ExSim en eHealth-KD Challenge 2020

Zainab H. Almugbela, n.

aUniversidad de Leeds, Leeds, LS2 9JT, Reino Unido

[b]Universidad Imam Abdulrahman Bin Faisal, POBox 1982, Dammam, Arabia Saudita

Resumen Este

artículo describe el sistema presentado al eHealth-KD Challenge 2020-Tarea A: reconocimiento de entidades. El sistema utiliza una metodología de aprendizaje supervisado para reconocer entidades dentro de textos en español; es decir, aplica técnicas de PNL y word2vec para crear un diccionario etiquetado único de entidades en el conjunto de entrenamiento. Estas etiquetas se propagan a nuevas entidades que se encuentran en el conjunto de pruebas mediante medición de similitud semántica. La simplicidad de nuestro sistema muestra un bajo rendimiento con F1=0,32, precisión= 0,29 y recuperación=0,34. Finalmente, se discute el sistema desde diferentes aspectos: desafíos, intentos anteriores, características del sistema actual y posibles trabajos futuros.

Palabras clave

reconocimiento de entidades, PNL, word2vec

1. Introducción

El reconocimiento de entidades (RE) juega un papel vital en el análisis de textos no estructurados. ER se utiliza en varios dominios para diferentes aplicaciones, como sistemas de recomendación y recuperación de información; Específicamente, ER se utiliza ampliamente para reconocer entidades relacionadas con la salud en el ámbito de la atención médica, por ejemplo, nombres de enfermedades y medicamentos [1].

Debido a que la ER se aplica ampliamente, las investigaciones proponen varios enfoques para abordar este problema. Estos enfoques se pueden clasificar en dos secciones. La primera sección se llama enfoques de aprendizaje supervisado. Requieren un conjunto de entrenamiento, como la aplicación de clasificadores en palabras anotadas [2]. La segunda sección se llama enfoques de aprendizaje no supervisados. Requieren modelos estadísticos, como la clasificación de entidades según el esquema TF-IDF [3].

La Tarea A de eHealth-KD 2020 [4] tiene como objetivo reconocer entidades en frases de salud españolas no estructuradas tomadas de Medline. Este artículo propone un enfoque de aprendizaje supervisado para abordar el problema de ER. Esto significa entrenar un texto anotado determinado (conjunto de entrenamiento) para reconocer y etiquetar entidades en texto nuevo (conjunto de prueba). Nuestro enfoque está motivado por la teoría que define el lenguaje como una bolsa de entidades etiquetadas [5] y la investigación que plantea la importancia de medir la similitud de significados (similitud semántica) entre estas entidades tanto en el campo de la lingüística como en la tarea de clasificación [6] .

En la investigación [6], se utiliza la similitud semántica para aprender a identificar verbos de diferentes tiempos o las capitales de países. Esto significa que si se proporciona la etiqueta de una entidad, se puede asignar a una nueva entidad similar. Por ejemplo, si la etiqueta del país es "sustantivo" en mayúsculas

Actas del Foro de Evaluación de Lenguas Ibéricas (IberLEF 2020) correo electrónico:

sczha@leeds.ac.uk (ZH Almugbel) orcid:

0000-0003-4570-7088 (ZH Almugbel)

ISSN 1613-0073 Actas del taller CEUR (CEUR-WS.org)

Por ejemplo, en el caso de países, esta etiqueta también se puede asignar a la etiqueta de la capital. Por tanto, la similitud semántica puede identificar entidades que pertenecen a la misma etiqueta. Sin embargo, una desventaja obvia de este enfoque es que requiere una gran cantidad de datos de entrenamiento para minimizar la posibilidad de errores de "palabra fuera del vocabulario".

En base a esto, el problema de ER puede tratarse como un problema de clasificación. Las clases son las etiquetas de las entidades que se toman del conjunto de entrenamiento; se les asigna uno de los siguientes valores: concepto, acción, predicado y referencia. La medida de similitud semántica se puede emplear como clasificador. Etiqueta las nuevas entidades según su medida de similitud con las entidades existentes. Cada nueva entidad está etiquetada con la etiqueta de la entidad con la puntuación de similitud más alta.

Así, las entradas del sistema son: 1. los textos anotados que incluyen las entidades y sus etiquetas, y 2. el texto no estructurado de destino que incluye entidades sin etiquetar. La salida del sistema son las etiquetas predicadas de las entidades identificadas que existen dentro del texto no estructurado de destino. Cada entidad tiene una sola etiqueta. Para identificar entidades y predecir sus etiquetas, el sistema aplica PNL [7] para la limpieza de texto e incrustaciones de palabras [8] para representar entidades como vectores numéricos. Luego, calcula la similitud del coseno entre estos vectores para medir su similitud semántica.

El resto del artículo presenta la descripción del sistema en la Sección 2, seguida de los resultados en la Sección 3. Luego, la discusión se divulga en la sección 4. Finalmente, la conclusión se expresa en la sección 5.

2. Descripción del sistema

Esta sección explica principalmente el sistema. Proponemos una idea sencilla para identificar las entidades médicas. Se trata de utilizar PNL y word2vec para crear un diccionario que contenga entidades distintas con sus etiquetas del conjunto de entrenamiento basado en la medición de similitud. Después de que el algoritmo 1 crea este diccionario, se utiliza para anotar las entidades de las oraciones en el conjunto de prueba. A continuación se analizan más detalles sobre el sistema. Está organizado en tres subsecciones con fines aclaratorios: algoritmo, limpieza de texto e incrustaciones de palabras.

2.1. El algoritmo

En esta parte se presenta el algoritmo y se explican sus principales parámetros y variables.

Algoritmo 1: algoritmo de reconocimiento de entidades

```
Resultado: una lista de entidades con sus etiquetas previstas
frases clave = [[ 1, 1], [ 2, 2],..., [ 2 Lista , ]] ;
de entrenamiento = [[ 11, 12, 13,. . . ][ 21, 22,. . . ]. . .[. . . 3 , ]] ;
modelo = palabra2vec (Lista de entrenamiento,...);
tokens probados=[[]];
por cada K,L en frases clave
    para cada oración s en el conjunto de pruebas, haga
        si coincidencia exacta de K en s entonces
            tokens probados.append([K, 1, L])
        fin
        preproceso(s);
        para cada token t en s do
            puntuación=modelo.similitud(t,K);
            si puntuación>0 entonces
                si no está en testTokens entonces
                    testTokens.append([t, puntuación, L]);
                de lo contrario, si la puntuación actual > la puntuación almacenada, entonces
                    testTokens.update([t, puntuación actual, L actual])
            fin
        fin
    fin
fin
```

El algoritmo 1 consta de tres listas que requieren aclaración:

- frases clave es una lista anidada que contiene cada entidad y su etiqueta anotada. Por ejemplo, es la entidad y es su etiqueta, donde la etiqueta es uno de los cuatro valores posibles (concepto, predicado, referencia, acción).
- TrainingList también es una lista anidada que contiene las entidades de oraciones. es el entidad j de la oración i; donde i es para las oraciones y j es para las entidades dentro de un oración, por ejemplo 11 es la primera entidad en la primera oración.
- La lista TestedTokens contiene una lista de tres elementos [, ,]. El elemento de cada lista. Es único porque cada entidad debe tener como máximo una etiqueta.
    1. El primer elemento es una entidad que se toma del conjunto de prueba.
    2. El segundo elemento es la puntuación de medición de similitud entre y entidad k.
    3. El tercer elemento es la etiqueta prevista para .

Tanto el preprocesamiento como el modelo se explican en las siguientes subsecciones.

2.2. Limpieza de texto

En este sistema, la limpieza de texto se aplica sólo en el conjunto de prueba. Esto se debe a que el modelo se entrena directamente en TrainingList que contiene las entidades anotadas de las oraciones en letras minúsculas. La función de preproceso se aplica a las oraciones del conjunto de prueba. Emplea Natural Language Toolkit (NLTK) para limpiar cada oración eliminando la puntuación, cualquier entidad entre paréntesis y palabras vacías [7]. Luego, divide la oración en una lista de entidades en letras minúsculas.

2.3. Incrustaciones de palabras

El sistema emplea la biblioteca Gensim para aplicar el modelo word2vec [8] para incrustaciones de palabras y medición de similitud. Primero, el modelo se entrena en la lista: TrainingList con los siguientes hiperparámetros:

- size=100: es el tamaño dimensional de los vectores que representan las entidades.
- ventana=5: es la distancia máxima entre la entidad actual y la prevista dentro una sentencia.
- min_count=5: es la frecuencia mínima considerada de entidades en el modelo de entrenamiento.
- trabajadores=10: es el número de hilos que utilizará el modelo de entrenamiento
- El modelo skip-gram se utiliza en el modelo de entrenamiento.
- epochs=20: es el número de iteraciones sobre el conjunto de entrenamiento.

En segundo lugar, el modelo calcula la similitud del coseno entre una entidad del conjunto de entrenamiento y una entidad del conjunto de prueba. No se considera un valor umbral porque el sistema asigna la etiqueta de la entidad etiquetada similar más alta para cada entidad.

3. Resultados

Esta sección informa el desempeño del equipo en la tarea seleccionada; Específicamente, la Figura 1 muestra los resultados de nuestro sistema tanto para el conjunto de desarrollo como para el de prueba. La Figura 2 muestra los resultados de todos los equipos participantes en la misma tarea. En general, se puede notar que nuestro sistema no logró identificar entidades. Esto podría deberse a que el modelo está limitado por las entidades del conjunto de entrenamiento.

4. Discusión

Esta sección ilustra los desafíos que enfrentamos para participar en esta competencia, algunos intentos anteriores para mejorar el sistema y las características actuales del sistema.

En primer lugar, dos de los principales desafíos fueron el lenguaje y los recursos informáticos. En primer lugar, las barreras del idioma pueden deberse a dos aspectos: personal y técnico. En el aspecto personal, se tuvieron que realizar algunas traducciones para entender algunas entidades porque no estamos familiarizados.

Figura 1: Resultados de ExSim en la Tarea A

con español. En el aspecto técnico, cuando intentamos implementar el sistema desde cero, tuvimos un problema al escribir en los archivos de salida. El sistema muestra resultados en la consola pero los archivos de salida están vacíos. Después del fracaso de varios intentos de resolver este problema de codificación, se descuidó esta implementación anterior y en su lugar se utilizó la línea de base de eHealth-KD 2020.
En segundo lugar, el sistema utilizó los siguientes recursos informáticos: CPU i7-6500 de 2,50 GHz y 8 GB de RAM. Estos recursos tenían limitaciones por dos lados: el tiempo de ejecución y la capacidad de memoria, como se comenta en el segundo y tercer punto.

En segundo lugar, los intentos anteriores de mejorar el sistema incluyen lo siguiente:

- Se aplicó FastText [9] para incrustaciones de palabras, pero produjo puntuaciones de medición de similitud falsas para la mayoría de las entidades. Esto provocó que se etiquetaran la mayoría de las entidades con una sola etiqueta, por ejemplo, concepto. Modificamos sus hiperparámetros para comprobar cómo esto podría influir en las puntuaciones de similitud, pero las puntuaciones no mejoraron. Por lo tanto, se reemplaza por word2vec que otorga puntuaciones más confiables. El objetivo principal de utilizar FastText es que funciona al nivel de los personajes. Esto podría facilitar la identificación de las entidades que comparten un número específico de caracteres, lo que a su vez mejora la coincidencia de entidades similares de diferentes longitudes.

- En otro intento anterior, se consideraron tanto los unigramas como los bigramas, pero esto también se canceló porque no mejoraba el rendimiento del sistema. Los siguientes pasos aclaran nuestro enfoque para medir la similitud semántica entre los bigramas:

  1. Los bi-gramas (Frases de entrenamiento) con sus etiquetas son extraídos del entrenamiento.

colocar.

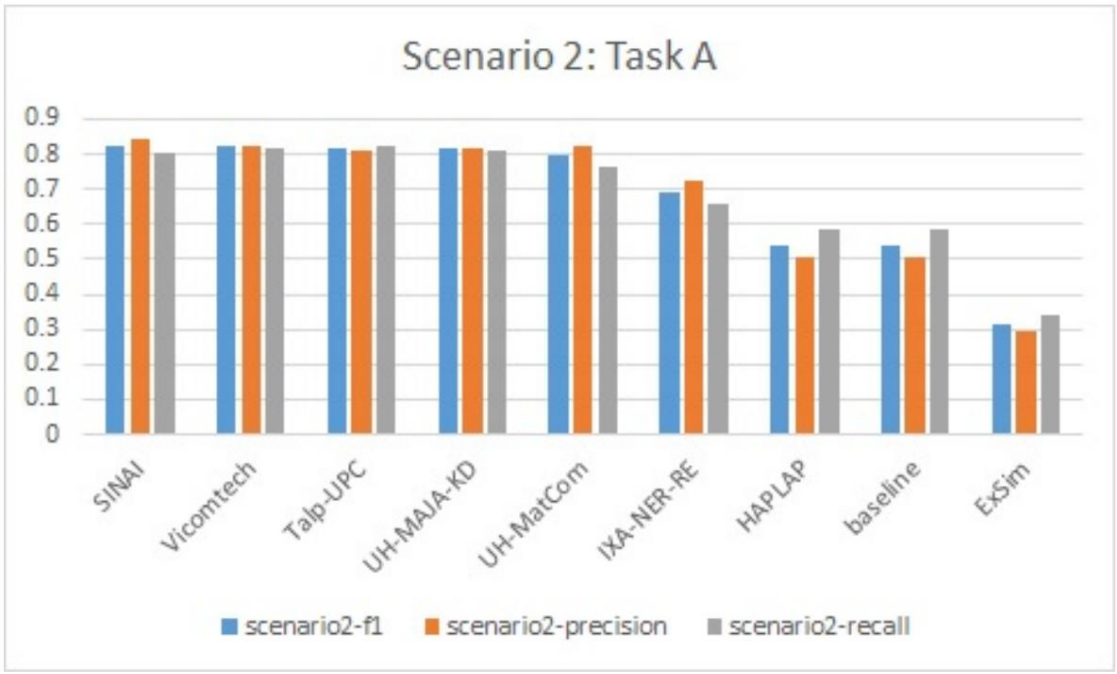

Figura 2: Resultados finales de los equipos en la Tarea A. Los resultados se toman de eHealth-KD Challenge 2020. (https://knowledge-learning.github.io/ehealthkd-2020/results)

2. Se crea una lista de los posibles bigramas (frases de prueba) para cada oración a partir de el conjunto de pruebas.
3. Se agrega la similitud semántica entre las entidades de frases de entrenamiento y frases de prueba ; luego se promedia.
4. El valor de la etiqueta de la frase de entrenamiento se asigna a la etiqueta de la frase de prueba según en la puntuación media más alta.

- También se consideró el uso de un modelo entrenado externo [10] , pero el sistema se quedó "fuera de servicio". error de memoria".

En tercer lugar, las características actuales del sistema son las siguientes:

- La última versión del sistema no ha considerado el etiquetado de POS y la lematización de palabras.
- Los hiperparámetros del último modelo se establecen después de varias modificaciones:
  - El parámetro min_count ha sido probado para los valores 1 a 5. Se elige el valor 5. Aunque esto provoca que se omitan algunas entidades, reduce el tiempo de ejecución a 12 horas.
  - También se han probado otros parámetros para diferentes valores pero no hemos observado ninguna mejora en el sistema. Estos incluyen los siguientes parámetros:

    el tamaño se probó para los valores 60-300.

los trabajadores fueron evaluados para los valores 5-10.

épocas se probaron para los valores 10-20

5. Conclusión

Este artículo presenta el sistema que se implementa para abordar la tarea de reconocimiento de entidades. Se basa en técnicas de PNL y de incrustación de palabras. Dado que el sistema tiene un bajo rendimiento, requiere mejoras desde diferentes aspectos. En primer lugar, se recomiendan mejores recursos informáticos. Esto aporta muchos beneficios al sistema: 1. reducir el tiempo de ejecución del sistema y 2. permitir el uso de modelos previamente entrenados y recursos externos. En segundo lugar, se debe aplicar etiquetado en el punto de venta para eliminar las palabras que no sean importantes. En tercer lugar, se recomienda realizar una revisión de la literatura. Esto permite aprovechar las ventajas de las técnicas ER anteriores de alto rendimiento.

Expresiones de gratitud

Agradecemos a los organizadores y revisores de eHealth-KD 2020. Un agradecimiento especial a los supervisores, el Prof. Eric Atwell, por sus valiosas sugerencias y al Dr. Mohammad Ammar Alsalka, por sus útiles comentarios.

Referencias

[1] IJ Unanue, EZ Borzeshi, M. Piccardi, Redes neuronales recurrentes con incrustaciones de palabras especializadas para el reconocimiento de entidades nombradas en el dominio de la salud, Journal of biomedical informatics 76 (2017) 102–109.

[2] R. Florian, A. Ittycheriah, H. Jing, T. Zhang, Reconocimiento de entidades nombradas mediante combinación de clasificadores, en: Actas de la séptima conferencia sobre aprendizaje del lenguaje natural en HLT-NAACL 2003-Volumen 4, Asociación de Lingüística Computacional , 2003, págs. 168-171.

[3] S. Zhang, N. Elhadad, Reconocimiento de entidades biomédicas con nombre no supervisadas: experimentos con textos clínicos y biológicos, Journal of biomedical informatics 46 (2013) 1088–1098.

[4] A. Piad-Morffis, Y. Gutiérrez, H. Cañizares-Diaz, S. Estevez-Velarde, Y. Almeida-Cruz, R. Munoz, A. Montoyo, Panorama general del desafío de descubrimiento de conocimientos en esalud en iberlef 2020, en: Actas del Foro Ibérico de Evaluación de Lenguas (IberLEF 2020),

[5] V. Burr, Una introducción al construccionismo social, Routledge, 2006.

[6] D. Jatnika, MA Bijaksana, AA Suryani, análisis del modelo Word2vec para similitud semántica vínculos en palabras en inglés, Procedia Computer Science 157 (2019) 160–167.

[7] S. Bird, E. Klein, E. Loper, Procesamiento del lenguaje natural con Python: análisis de texto con el kit de herramientas del lenguaje natural, "O'Reilly Media, Inc.", 2009.

[8] R. Řehůřek, P. Sojka, Marco de software para modelado de temas con grandes corporaciones, en: Actas del taller LREC 2010 sobre nuevos desafíos para los marcos de PNL, ELRA, Valletta, Malta, 2010, págs. http://is.muni.cz/publication/884893/en.

[9] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriquecimiento de vectores de palabras con subpalabra información, preimpresión de arXiv arXiv:1607.04606 (2016).

[10] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, Aprendizaje de vectores de palabras para 157 idiomas, en: Actas de la Conferencia Internacional sobre Evaluación y Recursos Lingüísticos (LREC 2018), 2018.