ANÁLISIS DE SIMILITUD EN DOCUMENTOS DE TEXTO MEDIANTE TÉCNICAS DE CIENCIA DE DATOS BASADAS EN APRENDIZAJE PROFUNDO (DEEP LEARNING)

Calibar, Andrea Belén; Holleger, Jorge; Klenzi, Raúl Oscar
Instituto de Informática / Departamento Informática / Facultad de Ciencias Exactas Físicas y Naturales / Universidad Nacional de San Juan
Domicilio: Av. Ignacio de la Roza 590 (O), Complejo Universitario "Islas Malvinas", Rivadavia, San Juan, CPA: J5402DCS, 0264-260353 0264-4260355
{veneno76;wikituky;rauloscarklenzi}@gmail.com

RESUMEN

La presente propuesta pretende ingresar a un área de conocimiento actual y de creciente aplicabilidad en la extracción de conocimiento subyacente en datos de diferentes tipologías, cantidades y calidad, como es el aprendizaje profundo (Deep Learning –DL-). Aquí se propone una línea de investigación que habrá de contrastar los grados de similitud entre documentos de texto alcanzados, por medio de tres métodos y herramientas de software diferentes. Se considerará inicialmente el pluggin de aprendizaje profundo *Deeplearning4J* del entorno de software libre de aprendizaje de máquina (Machine Learning -ML-) KNIME ANALYTICS 3.5.2. Una segunda alternativa a utilizar será la biblioteca GENSIM de Python, para finalmente trabajar con una versión adaptada de red recurrente creada a partir de TENSORFLOW. Se comparará el rendimiento de estas herramientas sobre datos de reservorios existentes en internet con el fin de integrarlas y explotarlas simultáneamente en entornos de hardware con CPU multinúcleos y GPU computing.

CONTEXTO

Esta propuesta se da en el marco de los proyectos CICITCA_UNSJ “Ciencia de los Datos aplicada a grandes colecciones de datos” llevado adelante en el bienio 2016_2017 y la propuesta actual de proyecto de investigación, en evaluación, “Visualización y Deep Learning en Ciencia de los Datos”.

La elección de las herramientas a utilizar se centra esencialmente en el hecho de que KNIME es una plataforma cohesionada para científicos de datos de todos los niveles de habilidades, que proporciona un marco de ciencia de datos único y consistente. Ofrece capacidades de acceso y manipulación de datos de alta calificación, una amplia y completa gama de algoritmos y herramientas de aprendizaje automático adecuada desde principiantes hasta científicos de datos experimentados. La plataforma de KNIME se integra con otras herramientas y plataformas, como R, Python, Spark, H2O.ai, Weka, DL4J y Keras. La ayuda contextual de KNIME es más flexible que los "asistentes" fijos. La interfaz de usuario y los extensos ejemplos proporcionados con la plataforma atraen a la comunidad de científicos de datos [1].

Fig 1: Interfaz de KNIME ANALYTICS

La Fig 1 presenta la interfaz de usuario correspondiente a KNIME 3.5.2 y donde se visualizan diferentes cuadros que hacen al entorno de trabajo propiamente dicho, accesibilidad a diferentes módulos de trabajo como también ejercicios de ejemplificación, aplicación y la ayuda correspondiente a cada módulo utilizado.

1. INTRODUCCIÓN

El análisis de similitud de textos es fundamental para una amplia gama de tareas en el área de Procesamiento de Lenguaje Natural (PLN).Encontrar la similitud entre pares de textos se ha convertido en un gran reto para los especialistas del área, pudiéndose aplicar en diferentes tareas de PLN, tales como máquinas de traducción, construcción automática de resúmenes, atribución de autoría, pruebas de lectura comprensivas, recuperación de información, análisis de tendencias de investigación en el dominio académico, recomendación de cita y muchas otras, donde es prioritario medir el grado de similitud entre dos textos dados. Mayoritariamente, las métricas de similitud entre documentos han girado alrededor de similitudes sintácticas, problemática que se torna aún más compleja cuando se desea hallar la similitud semántica entre textos, los cuales en general difieren en longitud. En particular esta problemática se ve reflejada cuando se desea encontrar el grado de similitud entre un párrafo y una sentencia, una sentencia y una frase, una frase y una palabra y una palabra y un sentido [2].

La resolución a este tipo de tareas forma parte de un gran número de problemáticas a las que el DL pretende dar solución transformándose en un área de investigación extremadamente activa que está allanando el camino para el aprendizaje de máquina moderno [3].
Dentro de la rama de la inteligencia artificial y como se aprecia en la Fig 2, el DL es un conjunto de algoritmos de *machine learning* que intenta modelar abstracciones de alto nivel en datos usando arquitecturas compuestas de transformaciones no lineales múltiples [4].

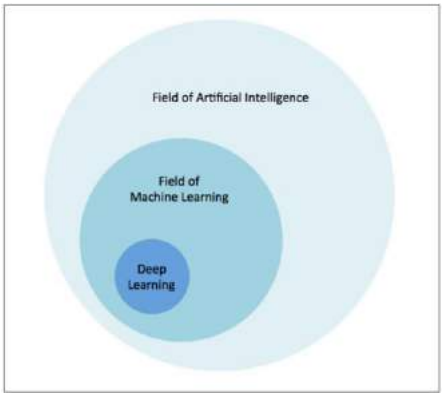

Fig 2: Deep learning como campo de Machine Learning[5].

Este tipo de aprendizaje tiene entre muchas otras características: mayor cantidad de neuronas que las redes neuronales convencionales, formas más complejas de conectar capas / neuronas en las redes, cantidad de potencia de cálculo

disponible para entrenar y extracción automática de características [5].

Una transformación previa al análisis profundo de los datos, cuando estos se presentan en forma de texto, es la incrustación de palabras (Word Embedding).

La incrustación de palabras es el nombre colectivo para un conjunto de técnicas de modelado de lenguaje y aprendizaje de características en el (PLN) donde las palabras o frases del vocabulario se asignan a vectores de números reales [9]. Cuando esto sucede, las palabras que emanan del mismo contexto o de un contexto similar pueden asociarse entre sí.

La transformación de palabras a números es necesaria porque muchos algoritmos de aprendizaje automático (incluidas las redes profundas) requieren en su entrada vectores de valores continuos, en lugar de cadenas de texto sin formato.

Una técnica de modelado de lenguaje natural de Word Embedding es word2vect, que es un grupo de modelos relacionados que se utilizan para crear incrustaciones de palabras. Estos modelos son redes neuronales que están capacitadas para reconstruir contextos lingüísticos de palabras [6].

Word2vec toma como entrada un gran corpus de texto y produce un espacio vectorial multidimensional, asignándosele a cada palabra en el corpus, un vector correspondiente en el espacio. Los vectores de palabras se ubican en el espacio vectorial de forma tal que las palabras que comparten contextos comunes en el corpus se ubican muy cerca la una de la otra en el espacio[6]

Esta y otras técnicas similares se estudiarán conformando un modelo apto para realizar el procesamiento de texto.

En la presente investigación se realizará el análisis de similitud de textos con entrenamiento de los respectivos modelos, enfocados en documentos cuyos temas están relacionados al ámbito de las ciencias de la computación y serán analizados mediante entornos aptos para utilizar técnicas de aprendizaje profundo en el procesamiento de texto.

En este contexto existen diferentes entornos de software de aprendizaje de máquina que permiten el procesamiento de dichos datos.
Para el tratamiento y análisis de los datos, se prevé utilizar, entre otras herramientas libres:

- KNIME ANALYTICS 3.5.2: Con la integración KNIME Deeplearning4J o DL4J versión 0.9.1 permite utilizar redes neuronales profundas.Se descarga de https://www.knime.com
- GENSIM: (librería de Python): Desarrollado originalmente por Radim Řehůřek. Gensim está bajo la licencia GNU LGPLv2.1 aprobada por OSI. Se descarga de https://radimrehurek.com/gensim/install.html
- TENSORFLOW: (librería de Python): desarrollado originalmente por investigadores e ingenieros del equipo Brain de Google dentro de la organización de investigación en Inteligencia Artificial de Google para realizar investigaciones de aprendizaje automático y redes neuronales profundas. Se descarga de https://www.tensorflow.org

En particular esta propuesta pretende una comparación de rendimiento analizando la similitud de documentos, entre el uso del módulo de aprendizaje profundo existente en la herramienta KNIME ANALYTICS 3.5.2 DL4J 0.9.1, La extensión consiste en un conjunto de nodos nuevos que permiten ensamblar de forma modular una arquitectura de red

neuronal profunda, entrenar la red con datos y utilizar la red capacitada para las predicciones. Además, es posible escribir/leer una red entrenada o sin entrenar hacia/desde un disco que permite compartir y reutilizar las redes creadas [7].

Utilizando la posibilidad de este último de extender sus capacidades mediante el uso de DL4J 0.9.1 se pretende compararla con el uso de una biblioteca de propósitos específicos como GENSIM.

GENSIM es un robusto conjunto de herramientas de modelado de temas y modelado de espacios vectoriales de código abierto implementado en Python. Utiliza NumPy, SciPy y, opcionalmente, Cython para el rendimiento. GENSIM está específicamente diseñado para manejar grandes colecciones de texto, usando transmisión de datos y algoritmos incrementales eficientes, lo que lo diferencia de la mayoría de los otros paquetes de software científico que solo se enfocan en procesamiento por lotes y en memoria [8].

Sin embargo, GENSIM no permiten utilizar el paralelismo aportado por la GPU de los actuales equipos de hardware. Aun así, ha demostrado un gran rendimiento en estudios previos. A diferencia de esta última KNIME admite GPU y esta posibilidad será explotada en la presente investigación.

La última instancia de comparación habrá de darse desde el uso de redes recurrentes contenidas en el entorno de software TENSORFLOW.

TensorFlow es una biblioteca de software de código abierto para el cálculo numérico utilizando gráficos de flujo de datos. Los nodos en el gráfico representan operaciones matemáticas, mientras que los bordes del gráfico representan los conjuntos de datos multidimensionales (tensores) comunicados entre ellos. Su arquitectura flexible permite implementar cálculos en una o más CPU o GPU de una computadora de escritorio, servidor o dispositivo móvil con una sola API. TensorFlow fue desarrollado originalmente por investigadores e ingenieros del equipo Brain de Google dentro de la organización de investigación de Inteligencia Artificial de Google para realizar investigaciones de aprendizaje automático y redes neuronales profundas, pero el sistema es lo suficientemente general como para aplicarse en una amplia variedad de otros dominios [8].

Para efectuar el correspondiente entrenamiento, se hará uso de una gran cantidad de datos de reservorios existentes en internet y se realizará su posterior testeo con material provisto en el marco del proyecto de investigación.

2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

En el marco de la investigación se pretende:

- Analizar y describir el conjunto de estudios y prácticas requeridos en Ciencia de Datos.
- Construir y validar las diferentes instancias de modelación para la evaluación de similitudes de documentos de texto, en el mismo entorno de hardware.
- Estudiar y analizar los diferentes resultados conforme las diferentes capacidades de visualización existentes en las herramientas.

3. RESULTADOS ESPERADOS

A los efectos de la utilización del hardware y software correspondiente, se

prevé la ejecución y prueba, en forma simultánea de diferentes ejemplos de aplicación en los tres entornos de software (centrados u operados desde la interfaz de usuario de KNIME ANALYTICS) y sobre una plataforma de hardware con multiprocesamiento a nivel de CPU y GPU computing. De esta aplicación se espera encontrar mejoras en la instancia programada sobre GPU computing basada en el modelo de aprendizaje Redes Neuronales Recurrentes.

4. FORMACIÓN DE RECURSOS HUMANOS

La presente propuesta da continuidad a los proyectos que la contienen y desde allí se ha dado la posibilidad de obtener becas de investigación de alumnos avanzados actualmente en ejecución, así como la conclusión y propuestas de nuevos trabajos finales de grado y posgrado lo que sin dudas mejora la calidad y perfil de cada uno de los integrantes del grupo de trabajo Esto permitirá el enriquecimiento de los distintos espacios curriculares de los que son responsables los diferentes integrantes del proyecto.

5. BIBLIOGRAFÍA

[1] Gartner. **Magic Quadrant for Data Science and Machine-Learning Platforms**. Feb 2018

[2] Darnes Villareño, Mireya Tovar, Beatriz Beltrán, Saúl León**Un modelo para detectar la similitud semántica entre textos de diferentes longitudes**. Benemérita Universidad Autónoma de Puebla,Facultad de Ciencias de la Computación, Puebla México

[3]Nikhil Buduma. **Fundamentals of Deep Learning**-O'Reilly (2017)

[4]Y. Bengio, A. Courville, and P. Vincent., "**Representation Learning: A Review and New Perspectives,"** IEEE Trans. PAMI, special issue Learning Deep Architectures, 2013.

[5] Josh Patterson and Adam Gibson**Deep Learning A Practitioner's Approach**

[6]Mikolov, Tomas; et al. "**Efficient Estimation of Word Representations in Vector Space**"

[7] https://www.knime.com/deeplearning4j

[8]https://radimrehurek.com/gensim

[9]https://es.wikipedia.org/wiki/Word_embedding