DeepText

Nueva generación de modelos neuronales de inteligencia artificial para transformar las tecnologías de la lengua en la industria del País Vasco

Entregable E2.2: Nuevos modelos de lenguaje monolingües para el castellano y euskara basadas en las últimas arquitecturas neuronales

Responsable	IXA
Tipo	Recurso
Ejercicio	2020

ELKARTEK 2020
Proyectos de investigación fundamental colaborativa
DeepText – Entregable E2.2

Índice

ELKARTEK 2020
Proyectos de investigación fundamental colaborativa

DeepText – Entregable E2.2

Introducción

Este entregable describe los modelos entrenados dentro de la tarea T2.2 (Entrenamiento de modelos de lenguaje contextuales monolingües con corpus de PT1) del proyecto DeepText. Estos modelos entrenados con los corpus recogidos en el paquete de trabajo 1, han supuesto un avance en el estado del arte para el euskera y el castellano y suponen un hito importante para el desarrollo de aplicaciones basadas en redes neuronales para el euskera y castellano. Avanzando así además, en el estado del arte de las aplicaciones de Procesamiento de Lenguaje Natural (PLN) para estas lenguas.

Descripción

Se han entrenado modelos monolingües en euskera y castellano con parte del corpus de PT1. A continuación se detallan los experimentos llevados a cabo en ambos idiomas: para euskera se han creado tres modelos y en el caso del castellano dos.

Modelos para el euskera

El corpus utilizado en todos los modelos en euskera ha sido el mismo:

	Número de tokens
Wikipedia	35 M
Berria	81 M
EITB	28 M
Argia	16 M
Armiarma	2 M
Web Elhuyar	62.6 M
	224.6 M

Tabla 1: Corpus de entrenamiento para euskera

El corpus comprende artículos de noticias de periódicos en euskera y Wikipedia. El corpus de entrenamiento, contiene 224.6 millones de tokens, de los cuales 35 millones provienen de Wikipedia. El entregable E1.1 (Corpus monolingües de gran tamaño para euskera y castellano para los cinco dominios especificados) describe con detalle las características de estos corpus, así como ofrece los enlaces para poder acceder a los mismos.

Hemos entrenado un modelo basado en la arquitectura BERT (Devlin et al., 2019) para euskera utilizando el corpus arriba presentado, ya que la representación del euskera en el modelo BERT multilíngüe original de Google (Devlin et al., 2019) tiene carencias como se describe en (Agerri et al, 2020). Las principales diferencias entre nuestro modelo y la implementación original son el corpus utilizado para el preentrenamiento, el algoritmo para la creación del vocabulario de subtokens y el uso de una estrategia de enmascaramiento diferente. Hemos llamado a este modelo **BERTeus**.

El modelo recibe como entrada la concatenación de dos frases del corpus, en el 50% de las veces la concatenación la forman dos frases aleatorias y en el 50% restante de las veces, la segunda frase es la subsecuente a la primera. El preentrenamiento consta de dos tareas que se aprenden a la vez: Por una parte se enmascaran al azar el 20% de las palabras de entrada y el modelo debe predecirlas (*Masked Language Model*). Por otra parte, el modelo debe adivinar si las frases introducidas son subsecuentes (*Next Sentence Prediction*).

Las frases no se representan como secuencias de palabras, sino como secuencias de subtokens. Para la creación del vocabulario correspondiente a dichos subtokens se ha utilizado el algoritmo *Sentence-Piece*, que divide las palabras en los subtokens mas frecuentes del corpus. Durante el entrenamiento, y como nuestro vocabulario consiste en unidades de subtokens, utilizamos *Whole Word Masking*, que aplica el enmascaramiento a palabras completas, haciendo así la tarea de *Masked Language Model* más difícil pero obteniendo beneficios sustanciales. Al aplicar esta estrategia, si el subtoken enmascarado es parte de una palabra, se enmascara la palabra completa.

De este modo, se ha creado un vocabulario de subtokens que contiene 50.000 elementos. Además, como se puede observar en la tabla 2, en el entrenamiento, se ha utilizado Adam con una tasa de aprendizaje (*learning rate*) de 1e-4 y 1.000.000 de actualizaciones: el 90% de las actualizaciones con lotes (*batch*) de 256 y secuencia de 128 y el 10% restante con lotes de 256 y secuencia de 512.

Modelo	**Tasa de aprendizaje**	**Actualizaciones**	**Lotes**	**Secuencias**
BERTeus	1e-4	1.000.000	256 (90%)	128 (90%)
			256 (10%)	512 (10%)
RoBERTeus v1	3e-4	40,000	2048	512
RoBERTeus v2	3e-4	150,000	2048	512

Tabla 2: Datos de entrenamiento de los modelos para euskera

Además del modelo basado en BERT, hemos entrenado dos modelos basados en la arquitectura RoBERTa (Liu et al., 2019) para euskera utilizando el corpus arriba presentado[1].

1En este caso no se ha utilizado los textos de Argia por problemas de codificación.

Hemos llamado a estos modelos RoBERTeus v1 y v2. RoBERTeus repite el procedimiento de preentrenamiento de BERT e incluye un mayor tiempo de entrenamiento del modelo, con lotes más grandes y sobre más datos. Además, siguiendo las indicaciones del artículo original, se ha eliminando el objetivo de predicción de la siguiente oración. Por otra parte, éste modelo se ha entrenado en secuencias más largas y se ha cambiado dinámicamente el patrón de enmascaramiento aplicado a los datos de entrenamiento. El primer modelo entrenado siguiendo estas pautas contiene 40.000 actualizaciones de parametros, con el objetivo de replicar el modelo BERTeus. En este caso se ha preprocesado el corpus con el tokenizador *Byte-Pair Encoding* (*BPE*) (Cased - Manteniendo las mayúsculas y minusculas) y con un vocabulario de 50,000 subtokens. En el entrenamiento, se ha utilizado Adam con una tasa de aprendizaje de 3e-4 y 40.000 actualizaciones: las actualizaciones con lotes 2048 y secuencia de 512.

El segundo modelo entrenado contiene 150.000 actualizaciones con el objetivo de obtener un modelo similar al presentado en Liu et al, (2019). Así, la única diferencia respecto al modelo con 40,000 actualizaciones, son los detalles de entrenamiento. Se ha utilizado Adam con una tasa de aprendizaje de 3e-4, *Whole Word Masking* y 150.000 actualizaciones con lotes de 2048 y secuencias de 512.

Todos los modelos se han probado en cuatro tareas PLN básicas para el euskera: etiquetado gramatical (POS), reconocimiento de entidades nombradas (NER), análisis de sentimientos y clasificación de temas. El etiquetado gramatical es el proceso de asignar a cada una de las palabras de un texto su categoría gramatical. El reconocimiento de entidades nombradas busca clasificar en categorías predefinidas, como personas, organizaciones, lugares, expresiones de tiempo y cantidades, las entidades nombradas encontradas en un texto. El análisis de sentimiento es una tarea de clasificación en función de la connotación positiva, negativa o incluso neutra del lenguaje en un texto. La clasificación de temas consiste en la identificación del tema de un texto dados unos temas predefinidos.

En aquellos casos en los que ha sido posible, las tareas se han evaluado con conjuntos de datos existentes y previamente utilizados. Es el caso del etiquetado gramatical y el reconocimiento de entidades nombradas. El POS se ha evaluado con los datos de Universal Dependencies 1.2. Los datos para el euskera (Aranzabe et al., 2015) se basan en una conversión de una parte del Basque Dependecy Treebank (BDT) (Aduriz et al., 2003). Para evaluar las entidades nombradas, se ha utilizado el gold-standard EIEC[2] , un corpus para NER en euskera (Alegria et al., 2006). El corpus contiene 44.000 tokens para entrenamiento (3817 entidades únicas) y 15000 para test (931 entidades), con cuatro tipos de entidades (Ubicación, Persona, Organización y Varios (Miscelaneous)),

2 http://ixa2.si.ehu.eus/eiec/eiec_v1.0.tgz

Para evaluar la tarea de clasificación de temas se ha recopilado un conjunto de datos que contiene 12.000 titulares de noticias de Argia clasificados según 12 categorías temáticas. El dataset está disponible públicamente bajo licencia CC[3]. La clasificación de sentimientos se evalúa utilizando un corpus de tweets que contienen mensajes relacionados con el dominio cultural. El corpus contiene anotaciones para tres clases (positivas, negativas y neutras) y un total de 2.936 ejemplos[4]. El modelo que ha obtenido los mejores resultados es BERTeus, mejorando el estado del arte para todas las tareas. El resumen de resultados se puede consultar en la tabla 3:

Tarea	**BERTeus**	**mBERT**	**Estado del arte previo**
Clasificación de temas	**76.77**	68.42	63.00
Análisis de sentimientos	**78.10**	71.02	74.02
POS	**97.76**	96.37	96.10
NER	**87.06**	81.52	76.72

Tabla 3: Resultados para POS, NER, análisis de sentimientos y clasificación de temas para el euskera

Mostramos que, en el caso del euskera, los modelos disponibles públicamente pueden ser superados por modelos análogos entrenados con corpus apropiados. Mejoramos el estado del arte en cuatro tareas para el euskera. En el caso de BERT, mostramos que nuestro modelo BERTeus supera al BERT multilingüe oficial con un margen amplio, con una mejora de hasta 10 puntos absolutos en las cuatro tareas. Nuestros experimentos se han realizado con modelos de entrenamiento análogos a los correspondientes modelos pre-entrenados existentes. Esto permite tener modelos comparables y concluir que el corpus de entrenamiento es clave. En el caso de BERT, entrenar un modelo monolingüe en lugar de uno multilingüe compartido con otros 103 idiomas es un factor clave. BERTeus ha sido publicado en el artículo “Give your Text Representation Models some Love: the Case for Basque” (Agerri et al, 2020) y se puede descargar desde la página de huggingface (https://huggingface.co/ixa-ehu/berteus-base-cased).

3 https://hizkuntzateknologiak.elhuyar.eus/assets/files/BHTC.tgz
4 https://hizkuntzateknologiak.elhuyar.eus/es/recursos

Modelos para el castellano

Para el castellano, hemos seguido las mismas pautas para crear los modelos. Por un lado, hemos entrenado un modelo basado en la arquitectura BERT (IXABERTes v1) y por otro lado un modelo basado en RoBERTa (IXABERTes v2). En el caso del modelo BERT el corpus de entrenamiento lo componen Gigaword y Wikipedia. La tokenización se ha llevado a cabo con el tokenizador *Sentence Piece* con las opciones de *Cased* (manteniendo las mayúsculas y minúsculas) y *Normalized* (normalizando la oración de entrada con una variante de Unicode NFKC) y también se ha creado un vocabulario de 50.000 subtokens. Para el entrenamiento se ha utilizado una tasa de aprendizaje de 1e-4, *Whole Word Masking* y *Next Sentence Prediction*. El 1.000.000 de actualizaciones se han hecho con lotes de 256 y secuencias de 512.

El modelo RoBERTa por el contrario está entrenado con el corpus OSCAR. La tokenización se ha llevado a cabo con el tokenizador *Byte-Pair Encoding* con la opción de *Cased* y se ha vuelto a crear un vocabulario de 50.000 subtokens. Para el entrenamiento se ha utilizado una tasa de aprendizaje de 7e-4 para hacer 120.500 actualizaciones con lotes de 2048 y secuencias de 512.

Modelo	Tasa de aprendizaje	Actualizaciones	Lotes	Secuencias
IXABERTes v1	1e-4	1.000.000	256	512
IXABERTes v2	7e-4	120.500	2048	512

Tabla 4: Datos de entrenamiento de los modelos para castellano

Como en el caso del euskera, se han probado los modelos en cuatro tareas PLN para el castellano: clasificación de temas, análisis de sentimientos, POS y NER. Todas las tareas se han evaluado con conjuntos de datos existentes y previamente utilizados en castellano. Para la clasificación de temas se ha utilizado el corpus MLDoc[5], un corpus para la clasificación de documentos multilingües en ocho idiomas. El etiquetado gramatical se ha evaluado con los datos de Universal Dependencies 1.2 y para la evaluación de las entidades nombradas se han utilizado tres conjuntos de datos diferentes: CoNLL 2002, ANCORA y Capitel 2020. Por último el análisis de sentimientos, como en el caso del euskera, se evalúa utilizando un corpus de tweets que contienen mensajes relacionados con el dominio cultural.

Los mejores resultados se obtienen con el modelo BERT (IXABERTes v1), superando los resultados que obtiene mBERT en todas las tareas y BETO (Cañete et al., 2020) en POS y NER. Estas mejoras se han obtenido replicando y comparando los modelos mBERT y BETO en nuestros experimentos. Los resultados que no se mejoran respecto a BETO son en las tareas

5 https://github.com/facebookresearch/MLDoc

de clasificación de temas (aunque en el artículo de BETO presentan un valor de F-SCORE de 95.6) y análisis de sentimiento . La tabla 5 presenta dichos resultados.

Tarea	IXABERTes v1	mBERT	BETO
Clasificación de temas	95.97	95.37	**96.32 (Cased)**
Análisis de sentimientos	77.35	75.47	**79.87 (Cased)**
POS	**97.50**	96.35	97.35
NER - CoNLL	**88.17**	87.05	87.69
NER - ANCORA	**92.98**	91.47	91.66
NER - CAPITEL	**88.99**	87.95	88.71

Tabla 5: Resultados para POS, NER, análisis de sentimientos y clasificación de temas para el castellano

Descarga de los sistemas

Los modelos del lenguajes monolingües están accesibles en la siguiente página web. http://www.deeptext.eus/resources

Para poder utilizar los modelos, al ser todos modelos de hugginface, es suficiente con instalar Transformers de Huggingface (https://huggingface.co/transformers/). Transformers está testado en Python 3.6+, y PyTorch 1.1.0+ orTensorFlow 2.0+.

Del mismo modo, si por ejemplo se quiere utilizar el modelo BERTeus, que se encuentra en Huggingface, es suficiente con incluir las siguientes líneas de código:

```
tokenizer=BertTokenizer.from_pretrained('ixa-ehu/berteus-base-cased')
model=BertModel.from_pretrained('ixa-ehu/berteus-base-cased')
```

Bibliografía

Aduriz, I., Aranzabe, M. J., Arriola, J. M., Atutxa, A.,Diaz de Ilarraza, A., Garmendia, A., and Oronoz, M. (2003). Construction of a basque dependency treebank.InTreebanks and Linguistic Theories, Vaxjo, Sweden

Agerri R., San Vicente I., Campos J.A., Barrena A., Saralegi X., Soroa A., Agirre E. (2020) Give your Text Representation Models some Love: the Case for Basque. Proceedings of LREC.

Alegria, I., Arregi, O., Ezeiza, N., and Fern ´andez, I.(2006). Lessons from the development of a named en-tity recognizer for Basque.Procesamiento del lenguajenatural, 36:25–37.

Aranzabe, M. J., Atutxa, A., Bengoetxea, K., de Ilarraza,A. D., Goenaga, I., Gojenola, K., and Uria, L. (2015).Automatic conversion of the basque dependency treebank to universal dependencies. In Proceedings of thefourteenth international workshop on treebanks an linguistic theories (TLT14), pages 233–241.

Cañete J., Chaperon G., Fuentes R., Ho J., Kang H., Pérez J. (2020) Spanish Pre-Trained BERT Model and Evaluation Data. PML4DC at ICLR 2020

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.(2019). BERT: Pre-training of Deep Bidirectional Trans-formers for Language Understanding. InProceedings ofthe 2019 Conference of the North American Chapter ofthe Association for Computational Linguistics: HumanLanguage Technologies, Volume 1 (Long and Short Pa-pers), pages 4171–4186.

Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V. (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach. CoRR abs/1907.11692