisión de la literatura necesaria para el avance de la investigación. Se presentará el *dataset* que se usará a través de la investigación, la arquitectura y el entrenamiento del modelo, el estudio sintáctico en el que se basa la tesis, y

una crítica y modificación a éste para su mejor funcionamiento. Se terminará resumiendo cómo se usarán estos avances para el desarrollo de este trabajo.

En el Capítulo 3 se realizará un estudio de la sintaxis de BETO, basado en las herramientas postuladas por Clark et al., 2019. Con éste se comparará las capacidades de BETO con las de BERT, y se definirá el concepto de Mapas de Conocimiento.

El Capítulo 4 presentará un estudio sobre el comportamiento completo del modelo. En base a los Mapas de Conocimiento, se investigará si los patrones que se presentan en los Mapas son semejantes entre Mapas de relaciones similares, y se verá a qué factores se deben que ocurran estas similitudes.

Seguido, en el Capítulo 5 se revisará una serie de factores que podrían afectar el reconocimiento de relaciones sintácticas, y se evaluará cuáles causan que la predicción falle y cómo estos varían según la relación que está siendo analizada.

El Capítulo 6 se enfoca en qué se predice cuando una relación no es reconocida correctamente. Se analiza si BETO tiene tendencias a predecir lo mismo para distintas relaciones, y se estudia si el modelo aun posee la capacidad de comprender una frase incluso cuando no logra descifrar sus dependencias sintácticas.

El último experimento se presentará en el Capítulo 7. Éste determinará si el modelo es capaz de combinar las habilidades sintácticas de un subconjunto de sus partes para generar un reconocimiento superior al utilizado, dando origen al conocimiento distribuido.

Para terminar, se procederá a entregar una conclusión en el Capítulo 8. Se verá como los resultados de los distintos experimentos ayudan a formar una idea del conocimiento sintáctico presente en el modelo, qué medidas se pueden tomar para superar las limitaciones encontradas, y cuáles son los trabajos planteados a futuro.

2. REVISÓN DE LA LITERATURA

Antes de presentar los estudios realizados, es necesario describir su contexto. A continuación, se presentarán distintos conceptos que serán claves para la comprensión de los experimentos. Primero se describirá el *dataset* que será utilizado, junto al significado en sus anotaciones sintácticas. Seguido, se mostrará en detalle la estructura interna de BERT. Por último, se mostrará un estudio de la sintaxis de BERT, además de una crítica al estudio que presenta una modificación útil para nuestro análisis.

2.1. GSD-Spanish y relaciones sintácticas

GSD-Spanish es un *treebank* de texto en español proveniente de noticias, blogs, críticas y Wikipedia, que incluye las etiquetas de Universal Dependencies (McDonald et al., 2013). Universal Dependencies (UD) es un *framework* de anotaciones gramáticas consistentes universalmente a través de distintos idiomas basado en las dependencias de Stanford (de Marneffe & Manning, 2008). Esto incluye etiquetas de *Part-of-Speech*, características morfológicas y relaciones sintácticas, en 104 lenguajes diferentes.

El *treebank* incluye 431,587 palabras en español utilizadas en 16,013 frases, con sus anotaciones correspondientes. Estas son previamente divididas en *sets* de entrenamiento (14,187 frases), testeo (426 frases) y desarrollo (1,400 frases). Hay 33 relaciones sintácticas presentes, todas siendo etiquetadas manualmente. Sin embargo, solo 26 de éstas aparecen consistentemente (más de 100 veces). Una lista de estas 26 relaciones sintácticas se encuentra en el Apéndice A.

Las relaciones sintácticas generan un árbol a partir de las palabras de alguna frase. Las distintas formas en que las palabras se conectan dan origen a las distintas relaciones encontradas. UD les atribuye un nombre universal a cada relación. Una ilustración de este árbol con los nombres de las relaciones está dada en la Figura 2.1. En una relación sintáctica, se le llama al nodo padre la cabeza de la relación, y al nodo hijo la dependencia.

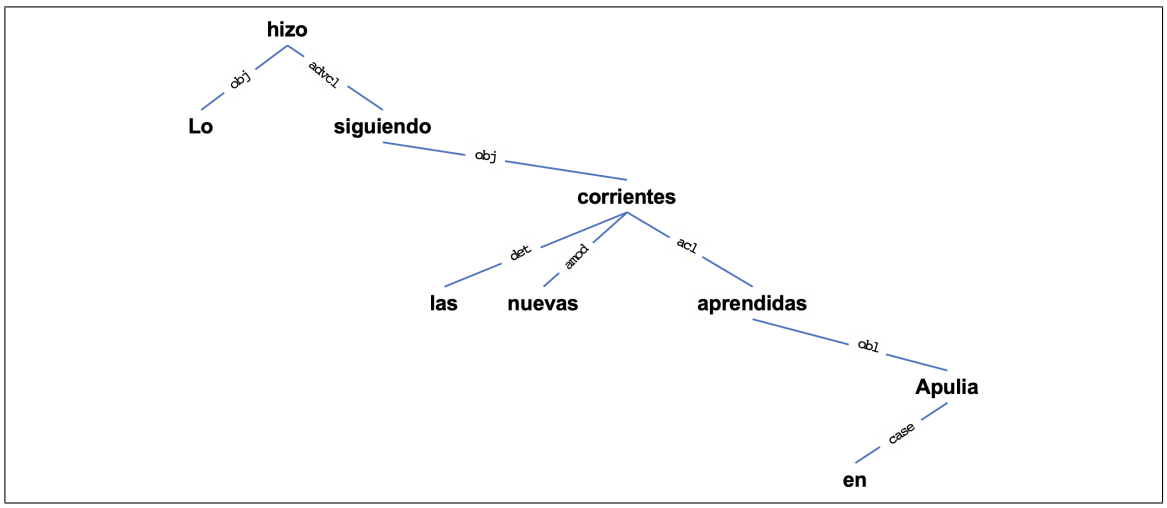


Figura 2.1. Ejemplo de un árbol sintáctico. Se presenta el árbol generado por la frase "Lo hizo siguiendo las nuevas corrientes aprendidas en Apulia". Se observa que una palabra puede tener muchos nodos hijo, pero solo puede tener un nodo padre. Fuente: Elaboración propia.

2.2. Transformers y Atención

El *Transformer* (Vaswani et al., 2017) es un modelo de transducción de secuencias usado para la traducción de texto. En el momento de su publicación, el estado-del-arte en su categoría consistía de redes neuronales recurrentes o convolucionales que utilizaban el mecanismo de atención (Bahdanau, Cho, & Bengio, 2016). El *Transformer* por otro lado funciona únicamente en base a los que se llaman módulos de *self-attention*, sin el uso de recurrencias ni convoluciones. El modelo obtuvo resultados superiores al estado del arte, mientras que fue entrenado durante una fracción del tiempo que se entrenaron los otros modelos.

Inicialmente, el modelo recibe como entrada una frase. Ésta es separada en distintos *tokens*. Normalmente un *token* representa una palabra, pero palabras complejas y largas suelen ser separadas en dos o más. Estos *tokens* son luego representados como vectores a través de un *embedding*. A este *embedding* se le suma un *encoding* posicional, que le agrega información sobre la posición de cada *token*. Finalmente estos vectores son pasados al módulo de *self-attention*.

Antes de explicar el módulo de *self-attention*, corresponde explicar el módulo de atención. En este, existe un vector $y_i \in \mathbb{R}^d$ que se busca actualizar en base otros n vectores $X = \{x_1, x_2, ..., x_n\} \subset \mathbb{R}^d$. Para realizar esto, se definen 3 funciones distintas: *Query* $q(\cdot)$, *Key* $k(\cdot)$ y *Value* $v(\cdot)$:

$$q(y) := yW^Q + b^Q, \text{ con } W^Q \in \mathbb{R}^{d \times d'}, b^Q \in \mathbb{R}^{d'}$$
$$k(x) := xW^K + b^K, \text{ con } W^K \in \mathbb{R}^{d \times d'}, b^K \in \mathbb{R}^{d'}$$
$$v(x) := xW^V + b^V, \text{ con } W^V \in \mathbb{R}^{d \times d'}, b^V \in \mathbb{R}^{d'}$$

Donde los valores de los W y b son aprendidos. Estas funciones buscan hacer un símil con el proceso de hacer una consulta (*query*) a un *set* de datos ordenados por sus llaves (*keys*), donde se devuelven los valores (*values*) de los datos que responden la consulta.

Luego, se obtienen los *Keys* y *Values* de los vectores en X, además del *Query* de y_i. Ahora, el módulo de atención decide cuáles de los vectores en X son los de mayor importancia. Idealmente, se busca que el vector del *query* solo sea similar a los *keys* de los $x_j \in X$ importantes, para poder realizar el producto punto entre ambos vectores y obtener resultados similares a la importancia. Después de obtener los producto puntos, éstos se regularizan dividiéndose por $\sqrt{d'}$ y se pasan por una función *softmax*, función que usa exponenciales para simular la función *max* pero seguir siendo diferenciable (característica necesaria para las redes neuronales). La regularización ocurre para que la función *softmax* se comporte de manera razonable.

$$\alpha_{i,j} := \underset{x_j \in X}{\textit{softmax}} \left(\frac{q(y_i)k(x_j)^\top}{\sqrt{d'}} \right) \in \mathbb{R} \tag{2.1}$$

Por último, se usan estos valores, llamados los coeficientes de atención, para destacar los *Values* de los x_j importantes, que se suman y pasan por una última transformación O aprendida. Este último paso se muestra en la Ecuación 2.2

$$y'_i = \left(\sum_{j=1}^{n} \alpha_{i,j} v(x_j) \right) W^O + b^O, \text{ con } W^O \in \mathbb{R}^{d' \times d}, b^O \in \mathbb{R}^d \tag{2.2}$$

A todo este proceso se le llama módulo de atención. La única diferencia entre éste y los de *self-attention* es que el vector que atiende también es uno de los vectores atendidos. En otras palabras, en los módulos de *self-attention*, $y_i \in X$. Este proceso de *self-attention* ocurre sobre todos los *tokens* del *input*. Es decir, cada representación de un *token* es actualizada a partir de la representación de los otros *tokens*.

Los módulos de *self-attention* del *Transformer* además son *Multi-Head*. Es decir, en vez de realizar el proceso de *self-attention* una vez, obteniendo una representación de dimensionalidad d, el proceso se repite h veces para vectores de dimensionalidad d/h. Finalmente, los h vectores se concatenan y se pasan por la transformación O. Se usan estos submódulos, o *heads*, porque permiten al modelo atender a diferentes partes de la información en diferentes posiciones.

El modelo del *Transformer* trabaja con bloques *Encoder* y *Decoder*. En particular el *Encoder* consiste de un módulo de *Multi-Head self-attention* seguido por una conexión residual y una *layer normalization*. Después simplemente se pasan los estados a una *feed-forward network*, y se repite una conexión residual y otra *layer normalization*. El bloque *Encoder* es el importante en este estudio, ya que en éste se basa la arquitectura de BERT.

2.3. BERT & BETO

La arquitectura de BERT consiste en apilar L bloques de *Transformer Encoders* de A *heads* cada uno. Para procesar los inputs, se utiliza el mismo proceso de tokenización, *embedding* y *encoding* posicional usado en el *Transformer*. Además, se agrega un *embedding* de segmento, que se explicará más adelante. Para el modelo base, se definió que serían $L = 12$ bloques (o *layers*) de $A = 12$ *heads*. El estado oculto de cada *layer* es de tamaño

$H = 768$. Así, después de cada *layer* se obtienen 12 representaciones de cada *token* de dimensión $768/12 = 64$, o equivalentemente, una con su concatenación.

El modelo es pre-entrenado con dos tareas que requieren las capacidades de incorporar contexto en la representación del *input*:

La primera se trata de la predicción de un *token* enmascarado. Se decide enmascarar el 15% del *input* de manera aleatoria, reemplazándolo con un *token* "[MASK]", y que el modelo prediga cuál era el *token* original. Sin embargo, si solo se predicen los *tokens* que fueron enmascarados, no necesariamente se producirían buenas representaciones para los otros *tokens*. Como solución se decide modificar el proceso de enmascarar: del 15% de *tokens* seleccionados, 80% se reemplaza con un [MASK], 10% se reemplaza con un *token* cualquiera, y el 10% restante, el *token* se mantiene como está. Esta tarea suele llamarse *Cloze* (Taylor, 1953), pero en el trabajo de BERT se llama *Masked LM* (MLM).

La segunda tarea se llama *Next Sentence Prediction* (NSP). A BERT se le entregan dos frases. En el 50% de los casos, la segunda frase es la continuación de la primera en el documento original. En el otro 50%, es otra frase aleatoria. Se pide al modelo predecir si la segunda frase es o no la continuación de la primera frase. Para marcar la separación de las frases, se agrega un *token* "[SEP]" al final de cada una. Además, el *embedding* de segmento mencionado anteriormente incluye la información de a qué frase pertenece cada *token*. También se incluye un *token* "[CLS]" al principio del *input*, en donde se busca que el modelo responda con su predicción.

Finalmente se espera que en la representación final de [CLS] se obtenga la predicción de NSP, mientras que la representación de cada otro *token* sería la predicción de su *token* original para la tarea de MLM. La arquitectura de BERT, junto a las salidas de pre-entrenamiento, se muestran en la Figura 2.2

Para BETO, la arquitectura es la misma. La mayor diferencia es que el pre-entrenamiento se realizó con texto en español. En particular, se usó todo el texto en Wikipedia y todas las fuentes del proyecto OPUS (Tiedemann, 2012) que estuvieran en español.

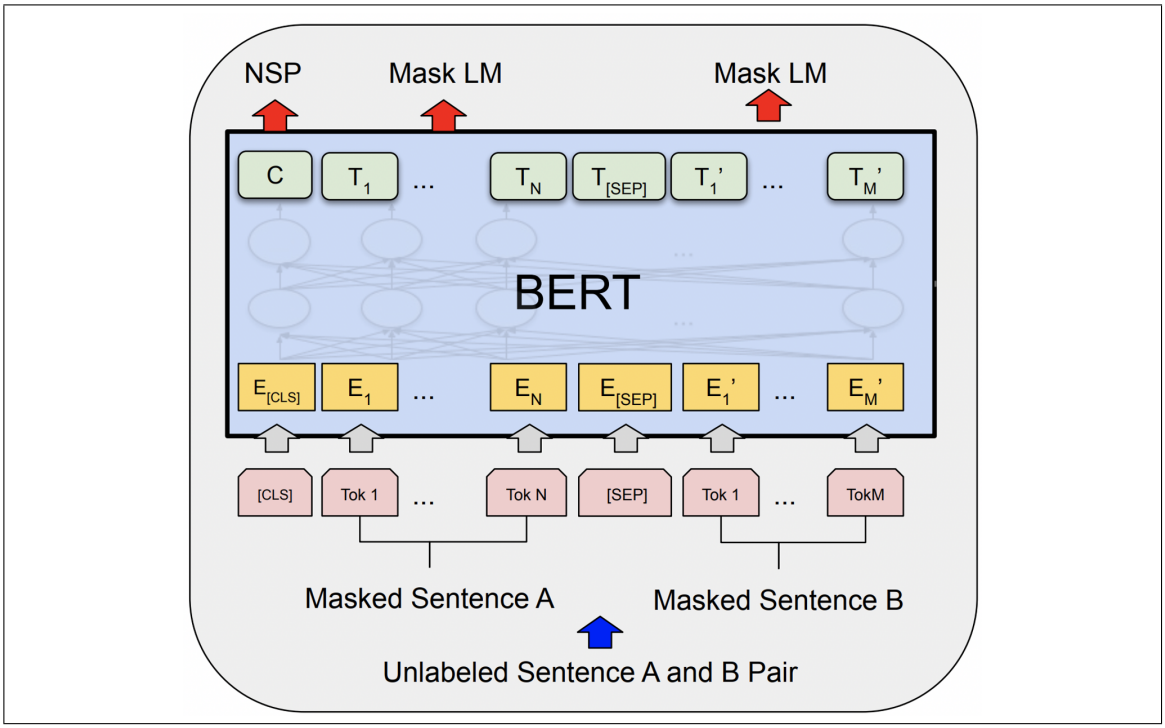


Figura 2.2. Ilustración de la Arquitectura de BERT. Se muestran además las salidas consideradas en el pre-entrenamiento. Fuente: Devlin et al., 2018.

El proceso de pre-entrenamiento también recibió algunas modificaciones. Principalmente, se realizó un *Dynamic Masking*, que modifica el *masking* que recibe un *input* para cada época, además de un *Whole-Word Masking* (WWM) que enmascara además del *token* original, todos los *tokens* de la misma palabra.

BETO resultó tener un rendimiento superior a mBERT (la versión de BERT pre-entrenada en múltiples lenguajes) en la mayoría de las tareas estudiadas y comparables en el resto. La Tabla 2.1 presenta la comparación de rendimiento entre mBERT y BETO *cased*, es decir, BETO usando un *embedding* que diferencia las mayúsculas en las palabras.

Tabla 2.1. Rendimiento de mBERT y BETO *cased* en múltiples tareas de NLP. El "*" indica un nuevo estado-del-arte en la tarea en el momento de su publicación. Los valores de mBERT fueron obtenidos por (a) Wu & Dredze, 2019 y (b) Yang et al., 2019. Los valores de BETO fueron obtenidos por Cañete et al., 2020

Modelo	XNLI	PAWS-X	NER	POS	MLDoc
mBERT	78.50^a	89.00^b	87.38^a	97.10^a	$\mathbf{95.70^a}$
BETO cased	**82.01**	**89.05**	**88.43**	**98.97***	95.60

2.4. SQuAD, MLQA & SQuAD-es

2.4.1. SQUAD

SQuAD (*Stanford Question Answering Dataset*) (Rajpurkar et al., 2016) es un *dataset* de más de 100,000 preguntas en inglés basadas en distintos segmentos de artículos de Wikipedia, donde sus respuestas se encuentran dentro de dicho segmento. La Figura 2.3 presenta un ejemplo de SQuAD, mostrando el segmento, la pregunta y la respuesta presente en el segmento.

Para tener una perspectiva sobre la dificultad de SQuAD, Rajpurkar et al., 2016 implementaron un modelo en base a una regresión lineal con distintas características. Este simple modelo logra un puntaje F1 de 51.0. En contraste, los autores encuentran que el rendimiento humano logra un puntaje F1 de 86.8, en base al acuerdo que tenían los anotadores del *dataset*. Cuando BERT fue recién publicado, logró resultados estado-del-arte en SQuAD, obteniendo un puntaje F1 de 93.2.

2.4.2. MLQA

Problemáticamente, SQuAD solo existe en inglés, significando una limitación para el área de NLP en otros idiomas. Para combatir esto, Lewis et al., 2019 crearon el *dataset*

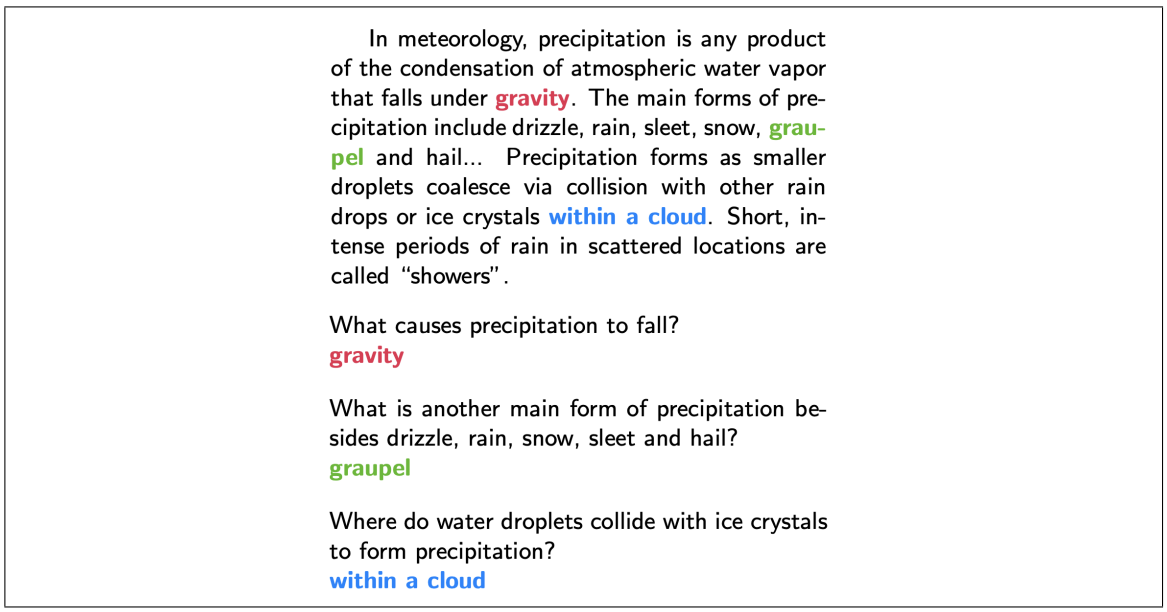


Figura 2.3. Ejemplo de una segmento usado en SQuAD. Se presentan tres preguntas con sus respuestas, y se destaca la presencia de las respuestas en el segmento. Se ve que las preguntas pueden ser de más de una palabra. Fuente: Rajpurkar et al., 2016.

MLQA, que busca evaluar la capacidad de responder preguntas en múltiples lenguas. Consta de más de 5,000 preguntas con el formato de SQuAD en siete lenguajes distintos, entre ellos el español.

Los autores realizan un *fine-tuning* sobre mBERT con el *dataset* de SQuAD de un tamaño mayor, y evaluan directamente en MLQA. Como mBERT fue pre-entrenado con múltiples lenguas, se busca medir su capacidad de transferir lo aprendido durante su *fine-tuning* a otros idiomas. Particularmente en español, mBERT obtiene un puntaje F1 de 64.3 y logra responder exactamente la respuesta correcta (EM) un 46.6% de las veces.

2.4.3. SQuAD-es

La tarea de traducir SQuAD a otras lenguas es compleja, principalmente porque al traducir la pregunta y el segmento, las traducciones suelen tener pequeñas variaciones. Pero como el formato de SQuAD requiere que la respuesta esté dentro del segmento,

esto no es aceptable. Como solución, Carrino et al., 2019 implementaron un método para traducir SQuAD al español, tomando en consideración este requisito. El método *Translate-Align-Retrieve* (TAR) consiste en entrenar un modelo de traducción de texto y un modelo alineamiento de palabras. Con éstos, se procede a traducir los ejemplos de SQuAD y obtener un mapeo de las palabras desde el segmento en inglés al que está en español. Por último se ejecuta un mecanismo de obtención de respuestas en el segmento en base al mapeo obtenido en el paso anterior. Con este método logran traducir cerca del 100% de SQuAD, formando el *dataset* SQuAD-es.

Al tener acceso al nuevo *dataset*, Carrino et al., 2019 realizaron un *fine-tuning* sobre mBERT utilizando SQuAD-es, y lo evaluaron en las preguntas de español de MLQA. El modelo logró un rendimiento estado-del-arte, obteniendo un puntaje F1 de 68.1 y un EM de 48.3%.

2.5. Análisis de la atención de BERT

Clark et al., 2019 proponen una serie de métodos que tienen como finalidad el estudio del uso de la atención en redes de *Transformers* como BERT. En cada *head* de BERT, se generan coeficientes de atención entre todas las representaciones de *tokens*, y en base a estas atenciones se genera una nueva representación para cada *token*. Con esto en mente, los autores se basan en la premisa que los coeficientes de atención indican qué tan importante es cada *token* para la formación de la representación resultante. Con esto desarrollan un análisis sintáctico de la atención y un análisis general.

Durante el estudio sintáctico, se analizan por separado cada *head* del modelo, en búsqueda de encontrar el conocimiento lingüístico que desarrolló. Como el análisis es sobre la sintaxis de las palabras en el *input*, es necesario transformar los coeficientes de atención desde uno a nivel de *tokens* a uno a nivel de palabras. Para hacer esto, cuando se trate de la atención proveniente de una palabra de múltiples *tokens*, se considerará como la atención de la palabra el promedio de las atenciones de todos sus *tokens*. Por otro lado,

cuando una palabra de múltiples *tokens* es atendida, se considera la suma de la atención recibida en sus *tokens* como la atención sobre la palabra. Estas decisiones logran mantener la propiedad que la atención proveniente de cada palabra suma 1, independiente de cuántos *tokens* eran originalmente. Este proceso se describe con mayor formalidad en la Ecuaciones 2.3

$$\alpha_{i,\text{word}_j} = \sum_{t \in \text{word}_j} \alpha_{i,t} \tag{2.3}$$

$$\alpha_{\text{word}_i,\text{word}_j} = \sum_{t \in \text{word}_i} \frac{\alpha_{t,\text{word}_j}}{|\text{word}_i|}$$

En el contexto de coeficientes de atención a nivel de palabras, se dice que dado un *head* h y una palabra p_1, se predice la palabra p_2 si p_1 atiende mayoritariamente a p_2 con los coeficientes de atención de h. Bajo esta definición se busca si algún *head* predice alguna relación sintáctica. Para realizar esto, se utiliza un *dataset* de frases en inglés que contenga sus relaciones sintácticas etiquetadas. En este caso se usa el Penn Treebank (Marcus, Santorini, & Marcinkiewicz, 1993) etiquetadas con las dependencias de Stanford. Luego, para cada *head* se verifica si las predicciones de las palabras corresponden a sus cabezas sintácticas, obteniéndose un porcentaje de reconocimiento para cada relación. Para comparar se desarrolla como *baseline* la predicción del *offset* más común: ciertas relaciones son más simples que otras. Algunas tienen un comportamiento bastante predecible, como por ejemplo casi siempre relacionar dos palabras consecutivas. En este caso, un *offset* de 1 caracterizaría este comportamiento (o -1, dependiendo del sentido de la relación). Así, las capacidades predictivas de las *heads* de BERT se compararán con los *offsets* más común para cada relación, para respaldar los resultados encontrados.

Tabla 2.2. Rendimiento sintáctico de las mejores *heads* de BERT. La notacion L-H indica que se trata de la *head* H del *layer* L. El rendimiento se mide como el porcentaje de veces que la relación fue correctamente predicha por cada *head*.

Relación	*Head*	Rendimiento	*Baseline*
poss	7-6	80.5	47.7
auxpass	4-10	82.5	40.5
ccomp	8-1	48.8	12.4
mark	8-2	50.7	14.5
prt	6-7	99.1	91.4

Ciertas *heads* tienen un rendimiento considerablemente mejor que el *baseline* del *offset* más común. La Tabla 2.2 presenta cinco de estos casos. Cabe destacar que también existen relaciones en donde las predicciones de las mejores *heads* no son tan buenas como en estos ejemplos, y se asemejan al resultado del *baseline*.

Con estos resultados, Clark et al., 2019 concluyen que BERT aprendió hasta cierto nivel un conocimiento sintáctico, lo cual es muy llamativo cuando uno considera que el pre-entrenamiento del modelo no incluye estas etiquetas, por lo que esta capacidad es una consecuencia secundaria del pre-entrenamiento.

Una vez que se estima que algunas *heads* tienen cierto conocimiento sintáctico específico, se busca si el modelo logra expresar conocimiento sintáctico general. Así, se desarrolla un conjunto de *probing classifiers* que intentan predecir el árbol sintáctico completo. Estos clasificadores, también llamados *probes* (o sondas) se basan en distintas fuentes de información relacionadas con el modelo. Éstas se procesan y producen para cada palabra de la frase una distribución de probabilidad sobre las otras palabras, indicando qué tan probable es que esta segunda palabra sea la cabeza sintáctica de la primera.

Si el *probe* logra reconocer las relaciones sintácticas, independiente de cuál es la relación específica reconocida, decimos que se está observando un conocimiento sintáctico general.

En particular, el *Attention-Only Probe* aprende una combinación lineal de los coeficientes de atención de cada *head*, tomando en consideración ambos sentidos de la atención (atender o ser atendido). Con este *probe*, la estimación de la probabilidad de que la palabra i sea la cabeza sintáctica de la palabra j esta dada por:

$$p(i \mid j) \propto \exp\left(\sum_{k=1}^{n} w_k \alpha_{ij}^k + u_k \alpha_{ji}^k\right) \tag{2.4}$$

donde α_{ij}^k es el coeficiente de atención a nivel palabras de la palabra i hacia la palabra j en la *head* k, y los vectores w_k y u_k son aprendidos durante el entrenamiento del *probe*.

Con este *probe*, en un 61% de los casos se logra predecir correctamente qué palabra es la cabeza sintáctica de una dependencia. Estos resultados son satisfactorios y supera los *baselines* planteados en su estudio. El rendimiento resultante sugiere que BERT tiene un conocimiento general de la sintaxis en inglés bastante completo.

Por último, Clark et al., 2019 también realizaron un análisis general de los coeficientes de atención de BERT, ahora a nivel *token*. Se estudia si hay una tendencia a atender a *token* especial, como lo son [CLS], [SEP] y los *tokens* de puntuación (“.” y “,”). La Figura 2.4 muestra la atención dirigida a estos *tokens* en cada una de las *heads* de BERT, ordenadas según el *layer* al que pertenecen.

Se ve que la atención se concentra en el *token* [CLS] en las capas iniciales, [SEP] en las capas del medio, y los *tokens* de puntuación al final. Estas altas atenciones no son intuitivas, al considerar que estos *tokens* no tienen contenido lingüístico. Los autores argumentan que las *heads* del modelo que reconocen características sintácticas atienden a los [SEP] cuando la característica no está presente. Así, atender a [SEP] se usa como una omisión de una operación sintáctica. Sin embargo, Kobayashi et al., 2020 se cuestionan cómo puede ser que atender a [SEP] sea no actuar cuando se trata de una atención tan alta.

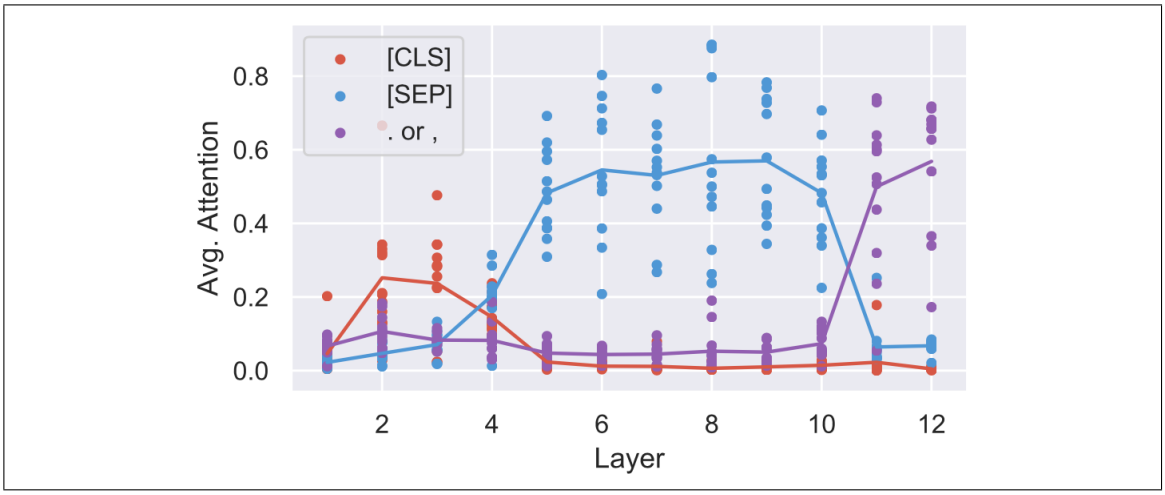


Figura 2.4. Atención a *tokens* especiales en BERT, según el *layer* de la atención. Durante gran parte del modelo, más del 50% de la atención se dirige a estos *tokens*. en Fuente: Clark et al., 2019.

2.6. Otra forma de medir el enfoque de BERT

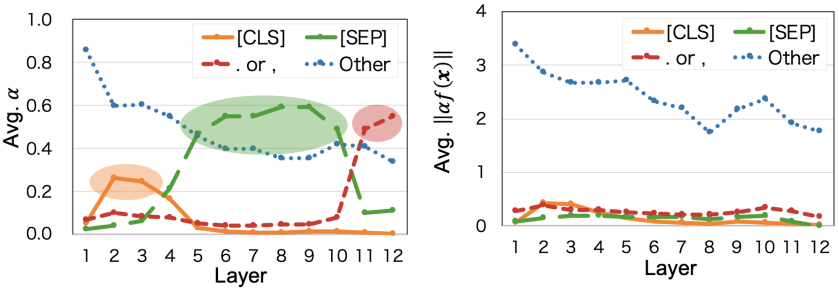
El estudio de Kobayashi et al., 2020 se basa en una crítica al uso de coeficientes de atención como representación de lo importante para el modelo. En la ecuación 2.2 se ve que el vector resultante depende de ambos los coeficientes de atención obtenidos en la ecuación 2.1 y los vectores $v(x_j)$ que representan los *valores* de cada *token*. Es completamente posible que el vector $v(x_j)$ con el mayor coeficiente de atención $\alpha_{i,j}$ tenga una norma muy pequeña y así afecte insignificantemente al valor final de y'_i. Sin embargo, al solo considerar los coeficientes de atención, tendríamos la idea que el *token* t_j es muy importante para la formación de y'_i cuando no lo es. Para combatir este problema, los autores plantean una reformulación a la ecuación 2.2. Dada la linealidad de la multiplicación matricial con W^O, es posible definir una función f

$$f(x) := v(x)W^O$$

Con la que la ecuación 2.2 resultaría equivalente a

$$y'_i = \sum_{j=1}^{n} \alpha_{ij} f(x_j) + b^O$$

Con esto, se plantea analizar la influencia de los *tokens* a partir de la norma euclidiana de $\alpha f(x)$ en vez de solo sus coeficientes de atención. Se realiza el mismo estudio de Clark et al., 2019 con respecto a la importancia de *tokens* especiales, solo que esta vez además se utilizan las $\|\alpha f(x)\|$, y sus resultados se muestran en la Figura 2.5. Con esto, los comportamientos previamente no explicables ahora son intuitivos: aunque los coeficientes de atención a *tokens* especiales sean altos, estos se contraponen con valores $\|f(x)\|$ minúsculos, llegando a un aporte total muy bajo. Así, se obtiene una explicación a la idea de Clark et al., 2019 sobre atender a [SEP] para no actuar. Atender a [SEP] es omitir actuar, porque el efecto sobre y'_i vendría siendo nulo.



(a) Weight-based analysis.

(b) Norm-based analysis.

Figura 2.5. Comparación entre los análisis en base a coeficientes de atención y $\|\alpha f(x)\|$. Previamente los *tokens* especiales tenían la mayoría de la influencia. Con el análisis de $\|\alpha f(x)\|$ se ve que este no es el caso. Fuente: Kobayashi et al., 2020.

2.7. Objetivos de la investigación

Los análisis presentados anteriormente son en su mayoría específicos a BERT y su relación con el idioma del inglés. Así, los estudios del área se quedan fuera de la zona

hispano-hablante, y sus avances no son aplicables en ésta. Por lo tanto, existe una necesidad de comprensión del comportamiento de BETO para desarrollar un modelo más interpretable para el español.

Además, ciertas preguntas con respecto a la sintaxis quedan aun abiertas. Los avances que presentan Kobayashi et al., 2020 sugieren que los coeficientes de atención podrían no ser los mejores representantes del conocimiento sintáctico de BERT y que un estudio en base a $\|\alpha f(x)\|$ podría resultar más fructífero. También, no se logra determinar las causas de las limitaciones sintácticas del modelo, y su investigación podría llevar a prevenir sus problemas. Por último, no queda claro cómo interactúan las *heads* del modelo con respecto a su conocimiento sintáctico. Una descripción más clara de esto podría abrir puertas a una comprensión del comportamiento del modelo más profunda.

En esta tesis se busca desarrollar una comprensión del modelo de BETO por medio del análisis de sus capacidades sintácticas, utilizando los procesos ya existentes para su estudio y resolviendo las dudas que quedan abiertas. El estudio consiste en realizar análisis similares a los desarrollados por Clark et al., 2019, para obtener una descripción base del conocimiento sintáctico del modelo. Los resultados se usarán como punto de partida para luego estudiar si algunos patrones de activación del modelo se repiten para distintas relaciones, cuáles son los factores que afectan al conocimiento sintáctico, la capacidad del modelo de comprender alguna frase cuando este conocimiento no está presente, y si existe alguna interacción entre las *heads* de BETO que indiquen un comportamiento sintáctico más avanzado.

3. ANÁLISIS SINTÁCTICO BASE DE BETO

El primer estudio será en base a la metodología planteada por Clark et al., 2019, pero esta vez en base a los coeficientes de atención entregados por BETO. Estos tendrán la finalidad de determinar si las *heads* de BETO tienen efectivamente un conocimiento sintáctico, y si el modelo tiene la capacidad de reproducir el árbol sintáctico completo de una frase cualquiera. Los resultados darían un respaldo a la investigación general de la tesis, y servirían como base para profundizar el estudio de la sintaxis aprendida.

Además, se realizarán ciertas modificaciones a los métodos planteados en búsqueda de mejores resultados. Ambos estudios sintácticos de Clark et al., 2019 solo toman en consideración los coeficientes de atención para determinar la presencia del conocimiento. En este experimento en cambio, se estudiará el comportamiento del modelo según los coeficientes de atención al igual que los valores de $\|\alpha f(x)\|$ descritos por Kobayashi et al., 2020. Con esto se espera descubrir cuál patrón tiene mayor utilidad para el estudio sintáctico de BETO, y usar éste para los siguientes experimentos.

También, se separarán los resultados encontrados en el *Attention-Only Probe* del modelo. El *probe* indica la capacidad del modelo completo para identificar las cabezas sintácticas de una frase, independientes de cuáles sean éstas. Además de obtener esta capacidad general de BETO, se separarán los resultados según la relación específica que está siendo evaluada. Así, se obtendrá la capacidad del modelo completo para predecir cada relación, a diferencia de solo obtener la capacidad de alguna *head* en particular.

3.1. Metodología

3.1.1. Conocimiento sintáctico en las *heads*

En este experimento se trabajará con el *set* de desarrollo de GSD-Spanish. Para partir, se busca si BETO tiene capacidades sintácticas representadas en los coeficientes de

atención de alguno de sus *heads*. En particular, se busca si la atención de alguna palabra se dirige mayoritariamente a otra palabra y, dado que el *dataset* incluye las anotaciones sintácticas de las frases, distinguir cuando estas dos palabras están relacionadas sintácticamente y bajo qué relación. Los pasos, entonces, son los siguientes:

- Primero, se entregan las frases como *input* a BETO. Este realiza una tokenización sobre cada *input* y luego lo procesa, generando los coeficientes de atención entre todos los *tokens* para todas sus *heads*.
- Segundo, como la sintaxis relaciona palabras, se necesita transformar los coeficientes de atención desde un nivel de *tokens* (*token-token*) a uno de palabras (*word-word*). La transformación fue descrita en la ecuación 2.3, y una ilustración de ésta se encuentra en la Figura 3.1.

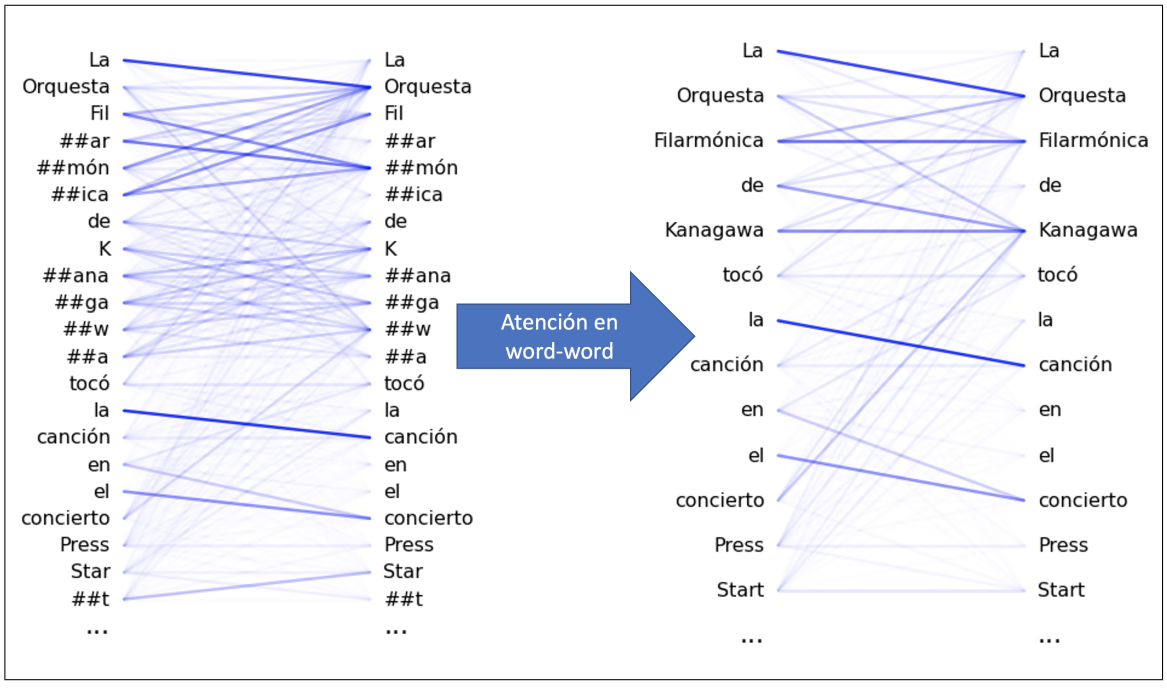


Figura 3.1. Transformación de coeficientes de atención de *token-token* a *word-word*. Se presentan los coeficientes generados por la *head* 5-3. Al trabajar con la atención entre palabras se puede comenzar a reconocer conexiones que podrían tener alguna característica lingüística. Fuente: Elaboración propia.

- Tercero, en base a los coeficientes de atención *word-word*, para cada palabra p_1 encontramos la palabra p_2 a la que más se atiende en cada *head*. Se dice entonces que la "predicción" de la palabra p_1 en el *head* h corresponde a la palabra p_2. Algunos ejemplos de predicciones se presentan en la Figura 3.2.

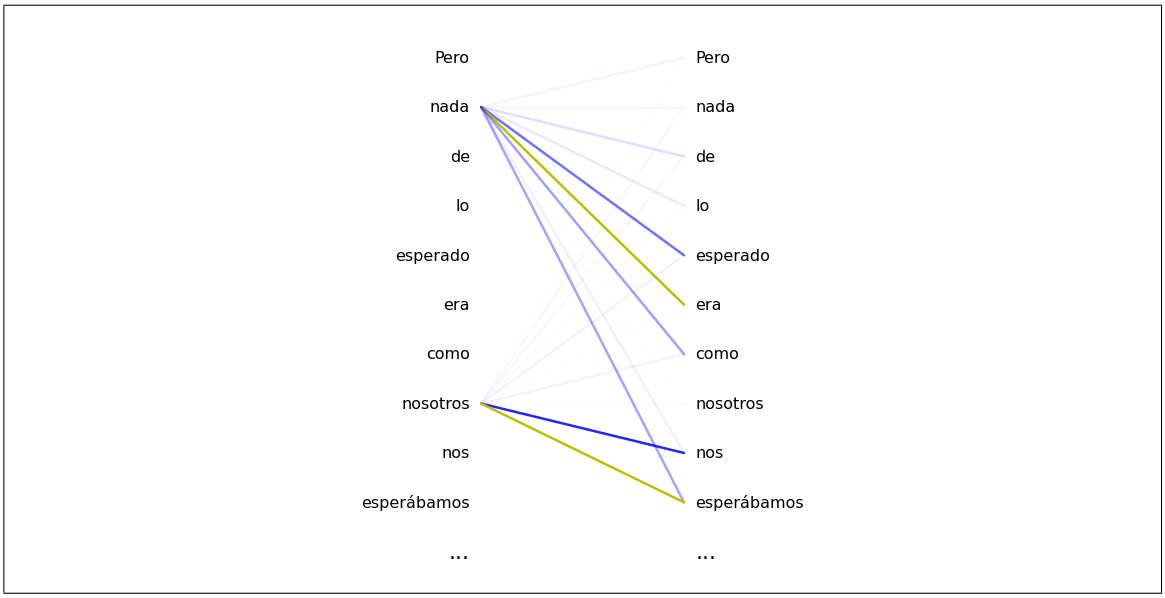


Figura 3.2. Predicción de una palabra según sus coeficientes de atención *word-word*. Un color más fuerte indica un coeficiente mayor. Se presentan los coeficientes de atención de "nada" y "nosotros" en la *head* 6-2, y se destacan las predicciones en amarillo. En este caso, ambas predicciones reconocen correctamente la relación sintáctica *nsubj*. Fuente: Elaboración propia.

- Por último, evaluamos el rendimiento de las *heads* en reconocer cada relación. Dada una relación r, revisamos cada caso en donde r se presente entre dos palabras, y determinamos cuál es la dependencia p_1 y cuál la cabeza sintáctica p_2. Luego, para cada *head* revisamos si la predicción de p_1 corresponde a p_2, obteniendo un porcentaje de reconocimiento de r. No investigamos la relación sintáctica en el otro sentido (es decir, cuando p_2 predice p_1) dado que Clark et al., 2019 reportan un mal rendimiento en esta dirección. A modo de ejemplo, en la Figura 3.2 presentada anteriormente se ve que las predicciones en la *head*

6-2 efectivamente corresponden a la relación sintáctica de *nsubj*, por lo que esta *head* podría tener un alto rendimiento en esta relación.

Ciertas relaciones suelen presentar la misma distancia entre las palabras involucradas. Por ejemplo, en más de un 87% de las apariciones de la relación *det*, la cabeza sintáctica es la palabra que sigue a la dependencia. Así, cuando se ve que una *head* reconoce a una relación con gran probabilidad, se necesita distinguir cuando esto se debe a algún conocimiento sintáctico en la *head*, y cuando simplemente se está prediciendo la palabra a una distancia fija. Con esto en mente, se desarrolla un *baseline* para cada relación en base a su *offset* más común. Esto ayudará a reconocer los comportamiento de las *heads* que sean más complejos que simplemente seguir un *offset*. Por ejemplo, es llamativo que una *head* reconozca la relación *det* solo si su rendimiento es mayor a un 87%.

Después de obtener el rendimiento de las *heads* para cada relación, se repite el experimento, pero se cambia la base de coeficientes de atención y ahora se analiza en base a los valores de $\|\alpha$f(x)$\|$. El proceso de transformación de los valores a un nivel *word-word* y la definición de la predicción son exactamente iguales. Con ambos experimentos realizados, se pueden comparar los resultados y determinar cuál patrón representa de mejor manera las capacidades sintácticas de BETO.

3.1.2. *Attention-Only Probe*

Seguido, se busca determinar si BETO presenta una capacidad sintáctica general en sus coeficientes de atención. Es decir, si a partir de los coeficientes se logra reconocer las relaciones sintácticas, independiente de qué tipo de relación sea. Los resultados se obtienen por medio del *Attention-Only Probe* presentado por Clark et al., 2019.

En este experimento se trabaja con GSD-Spanish. Por cada frase que se le entrega a BETO, éste genera un conjunto de coeficientes de atención por cada una de sus *head*. El *Attention-Only Probe* toma como *input* los coeficientes de atención de todas las *heads*

y aprende una combinación lineal de éstos para realizar una predicción sobre qué palabras están relacionadas sintácticamente. La predicción se compara con las anotaciones presentes en el *dataset*. Como se busca predecir relaciones entre palabras y no *tokens*, se utilizan los coeficientes de atención *word-word* obtenidos de la misma forma que los del análisis anterior. Se entrena el *probe* durante una época sobre el *set* de entrenamiento presente en el *dataset* y se evalúa en su *set* de testeo.

Los resultados que se obtienen muestran la capacidad de reconocer cualquier relación sintáctica. Pero también es de utilidad obtener la capacidad de reconocer alguna relación en particular. Para estudiar esto, se realiza el mismo experimento, pero se divide la evaluación según la relación que está siendo predicha. Así, se puede comparar estos resultados con los del análisis anterior, en búsqueda de un beneficio al usar todas las *heads* en vez de solo una.

Por último, nuevamente se repiten ambas variaciones del experimento por completo, pero utilizando los valores de $\|\alpha \mathrm{f(x)}\|$ en lugar de considerar solo los coeficientes de atención, con la finalidad de determinar de qué forma se describe mejor el conocimiento sintáctico.

3.2. Resultados

En la Tabla 3.1 se presentan los mayores porcentajes de reconocimiento de las 10 relaciones más comunes del *dataset*, y se entregan los resultados para todas las relaciones en el Apéndice B. Para estos 10 casos, los mejores rendimientos en base a los coeficientes de atención y los valores de $\|\alpha \mathrm{f(x)}\|$ fueron ambos obtenidos por las mismas *heads*. Así, se agregan también las mejores *heads* para cada relación. En el apéndice se especifica qué *head* obtuvo el rendimiento en cada caso, ya que en ciertas relaciones el mejor rendimiento corresponde a distintas *heads*. También se entrega el porcentaje de reconocimiento de la relación en base al *offset* más común, en conjunto el *offset* en cuestión.

Tabla 3.1. Rendimiento sintáctico de las mejores *heads* de BETO según los valores de $\|\alpha f(x)\|$ y los coeficientes de atención. Se destacan los rendimientos más altos para cada relación. En estos casos, todos corresponden a $\|\alpha f(x)\|$.

Relación	$\|\alpha f(x)\|$	Atención	*Head*	Offset	Apariciones
case	**83.14**	81.33	7-5	47.53 (2)	5832
det	**95.22**	95.15	7-5	87.53 (1)	5253
nmod	**55.54**	53.31	8-6	37.53 (-2)	3403
obl	**40.46**	38.98	4-1	27.38 (-3)	2155
amod	**83.80**	82.83	6-12	62.08 (-1)	1957
conj	**44.76**	44.42	8-7	23.13 (-2)	1461
nsubj	**47.03**	44.93	6-2	20.41 (1)	1399
cc	**66.10**	65.57	8-1	46.89 (1)	1382
obj	**76.53**	74.37	8-6	51.26 (-2)	1124
advmod	**58.32**	58.22	6-12	46.23 (1)	1108

Los resultados indican que tanto los valores de $\|\alpha f(x)\|$ y coeficientes de atención presentan un rendimiento superior al *baseline* dado por el *offset* más común, incluso en algunos casos duplicándolo. Esto es una clara indicación de que el modelo sí posee capacidades sintácticas en algunas de sus *heads*. Además, para 23 de las 26 relaciones analizadas, los valores de $\|\alpha f(x)\|$ muestran un mejor rendimiento que los coeficientes de atención, por lo que mayoritariamente el análisis en base a las normas logra una mejor representación de las capacidades sintácticas que el análisis en base a las atenciones.

La Tabla 3.2 muestra los resultados del *probe* en base a los coeficientes de atención y los valores de $\|\alpha f(x)\|$. Estos son, el porcentaje de reconocimiento de relaciones sintácticas en general, además del porcentaje para las 10 relaciones más comunes. Los resultados para todas las relaciones están en el Apéndice C.

Tabla 3.2. Rendimiento del *Attention-Only Probe* aplicado en BETO según el uso de $\|\alpha f(x)\|$ y coeficientes de atención. Se destaca el rendimiento más alto. En el resultado general, las normas $\|\alpha f(x)\|$ describen mejor la capacidad sintáctica del modelo. Sin embargo, los resultados separados por relación son más variados.

Relación	$\|\alpha f(x)\|$	Atención
General	**68.38**	67.57
case	**85.17**	83.34
det	94.46	**94.70**
nmod	64.29	**64.40**
obl	**44.70**	42.99
amod	85.39	**85.74**
conj	32.98	**33.40**
nsubj	**50.11**	46.14
cc	**65.75**	63.56
obj	70.23	**75.12**
advmod	**68.52**	68.52

Se ve que el *probe* logra reconocer alguna dependencia sintáctica un 68.38% de las veces cuando se basa en las normas $\|\alpha f(x)\|$, demostrando que BETO contiene un conocimiento sintáctico robusto. Con respecto al reconocimiento de relaciones por separado, la mitad es mejor reconocida en base a los valores de $\|\alpha f(x)\|$ y la otra mitad en base a los coeficientes de atención. Sin embargo, el reconocimiento general es mayor con las normas $\|\alpha f(x)\|$.

3.3. Conclusiones

BETO presenta una capacidad de reconocer relaciones sintácticas en el comportamiento de algunas de sus *heads*, además de un conocimiento sintáctico general robusto bajo ambos coeficientes de atención y valores de $\|\alpha f(x)\|$.

Al comparar los resultados de $\|\alpha f(x)\|$ y los coeficientes de atención, se ve que una gran mayoría de las veces, $\|\alpha f(x)\|$ representa de mejor manera las capacidades lingüísticas de las *heads* del modelo de reconocer alguna relación sintáctica en particular. También presenta un mejor rendimiento sintáctico general en el *probe*. En el análisis de Koboyashi et al. se muestra que los $\|\alpha f(x)\|$ ya no se dirigen a los *tokens* especiales, como lo hace la atención, si no que apuntan mayoritariamente a los *tokens* normales. Los *tokens* especiales no deberían llevar información lingüística, por lo que sería de esperar que atender a ellos sea una pérdida de recursos cuando se trata de reconocer relaciones sintácticas. Así, nuestros resultados son consistentes con estos supuestos de la ausencia de información lingüística en *tokens* especiales.

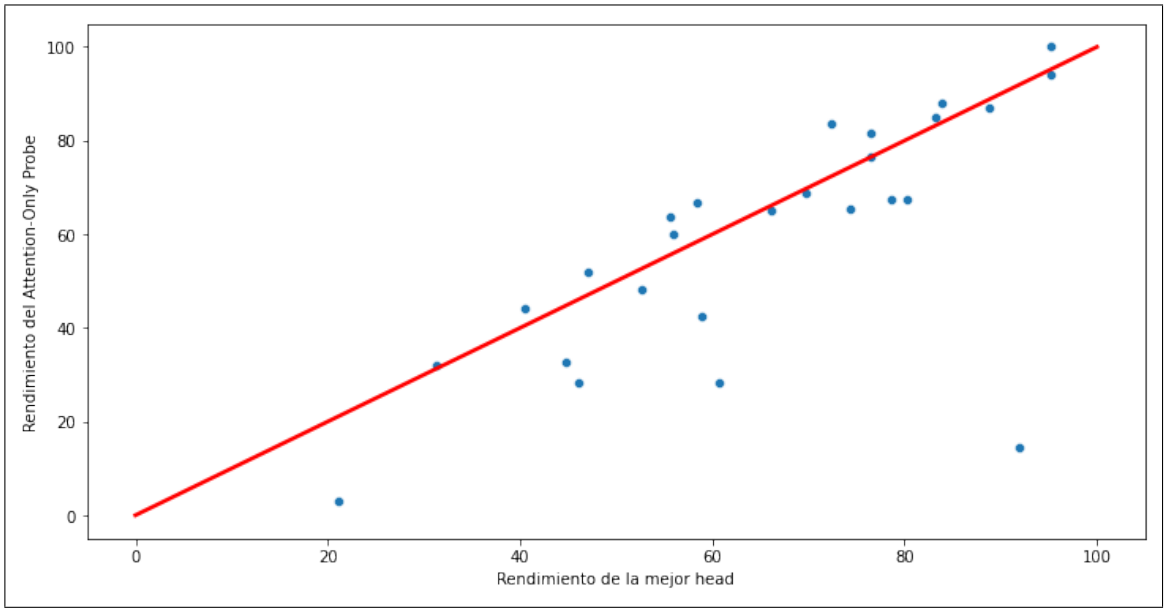


Figura 3.3. Comparación entre el rendimiento de las mejores *heads* y resultados del *Attention-Only Probe* para cada relación. Ambos rendimiento son muy similares, lo que sugiere una influencia directa.

Con respecto a la separación de los resultados del *probe*, se ve que los resultados de cada relación son muy similares a los de su mejor *head*. Esto se presenta de mejor manera en la Figura 3.3, donde se comparan los resultados de $\|\alpha f(x)\|$. Se ve una correlación de 0.71 y que los valores son cercanos a la línea que representa la identidad, mostrada en rojo.

Queda en duda si el conocimiento sintáctico reconocido por el *probe* está disperso en todo BETO, o si el *probe* simplemente reconoce cuando la mejor *head* para alguna relación está activa y en base a eso realiza su predicción. Esta duda será estudiada en la Sección 7 más adelante.

3.4. BERT vs BETO

Cabe destacar que para el mismo *probe* en base a los coeficientes de atención, BETO obtuvo 6.57 puntos más que BERT, indicando que el modelo español tiene una capacidad sintáctica mayor a su contraparte en inglés.

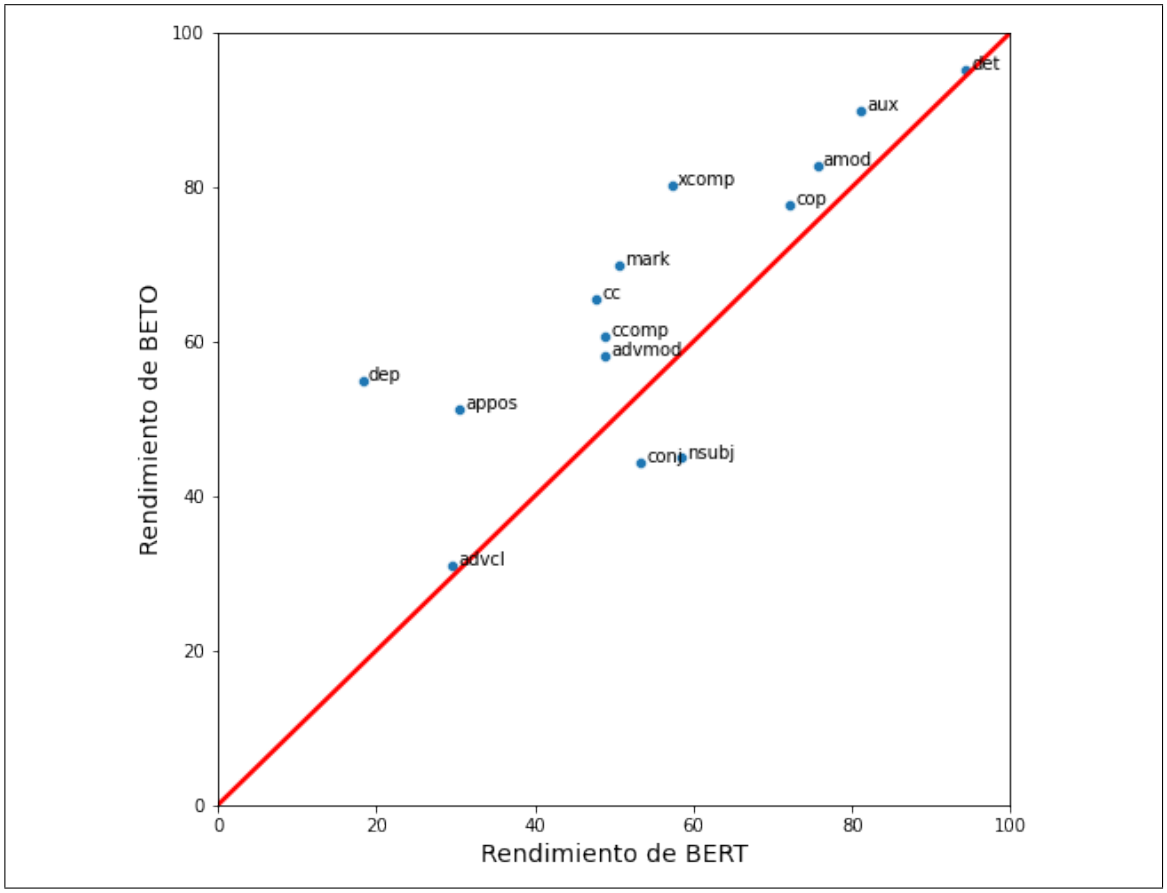


Figura 3.4. Rendimiento de la mejor *head* en BETO y BERT para cada relación. Hay una alta correlación en el reconocimiento de las relaciones en ambos idiomas.

En su estudio, Clark et al., 2019 se preguntan si las mismas relaciones reconocidas por BERT serían igualmente reconocidas en modelos de otros lenguajes, y lo plantean como trabajo futuro. Con los resultados encontrados ahora, es posible responder esto en parte. Solo algunas relaciones son publicadas en el estudio de BERT, y de esas, solo algunas están presentes en el *dataset* utilizado en este experimento. La Figura 3.4 muestra los rendimientos para las relaciones presentes en ambos estudios. Aunque sean pocas, éstas muestran una alta correlación de 0.78. Sin embargo, aun no queda claro por qué las relaciones que BERT predice, también las predice BETO, ya que aunque se traten de las mismas relaciones, éstas se pueden estructurar de maneras distintas para distintos lenguajes. Así, surge la duda de si este patrón se mantiene entre idiomas que tengan orígenes más distintos que el del español y el inglés.

3.5. Mapas de Conocimiento

En base a los experimentos recién presentados, se define el concepto de Mapas de Conocimiento. Un Mapa de Conocimiento de la relación sintáctica r es una matriz del tamaño igual al modelo (*layers* $\times$ *heads*), donde la posición (L, H) es el rendimiento de la predicción de r según la *head* L-H. Como se mostró que las $\|\alpha$f(x)$\|$ representan de mejor manera las capacidades sintácticas, se usarán éstas como base para la predicción mencionada. Los Mapas del Conocimiento ayudan a visualizar las capacidades de reconocimiento de las *heads* de BETO cuando se presentan como mapas de calor, además de permitir el análisis sintáctico de todas las *heads* al mismo tiempo. La Figura 3.5 muestra ejemplos de Mapas de Conocimiento para algunas relaciones sintácticas, y el Apéndice D contiene los Mapas de todas las relaciones estudiadas. Por ejemplo, para el Mapa de Conocimiento de la relación *case* se observa que la *head* 7-5 es la que mejor reconoce la relación. Sin embargo, también existe un análisis entre *heads*. Se observa que el rendimiento de todas las *heads* es bastante similar al rendimiento de las *heads* para predecir *det*, y muchos patrones presentes en el Mapa de Conocimiento de *case* también están en

el de *det*. Esta observación podría sugerir similitud entre estas dos relaciones sintácticas. En la Sección 4, se estudiará este fenómeno con mayor profundidad.

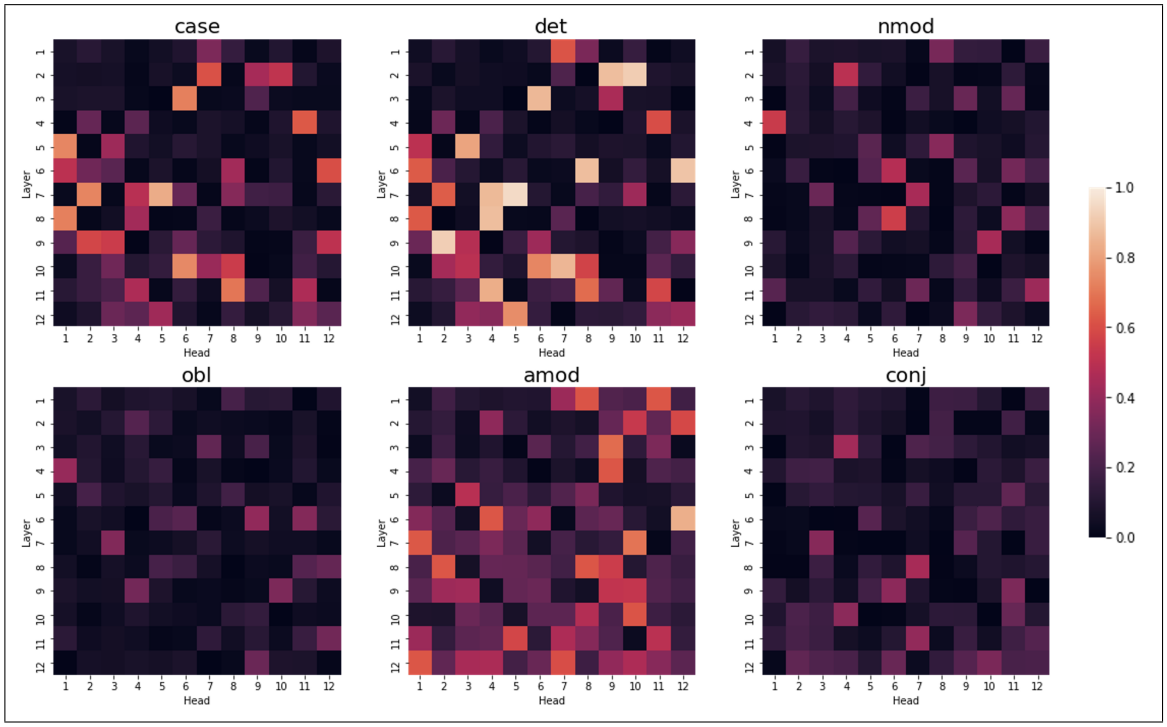


Figura 3.5. Ejemplos de Mapas de Conocimiento de BETO sobre distintas relaciones. Se ve que el reconocimiento de relaciones forma patrones que se podrían repetir.

4. *CLUSTERS* DE MAPAS DE CONOCIMIENTO

Los patrones formados en los Mapas de Conocimiento corresponden al comportamiento completo de BETO al reconocer alguna relación sintáctica. Sin embargo, muchos de estos patrones se ven repetidos en distintos Mapas. En esta sección se estudiará si las similitudes entre Mapas de Conocimiento indican similitudes en las relaciones mismas. De ser así, esto indicaría que, además de reconocer relaciones sintácticas, BETO usa un patrón similar para codificar relaciones sintácticas parecidas.

Además, se revisará la importancia del *offset* más común de cada relación para sus Mapas de Conocimiento, y cómo afecta eliminar este *offset* a la similitud entre patrones generados para relaciones parecidas. Esto daría una estimación a las limitaciones presentes en los Mapas de Conocimiento del modelo.

4.1. Metodología

Esencialmente, el experimento consistirá en encontrar qué relaciones son similares entre ellas, qué Mapas de Conocimientos se asemejan, y revisar si coinciden las semejanzas.

Para encontrar qué relaciones son similares entre ellas, se requieren resolver dos problemas. El primero vendría siendo definir qué significa que dos relaciones sintácticas sean similares. Para evitar complicaciones teóricas de la sintaxis fuera del alcance de esta tesis, se trabaja bajo el supuesto que las relaciones sintácticas están bien caracterizadas por la distribución de distancias entre las palabras involucradas en la relación. Como se mencionó anteriormente, ciertas relaciones suelen manifestarse a alguna distancia específica en un alto porcentaje de sus apariciones, lo que se llamó su *offset* más común. Sin embargo, se podría hacer un segundo análisis a esto, y obtener todos los *offsets* que presenta cada relación en el *dataset*, junto con su frecuencia. Así se obtienen representaciones más completas de cómo es una relación. La Figura 4.1 muestra las distribuciones de distancia para algunas relaciones. Por ejemplo, se observa que la relación *xcomp* tiene como *offset*

más común la distancia -1. También se ve que sus distancias son solo negativas, significando que la cabeza sintáctica siempre aparece antes que la dependencia. En contraste, la relación *compound* puede aparecer en ambos sentidos, pero no relaciona palabras tan distantes como lo hace *xcomp*. Luego, se pueden apreciar diferencias entre estas relaciones en base a sus distribuciones de distancias.

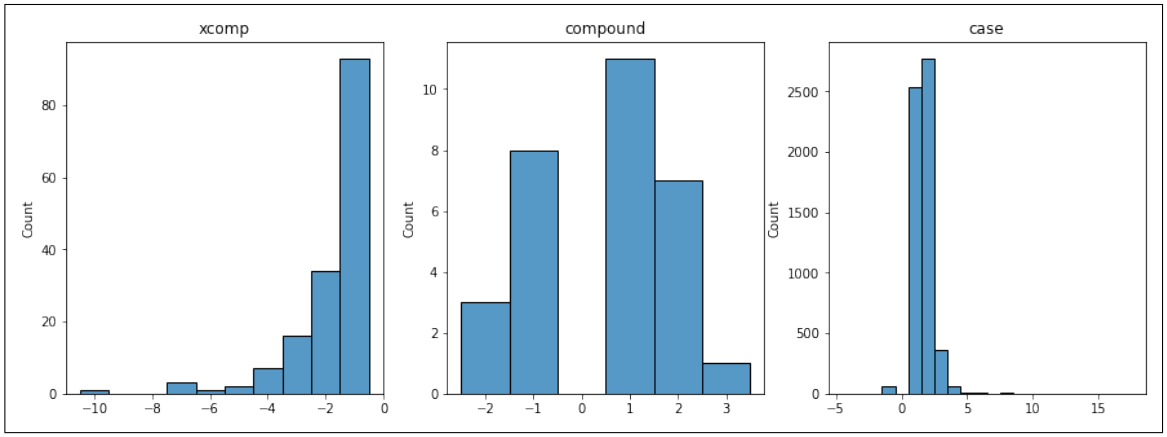


Figura 4.1. Ejemplos de distribuciones de distancias para relaciones sintácticas. Se puede ver que existen distintos comportamientos para distintas relaciones.

Al trabajar con las distribuciones de distancia se logra definir las relaciones sintácticas de una manera cuantificable, permitiendo una comparación numérica entre ellas. Luego, se puede utilizar la distancia de Jensen-Shannon, métrica usada comúnmente para distribuciones de probabilidad, para obtener una distancia entre las relaciones.

Ahora que se tiene una definición numérica para las similitudes entre las relaciones, se necesita identificar cuáles relaciones son similares. Una forma de realizar esto es encontrando *clusters* sobre puntos que representen a las relaciones. Así, se obtiene una clara definición de grupos dentro de las relaciones. Luego, se puede realizar el mismo proceso sobre los Mapas de Conocimiento de las relaciones y revisar si las mismas relaciones pertenecen a los mismos grupos. Para lograr esto, el primer paso será obtener puntos que representen a cada una de las relaciones. Los algoritmos de clusterización suelen trabajar de mejor manera cuando la dimensionalidad de los puntos es baja. Además, trabajar con

dos dimensiones permite visualizar los puntos y los *clusters* que son formados. Así, se usa la técnica de *multidimensional scaling* (MDS) (Kruskal, 1964) para obtener puntos 2-Dimensionales que representen las relaciones sintácticas en base a las distancias entre ellas mismas. El algoritmo usa las distancias Jensen-Shannon obtenidas anteriormente, y los puntos devueltos mantienen las distancias relativamente invariantes. Luego, sobre los puntos obtenidos se usa la técnica de clusterización llamada *K-Means* (Lloyd, 1982), con $k = 3$, con la cual se determinan los tres *clusters* que mejor separan las relaciones. Estos *clusters* logran especificar qué relaciones son similares entre ellas. Al trabajar con puntos 2-Dimensionales, es posible determinar la formación de *clusters* visualmente. Sin embargo, se usa la técnica de *K-Means* para confirmar o desafiar las identificaciones manuales de una manera más objetiva.

Para los Mapas de Conocimiento es necesario realizar el mismo proceso. Los puntos obtenidos por MDS también son 2-Dimensionales, pero el algoritmo utiliza la distancias Frobenius, métrica usada comúnmente para comparar matrices. Igualmente, se obtienen tres *clusters* usando *K-Means* con $k = 3$. Así, el resultado es la definición de tres grupos de Mapas de Conocimiento, donde los miembros de cada grupo son similares entre ellos.

Por último, se observan las similitudes en los grupos formados. Para cada grupo de relaciones, se revisa si el grupo se repite en los Mapas de Conocimientos y determinamos cuántas relaciones fueron clasificadas por los Mapas correctamente.

Además, se busca medir la importancia del *offset* más común en los Mapas de Conocimiento. En particular, se quiere ver si Mapas de Conocimiento formados sin los ejemplos que presentan el *offset* más común poseen la misma capacidad de reconocer similitudes entre las relaciones. Para esto, se filtran del *dataset* las apariciones de las relaciones en su *offset* más común, y en base al *dataset* filtrado se obtienen los rendimientos que definen a un Mapa de Conocimiento nuevo. Luego, se realiza el mismo proceso de generación de puntos y clusterización que se realizó para los Mapas de Conocimiento no filtrados. Por último, se mide la similitud entre los *clusters* formados y los *clusters* de las relaciones, comparándolo con el análisis de los Mapas de Conocimiento originales.

4.2. Resultados

En la Figura 4.2 se presentan las puntos que representan las relaciones sintácticas. Los colores indican a qué *cluster* pertenece la relación.

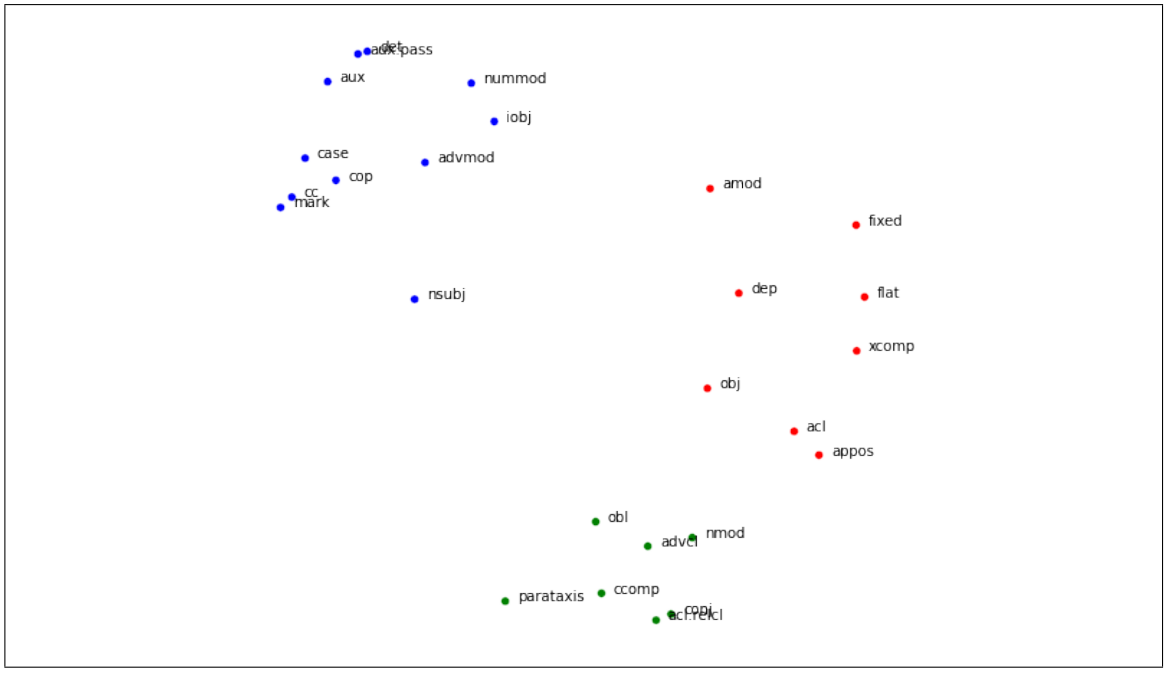


Figura 4.2. *Clusters* de relaciones sintácticas similares. Se ven tres claros grupos entre la relaciones. El algoritmo con $k = 3$ entrega la misma clusterización que se obtiene visualmente.

Con estos resultados se puede determinar qué relaciones sintácticas son similares entre ellas: las relaciones son similares a las que pertenecen al mismo *cluster*.

A continuación, la Figura 4.3 muestra los puntos que representan los Mapas de Conocimiento para cada relación. Nuevamente, el color indica el *cluster* al que pertenece. Además, se agrega el Mapa Vacío, que vendría siendo un Mapa de Conocimiento que no reconoce ninguna relación en ninguna *head*. Su adición sirve como punto de referencia, para determinar dónde el conocimiento es en general menor. Así, el Mapa Vacío será útil cuando se analice las modificaciones de los *clusters* al eliminar el *offset* más común.

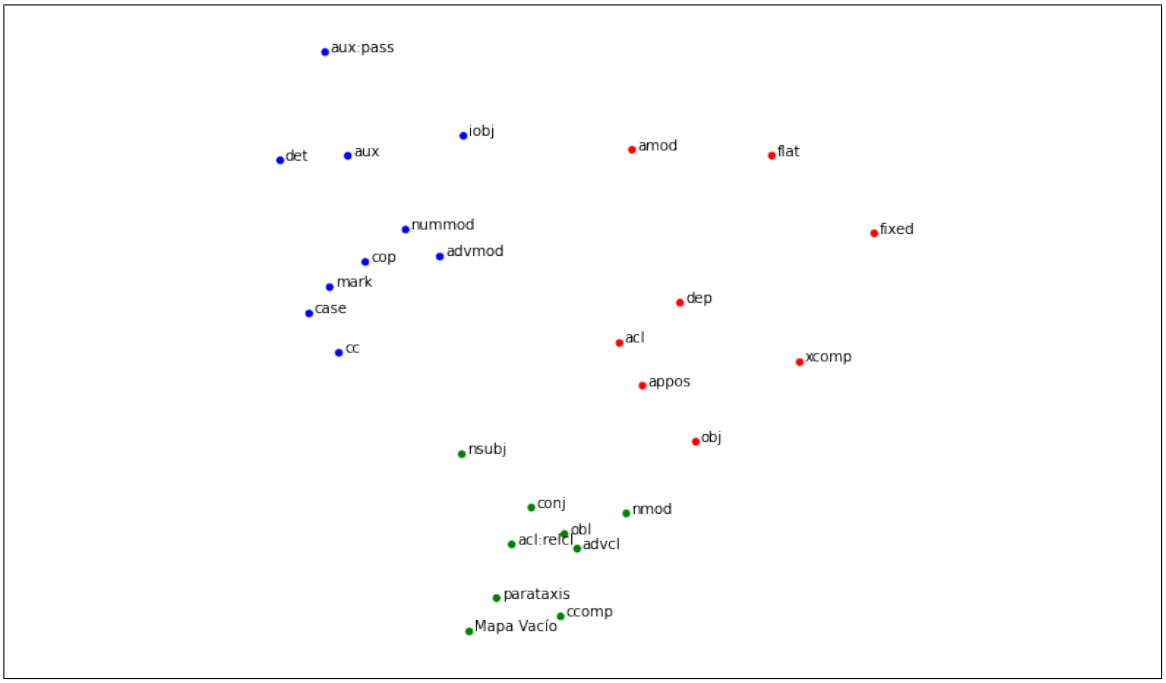


Figura 4.3. *Clusters* de Mapas de conocimiento similares. Se observa que los *clusters* resultantes son muy similares a los *clusters* de relaciones con respecto a qué relación se incluye en cada *cluster*.

Inmediatamente se reconoce que las dos clusterizaciones son casi idénticas. En específico, se observa que cada *cluster* de Mapas de Conocimiento incluyen a todas las relaciones que fueron encontradas similares en el resultado anterior, a excepción de la relación *nsubj* que cambió de un *cluster* a otro. Es decir, para 25 de las 26 relaciones analizadas, los Mapa de Conocimiento de BETO clasifican a las relaciones de la misma forma que las clasifican sus distancias.

Por último, la Figura 4.4 presenta las representaciones de los Mapas de Conocimientos filtrados de los *offsets* más comunes. Nuevamente, se agrega el Mapa Vacío.

Al eliminar el *offset* más común de la formación de los Mapas de Conocimiento, se ve que las capacidades de reconocer similitudes entre relaciones se reduce bastante. Los

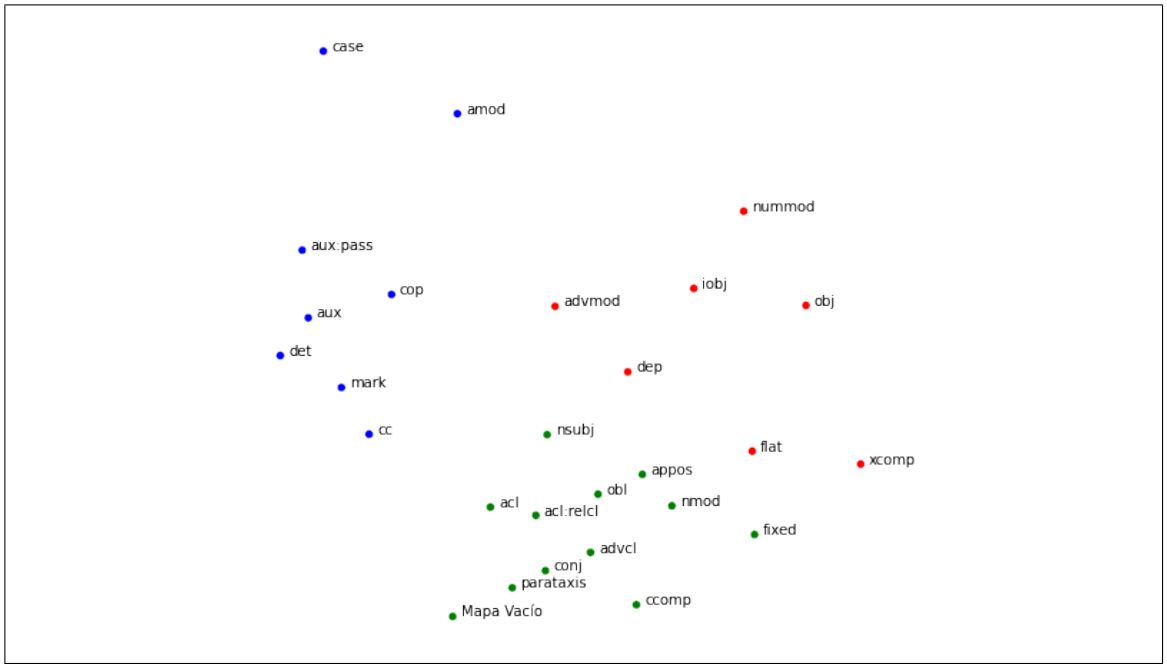


Figura 4.4. *Clusters* de Mapas de Conocimiento filtrados. Se ve como las clusterización es menos obvia y como los miembros de distintos *clusters* están más cerca.

clusters que encuentra *K-Means* son menos intuitivos y están más cerca entre ellos. Específicamente, 8 relaciones fueron atribuidas a algún *cluster* incorrecto. Además, los puntos de los Mapas en general se acercaron al Mapa Vacío, indicando que el conocimiento de cada Mapa disminuyó al eliminar el *offset*.

Sin embargo, cabe destacar que aunque la capacidad de reconocer similitudes en las relaciones decrece bastante, aun así sigue estando presente. En particular, es importante notar la observación contraria: 18 de las 26 relaciones fueron clasificadas correctamente por los Mapas de Conocimiento, a pesar de no incluir el conocimiento del *offset* más común para cada relación.

4.3. Conclusiones

Se concluye que los Mapas de Conocimiento se comportan similarmente para relaciones sintácticas parecidas. Esto podría indicar que la información que asemeja las relaciones está incorporada en los Mapas de Conocimiento. Se observa que este efecto se debe en gran parte a incluir al *offset* más común en el conocimiento que genera al Mapa, pero que sin embargo, incluso sin este conocimiento los Mapas manifiestan las similitudes. Así, la comprensión sintáctica de BETO es más compleja que solo reconocer las relaciones: el modelo también captura la asociación entre las relaciones sintácticas.

5. ANÁLISIS DE FALLAS EN LAS PREDICCIONES SINTÁCTICAS

En el análisis sintáctico base se ve que ciertas *heads* tienen una alta capacidad de reconocer alguna relación sintáctica específica. Más aún, para cualquier relación sintáctica existe una *head* de BETO que reconoce esta relación con una probabilidad mayor al *baseline* de su *offset* más común. Estos resultados son muy interesantes, pero dejan abierta la pregunta: ¿A qué se debe el porcentaje de relaciones que no se logran predecir correctamente? En este experimento, se dará cuenta de distintos factores que podrían afectar al reconocimiento de relaciones, y se analizará cuáles efectivamente resultan en predicciones fallidas para las mejores *heads* del modelo.

5.1. Metodología

Los factores a analizar tomarán en consideración cada relación en base a las palabras que relaciona. Luego, los factores de relaciones con predicciones correctas se compararán con los factores de las relaciones con predicciones incorrectas. Finalmente, los resultados se dividirán según la relación que se esté analizando. Cada instancia de relación vendrá de las frases perteneciente al *set* de desarrollo de GSD-Spanish.

5.1.1. Grado de tokenización

El primer factor a analizar es el grado de tokenización. Al transformar una palabra a *tokens*, ésta podría resultar dividida en distintas cantidades. A esta cantidad se le denomina el grado de tokenización. Por ejemplo, la palabra "expectación" al ser tokenizada, se separa entre los *tokens* "expec" y "##tación" (los $\#\#$ indican que el segundo *token* no proviene de otra palabra, si no de la misma que el *token* anterior), por lo que su grado de tokenización es 2. Por otro lado, la palabra "libros" solo se transforma en el *token* "libros", por lo que su grado de tokenización es 1. El grado de tokenización podría significar una complicación adicional para BETO, por lo que se analizará si juega un rol en las fallas del reconocimiento sintáctico. A cada ejemplo de una relación del *dataset* se le asigna un

puntaje equivalente al promedio del grado de tokenización de las palabras que asocia. La Figura 5.1 ilustra la asignación de puntaje para dos ejemplos de relaciones.

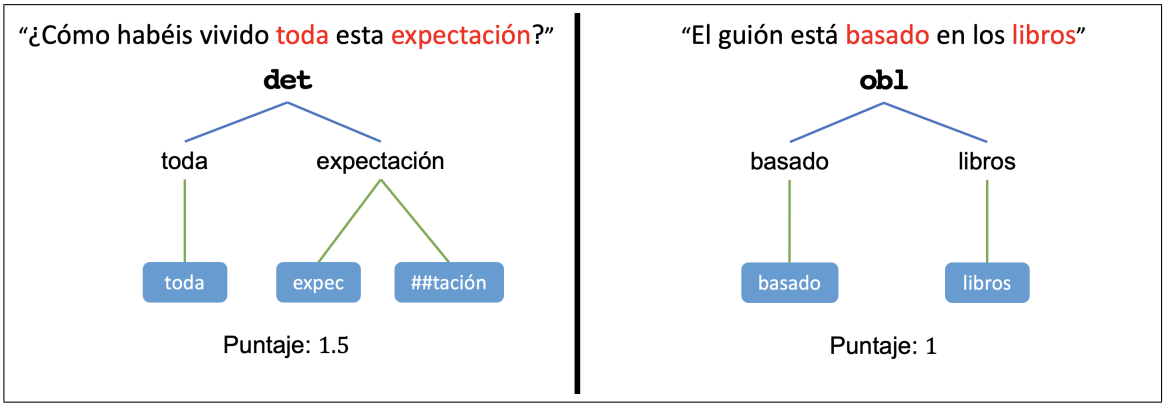


Figura 5.1. Puntajes de grado de tokenización para relaciones. Las líneas verdes simbolizan el proceso de tokenización de las palabras. Se ven dos casos en donde palabras están relacionadas sintácticamente. El puntaje asociado a la relación es el promedio del puntaje de las palabras. Fuente: Elaboración propia.

Luego, para cada relación analizada se considera la *head* que mejor reconoce la relación. Se obtiene la predicción de la *head* para cada ejemplo de la relación, y se separan los ejemplos según la correctitud de la predicción. Por último, se calcula el puntaje promedio para ambos grupos de ejemplos. Si hay una diferencia grande entre los puntajes, esto sugeriría que el grado de tokenización es un factor importante en las fallas de predicciones sintácticas de la relación.

5.1.2. Frecuencia de las palabras

El siguiente factor a estudiar es la frecuencia de las palabras utilizadas. Ciertas palabras son bastante comunes en algunas relaciones, y otras aparecen muy pocas veces. Así, BETO puede intentar de no relacionar dos palabras bajo cierta relación si alguna de las palabras no es usualmente usada en ella. Luego, se analizará si la frecuencia de las palabras en cada relación afecta en la predicción.

Similar al factor anterior, se le otorga un puntaje a cada palabra, pero esta vez el puntaje es asociado a la frecuencia de ésta y a la relación analizada. Un posible puntaje sería el porcentaje que aparece cada palabra en cada relación, y al analizar una relación, usar su puntaje asociado. Sin embargo, ciertas palabras son demasiado comunes en el *dataset*, por lo que dominarían sus puntajes en todas las relaciones. Por esto, se decide normalizar los puntajes de cada palabra por su porcentaje en todo el *dataset*. Así, el puntaje asociado a una palabra en una relación sería una proporción, donde su valor es 1 si la palabra aparece tanto en la relación como en el *dataset*, menor que 1 si aparece menos en la relación, y mayor que 1 si aparece más.

Para cada relación, se analiza el puntaje más bajo de las dos palabras relacionadas. Se elige el más bajo porque se quiere estudiar si una palabra poco común afecta en las predicciones.

Nuevamente, en base a la separación de los ejemplos según la correctitud de la predicción, se calcula los puntajes promedio para cada grupo. Una diferencia considerable significaría que al usar palabras poco comunes para alguna relación, se afectaría la capacidad del modelo de reconocerla.

5.1.3. Presencia del token [UNK]

El tercer factor es la presencia del *token* especial "[UNK]". Durante la tokenización, puede ocurrir que ciertos caracteres de una palabra no son reconocidos, y como consecuencia se le atribuye el *token* [UNK] a la palabra. La presencia de este *token* indica cierta falta de información, la cual podría afectar al rendimiento sintáctico de las *heads* de BETO. Para cada relación, se toma la división de sus ejemplos según la correctitud de su predicción, y se calcula qué porcentaje de los ejemplos involucra alguna palabra que es tokenizada como [UNK]. Si se encontrara una diferencia grande, se concluiría que el [UNK] es un factor importante para el reconocimiento sintáctico.

5.1.4. Distancia de las palabras

Por último, el cuarto factor es la distancia entre las palabras de la relación. Como se ha mencionado anteriormente, las relaciones conectan palabras a una variedad de distancias, y el relacionar palabras a distancias poco comunes para la relación, se podría afectar a la predicción de las *heads* de BETO. Para cada relación, se considera la distribución de distancias obtenida en el experimento de la Sección 4. Para cada una de las distancias, se medirá el rendimiento de reconocimiento solo sobre los ejemplos que presenten esa distancia en particular. Diferencias entre el rendimiento de las distancias podría indicar que BETO reconoce mejor la relación solo cuando ésta se comporta de cierta manera.

5.2. Resultados

Para los resultados involucrando el grado de tokenización, la frecuencia de palabras y la presencia del [UNK], solo se reportarán los resultados que presenten una diferencia considerable entre los puntajes. Para ser considerados, los puntajes de predicciones correctas e incorrectas deben diferir en un mínimo de 25%. También, como el *token* [UNK] es poco común, solo se considerarán las relaciones en donde el *token* apareció más de 20 veces.

Con respecto al análisis del grado de tokenización, no se encontró ninguna relación que tuviera una diferencia considerable entre los grupos de predicción correcta e incorrecta. Así, se determina que el grado de tokenización no afecta en gran medida las capacidades sintácticas del modelo.

Para la frecuencia de las palabras, 6 relaciones muestran diferencias considerables entre los puntajes promedio. En éstas, las palabras involucradas en cada relación suelen ser menos frecuentes cuando la predicción falla. Esto indicaría que el modelo reconoce menos una instancia de una relación cuando se usan palabras que no suelen ser usadas en ese contexto. Los puntajes para éstas 6 relaciones se encuentran en la Tabla 5.1.

Tabla 5.1. Diferencias considerables en la frecuencia de aparición de las palabras. Las relaciones predichas correctamente suelen tener palabras más comunes para la relación.

Relación	Predicción correcta	Predicción incorrecta	Diferencia
advmod	13.63	10.74	26.91%
iobj	24.14	17.43	38.49%
fixed	10.62	6.68	58.89%
aux:pass	43.44	33.92	28.06%
parataxis	75.57	56.26	34.33%
dep	111.20	74.19	49.89%

Así, la frecuencia de las palabras es un factor que afecta el bastante el rendimiento del modelo en múltiples relaciones.

Los resultados también son llamativos al analizar la presencia del *token* [UNK]. Existen 9 relaciones que presentan diferencias considerables entre las predicciones correctas y las incorrectas. Éstos se presentan en la Tabla 5.2.

Tabla 5.2. Diferencias considerables en la presencia del *token* [UNK]. Se presentan 9 relaciones con más de 20 apariciones del *token* y con diferencias muy grandes.

Relación	Predicción correcta	Predicción incorrecta	Diferencia	Apariciones
det	0.23%	31.47%	13019.46%	91
case	0.49%	3.15%	537.16%	55
appos	2.21%	3.82%	72.56%	23
nmod	0.95%	2.37%	149.83%	54
nsubj	6.61%	4.23%	56.21%	74
nummod	12.93%	40.00%	209.21%	123
dep	67.85%	9.09%	646.43%	42
conj	0.91%	3.34%	264.68%	33
flat	2.42%	5.26%	117.11%	21

Se destacan las relaciones *det* y *nummod*, donde el 31.47% y 40.00% de las predicciones incorrectas respectivamente, involucran a un *token* [UNK]. Revisando en específico a las dos relaciones, se ve que ambas involucran bastante al caracter "%". Como ejemplo, en la frase:

"El riesgo se redujo en un 50 %."

la palabra "un" está relacionada con "%" por medio de la relación *det*. El proceso de tokenización de BETO no reconoce al símbolo, por lo que le atribuye el *token* especial. En 79 ocasiones la relación *det* involucra al *token* [UNK] y no es reconocida. En 78 de estas 79, originalmente el *token* correspondía a "%". Sería interesante analizar si agregar el "%" a los caracteres reconocidos en la tokenización podría ayudar en el reconocimiento de estas relaciones.

Además de presentar diferencias muy altas en *det* y *nummod*, también presenta diferencias considerables en múltiples otras relaciones. Se concluye entonces que la presencia del [UNK] es un factor que afecta altamente al reconocimiento de relaciones sintácticas.

Por último se obtienen las diferencias de rendimiento de cada relación según la distancia de las palabras involucradas. En la Figura 5.2 se presentan los resultados de las 6 relaciones más comunes en el *dataset*. Estos consisten de la distribuciones de distancias representadas como histogramas, donde además el color de cada barra indica el reconocimiento de la relación a la distancia que corresponde la barra. Los resultados para todas las relaciones se muestran en el Apéndice E.

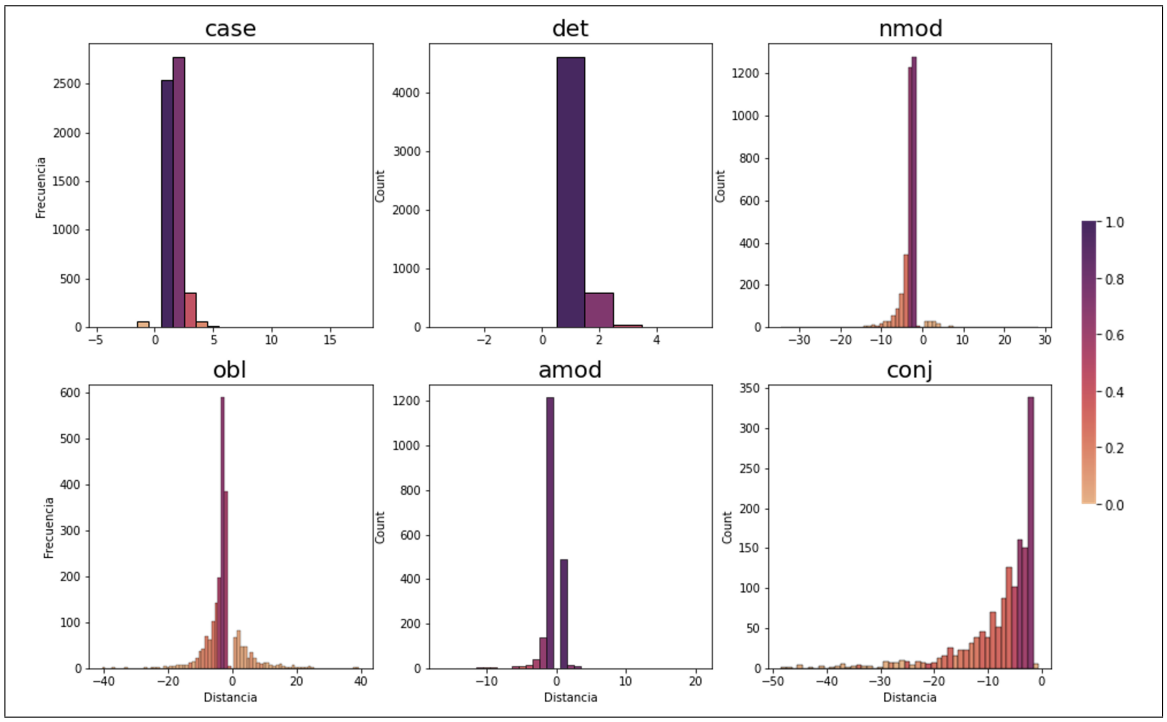


Figura 5.2. Reconocimiento de una relación según su distancia. Se observa una clara tendencia a un mayor reconocimiento de la relación cuando ésta tiene una distancia común.

Las relaciones muestran un claro aumento en su reconocimiento cuando se manifiestan en distancias frecuentes. Para realizar un mejor análisis de esto, se separan las distancias en dos categorías: Las distancias comunes de una relación son las 5 distancias en la que

relación más se presenta, mientras que las distancias no comunes son el complemento. La Tabla 5.3 presenta el rendimiento de las mejores *heads* de BETO para cada relación, además del rendimiento cuando se trata de una distancia común, y cuando no. Solo se muestran los resultados para las 10 relaciones más comunes, y todos los resultados se mostrarán en el Apéndice F.

Tabla 5.3. Rendimiento de la predicción de relaciones según el tipo de distancia que posee.

Relación	General	Distancias comunes	Distancias no comunes
case	83.14%	83.58%	6.06%
det	95.22%	95.35%	22.22%
nmod	55.54%	60.34%	7.44%
obl	83.80%	84.66%	8.83%
amod	83.80%	84.66%	53.70%
conj	44.76%	58.49%	24.14%
nsubj	47.03%	56.22%	28.45%
cc	66.10%	70.23%	21.88%
obj	76.53%	78.83%	48.19%
advmod	58.32%	62.70%	23.85%

El reconocimiento de la relación decrece críticamente cuando la relación se presenta por medio de una distancia no común. Se reconoce, entonces, que la distancia de la relación afecta extremadamente el reconocimiento de la relación.

A partir de este resultado, es interesante buscar alguna solución para obtener un mejor rendimiento en el reconocimiento de relaciones sintácticas, evitando el problema de las distancias. La solución más directa es simplemente evitar las distancias no comunes cuando esto sea posible. Como ejemplo, se considera la frase:

"De niña se trasladó a Inglaterra para educarse [...]."

La palabra "niña" es la dependencia de "trasladó" por medio de la relación *obl* a una distancia de 2. Sin embargo, la relación *obl* no suele estar a una distancia 2, pero sí es muy común en una distancia de -2. Luego, podría ser de utilidad modificar la frase para que sea de esta manera:

"Se trasladó de niña a Inglaterra para educarse [...]."

En ésta, la relación entre "niña" y "trasladó" es la misma, pero se presenta a una distancia -2. Se entregaron como *input* ambas frases a BETO, y se revisó la mejor *head* para *obl*. Efectivamente, la *head* no reconoce la relación en el primer caso, pero sí la reconoce en el segundo. Así, se logró modificar el *input* que se le entrega a BETO para poder asegurar una comprensión superior por parte del modelo. No todas las relaciones son tan modificables como la del ejemplo, pero las que sí pueden recibir un aumento en su reconocimiento sintáctico con pocos cambios.

5.3. Conclusiones

Se identifican tres factores que afectan bastante a la capacidad del modelo de reconocer alguna relación. Estos son la frecuencia de las palabras involucradas en la relación, el uso del *token* especial [UNK], y la distancia en que se encuentran las palabras de la relación.

La frecuencia de las palabras juega un rol importante en la predicción de la relación. Este factor destaca la naturaleza del aprendizaje del modelo: la memorización. Si BETO no ve alguna palabra involucrada en cierto contexto sintáctico durante su entrenamiento, entonces no aprende ese uso de la palabra.

Con respecto a la distancia de las relaciones, ésta resulta ser el factor predominante con respecto a predecir relaciones correctamente: el modelo reduce su rendimiento críticamente cuando la relación a predecir presenta una distancia poco frecuente. Se muestra que es posible realizar pequeñas modificaciones a la frase, manteniendo su significado original,

de tal manera que la frase es más fácil de entender sintácticamente para las *heads* de BETO. Sin embargo, queda como trabajo futuro implementar una metodología que logre realizar estas modificaciones de manera sistemática.

Por último, es importante recordar que el modelo no conoce la definición de las relaciones si no que posiblemente reconoce los patrones que le son útiles para capturar el contexto de la frase. Así, podría ser que BETO solo esté interesado en un subconjunto de casos para cada relación, mientras que otro subconjunto puede que no le resulte tan apropiado. Por ejemplo, se tiene la relación *iobj* que corresponde al objeto indirecto. En más del 65% de los casos la relación presenta una distancia de 1, y de estos, más del 99.7% involucra un pronombre reflexivo (se, me, te, nos), como en la frase:

"Con esta obra la soprano [...] se despidió de la escena."

Luego cuando se considera la relación *iobj* presente en:

"Los nobles catalanes le concedieron su apoyo al rey."

Aunque técnicamente la relación sea la misma, cada ejemplo pertenece a un subconjunto distinto de casos para la relación. Luego, un bajo rendimiento en la relación $iobj$ podría significar que el modelo sí reconoce la relación, pero solo lo hace cuando ésta tiene una estructura específica.

6. CONOCIMIENTO SINTÁCTICO AL RESPONDER PREGUNTAS

En el análisis anterior, se estudian distintos factores que llevan a las mejores *heads* del modelo a fallar en la predicción de relaciones sintácticas. En éste, el enfoque estará en qué se predice cuando falla una predicción, y si esto conlleva carencias en las capacidades del modelo.

Cuando se tiene una dependencia sintáctica relacionada a su cabeza, y la predicción falla, un patrón interesante para estudiar podría ser si la predicción se salta la cabeza sintácticamente y relaciona directamente a la dependencia con su abuelo sintáctico. Así, se revisa si alguna relación presenta este patrón en sus predicciones incorrectas. Tres relaciones (*amod*, *advmod* y *nummod*) resultan destacar, presentando este patrón más de un 20% de sus predicciones incorrectas, como muestra la Tabla 6.1.

Tabla 6.1. Porcentaje de predicciones incorrectas dirigidas al abuelo sintáctico. Las tres relaciones presentan el patrón una gran cantidad de veces, mientras que el resto de las relaciones solo lo muestran un 3.79% en promedio. Todos los porcentajes de las relaciones son presentadas en el Apéndice G.

Relación	Predicciones incorrectas dirigidas al abuelo sintáctico
amod	20.19%
advmod	27.30 %
nummod	36.67 %

De manera interesante, las tres relaciones que presentan este patrón son modificadores de sustantivos, ya sea el adjetivo (*amod*), adverbio (*advmod*) o modificador numérico (*nummod*). Así, la estructura de la frase podría estar siendo comprendida en un orden que no es correcto.

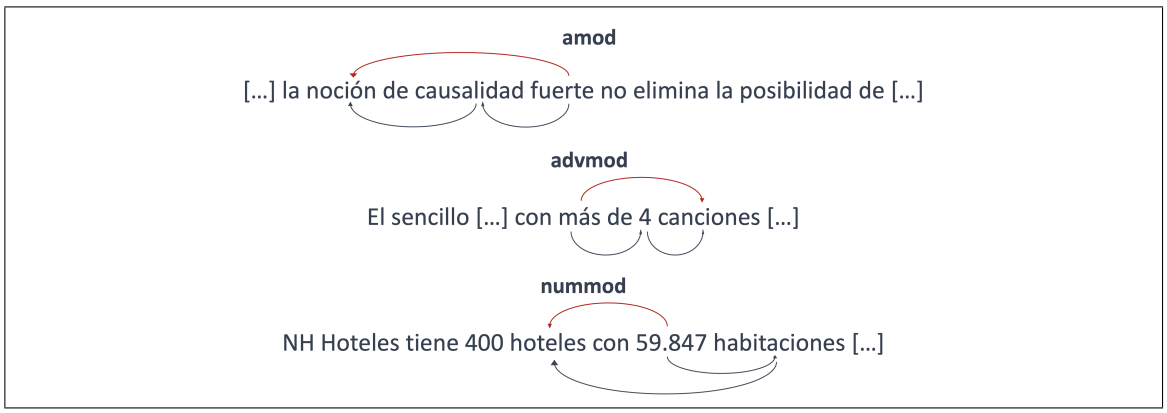


Figura 6.1. Predicciones incorrectas que apuntan al abuelo sintáctico. En azul se muestra el árbol sintáctico original, mientras que en rojo se muestra la predicción incorrecta. Se ve cómo las frases podrían significar algo distinto si el modificador se le asocia al abuelo sintáctico o bien si el modificador no se aplica a la cabeza sintáctica. Fuente: Elaboración propia.

Algunos ejemplos de estos se presentan en la Figura 6.1. Aquí se ve cómo podrían cambiar ciertos significados dentro de cada frase si el modificador apunta al abuelo sintáctico o si no se dirige a la cabeza sintáctica. En el caso de *advmod*, el número de canciones del sencillo es mínimo 5, y en el caso de *nummod* hay 59.847 habitaciones, no 59.847 hoteles. También podría darse que el significado de la frase no cambie mucho, como ocurre en el ejemplo de *amod*.

Se busca poner en prueba si el modelo comprendió correctamente el significado de la frase, incluso cuando le atribuye relaciones sintácticas incorrectas. Una forma de realizar esto sería generar preguntas sobre estos ejemplos, donde la pregunta requiera la correcta comprensión de estos modificadores, y ver si el modelo es capaz de responder correctamente. En este experimento se generará dicho *dataset* de preguntas, y se medirá la capacidad de BETO de responderlas.

6.1. Metodología

Este experimento requiere dos procesos: el *fine-tuning* de BETO para que sea capaz de responder preguntas, y la generación del *dataset* de dichas preguntas.

6.1.1. *Fine-tuning* de BETO

Se realiza un *fine-tuning* con el *dataset* de SQuAD-es (Carrino et al., 2019). Esto se repite por 3 épocas, con un *learning rate* de $5e-5$, siguiendo a las configuraciones realizadas por Devlin et al., 2018.

Luego, para obtener un punto de referencia con respecto al rendimiento del modelo, se evalúa a BETO en el *dataset* MLQA (Lewis et al., 2019), en la sección en español. El modelo obtiene un puntaje F1 de 68.58, y responde exactamente la respuesta correcta (EM) un 48.14% de las veces.

6.1.2. Generación del *dataset*

El primer paso es conseguir los ejemplos que presenten el comportamiento buscado. Así, se obtienen los ejemplos de predicciones incorrectas de las relaciones *amod*, *advmod* y *nummod*, en donde la predicción se dirija al abuelo sintáctico. Se obtienen 64 ejemplos para *amod*, 110 para *advmod* y 55 para *nummod*, es decir, un total de 229 ejemplos.

Luego, para cada ejemplo de una predicción errónea se genera una pregunta que requiera relacionar la dependencia con su cabeza sintáctica. En la Figura 6.2 se presentan las preguntas asociadas a los ejemplos presentados anteriormente.

6.2. Resultados

Se evalúa BETO con el *fine-tune* de SQuAD-es sobre el *dataset* generado, y se obtienen el puntaje F1 y EM para todo el *dataset* y para cada subdivisión de éste según la relación analizada. Los resultados se presentan en la Tabla 6.2.

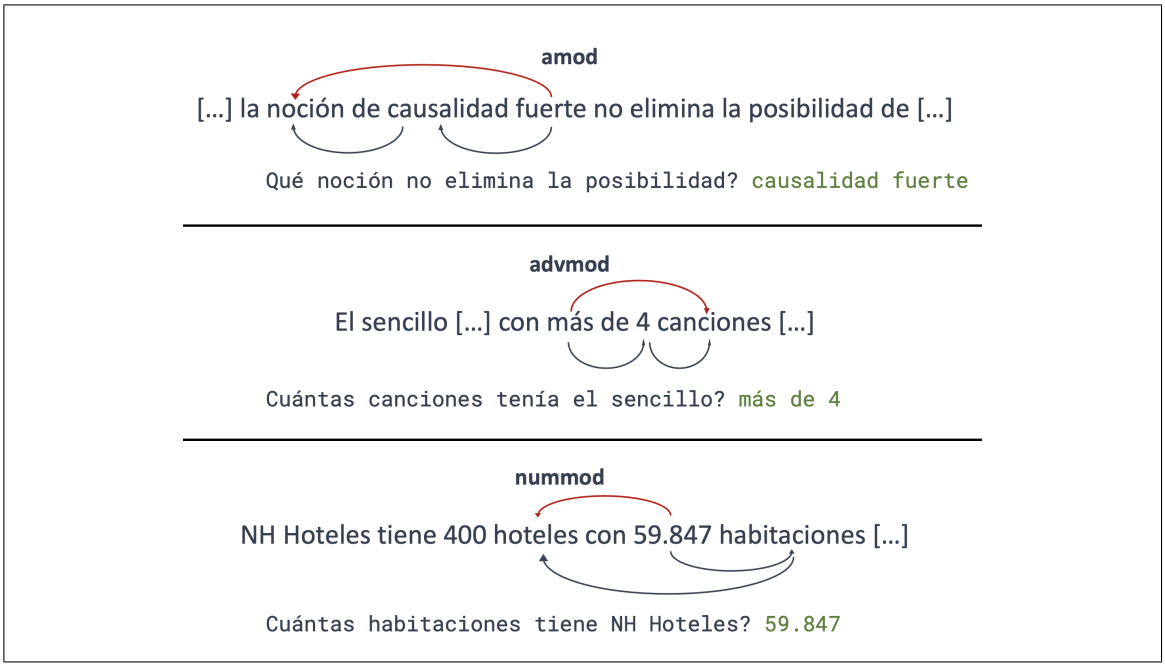


Figura 6.2. Ejemplos de preguntas y respuestas generados para el *dataset*. En todos los ejemplos, es necesario relacionar la dependencia sintáctica con su cabeza para obtener la respuesta correcta. Fuente: Elaboración propia.

Tabla 6.2. Rendimiento de BETO+SQuAD-es sobre el dataset generado, y según la relación analizada. El *dataset* generado se identifica como Mod-QA. Se agrega el rendimiento del mismo modelo en MLQA. Se observa que el rendimiento es mejor en Mod-QA que en MLQA, pero al dividirlo por relación vemos que esto solo ocurre en dos de las tres relaciones.

Dataset	Puntaje F1	EM
Mod-QA	72.84	51.09%
Mod-QA, *amod*	81.72	70.31%
Mod-QA, *advmod*	65.21	37.27%
Mod-QA, *nummod*	84.31	63.64%
MLQA	68.58	48.14%

En los resultados se ve que el modelo obtiene un mejor rendimiento en el *dataset* generado que en MLQA. Así, el modelo no considera las preguntas relacionadas a los modificadores como más difíciles que la pregunta promedio de MLQA, sugiriendo que sí posee la capacidad de relacionar las dependencias con sus cabezas sintácticas.

Sin embargo, al subdividir los resultados, se ve que las preguntas que involucran a la relación *advmod* tienen un rendimiento menor que MLQA en puntaje F1 y una diferencia mucho mayor EM. Luego, las capacidades inferidas de BETO se ven limitadas a las relaciones de *amod* y *nummod*, mostrando una carencia de éstas cuando se trata de *advmod*. La Figura 6.3 muestra algunas respuestas entregadas por el modelo cuando falla la predicción de *advmod*.

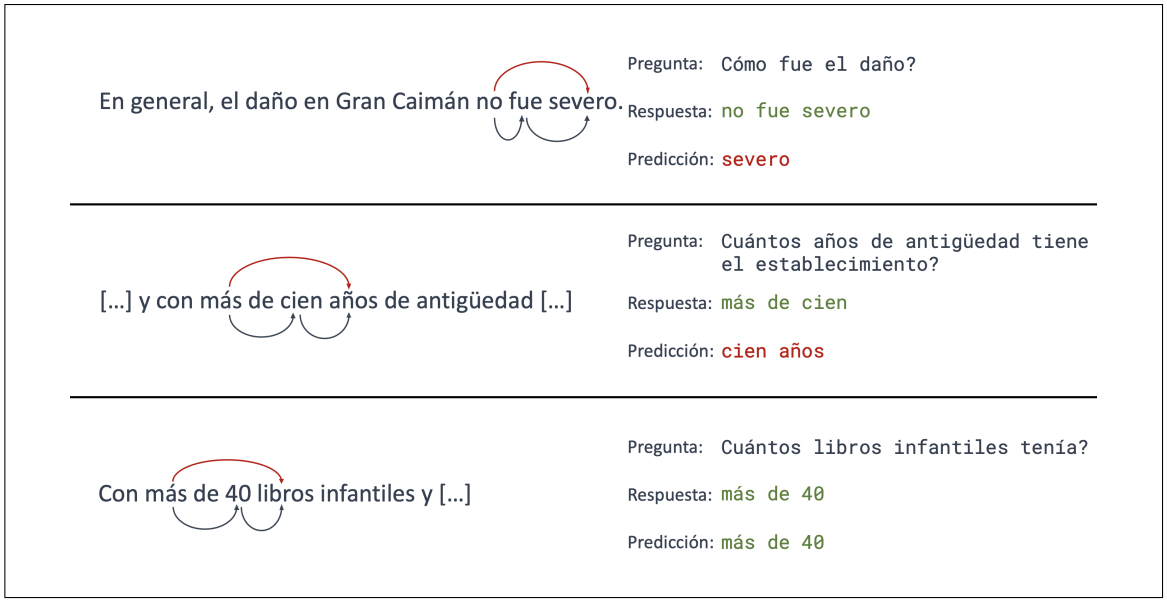


Figura 6.3. Ejemplos de respuestas de BETO+SQuAD-es sobre las preguntas de *advmod*. Suele ocurrir que éstas no incluyen a los modificadores, por lo que se le da un significado distinto a la respuesta. Fuente: Elaboración propia.

6.3. Conclusiones

Los resultados de *amod* y *nummod* sugieren que, aunque las mejores *heads* del modelo fallen en su predicción sintáctica, el modelo aun posee, hasta cierto punto, la capacidad de distinguir cómo interactúan las palabras, y logra comprender el significado de la frase compuesta por ellas. Una posible explicación para esto es que el modelo tenga un desacuerdo con respecto a las convenciones en las anotaciones sintácticas provenientes de GSD-Spanish. Un comportamiento similar es destacado por Clark et al., 2019, donde BERT consideraba que en el inglés, las relaciones *poss* (posesivas) deberían tener como dependencia sintáctica a *'s* en vez de a su complemento, como lo hacen las anotaciones originales. Estos comportamientos resaltan el hecho que las capacidades sintácticas de estos modelos no son aprendidas directamente y de manera supervisada, si no como consecuencia de su aprendizaje en donde el árbol sintáctico está implícito.

Con respecto a la relación *advmod*, los resultados indican que el modelo carece en gran parte el reconocimiento de las interacciones estudiadas. Cabe destacar que estas carencias no solamente significan que la mejor *head* para *advmod* falla en predecir la cabeza sintáctica, si no también que suelen llevar al modelo a obtener una errónea comprensión del texto procesado. Esto muestra la importancia que tiene el desarrollo de una correcta sintaxis dentro de modelos de lenguaje.

7. RESILIENCIA DEL CONOCIMIENTO DISTRIBUIDO

Con respecto a los resultados del *Attention-Only Probe* sobre BETO en el primer análisis, la Figura 3.3 levantó una duda sobre su funcionamiento: Se muestra que los resultados del *probe* separados por relación son muy similares a los resultados de las mejores *heads* para cada relación. Esto deja la duda de si la predicción sintáctica del *probe* se debe a un enfoque único en el comportamiento de cada mejor *head*, o si el conocimiento que llevó a la predicción vino de una cooperación entre distintas *heads* con distintos niveles de capacidades sintácticas, es decir, un conocimiento distribuido.

La pregunta planteada es de gran interés. Se ha visto que ciertas *heads* del modelo tienen altas capacidades sintácticas, sirviendo como explicación para algunos de los comportamientos de BETO. Sin embargo, la presencia de un conocimiento distribuido abriría puertas para estudios más profundos del conocimiento desarrollado, involucrando patrones entre *heads* en vez de dentro de ellas.

Así, en este último experimento se busca investigar las interacciones del conocimiento sintáctico presente en BETO. En particular, se modificará el funcionamiento del *probe* para poder medir el efecto de las mejores *heads* sobre éste, y determinar si el conocimiento sintáctico de peores *heads* en conjunto son capaces de realizar una buena predicción.

7.1. Metodología

El *probe* estudiado en el experimento base busca predecir relaciones sintácticas en base al comportamiento de la *heads* del modelo. Recibe como *input* todos los valores de $\|\alpha f(x)\|$ generados por todas las *heads* de BETO cuando éste procesa una frase, y entrena un conjunto de pesos para poder determinar qué palabras de la frase están relacionadas sintácticamente, sin importar el tipo de relación que presenten. Durante su testeo, se obtiene el rendimiento de reconocer relaciones en general, además del rendimiento separado para cada relación en particular. Para estudiar el efecto del conocimiento de las mejores

heads en el *probe*, es posible eliminar las normas $\|\alpha f(x)\|$ que generan estas *heads*, entrenar el *probe* con los *inputs* filtrados, y medir los cambios en el rendimiento.

Se considera alguna relación sintáctica cualquiera. Cada *head* de BETO tiene una distinta capacidad para reconocer dicha relación. Para poder evaluar el efecto de las mejores *heads*, se ordenan éstas según su rendimiento y se van eliminando acumulativamente del *input* del *probe*. Así, inicialmente se filtran las $\|\alpha f(x)\|$ generadas por la mejor *head* para la relación, se entrena el *probe* con los *inputs* filtrados, y se mide el rendimiento de predecir la relación. Luego, se eliminan además las $\|\alpha f(x)\|$ generadas por la segunda mejor *head*, y se repite el entrenamiento y la evaluación. El proceso se repite hasta que el *input* del *probe* esté vacío.

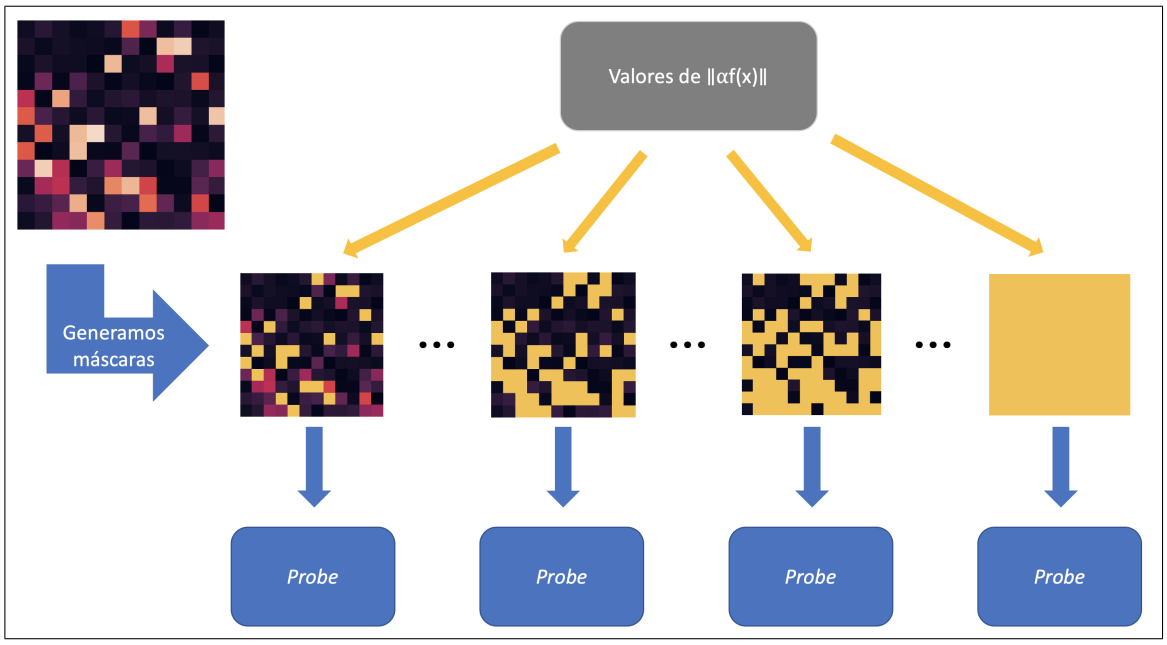


Figura 7.1. Ilustración de la generación de filtros en el *input* del *probe*. En este ejemplo, se ve como el Mapa de Conocimiento de *det* genera los filtros, partiendo por las *heads* con mayor capacidad de reconocimiento, y terminando con un filtro total. Fuente: Elaboración propia.

En la Figura 7.1 se ilustra el proceso recién descrito: el Mapa de Conocimiento sirve como representación del rendimiento de cada *head*. Así, se generan filtros acumulativos

que eliminan las *heads* en el orden de su rendimiento. Los valores de $\|\alpha f(x)\|$ son entonces filtrados para luego entrenar y evaluar el *probe*.

Todas las iteraciones de entrenamiento y evaluación se realizan con los *sets* de entrenamiento y testeo de GSD-Spanish, respectivamente.

7.2. Resultados

Los resultados del experimento están separados por relación, y se muestran en gráficos donde el eje X es la cantidad de *heads* filtradas, y el eje Y es el rendimiento. Cada gráfico incluye:

- El rendimiento del *probe* en base al *input* filtrando las mejores *heads*, en azul. Este valor representa la capacidad sintáctica del modelo con solo las *heads* presentes en el *input*.
- El rendimiento de la mejor *head* dentro del *input*, en verde. Si el modelo no tuviese una capacidad sintáctica distribuida, este valor debería ser siempre mayor o igual al rendimiento del *probe*.
- Además, se agrega una línea horizontal que marca el *baseline*, indicando cuándo una predicción tiene un rendimiento mayor o menor que el *offset* más común.

La Figura 7.2 presenta los resultados para las 6 relaciones más comunes, y el Apéndice H los muestra para todas las relaciones estudiadas.

Para la mayoría de las relaciones se observa un patrón similar: Inicialmente, el resultado del *probe* con todas las *heads* del modelo es muy similar al de la mejor *head*. Sin embargo, a medida que se van eliminando las *heads* del *input*, el rendimiento de la mejor *head* presente decrece rápidamente mientras que el rendimiento del *probe* disminuye de una manera más lenta. Así, se identifica que el alto rendimiento del *probe* no proviene únicamente de las mejores *heads*, si no que después de eliminar éstas, el *probe* sigue siendo capaz de predecir la relación.

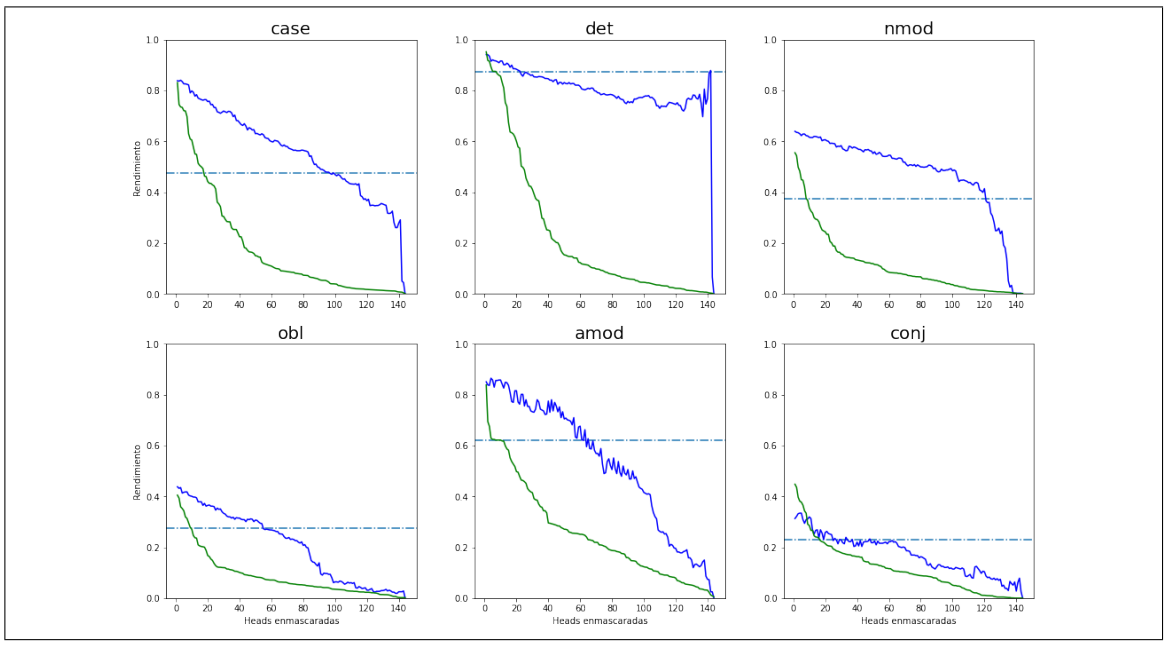


Figura 7.2. Resultados del rendimiento del *probe* con *inputs* filtrados. Para la mayoría de las relaciones, el *probe* tiene un rendimiento constantemente superior a la mejor *head* en el *input*.

El análisis es más interesante cuando se realiza en contraste con el *baseline* del mejor *offset* como referencia. En la Figura 7.3 se ejemplifica esto con los resultados de la relación *nmod*. Se destaca el periodo en donde la mejor *head* del *input* tiene un rendimiento bajo el *baseline*, y el *probe* tiene uno superior.

En el periodo destacado, el *probe* solo está siendo informado por *heads* con un rendimiento inferior al *baseline* y, sin embargo, está logrando rendimiento superior a éste. Con esto, se sugiere que BETO posee un conocimiento sintáctico distribuido por medio de sus *heads*, donde la interacción de las peores *heads* logra un rendimiento similar al de las mejores. En el ejemplo, el conocimiento distribuido de *nmod* es mayor que el *baseline* incluso cuando se filtran las 100 mejores *heads* para la relación. Cabe mencionar que cuando se trata de otras relaciones , este periodo no siempre es tan largo ni la diferencia entre los rendimientos tan grande. Incluso a veces, este periodo no existe. Sin embargo, sí se descubre su presencia en distintos grados para la mayoría de las relaciones.

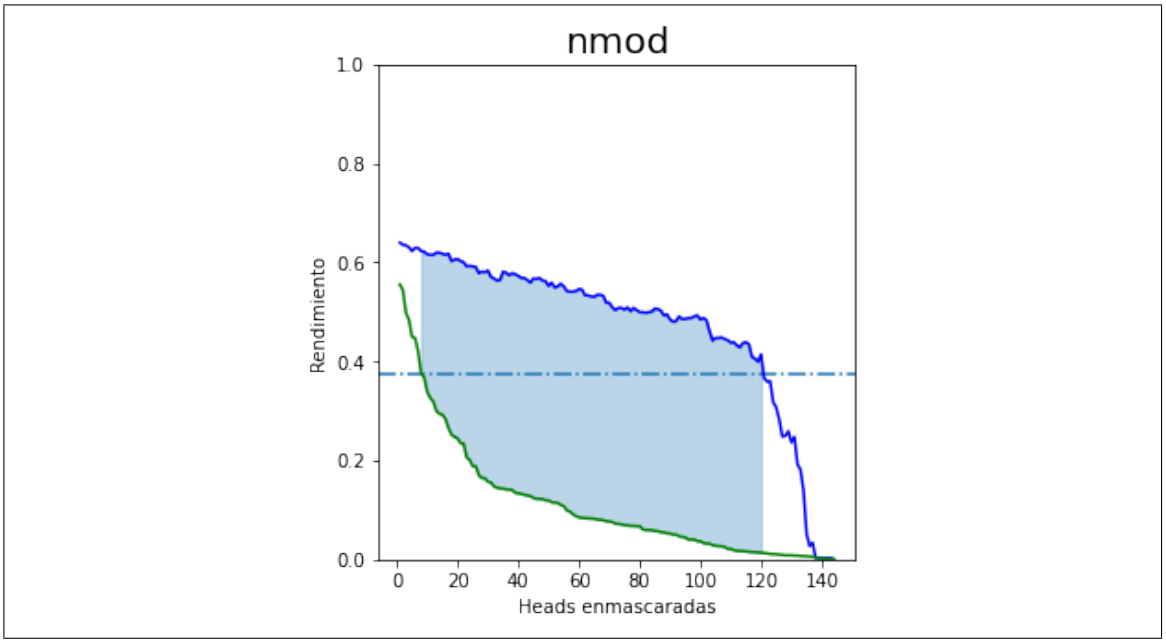


Figura 7.3. Resultados del *probe* filtrado para la relación *nmod*. Durante más de 100 iteraciones el *probe* recibió el *input* de *heads* con conocimiento sintáctico relativamente bajo, y sin embargo logra un rendimiento mayor que el *baseline*.

7.3. Análisis sobre el *probe*

Es importante notar que si el *probe* es suficientemente complejo, parte de su rendimiento podría estar proviniendo no de los coeficientes de $\|\alpha f(x)\|$ si no de algún sesgo en los datos, como siempre predecir las relaciones más comunes o basarse en el largo de la frase. Para verificar que los resultados reportados provengan mayoritariamente de los coeficientes $\|\alpha f(x)\|$, se realiza el mismo proceso del experimento original, pero se reemplazan las normas $\|\alpha f(x)\|$ por valores aleatorios. Luego se mide el rendimiento del *probe* para cada relación. El filtro de las *heads* se realiza en el mismo orden para todas las relaciones ya que, como todos los valores son aleatorios, no existe ninguna *head* mejor que otra.

La Figura 7.4 muestra los resultados del análisis para las 6 relaciones más comunes, junto al *baseline* del *offset* más común para cada una de ellas. En el Apéndice I se encuentran los resultados para todas las relaciones.

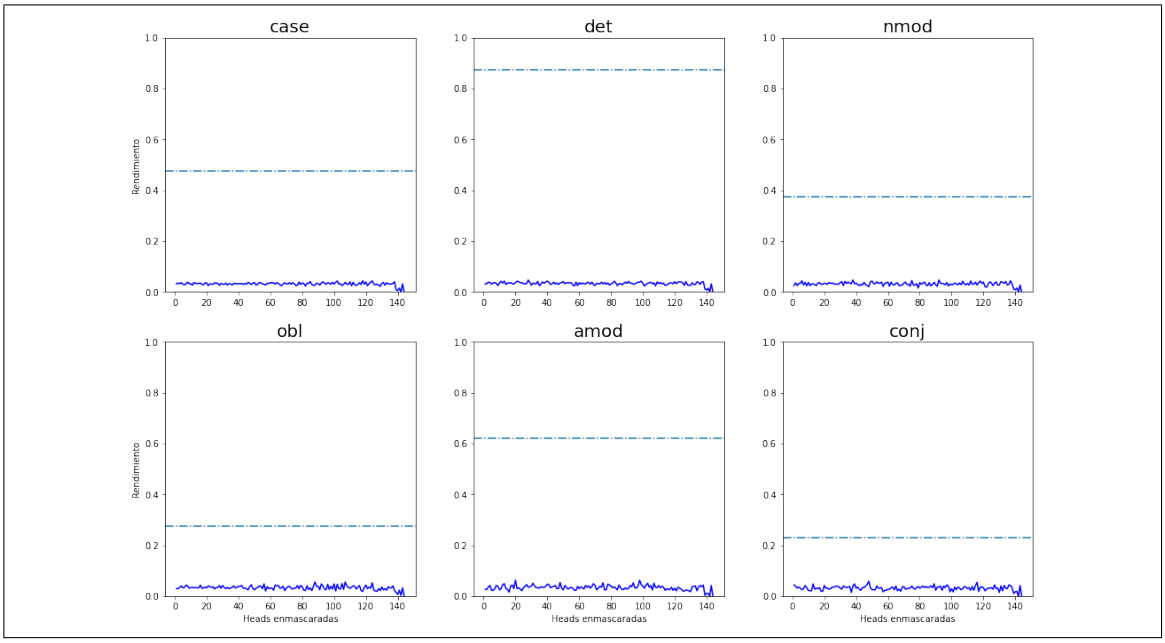


Figura 7.4. Resultados del análisis del *probe* con *inputs* aleatorios. El *probe* no es capaz de reconocer ninguna relación sin los coeficientes de $\|\alpha f(x)\|$.

Para todas las relaciones los resultados son bastante similares: El *probe* no logra reconocer ninguna relación cuando se eliminan los coeficientes de $\|\alpha f(x)\|$, y los resultados no cambian a medida que avanza el filtro en el *input*. Esto sirve como evidencia de que los defectos del *probe* no afectan considerablemente en el rendimiento del experimento original, si no que provienen de la información presente en los coeficientes $\|\alpha f(x)\|$.

7.4. Conclusiones

Los resultados del experimento sugieren que la capacidad sintáctica no se encuentra únicamente contenida en sus *heads*, si no también distribuida entre ella.

Existen distintas variaciones sobre modelos como BERT donde eliminan varias *heads* para obtener un modelo más rápido, manteniendo un rendimiento muy similar al modelo original. La presencia del conocimiento distribuido ayuda a dar una explicación de por qué estas variaciones funcionan: Es posible eliminar la mayoría de las *heads* con capacidades

sintácticas específicas e igual obtener buenos resultados. Así, estos modelos aun pueden poseer un conocimiento sintáctico considerable.

Por último, se presenta una nueva forma de analizar las capacidades sintácticas de BETO. El estudio del conocimiento sintáctico distribuido del modelo podría llevar a una mejor comprensión de su comportamiento, además de atribuirle una interpretación de gran utilidad.

8. CONCLUSIÓN

En la literatura existen estudios que presentan las capacidades sintácticas de modelos como BERT, resultando en una mejor comprensión e interpretabilidad del modelo. Sin embargo, estos avances solo son aplicables al modelo entrenado en inglés, dejando sus beneficios fuera del alcance de la zona hispano-hablante. En esta tesis se aborda esta brecha entre los idiomas, presentándose una investigación del conocimiento sintáctico de BETO, en conjunto a sus capacidades, limitaciones y posible estructura de cómo se presenta.

Se concluye que BETO también posee una comprensión sintáctica de las frases que procesa, e incluso muestra una mayor capacidad que su contraparte en inglés. Se muestra que los valores de $\|\alpha f(x)\|$ son una forma más fructífera de presentar las habilidades lingüísticas aprendidas por el modelo, y posee un mejor rendimiento que solo usar los coeficientes de atención, como lo hace la literatura antes del trabajo de Kobayashi et al., 2020. Además, se observa que los patrones sintácticos no solo están en ciertas *heads* del modelo, si no en el comportamiento de todas al mismo tiempo, y que este comportamiento es similar para relaciones similares. Con respecto a sus limitaciones, se muestra que el principal factor para una predicción sintáctica fallida es que la relación predicha se estructure de una manera poco común, pero otros factores como la frecuencia de las palabras usadas y la presencia de *tokens* [UNK] también afectan. Analizando la comprensión del modelo por medio de preguntas, se ve que para ciertas relaciones el modelo muestra la capacidad de responder correctamente, aun cuando las relaciones involucradas no se reconocen, sugiriendo que BETO posee ciertos desacuerdos en las convenciones de anotaciones sintácticas del *dataset*. En otras relaciones, la falta del contexto sintáctico podría significar una falta de comprensión del texto, y así un menor rendimiento al responder preguntas sobre éste. Por último, se observa que el conocimiento sintáctico del modelo además de estar en cada *head*, se presenta de una manera distribuida entre las *heads*, resultando en un rendimiento mayor que el de sus partes. Así, el modelo muestra

altas habilidades sintácticas incluso cuando se eliminan una gran cantidad de las mejores *heads* para la predicción.

Como trabajo futuro, sería interesante desarrollar un método sistemático de reordenamiento de relaciones sintácticas para su mejor reconocimiento por parte del modelo. Además, se podría estudiar en profundidad las conexiones entre las *heads* que permiten el desarrollo del conocimiento distribuido. Por último, la universalidad de las anotaciones entregadas en el *dataset* permite replicar este estudio en múltiples idiomas. Así, se podría realizar una búsqueda de patrones repetidos globalmente que nos den una idea de qué se necesita para entender algún idioma cualquiera.

Se espera que estos resultados sean útiles para darle una mayor comprensión a modelos de lenguaje como BETO, y que esto a la vez permita el desarrollo de tecnologías de inteligencia artificial más avanzadas y seguras para el español.

REFERENCIAS

Bahdanau, D., Cho, K., & Bengio, Y. (2016). *Neural machine translation by jointly learning to align and translate.* Retrieved from https://arxiv.org/abs/1409.0473

Bhardwaj, R., Majumder, N., & Poria, S. (2020). *Investigating gender bias in BERT.* Retrieved from https://arxiv.org/abs/2009.05021

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). *Language models are few-shot learners.* Retrieved from https://arxiv.org/abs/2005.14165

Carrino, C. P., Costa-jussà, M. R., & Fonollosa, J. A. R. (2019). *Automatic spanish translation of the squad dataset for multilingual question answering.* Retrieved from https://arxiv.org/abs/1912.05200

Cañete, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H., & Pérez, J. (2020). Spanish pre-trained BERT model and evaluation data. In *PML4DC at ICLR 2020.*

Clark, K., Khandelwal, U., Levy, O., & Manning, C. D. (2019). *What does BERT look at? an analysis of BERT's attention.* Retrieved from https://arxiv.org/abs/1906.04341

de Marneffe, M.-C., & Manning, C. D. (2008). The Stanford typed dependencies representation. *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, 1-8.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*.

de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., & Nissim, M. (2019). *BERTje: A dutch BERT model.* Retrieved from https://arxiv.org/abs/1912.09582

Ettinger, A. (2019). What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *CoRR*, *abs/1907.13528*. Retrieved from http://arxiv.org/abs/1907.13528

Hao, Y., Dong, L., Wei, F., & Xu, K. (2020). *Self-attention attribution: Interpreting information interactions inside transformer.* Retrieved from https://arxiv.org/abs/2004.11207

Hoover, B., Strobelt, H., & Gehrmann, S. (2019). *exBERT: A visual analysis tool to explore learned representations in transformers models.* Retrieved from https://arxiv.org/abs/1910.05276

Kobayashi, G., Kuribayashi, T., Yokoi, S., & Inui, K. (2020). *Attention is not only a weight: Analyzing transformers with vector norms.* Retrieved from https://arxiv.org/abs/2004.10102

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, *29*, 1-27. doi: https://doi.org/10.1007/BF02289565

Lewis, P., Oğuz, B., Rinott, R., Riedel, S., & Schwenk, H. (2019). MLQA: Evaluating cross-lingual extractive question answering. Retrieved from https://arxiv.org/abs/1910.07475

Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*, 129-137. doi: 10.1109/TIT.1982.1056489

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, *19*(2), 313–330.

Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., de la Clergerie, E., ... Sagot, B. (2020). CamemBERT: a tasty french language model. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., ... Lee, J. (2013). Universal dependency annotation for multilingual parsing. *Proceedings of ACL 2013*.

Nayak, P. (2019). Understanding searches better than ever before. Retrieved from `https://blog.google/products/search/search-language-understanding-bert/`

Parliament and Council of the European Union. (2016). *General data protection regulation (GDPR)*. Retrieved from `https://eur-lex.europa.eu/eli/reg/2016/679/oj`

Radford, A., Narasimhan, K., Salimans, T., & Suutskever, I. (2018). Improving language understanding by generative pre-training. Retrieved from `https://openai.com/blog/language-unsupervised/`

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. Retrieved from `https://openai.com/blog/better-language-models/`

Rajpurkar, P., Jia, R., & Liang, P. (2018). *Know what you don't know: Unanswerable questions for SQuAD.*

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *SQuAD: 100,000+ questions for machine comprehension of text.*

Taylor, W. L. (1953). Cloze procedure: A new tool for measuring readability. *Journalism Bulletin*, *30(4)*, 415-433.

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. In *Proceedings of the eighth international conference on language resources and evaluation (LREC'12)* (pp. 2214–2218).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). *Attention is all you need.*

Williams, A., Nangia, N., & Bowman, S. R. (2018). *A broad-coverage challenge corpus for sentence understanding through inference.*

Wu, S., & Dredze, M. (2019). *Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT.*

Yang, Y., Zhang, Y., Tar, C., & Baldridge, J. (2019). *PAWS-X: A cross-lingual adversarial dataset for paraphrase identification.*

ANEXOS

A. LISTA DE RELACIONES SINTÁCTICAS ANALIZADAS

Relación
case
det
nmod
obl
amod
conj
nsubj
cc
obj
advmod
mark
appos
flat
iobj
nummod
advcl
cop
acl:relcl
aux
fixed
acl
aux:pass

Relación
xcomp
parataxis
ccomp
dep

B. RENDIMIENTO DE LAS MEJORES *HEADS* DE BETO PARA CADA RELACIÓN

Relación	$\|\alpha f(x)\|$	Atención	Offset	
case	**83.14** (7-5)	81.33 (7-5)	47.53 (2)	5853
det	**95.22** (7-5)	95.15 (7-5)	87.53 (1)	5253
nmod	**55.54** (8-6)	53.31 (8-6)	37.53 (-2)	3403
obl	**40.46** (4-1)	38.98 (4-1)	27.38 (-3)	2155
amod	**83.80** (6-12)	82.83 (6-12)	62.08 (-1)	1957
conj	**44.76** (8-7)	44.42 (8-7)	23.13 (-2)	1461
nsubj	**47.03** (6-2)	44.93 (6-2)	20.41 (1)	1382
cc	**66.10** (8-1)	65.57 (8-1)	46.89 (1)	1124
obj	**76.53** (8-6)	74.37 (8-6)	51.26 (-2)	1108
advmod	**58.32** (6-12)	58.22 (6-12)	46.23 (1)	967
mark	69.76 (11-8)	**69.86** (11-8)	49.74 (1)	949
appos	**52.59** (11-7)	51.30 (11-7)	35.23 (-1)	772
flat	**74.32** (8-2)	74.32 (8-2)	73.72 (-1)	666
iobj	**71.94** (1-7)	71.01 (1-7)	65.12 (1)	645
nummod	76.45 (6-12)	**77.39** (6-12)	75.04 (1)	637
advcl	**31.33** (4-1)	31.01 (4-1)	22.89 (-2)	616
cop	**78.53** (6-12)	77.66 (6-12)	41.71 (1)	573
acl:relcl	**46.08** (7-3)	41.09 (7-3)	23.04 (-3)	421
aux	88.81 (6-12)	**89.89** (6-12)	76.17 (1)	277
fixed	**91.95** (2-12)	91.95 (2-12)	90.42 (-1)	261
acl	**58.91** (11-7)	57.36 (6-12)	39.53 (-1)	258
aux:pass	**95.18** (7-4)	95.18 (7-4)	87.95 (1)	166

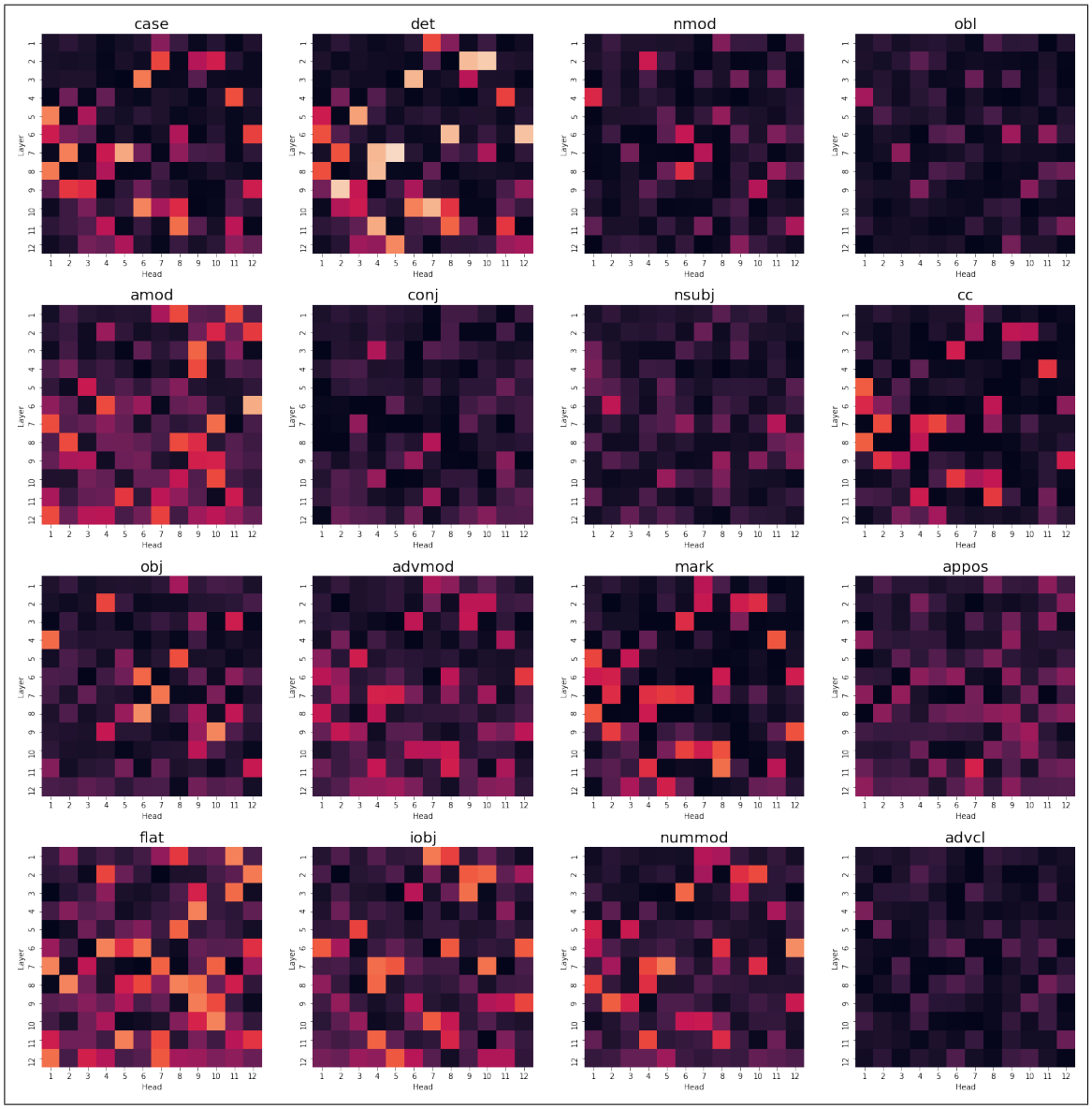
Relación	$\|\alpha f(x)\|$	Atención	Offset	
xcomp	**80.25** (2-4)	80.25 (6-6)	59.24 (-1)	157
parataxis	**21.17** (5-2)	21.17 (5-2)	6.57 (-2)	137
ccomp	**60.66** (6-11)	60.66 (6-11)	10.66 (-3)	122
dep	**56.00** (11-5)	55.00 (11-5)	48.00 (-1)	100

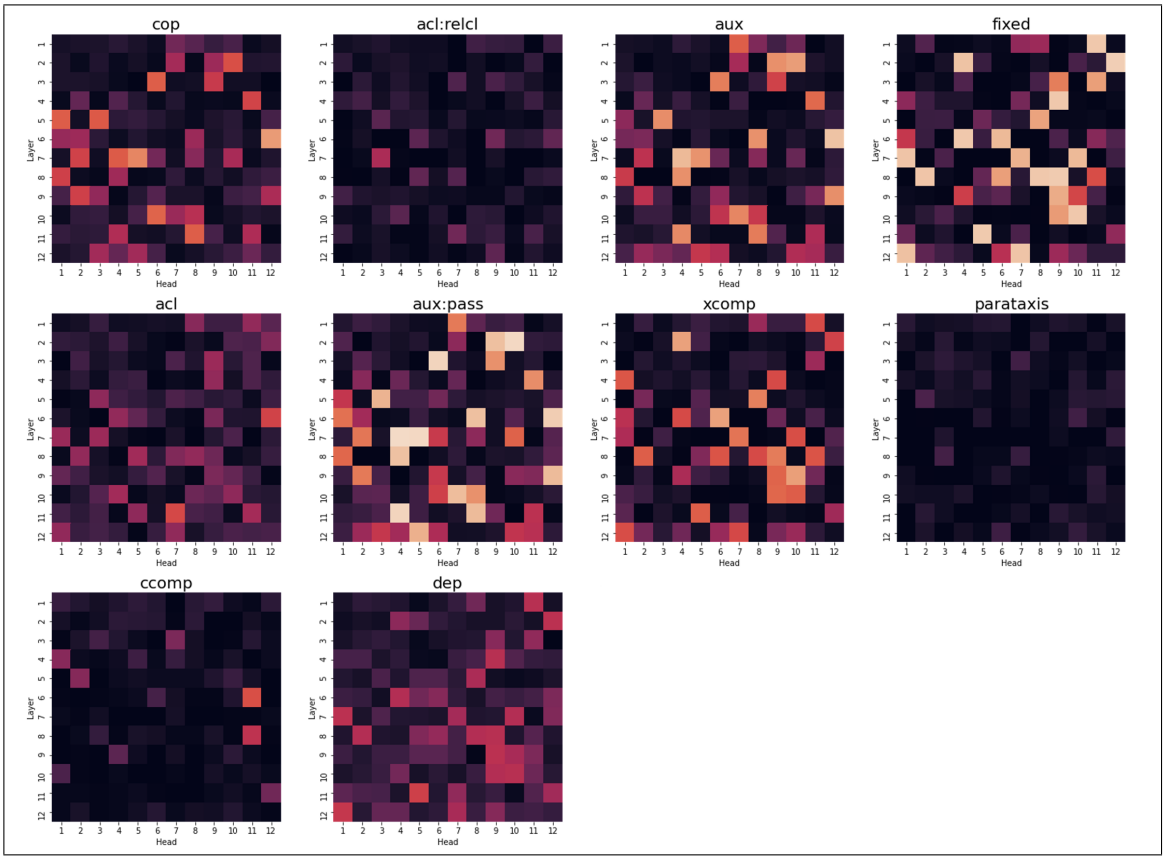
C. RESULTADOS DEL *PROBE* SEPARADOS SEGÚN RELACIÓN

Relación	$\|\alpha f(x)\|$	Atención
case	**85.17**	83.34
det	94.46	**94.70**
nmod	64.29	**64.40**
obl	**44.70**	42.99
amod	85.39	**85.74**
conj	32.98	**33.40**
nsubj	**50.11**	46.14
cc	**65.75**	63.56
obj	70.23	**75.12**
advmod	**68.52**	68.52
mark	**68.41**	67.05
appos	**48.09**	45.36
flat	65.32	**67.05**
iobj	**82.67**	82.67
nummod	79.19	**81.50**
advcl	**31.91**	27.23
cop	**70.37**	70.37
acl:relcl	**29.35**	27.72
aux	**87.66**	87.66
fixed	**15.32**	13.71
acl	42.59	**48.15**
aux:pass	**100.00**	100.00

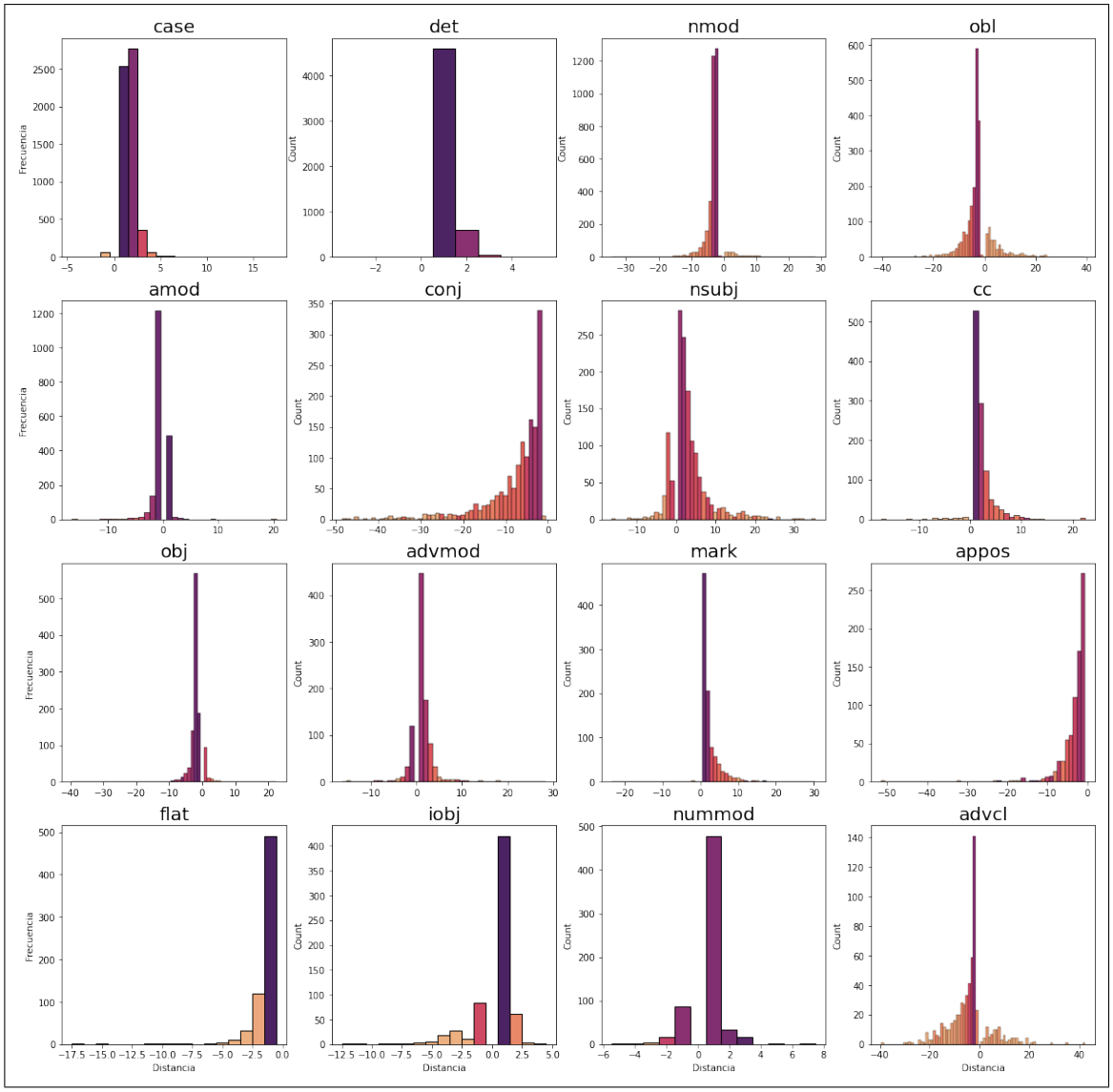
Relación	$\|\alpha f(x)\|$	Atención
xcomp	58.70	**61.96**
parataxis	**6.06**	4.55
ccomp	**28.21**	26.92
dep	**65.00**	65.00

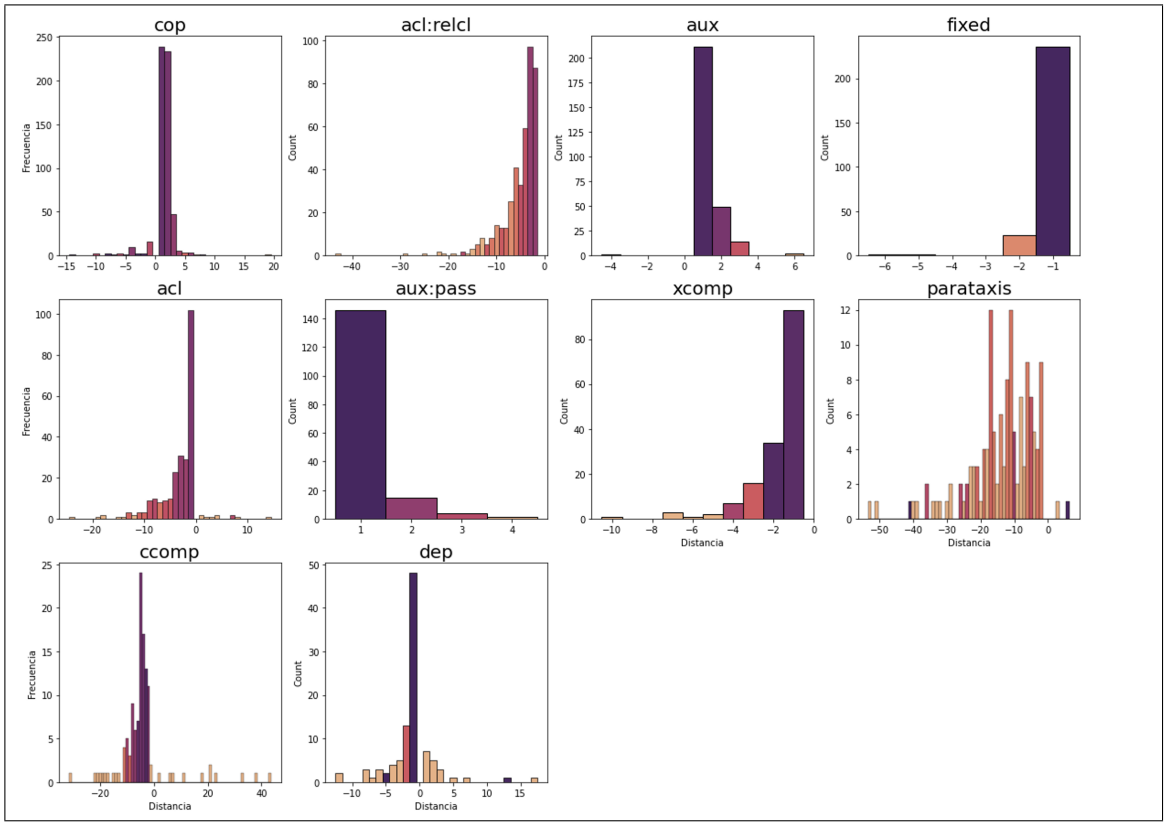
D. MAPAS DE CONOCIMIENTO DE TODAS LAS RELACIONES




cop
acl:relcl
aux
fixed
acl
aux:pass
xcomp
parataxis
ccomp
dep
Layer
Head

E. RENDIMIENTO PARA CADA RELACIÓN SEGÚN SU DISTANCIA




cop
acl:relcl
aux
fixed
acl
aux:pass
xcomp
parataxis
ccomp
dep
Frecuencia
Count
Distancia

F. RENDIMIENTO DE LA PREDICCIÓN DE RELACIONES SEGÚN EL TIPO DE DISTANCIA

Relación	General	Distancias comunes	Distancias no comunes
case	83.14	83.58	6.06
det	95.22	95.35	22.22
nmod	55.54	60.34	7.44
obl	40.46	56.87	8.83
amod	83.80	84.66	53.70
conj	44.76	58.49	24.14
nsubj	47.03	56.22	28.45
cc	66.10	70.23	21.88
obj	76.53	78.83	48.19
advmod	58.32	62.70	23.85
mark	69.76	74.65	26.80
appos	52.59	54.94	37.50
flat	74.32	75.23	0.00
iobj	71.94	75.94	0.00
nummod	76.45	76.67	57.14
advcl	31.33	56.62	7.01
cop	78.53	79.12	66.67
acl:relcl	46.08	54.57	20.19
aux	88.81	88.81	0.00
fixed	91.95	91.95	0.00
acl	58.91	68.21	30.16
aux:pass	95.18	95.18	0.00

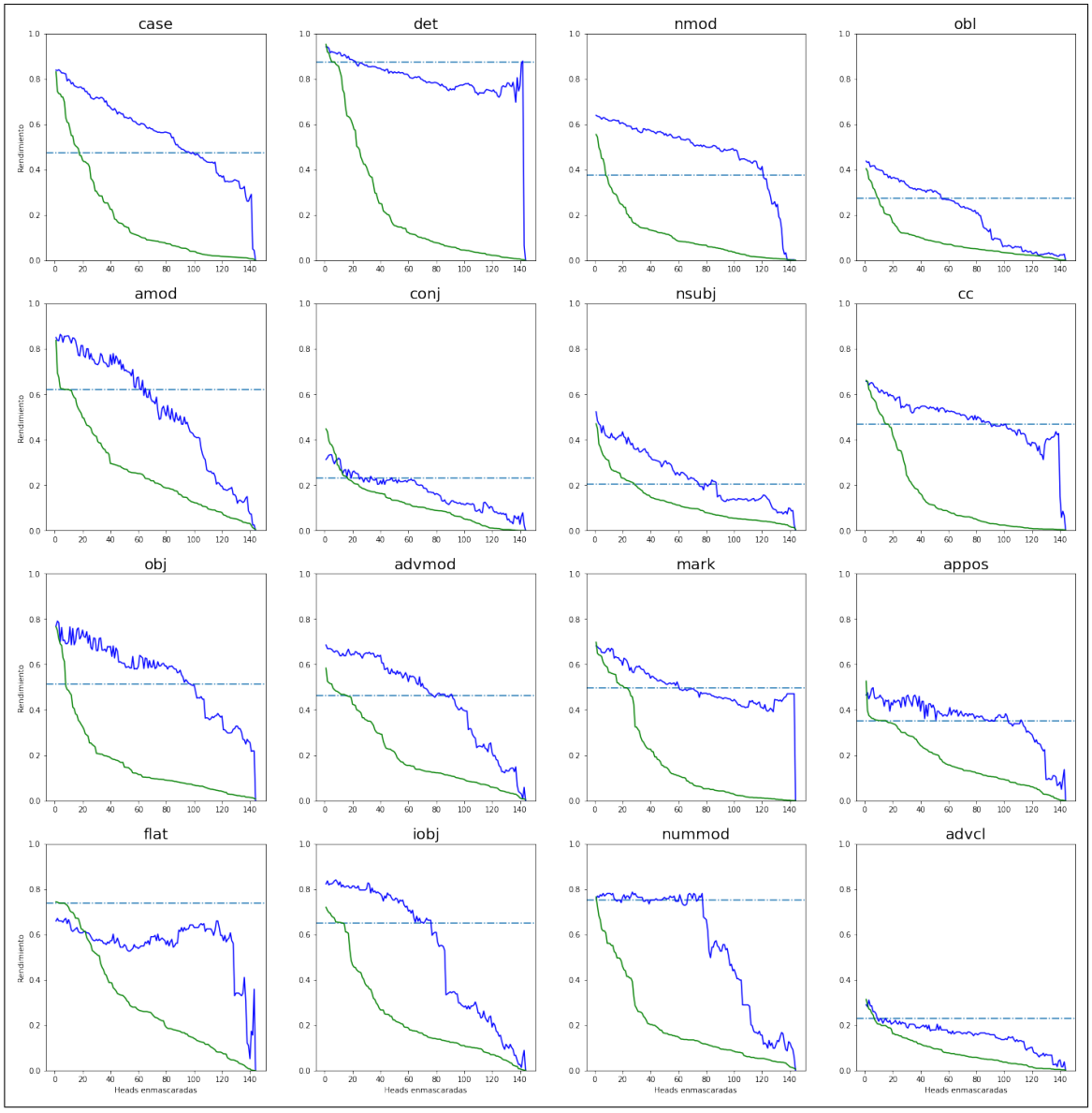
Relación	General	Distancias comunes	Distancias no comunes
xcomp	80.25	82.35	0.00
parataxis	21.17	26.00	18.39
ccomp	60.66	79.73	31.25
dep	56.00	67.95	13.64

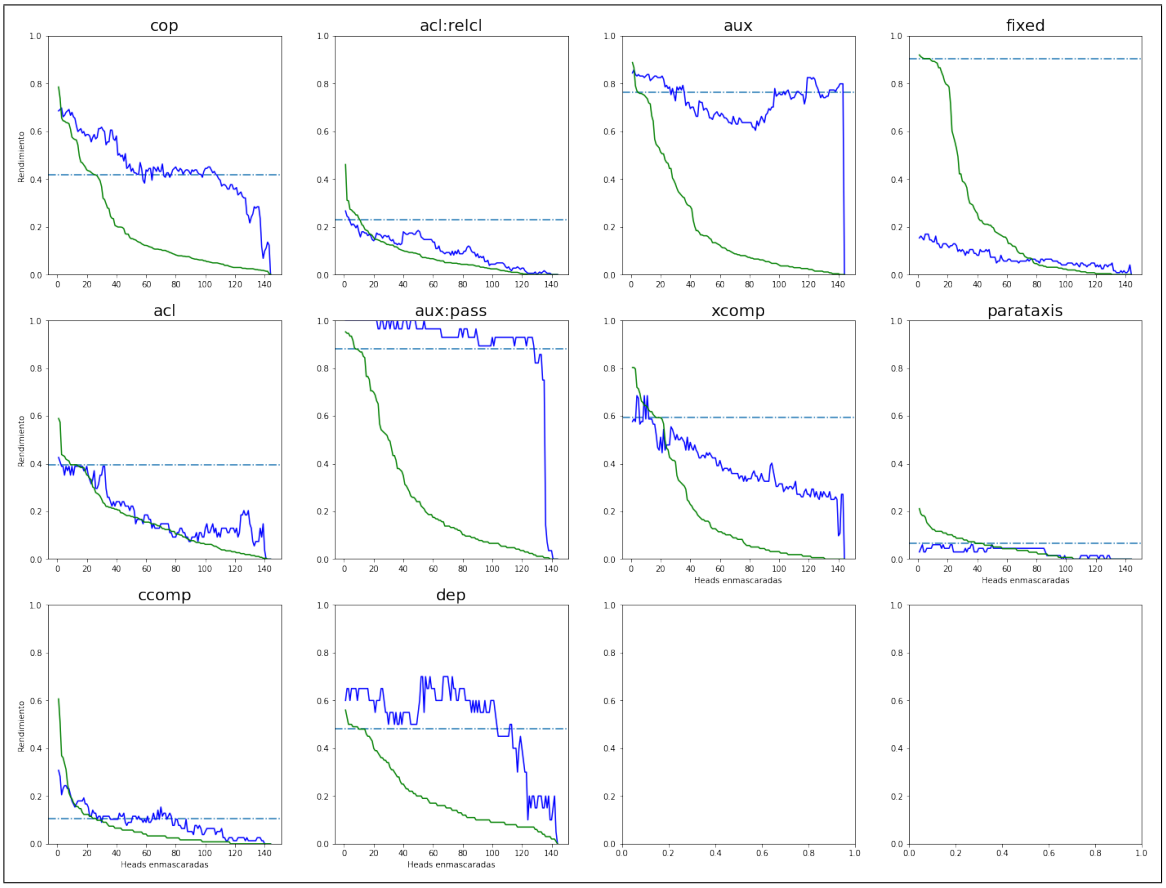
G. PORCENTAJE DE PREDICCIONES INCORRECTAS DIRIGIDAS AL ABUELO SINTÁCTICO

Relación	Predicciones incorrectas dirigidas al abuelo sintáctico
case	4.58%
det	5.18%
nmod	6.08%
obl	5.77%
amod	20.19%
conj	6.32%
nsubj	0.41%
cc	0.00%
obj	6.92%
advmod	27.30%
mark	3.83%
appos	9.56%
flat	0.00%
iobj	2.21%
nummod	36.67%
advcl	5.44%
cop	4.88%
acl:relcl	5.73%
aux	0.00%
fixed	0.00%
acl	6.60%
aux:pass	0.00%

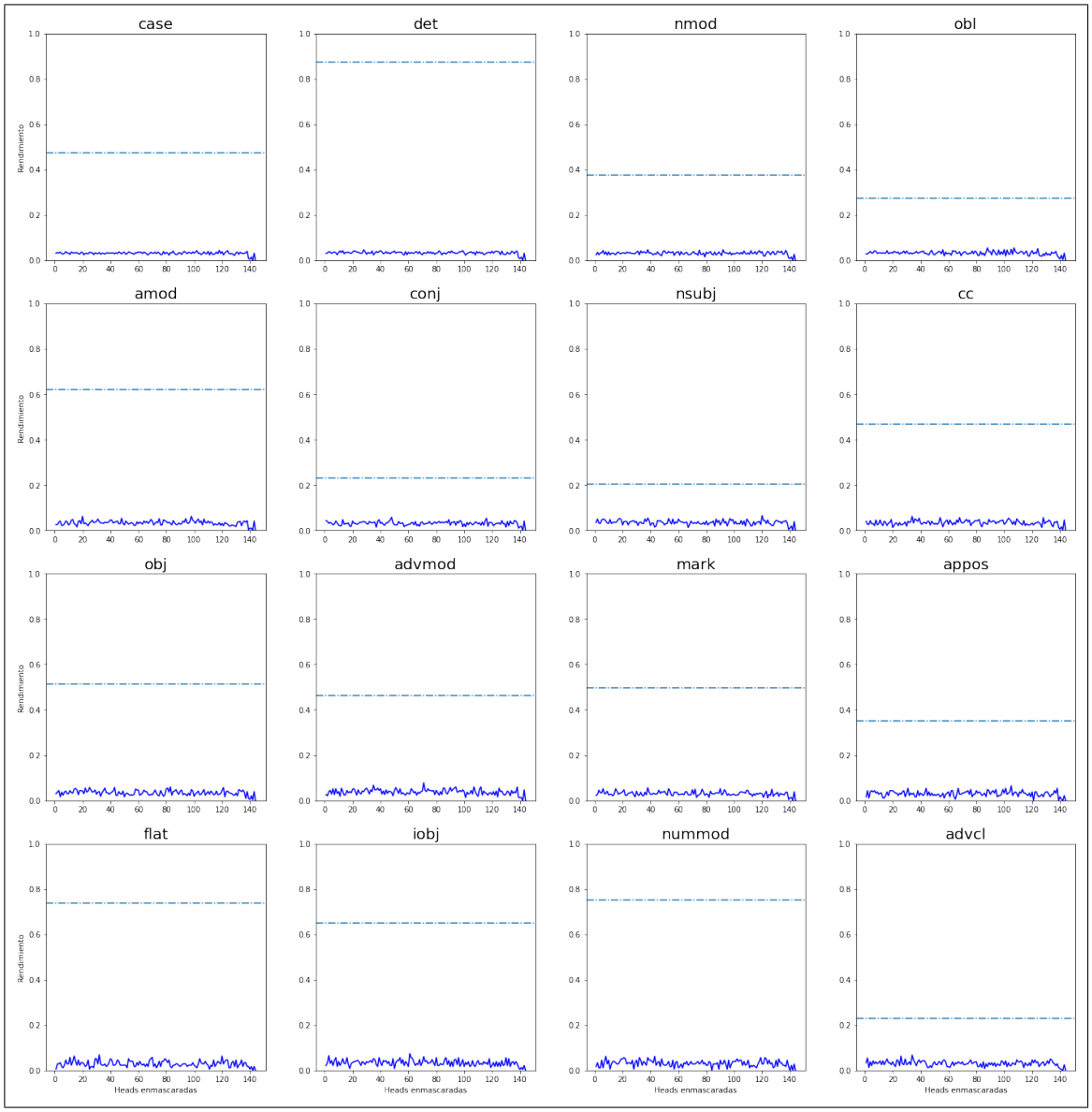
Relación	Predicciones incorrectas dirigidas al abuelo sintáctico
xcomp	3.23%
parataxis	1.85%
ccomp	6.25%
dep	2.27%

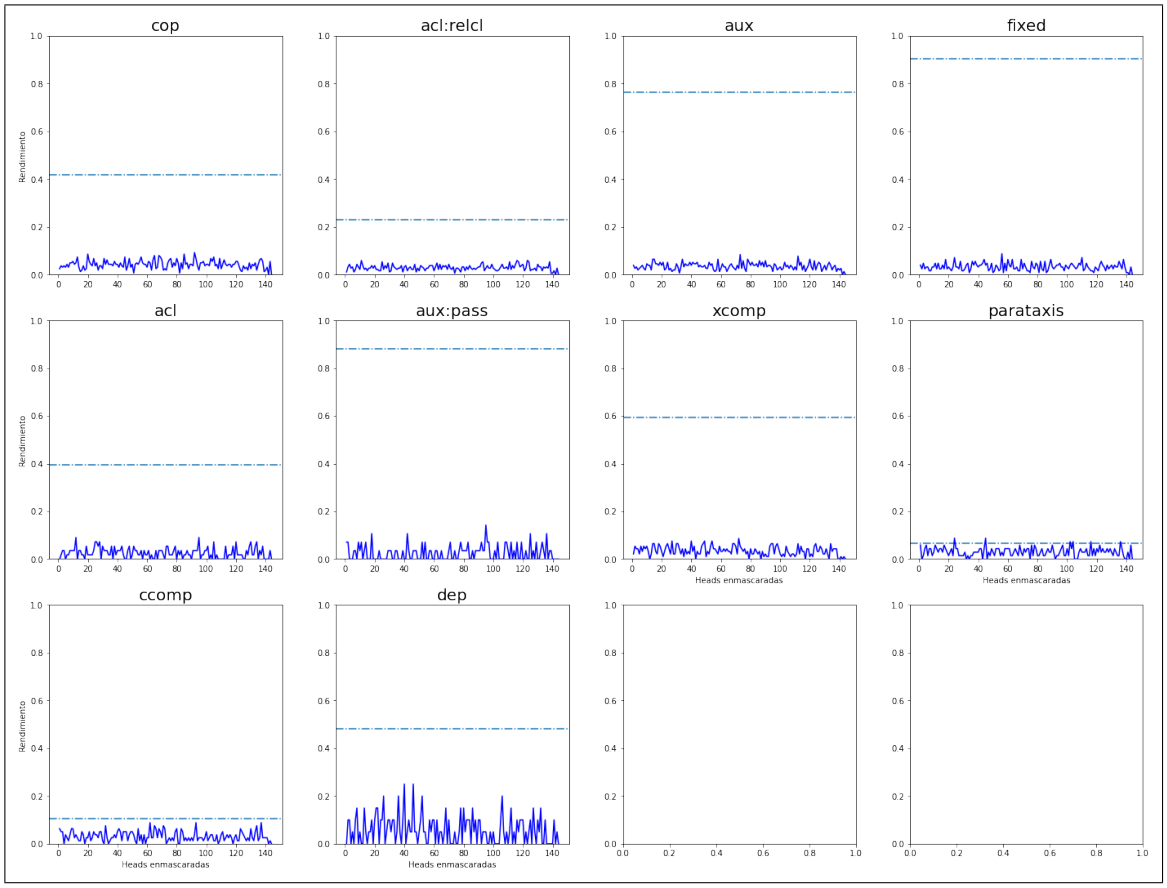
H. RESILIENCIA DEL CONOCIMIENTO DISTRIBUIDO SEGÚN RELACIÓN




cop
acl:relcl
aux
fixed
acl
aux:pass
xcomp
parataxis
ccomp
dep
Rendimiento
Heads enmascaradas

I. RENDIMIENTO DEL EXPERIMENTO DE CONOCIMIENTO DISTRIBUIDO CON VALORES ALEATORIOS




cop
acl:relcl
aux
fixed
acl
aux:pass
xcomp
parataxis
ccomp
dep
Rendimiento
Heads enmascaradas