

Mejora de la extracción de eventos Zero-Shot a través de la simplificación de oraciones

Sneha Mehta
Independiente
San Francisco, CA
EE.
UU. snehamehta@twitter.com

Huzefa Rangwala
Universidad George
Mason
Fairfax, VA EE. UU. rangwala@gmu.edu

Naren Ramakrishnan
Virginia Tech
Arlington, VA
EE.
naren@cs.vt.edu

Abstracto

El éxito de sitios como ACLED y Our World in Data ha demostrado la enorme utilidad de extraer eventos en formatos estructurados de grandes volúmenes de datos textuales en forma de noticias, redes sociales, blogs y foros de discusión. La extracción de eventos puede proporcionar una ventana a las crisis geopolíticas en curso y generar inteligencia procesable.

En este trabajo, presentamos la extracción de eventos de conflicto sociopolítico como una tarea de comprensión de lectura automática (MRC). En este enfoque, la extracción de actores y objetivos sociopolíticos de una oración se enmarca como un problema de respuesta a una pregunta extractiva condicionado a un tipo de evento. Hay varias ventajas de usar MRC para esta tarea, incluida la capacidad de aprovechar grandes modelos de idiomas multilingües preentrenados y su capacidad para realizar una extracción de tiro cero.

Además, encontramos que el problema de las dependencias de largo alcance, es decir, la gran distancia léxica entre las palabras desencadenantes y argumentales y la dificultad de procesar oraciones sintácticamente complejas plagan los enfoques basados en MRC. Para abordar esto, presentamos un enfoque general para mejorar el rendimiento de la extracción de eventos basada en MRC mediante la simplificación de oraciones sin supervisión guiada por el propio modelo MRC. Evaluamos nuestro enfoque en el conjunto de datos de extracción de eventos geopolíticos de ICEWS, con atención específica a los roles de argumento 'Actor' y 'Objetivo'. Mostramos cómo dicha simplificación del contexto puede mejorar el rendimiento de la extracción de eventos basada en MRC en más del 5 % para la extracción de actores y más del 10 % para la extracción de objetivos.

1. Introducción

Con la proliferación de las redes sociales, los microblogs y las noticias en línea, podemos obtener una comprensión en tiempo real de los eventos que suceden en todo el mundo.

El trabajo se realizó cuando el autor era estudiante en Virginia Tech

Al ingerir grandes conjuntos de datos no estructurados y convertirlos en formatos estructurados como tuplas (actor, evento, objetivo), podemos avanzar rápidamente en los sistemas de pronóstico de eventos (Ramakrishnan et al., 2014), codificación de eventos en tiempo real (Saraf y Ramakrishnan, 2016) u otras aplicaciones que pueden otorgar a las organizaciones una ventaja estratégica. Históricamente, esto ha sido posible gracias a iniciativas como ICEWS1 y GDELT2.

Estos sistemas dependen de la tecnología de extracción de eventos para llenar sus bases de conocimientos. La Fig. 1 ofrece un ejemplo de un evento 'Entablar juicio contra' del conjunto de datos de ICEWS. La extracción implica la identificación de entidades (empresario, empleados) correspondientes a los roles de argumento 'Actor' y 'Objetivo'. El evento es desencadenado por el predicado 'sued' en la figura. La tecnología tradicional de extracción de eventos se basa en enfoques basados en patrones que

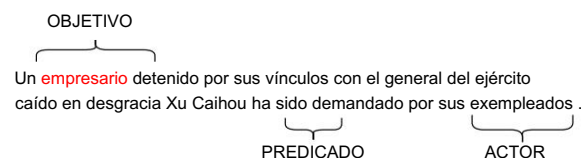


Figura 1: Un ejemplo de un evento del tipo 'Entablar demanda contra' del conjunto de datos de ICEWS. use patrones hechos a mano diseñados para extraer entidades y eventos (Boschee et al., 2013). Aunque los métodos basados en patrones tienen una alta precisión, no funcionan en tipos de eventos no vistos y con nuevas categorías de eventos. Por lo tanto, existe la necesidad de explorar métodos de extracción que puedan extenderse más allá de dominios y diccionarios fijos. Los enfoques modernos para la extracción de eventos (Chen et al., 2015; Nguyen et al., 2016; Wadden et al., 2019) se basan en notaciones detalladas y sufren problemas de escasez de datos y propagación de errores debido a los sistemas de canalización.

Con el éxito de los modelos de lenguaje preentrenado a gran escala en tareas de comprensión de lectura automática (MRC) (Devlin et al., 2019a; Liu et al., 2019; Huang

¹<https://dataverse.harvard.edu/dataverse/icews>

²<https://www.gdeltproject.org/>

et al., 2018), ha surgido un nuevo paradigma para la extracción de eventos basado en MRC (Du y Cardie, 2020; Liu et al., 2020). En este enfoque, la extracción de argumentos de eventos se plantea como un problema de extracción de intervalos de un contexto condicionado a una pregunta para cada argumento. Este enfoque es prometedor porque mitiga algunos de los problemas que enfrentan los enfoques tradicionales, como confiar en los sistemas ascendentes para extraer entidades/disparadores y, por lo tanto, eludir el problema de propagación de errores en los sistemas de canalización. También da lugar a la posibilidad de extracción de eventos de tiro cero y, por lo tanto, la capacidad de extenderse a nuevos dominios, lo que tradicionalmente es difícil debido a las dificultades para recopilar datos de entrenamiento etiquetados de alta calidad. Sin embargo, los modelos MRC luchan con dependencias de largo alcance y complejidades sintácticas. Por ejemplo, Liu et al. (2020) observan que un error típico de su sistema de extracción basado en MRC está relacionado con la dependencia de largo alcance entre un argumento y un disparador, lo que representa un 23,4 % de errores en el conjunto de datos de eventos ACE-2005 (Doddington et al., 2004) (aquí "largo alcance" indica que la distancia entre un disparador y un argumento es mayor o igual a 10 palabras). Du y Cardie (2020) observan que uno de los modos de falla de su sistema de extracción son las oraciones con estructuras de oraciones complejas que contienen múltiples cláusulas, cada una con disparador y argumentos. Estas observaciones constituyen un caso prometedor para la reducción de la complejidad o la simplificación del contexto para los sistemas MRC.

En este trabajo, planteamos la tarea de extracción de eventos de conflicto como una tarea de comprensión lectora mediante la generación de pares QA por argumento a extraer. Luego, para mitigar el problema de la dependencia de largo alcance y reducir la complejidad sintáctica, proponemos un enfoque de simplificación del contexto no supervisado que está guiado por una función de puntuación que incorpora la fluidez sintáctica, la simplicidad y la confianza de un modelo MRC (§ 2). Nuestras contribuciones clave son :

1. Enmarcar la extracción de eventos de conflicto como una tarea de comprensión de lectura mecánica y la exploración de la simplificación del contexto para ayudar a mitigar el problema de dependencia de largo alcance para la extracción de eventos basada en MRC (§ 2).
2. Demostramos empíricamente que la simplificación del contexto mejora el rendimiento de los sistemas MRC en entornos de entrenamiento de tiro cero y en el dominio.

2. Metodología

Dado que se ha detectado un evento en una oración, nos enfocamos en el problema de identificar los argumentos del evento detectado. Por ejemplo, en la Fig. 1, la tarea es identificar los argumentos 'Actor' y 'Objetivo' del evento 'Entablar demanda contra'. En correspondencia con cada tipo de evento, primero generamos pares de control de calidad correspondientes a los argumentos de actor y objetivo. El procedimiento de generación de QA para el conjunto de datos utilizado en este documento para la evaluación se describe en 4. La Tabla 1 muestra el par de QA generado para los argumentos Actor y Target para el evento que se muestra en la Fig. 1.

Los modelos de comprensión de lectura pueden ser frágiles ante cambios sutiles en el contexto. Pueden verse afectados por la complejidad sintáctica, especialmente cuando las preguntas no son específicas y no incluyen palabras que se superponen con el contexto. Además, las dependencias de largo alcance entre el disparador/predicado y el argumento son una fuente principal de error para los modelos MRC aplicados a la extracción de eventos como se describe en la sección 1. Para este propósito, proponemos un algoritmo de simplificación de oraciones no supervisada guiado por MRC (RUSS), que iterativamente realiza eliminaciones y extracciones del contexto en busca de un candidato de mayor puntuación. La función de puntaje incorpora componentes que aseguran la fluidez de las oraciones, la preservación de la información y la confianza del modelo MRC de destino. La figura 2 ofrece una descripción general del enfoque propuesto.

Tabla 1: Un ejemplo de un registro de control de calidad generado para un evento "Iniciar demanda contra" del conjunto de datos de ICEWS que se muestra en la Fig. 1. Los intervalos resaltados en rojo corresponden a los argumentos "Actor" y "Objetivo" del evento.

Sentencia Un	empresario detenido por sus vínculos con el general del ejército caído en desgracia Xu Caihou ha sido demandado por sus antiguos empleados.
Q-Actor ¿Quién demandó a alguien?	
Q-Target ¿Quién fue demandado por alguien?	

2.1 Algoritmo de simplificación de oraciones Dada

una oración de entrada s y una lista de preguntas $\{q_1, \dots, q_n\}$ correspondientes a diferentes argumentos, nuestro algoritmo realiza iterativamente dos operaciones en la oración: eliminación y extracción, en busca de una oración de mayor puntuación y genera una simplificación candidata c . Para generar candidatos, el algoritmo primero obtiene el árbol de análisis de distrito electoral del contexto utilizando un analizador de distrito electoral basado en intervalos (Joshi et al., 2018). Luego, secuencialmente por

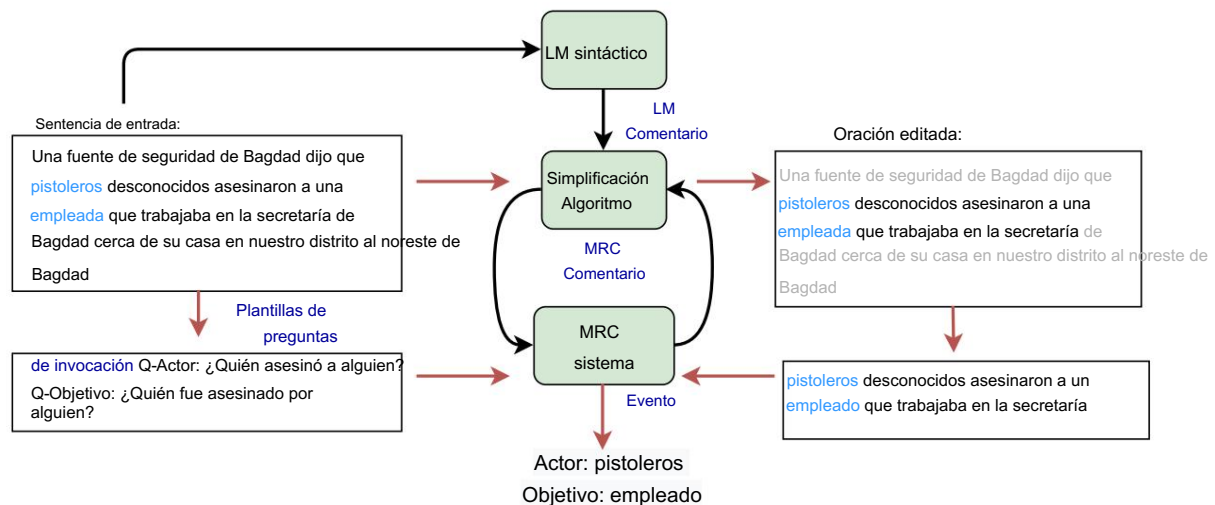


Figura 2: El enfoque de simplificación de oraciones RUSS.

forma dos operaciones en el árbol de análisis: eliminación y extracción.

Eliminación En esta operación, el algoritmo elimina secuencialmente subárboles del árbol de análisis que corresponden a diferentes frases. Tenga en cuenta que los subárboles con la etiqueta NP (Noun-Phrase) se omiten porque se espera que muchas entidades que forman argumentos de eventos sean frases nominales y eliminarlas de la oración resultaría en una pérdida significativa de información.

Extracción Esta operación simplemente extrae una frase, específicamente correspondiente a las etiquetas S y SBAR como la oración candidata. Este todo nos permite seleccionar diferentes cláusulas en una oración y eliminar la información periférica restante.

Estas operaciones generan múltiples candidatos. Candidatos con menos de un umbral de t palabras se filtran. Determinamos heurísticamente $t = 5$.

De los candidatos restantes, se elige el candidato con la puntuación más alta en función de la función de puntuación descrita en la siguiente sección (§ 2.2). El algoritmo finaliza si la puntuación máxima asignada a un candidato en la iteración actual no supera la puntuación máxima anterior. El algoritmo de simplificación RUSS se describe como Algoritmo 1 y el algoritmo de generación de candidatos se describe como Algoritmo 2 en el Apéndice.

2.2 Función de puntuación

Calificamos a un candidato como producto de diferentes puntajes correspondientes a la fluidez, la simplicidad y su adaptabilidad al modelo MRC posterior.

Puntaje LM (vlm) Este puntaje está diseñado para medir la fluidez del idioma y la simplicidad estructural de una oración candidata. En lugar de utilizar la perplejidad de LM, utilizamos la puntuación sintáctica de la razón de probabilidades logarítmicas (SLOR) (Pauls y Klein, 2012; Carroll et al., 1999) para medir la fluidez. También se demostró que SLOR es eficaz en la simplificación para mejorar la legibilidad del texto (Kann et al., 2018; Kumar et al., 2020). Dado un modelo de lenguaje entrenado (LM) y una oración s , SLOR se define como

$$\text{SLOR}(s) = \frac{1}{|s|} (\ln(\text{PLM}(s)) - \ln(\text{PU}(s))) \quad (1)$$

donde PLM es la probabilidad de oración dada por el modelo de lenguaje, $\text{PU}(s) = \prod_{w \in s} P(w)$ es el producto de la probabilidad del unigrama de una palabra w en la oración, y $|s|$ es la longitud de la oración. SLOR esencialmente penaliza la probabilidad de un LM simple por la probabilidad de unigrama y la longitud. Asegura que la puntuación de fluidez de una oración no se vea penalizada por la presencia de palabras raras. Un modelo de lenguaje probabilístico (LM) se usa a menudo como una estimación de la fluidez de las oraciones. En nuestro trabajo, en lugar de usar un LM simple, usamos un LM consciente de la sintaxis, es decir, además de las palabras, usamos etiquetas de dependencia y parte del discurso (POS) como entradas para el LM (Zhao et al., 2018). Para una palabra w_i , la entrada al LM consciente de la sintaxis es $[e(w_i); p(w_i); d(w_i)]$, donde $e(w_i)$ es la palabra incrustación, $p(w_i)$ es la incrustación de la etiqueta POS y $d(w_i)$ es la incrustación de la etiqueta de dependencia. Tenga en cuenta que nuestro LM está entrenado en el cuerpo del tren original. Por lo tanto, el LM consciente de la sintaxis ayuda a identificar candidatos que son estructuralmente agramaticales.

Puntaje de entidad (ventity) Las entidades ayudan a identificar la información clave de una oración y, por lo tanto, también son útiles para medir la preservación del significado. Los roles de argumento deseados también son entidades. Por lo tanto, si se omite cualquier entidad detectada en la oración original de un candidato, la puntuación de entidad para ese candidato es 0; de lo contrario, se establece en 1.

Puntuación de predicado (vpred) Esta puntuación conserva los predicados de evento en un candidato. Comprueba si un candidato contiene algún predicado de interés correspondiente al evento detectado (Tabla 5). Si no es así, vpred se establece en 0; de lo contrario, se establece en 1.

MRC Score (vrc) Los modelos MRC basados en transformadores pueden ser frágiles ante cambios sutiles en el contexto. Para hacer que el contexto sea robusto para el modelo MRC, esta puntuación nos permite controlar la complejidad del contexto con respecto a la confianza del modelo MRC. Se calcula por separado para cada rol.

Cada argumento de un evento es un lapso en el contexto. es la puntuación del mejor lapso en el contexto

rolei para el rol de argumento i, donde la puntuación de un candidato se define como $STx + ETy$ donde S es un vector inicial y E es un vector final como se define en Devlin et al. (2019b). Tx y Ty son las representaciones de la capa final del modelo BERT tokens x en el contexto. Tenga en cuenta que

para un lapso válido, $y > x$. Esta puntuación se calcula por separado para cada rol de argumento (Actor y Destinatario en el Ejemplo 1). La importancia de la i puede ser controlada por el exponente ri . La

contribución total de cada rol se calcula como el producto del puntaje correspondiente a cada rol, dado por El puntaje final de un candidato c es como lo siguiente:

$$v(c) = vlm(c)^a \cdot v_{entidad(c)}^b \cdot v_{pred(c)}^c \cdot \prod_{rolei} \text{---} (c) \quad (2)$$

Tenga en cuenta que b , c pueden ser 1 o 0 ya que ventity y vpred son binarios. En secciones posteriores, evaluaremos cómo se puede controlar la simplificación variando las constantes ri .

3 conjuntos de datos y métricas

Evaluamos a RUSS en el evento ICEWS conjunto de datos³ (Halkia et al., 2020) de los años 2013 a 2016. En este conjunto de datos, los datos de eventos consisten en interacciones codificadas entre actores sociopolíticos (es decir, acciones cooperativas u hostiles entre individuos,

³<https://dataverse.harvard.edu/dataverse/icews>

grupos, sectores y estados nacionales) asignados a los 4 CAMAFEO ontología. Preprocesamos los datos de ICEWS para extraer eventos triples que consisten en un actor de origen, un tipo de evento (según la taxonomía de eventos de CAMEO) y un actor de destino. Un registro ICEWS contiene una oración de evento, nombres de fuente y objetivo (actor y objetivo) y texto de evento, entre otros metadatos. Sin embargo, estos nombres de origen y destino están normalizados, es decir, es posible que los intervalos exactos de origen y destino no aparezcan en la oración de evento. Por ejemplo, un nombre de fuente en un registro ICEWS es "Organización del Tratado del Atlántico Norte", sin embargo, la oración del evento contiene su abreviatura "OTAN". Para recuperar los nombres exactos de origen y destino correspondientes a los intervalos que ocurren en la oración del evento, realizamos una desnormalización mediante el uso de los diccionarios de actores y agentes de ICEWS5 que contienen alias de diferentes entidades de origen y destino. Para el ejemplo anterior de "OTAN", el diccionario de actores contiene los siguientes alias " Organización del Tratado del Atlántico Norte, OTAN, Organización del Tratado del Atlántico Norte". Resolvemos el nombre de origen al alias que aparece dentro de la oración, que en este ejemplo es " OTAN". También eliminamos el nombre del país del paréntesis de los nombres de origen y destino: Ciudadano (Irak) → Ciudadano debido al formato en el que aparecen en los diccionarios. Después de la deduplicación y limpieza de los datos ICEWS obtenemos tuplas de actor, evento y destino para cada oración de evento.

El siguiente paso es generar pares de control de calidad para cada tupla según el tipo de evento.

4 Generación de conjuntos de datos de control de calidad

Primero agrupamos los registros de eventos ICEWS preprocesados por tipo de evento. Para cada tipo de evento, identificamos una lista de los predicados más comunes (desencadenantes) para ese tipo de evento mediante un enfoque heurístico, ya que las etiquetas de activación no están disponibles en el conjunto de datos de ICEWS. Usando este enfoque, obtuvimos una lista de predicados comunes correspondientes a los tipos de eventos y sus códigos CAMEO, como se muestra en la Tabla 5 del Apéndice. Por ejemplo, para el tipo de evento 'Demostrar o reunir', los predicados identificados son 'condenar', 'protestar', 'demostrar' y para el tipo de evento 'Acusar', los predicados son 'culpar', 'culpar', 'acusado', 'presunto', 'acusando'. Para cada uno de los predicados identificados para cada tipo de evento, usamos una plantilla de pregunta para cada uno de los dos roles de argumento Actor ⁴[https://](https://parusanalytics.com/eventdata/datos.dir/cameo.html)

⁴parusanalytics.com/eventdata/datos.dir/cameo.html

⁵<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/28118>

y objetivo. Para el rol de Actor, la plantilla usó una construcción activa '¿Quién \$predica\$ a alguien?' y para el mismo evento para el rol de Objetivo, la plantilla usó una construcción pasiva: '¿Quién fue \$predicado\$ por alguien?'. Esto da como resultado un total de 37 894 registros para los años 2013-2015 y 2953 registros para 2016 con una oración y dos preguntas, una para cada uno de los roles Actor y Target y los nombres Actor y Target como sus respuestas, respectivamente, distribuidos en 9 tipos de eventos. La distribución de entrenamiento/prueba de los registros de eventos sobre los diferentes tipos de eventos se muestra en la Tabla 6. Publicaremos las divisiones que usamos junto con las preguntas generadas, las respuestas y las compensaciones de intervalo para la reproducibilidad.

4.1 Evaluación

Realizamos una evaluación doble: 1) evaluamos el rendimiento de un sistema MRC antes y después de la simplificación en una configuración de disparo cero; 2) Entrenamiento en dominio: es decir, cuando hemos etiquetado los datos de entrenamiento en dominio disponibles, investigamos si la simplificación puede ayudar a mejorar el rendimiento cuando el sistema MRC ha sido entrenado en datos en dominio. En 1) emulamos un escenario sin recursos, es decir, usando el sistema MRC listo para usar en un dominio de destino. No ajustamos un modelo MRC previamente entrenado con el conjunto de datos de control de calidad generado. Más bien, el objetivo es evaluar el rendimiento del modelo en una configuración de disparo cero, sin utilizar ningún dato de entrenamiento del dominio de destino. Usamos el modelo BERT preentrenado ajustado en el conjunto de datos SQUAD 2.0 (Rajpurkar et al., 2016) y usamos la API predictora proporcionada aquí análisis de ⁶. Además llevamos a cabo seguimiento para estudiar la capacidad de control de la simplificación mediante la realización de análisis de ablación y la evaluación del rendimiento del modelo para diferentes valores de puntuación coeficientes de componente. Para configurar 1) usamos los datos de los años 2013-2015 para evaluación y para 2) usamos datos de los años 2013-2015 para tren y 2016 para prueba. Extrajimos los mejores lapsos predichos y calculamos una puntuación F1 de coincidencia exacta (Seo et al., 2017) que compara el lapso con la respuesta de verdad del terreno.

5 Resultados y Discusión

Los resultados de la extracción de tiro cero en el ICEWS conjunto de datos se describen en la Tabla 2. En las líneas de base utilizadas, la simplificación se realiza con exponentes de función de puntuación para vlm como $a = 1.5$ y ventily como

$b = 1$ se mantuvo constante mientras se variaba c para v_{pred} , r_1 para v_{actor} y r_2 para v_{target} . Sin simplificación, obtenemos puntajes F1 de 0,412 y 0,354 para los roles de actor y objetivo, respectivamente. Para la configuración más básica para la simplificación con $c = 0$, $r_1 = 1$ y $r_2 = 1$, las puntuaciones mejoran en un 4,6 % para la predicción del actor a 0,431 y en un 10,4 % a 0,391 para la predicción del objetivo, respectivamente, lo que muestra que la simplificación del contexto puede mejorar aún más un modelo poderoso como BERT en una configuración de tiro cero entre dominios. Para la predicción de actores, de 37.894 registros encontramos que para el 10,99 % de los registros, la puntuación F1 mejora después de la simplificación, para el 6,54 % de los registros F1 disminuyó después de la simplificación y para el resto la puntuación permaneció. Para la predicción de objetivos, para el 17,4% de los registros los puntajes mejoraron mientras que para el 7,9% de los registros los puntajes disminuyeron y para el resto de los registros, los puntajes permanecieron sin cambios. Tras introducir la puntuación del predicado ($c = 1$) vemos que estas mejoras descienden ligeramente. Esto es contrario a la intuición, porque uno esperaría que el rendimiento del modelo mejorara cuando los predicados relevantes están presentes en el contexto. Atribuimos este comportamiento al modelo MRC que aprovecha los idiomas previos en los datos de entrenamiento para predecir las respuestas. Por ejemplo, el modelo podría predecir el sujeto del predicado como respuesta a preguntas del tipo "Quién".

A continuación, aumentamos los coeficientes de los roles Actor y Target de 1 a 3. La razón por la que elegimos un número impar para este exponente es que, a veces, para los malos candidatos, las puntuaciones RC pueden ser negativas y dado que todas las puntuaciones se combinan en un número múltiple.

De manera replicativa, elevar una puntuación negativa a una potencia par revertiría el efecto deseado. Al observar los resultados en las filas 5 y 6 de la Tabla 2, podemos ver que ha aumentado el porcentaje de oraciones con las mismas puntuaciones antes y después de la simplificación. También observamos que el porcentaje de oraciones para las que las puntuaciones disminuyen después de la simplificación también han disminuido tanto para el actor (fila 5) como para el objetivo (fila 6), respectivamente. Podemos concluir que elevando los coeficientes de las puntuaciones específicas de los roles podemos hacer que los modelos de simplificación sean más robustos frente a simplificaciones inexactas para esos roles. También observamos, cuando $r_1 = 3$, obtenemos el F1 más alto para la predicción del actor, una mejora del 5,6% sobre la no simplificación y para $r_2 = 3$ obtenemos un F1 a la par con el más alto obtenido en la fila 2. Nuestros resultados indican claramente el beneficio de la simplificación sobre la no simplificación y también la mejora gradual en las puntuaciones cuando los coeficientes de argumento r_1 , r_2 son

⁶https://docs.allennlp.org/models/v2.4.0/modelos/rc/predictores/transformer_qa/

Tabla 2: Resultados de la extracción de eventos de tiro cero en el conjunto de datos ICEWS. coeficiente de vlm $a = 1,5$ y coeficiente de entidad $b = 1$ para todos los entornos en los que se realiza la simplificación. $\Delta +ve$ indica el % de registros para los que F1 mejora después de la simplificación, $\Delta -ve$ indica el % de registros para los que F1 empeora después de la simplificación y Δ mismo indica el % de registros para los que F1 permanece sin cambios.

Método	Actor				Objetivo			
	F1	$\Delta +ve$	$\Delta -ve$	Δ igual	F1	$\Delta +ve$	$\Delta -ve$	Δ igual
1 Sin simplificación	0.354	-	2	c = 0, $r_1 = 0,412$, $r_2 = 1$	0.431	10.99%	6.54%	82.45%
2 c = 1, $r_1 = 0$, $r_2 = 0.429$	10.81%	6.57%	82.61%	0.390	16.54%	7.53%	75.93%	4 c = 1, $r_1 = 1$, $r_2 = 1$
3 0.424	10.5%	6.3%	83.1%	0.387	16.29%	7.64%	76.05%	5 c = 1, $r_1 = 3$, $r_2 = 0$
4 0.43	5	9.72%	5.67%	84.6%	0.391	16.89%	7.97%	75.12%
5 6 c = 1, $r_1 = 0$, $r_2 = 3$	0.427	10.54%	6.95%	82.5%	0.391	16.12%	7.29%	76.59%

varió de 0 a 3.

5.1 Dependencias de largo alcance

La longitud media de las oraciones originales es de 32 palabras, mientras que la longitud media de las oraciones después de la simplificación es de 22 palabras (configuración de la fila 2). Esto indica que la simplificación no hace que las oraciones sean demasiado cortas como es intuitivo porque cortar información relevante dañaría el rendimiento.

A continuación, investigamos si la simplificación ha abordado el problema de la dependencia de largo alcance. Observamos las estadísticas relativas a la distancia entre el predicado y sus argumentos (Actor y Objetivo) para el ajuste $c = 0$, $r_1 = 1$, $r_2 = 1$, es decir, cuando la puntuación del predicado (vpred) no se tiene en cuenta. Como indica la Tabla 2 (fila 2), el 11 % de los registros aumenta el rendimiento después de la simplificación para Actor y el 17,35 % para Target. Encontramos que para esos registros la distancia promedio entre el predicado y su argumento Actor es de aproximadamente 13 palabras y la distancia promedio entre el predicado y el objetivo en el contexto simplificado es de aproximadamente 10 palabras. Para el argumento Objetivo, la distancia media entre el predicado y el objetivo es de unas 8 palabras para el original y unas 6 palabras para el contexto simplificado.

Vemos que RUSS recorta unas 3 palabras para la predicción del actor y 2 palabras para la predicción del objetivo en promedio. Concluimos que un cierto porcentaje de mejora proviene de reducir la distancia entre los predicados y los argumentos, mitigando así el problema de la dependencia de largo alcance.

5.2 Análisis cualitativo

La Tabla 3 enumera algunos casos en los que la simplificación ayuda a que el sistema MRC funcione mejor. En el primer ejemplo, el método propuesto eliminó la palabra 'personalmente' de la oración original (Sentence) para obtener la oración simplificada (Simplified) como se muestra

en la mesa. La pregunta planteada al modelo RC fue "¿A quién se está disculpando alguien?" y la respuesta de la verdad básica es "la oposición". Para el contexto original, el modelo extrae "Nawaz Sharif" como la respuesta incorrecta, mientras que después de eliminar el adverbio "personalmente", obtiene la respuesta correcta. Tenga en cuenta que esto disminuye la distancia entre el predicado disculpado de su argumento Nawaz Sharif. En el segundo ejemplo, el modelo RC extrae la frase nominal más cercana "Xu Caihou" como respuesta que es incorrecta. La simplificación elimina la frase preposicional "al general del ejército en desgracia Xu Caihou", lo que ayuda al modelo RC a extraer la respuesta correcta. Tenga en cuenta que en este caso era especialmente importante eliminar la frase anterior debido a la ambigüedad inherente de la construcción. Este caso también destaca las limitaciones de los sistemas RC actuales, ya que el sistema no pudo asociar con éxito a los empleados con el empresario y predijo el sintagma nominal más cercano al predicado demandado. En el tercer ejemplo, hubo un error de segmentación en el conjunto de datos de ICEWS y dos oraciones se unieron como se ve en la Tabla. RUSS eliminó con éxito la oración no relacionada que ayudó al sistema RC a extraer la respuesta correcta.

5.3 Análisis de errores

Del 6,54 % de los registros para los que la puntuación disminuyó después de la simplificación para la predicción del actor (fila 2 de la Tabla 2), para el 39,5 % de los registros, la predicción que usa el contexto original es una cadena de la predicción que usa el contexto simplificado. Esto significa que, en algunos casos, tanto el contexto original como el simplificado facilitan la respuesta correcta, pero la respuesta del contexto simplificado contiene información adicional por la que se penaliza durante el cálculo de la puntuación de F1. Por ejemplo, considere el contexto "una fuente de seguridad de Bagdad dijo que hombres armados desconocidos asesinaron a un empleado que trabajaba en l

Tabla 3: Ejemplos cualitativos del rendimiento de disparo cero del modelo RC antes y después de simplificar el contexto utilizando el algoritmo propuesto. Las palabras subrayadas son respuestas de verdad básica, las palabras enfatizadas son predicados (desencadenantes) y el tachado indica que el algoritmo eliminó las palabras.

Pregunta	¿A quién se está disculpando alguien?
Oración Scmp	El primer ministro de Islamabad, Nawaz Sharif, se disculpó personalmente con <u>la oposición</u> hoy por lo que calificó de comentarios desafortunados contra Aitzaz Ahsan Nawaz Sharif, del PPP.
Respuesta	
Oración Scmp	El primer ministro <u>simplificado</u> de Islamabad, Nawaz Sharif, se disculpó personalmente con <u>la oposición</u> hoy por lo que calificó de comentarios desafortunados contra Aitzaz Ahsan del PPP.
Respuesta	
Pregunta	¿Quién está siendo demandado por alguien?
Oración Scmp	Un <u>empresario</u> detenido por sus vínculos con el general del ejército caído en desgracia Xu Caihou, ha sido demandado por sus ex empleados
Respuesta Xu Caihou	
Oración Scmp	<u>Simplificado</u> Scmp, un <u>empresario</u> detenido por sus vínculos con el general del ejército caído en desgracia, Xu Caihou, ha sido demandado por sus ex empleado
Respuesta	empresario
Pregunta	¿Quién está siendo acusado de algo?
Sentencia Así	tras haber atacado a los dos elegidos de su partido, el árbitro Brice Hortefaux y Claude Goasguen, fue acusado de presiones e insultos. Rachida Dati ha acusado a <u>Claude Goasguen</u> de tomarla porque ella se negó a acostarse con él y esto durante un altercado probado por Canard Enchan.
Respuesta	rachida dati
Oración Scmp	<u>Simplificada</u> Así, <u>después de haber atacado a los dos elegidos a su partido</u> , el árbitro Brice Hortefaux y Claude Goasguen, fue acusado de presiones e insultos. Rachida Dati ha acusado a <u>Claude Goasguen</u> de tomarla porque ella se negó a acostarse con él y esto durante un altercado probado por Canard Enchan.
Respuesta	claude goasguen

de bagdad cerca de su casa en nuestro distrito al noreste de bagdad" que después de ejecutar el algoritmo de simplificación se acorta a "una fuente de seguridad de bagdad dijo que pistoleros desconocidos asesinaron a un empleado que trabajaba en la secretaría de bagdad cerca de su casa en nuestro distrito al noreste de bagdad". (El texto tachado representa el texto eliminado por el algoritmo propuesto). Para la pregunta; "¿Quién fue asesinado por alguien?" cuando se presenta con el contexto original, el modelo RC extrae "un empleado", mientras que después de eliminar el texto tachado, el modelo RC extrae "un empleado que trabaja en la secretaría". La respuesta de verdad básica para esto es "empleado". Como puede verse, ambas respuestas son correctas, pero el contexto simplificado se penaliza con palabras adicionales. Curiosamente, estos casos también representan el 48 % de los registros en los que el rendimiento mejora después de la simplificación, es decir, la predicción que usa el contexto original contiene la respuesta pero es más larga y la predicción que usa el contexto simplificado es más precisa. Esto es intuitivo, ya que el contexto se vuelve más breve y preciso después de la simplificación y, por lo tanto, uno espera que los modelos RC extraigan respuestas más precisas.

5.4 Capacitación en el dominio

En las secciones 5.1 a 5.3 vimos cómo RUSS mejoró el rendimiento en la configuración de disparo cero. En esta sección,

consideramos el escenario cuando hemos etiquetado datos de entrenamiento en el dominio disponibles y deseamos investigar si la simplificación puede ayudar a mejorar el rendimiento cuando el sistema MRC ha sido entrenado en datos en el dominio. Comparamos tres líneas de base.

BiLSTM-CRF (Huang et al., 2015; Halkia et al., 2020), modelo BertForQuestionAnswering de la biblioteca HuggingFace Transformers7 usando el modelo BERT base-cased como nuestro modelo base (BERT-RC), y use el mismo modelo después de la simplificación por el algoritmo RUSS (BERT-RC-Simple). Para el entrenamiento usamos el conjunto de datos ICEWS descrito anteriormente de años 2013-2015 y el año 2016 para la prueba.

BiLSTM-CRF Para esta línea de base, convertimos los tramos de actor y objetivo utilizando el esquema de etiquetado IOB en una secuencia de etiquetas. Usamos diferentes etiquetas para el actor y los objetivos (por ejemplo, B-ACT, B-TARG). El problema se convierte en el del etiquetado de secuencias sobre los tokens de la oración.

BERT-RC Para esta línea base, usamos la oración y los pares QA para el entrenamiento. Hay un total de 75.788 (37.894×2) ejemplos para entrenamiento y 5.906 (2.953×2) para prueba. Entrenamos todas las capas en lugar de solo la capa de clasificación, ya que observamos un gran

https://huggingface.co/transformers/v4.9.2/model_doc/bert.html?highlight=bertforquestionanswering#bertforquestionanswering

mejora en el primer caso en comparación con el segundo. Usamos una tasa de aprendizaje inicial de $3e-5$ y usamos la detención temprana con paciencia = 5 para encontrar el mejor modelo. Este modelo genera puntajes iniciales y finales de intervalo para cada token. Todos los tokens entre e incluyendo los tokens correspondientes a las puntuaciones máximas de inicio y final se extraen como el lapso previsto.

BERT-RC-Simple A continuación, usamos el algoritmo RUSS para obtener simplificaciones del conjunto de pruebas e informar el rendimiento de BERT-RC en este conjunto de pruebas simplificado.

La Tabla 4 indica el rendimiento del modelo en el conjunto de prueba original. Informamos F1 de coincidencia exacta para todas las líneas de base. Se puede observar que BERT RC funciona mejor que BiLSTM-CRF. La simplificación del contexto genera una mejora adicional (1,4 %) incluso en un modelo que se ajusta con precisión a los datos del dominio (BERT-RC-Simple).

Tabla 4: La tabla muestra el rendimiento de un modelo sin carcasa basado en BERT ajustado en un conjunto de datos en el dominio. Se puede ver que incluso después del ajuste fino, el enfoque RUSS mejora el rendimiento del modelo (BERT-RC-Simple).

Modelo	F1
BiLSTM-CRF	0,764
BERT-RC	0,776
BERT-RC-Simple	0,787

6 Trabajo relacionado

La extracción de eventos (EE) ha sido un área activa de investigación en la última década. En EE, los enfoques supervisados generalmente se basan en conjuntos de datos de entrenamiento etiquetados manualmente y ontologías hechas a mano. Li et al. (2013) utilizan los argumentos anotados y activadores de palabras clave específicas en el texto para desarrollar los enfoques supervisados también se han estudiado mediante el análisis de dependencia mediante el análisis de las relaciones de argumentos de eventos y el discurso de las interacciones de eventos (McClosky et al., 2011). Estos enfoques suelen estar limitados por la disponibilidad de los datos etiquetados de granularidad fina y requieren características elaboradas diseñadas. El trabajo reciente formula la extracción de argumentos de eventos como una tarea de MRC. Un desafío importante con este enfoque es generar un conjunto de datos de pares de control de calidad. Liu et al. (2020) proponen un método que combina traducción automática basada en plantillas y no supervisada para la generación de preguntas. Du y Cardie (2020) siguen un enfoque de plantilla y muestran que, cuanto más naturales sean las preguntas construidas, mejor será el rendimiento de extracción de eventos. Sin embargo, ninguno

de estos métodos apuntan directamente a abordar el problema de dependencia de largo alcance mediante la simplificación.

Los sistemas de simplificación automática de texto (ATS) tienen como objetivo transformar los textos originales en sus variantes léxica y sintácticamente más simples. La motivación para construir los primeros sistemas ATS fue mejorar el rendimiento de los sistemas de traducción automática y otras tareas de procesamiento de texto, por ejemplo, análisis sintáctico, recuperación de información y resumen (Chandrasekar et al., 1996). En el contexto de la extracción, Zhang et. Alabama. (Zhang et al., 2018) muestran que la poda de árboles de dependencia para eliminar estructuras irrelevantes puede mejorar el rendimiento de la extracción de relaciones. Se han realizado esfuerzos para incorporar dependencias sintácticas en los modelos en un esfuerzo por mitigar este problema 2016; 2018; 2020. Recientemente, Mehta et al. (2020) han utilizado la simplificación de oraciones como un paso previo al procesamiento para mejorar la traducción automática. La simplificación basada en la edición se ha investigado en gran medida para mejorar la legibilidad del texto (Kumar et al., 2020; Dong et al., 2019; Alva Manchego et al., 2017). Hasta donde sabemos, este es el primer trabajo que estudia la simplificación de oraciones para mejorar la extracción de eventos basada en MRC.

7 Conclusión y trabajo futuro

En este trabajo, motivamos la necesidad de un paradigma de extracción de eventos sociopolíticos/conflictos basado en MRC, especialmente para escenarios de tiro cero (§ 1). A continuación, discutimos el problema de dependencia de largo alcance que enfrentan los sistemas de extracción de eventos. Propusimos un algoritmo de simplificación para reducir la complejidad sintáctica del contexto con la ayuda de la retroalimentación del sistema MRC para abordar el problema (§ 2). Nuestros resultados indican que la simplificación no solo puede ayudar a

los sistemas MRC en una configuración de disparo cero (§ 5.1-5.3), sino también cuando se ajustan con precisión en los datos del dominio (§ 5.

En el trabajo futuro, planeamos escalar nuestro enfoque de generación de control de calidad para mejorar la cobertura en más tipos de eventos e idiomas. También podemos hacer que la simplificación de RUSS sea más eficiente al generar datos de entrenamiento paralelos para la simplificación usando el método RUSS fuera de línea y entrenar un modelo de simplificación usando los datos generados. De esta forma podemos obtener simplificaciones guiadas por inferencia sobre un modelo.

Reproducibilidad: liberamos nuestro código

⁸
8https://github.com/russ-event-extraction/russ_event_extraction

Referencias

- Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. [Aprender a simplificar a partir del etiquetado explícito de pares de texto complejo-simplificado](#). En Actas de la Octava Conferencia Internacional Conjunta sobre Procesamiento del Lenguaje Natural (Volumen 1: Documentos extensos), páginas 295–305, Taipei, Taiwán. Federación Asiática de Procesamiento del Lenguaje Natural.
- Elizabeth Boschee, Premkumar Natarajan y Ralph Weischedel. 2013. [Extracción Automática de Eventos de Texto de código abierto para pronóstico predictivo](#), páginas 51 a 67. Springer Nueva York, Nueva York, NY.
- John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin y John Tait. 1999. [Texto simplificado para lectores con problemas de lenguaje](#). En Novena Conferencia del Capítulo Europeo de la Asociación de Lingüística Computacional, páginas 269–270, Bergen, Noruega. Asociación de Lingüística Computacional.
- Raman Chandrasekar, Christine Doran y Bangalore Srinivas. 1996. Motivaciones y métodos para la simplificación de textos. En Actas de la 16ª conferencia sobre lingüística computacional, volumen 2, páginas 1041–1044. Asociación de Lingüística Computacional.
- Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng y Jun Zhao. 2015. Extracción de eventos a través de redes neuronales convolucionales de agrupación múltiple dinámica. En Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, volumen 1, páginas 167–176.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee y Kristina Toutanova. 2019a. Bert: Pre-entrenamiento de transformadores bidireccionales profundos para la comprensión del lenguaje. En NAACL-HLT.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee y Kristina Toutanova. 2019b. [BERT: Pre-entrenamiento de profundidad transformadores bidireccionales para la comprensión del lenguaje](#). En Actas de la Conferencia de 2019 del Capítulo de América del Norte de la Asociación de Lingüística Computacional: Tecnologías del Lenguaje Humano, Volumen 1 (Artículos largos y cortos), páginas 4171–4186, Minneapolis, Minnesota. Asociación de Lingüística Computacional.
- George Doddington, Alexis Mitchell, Mark Przytycki, Lance Ramshaw, Stephanie Strassel y Ralph Weischedel. 2004. [El programa de extracción automática de contenido \(ACE\): tareas, datos y evaluación](#). En LREC'04, Lisboa, Portugal. Asociación Europea de Recursos Lingüísticos (ELRA).
- Yue Dong, Zichao Li, Mehdi Rezagholizadeh y Jackie Chi Kit Cheung. 2019. [EditNTS: Una red neuronal modelo de programador-intérprete para la simplificación de oraciones a través de la edición explícita](#). En Actas de la 57.ª Reunión Anual de la Asociación de Lingüística Computacional, páginas 3393–3402, Florencia, Italia. Asociación de Lingüística Computacional.
- Xinya Du y Claire Cardie. 2020. [Extracción de eventos por respondiendo preguntas \(casi\) naturales](#). En Actas de la Conferencia de 2020 sobre métodos empíricos en el procesamiento del lenguaje natural (EMNLP), páginas 671–683, en línea. Asociación de Lingüística Computacional.
- Matina Halkia, Stefano Ferri, Michail Papazoglou, Marie-Sophie Van Damme y Dimitrios Thomakos. 2020. [Modelado de eventos de conflicto: Experimento de investigación y limitaciones de datos de eventos](#). En Actas del taller sobre extracción automatizada de eventos sociopolíticos de News 2020, páginas 42–48, Marsella, Francia. Asociación Europea de Recursos Lingüísticos (ELRA).
- Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel y Clare Voss. 2018. [Aprendizaje de transferencia Zero-shot para extracción de eventos](#). En Actas de la 56.ª Reunión Anual de la Asociación de Lingüística Computacional (Volumen 1: Documentos extensos), páginas 2160–2170, Melbourne, Australia. Asociación de Lingüística Computacional.
- Zhiheng Huang, Wei Xu y Kai Yu. 2015. [Modelos LSTM-CRF bidireccionales para el etiquetado de secuencias](#). CoRR, abs/1508.01991.
- Vidur Joshi, Matthew Peters y Mark Hopkins. 2018. [Extender un analizador a dominios distantes utilizando unas pocas docenas de ejemplos parcialmente anotados](#). En Actas de la 56.ª Reunión Anual de la Asociación de Lingüística Computacional (Volumen 1: Documentos extensos), páginas 1190–1199, Melbourne, Australia. Asociación de Lingüística Computacional.
- Katharina Kann, Sascha Rothe y Katja Filippova. 2018. [Evaluación de fluidez a nivel de oraciones: las referencias ayudan, ¡pero se pueden ahorrar!](#) En Proceedings of the 22nd Conference on Computational Natural Language Learning, páginas 313–323, Bruselas, Bélgica. Asociación de Lingüística Computacional.
- Dhruv Kumar, Lili Mou, Lukasz Golab y Olga Vechtomova. 2020. Simplificación de oraciones no supervisada basada en edición iterativa. preimpresión de arXiv arXiv:2006.09639.
- Qi Li, Heng Ji y Liang Huang. 2013. Extracción conjunta de eventos mediante predicción estructurada con características globales. En Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volumen 1: Documentos extensos), volumen 1, páginas 73–82.
- Jian Liu, Yubo Chen, Kang Liu, Wei Bi y Xiaojiang Liu. 2020. [Extracción de eventos como comprensión lectora automática](#). En Actas de la Conferencia de 2020 sobre métodos empíricos en el procesamiento del lenguaje natural (EMNLP), páginas 1641–1651, en línea. Asociación de Lingüística Computacional.
- Xiao Liu, Zhunchen Luo y Heyan Huang. 2018. [Extracción conjunta de múltiples eventos a través de la agregación de información gráfica basada en la atención](#). En Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, páginas 1247–1256, Bruselas, Bélgica. Asociación de Lingüística Computacional.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, et al. 2019. Roberta: un enfoque de preentrenamiento de bert sólidamente optimizado. preimpresión de arXiv arXiv:1907.11692.

Jie Ma, Shuai Wang, Rishita Anubhai, Miguel Ballesteros y Yaser Al-Onaizan. 2020. [Recursos mejorados modelo neuronal para la extracción de argumentos de eventos](#). En Findings of the Association for Computational Linguistics: EMNLP 2020, páginas 3554–3559, en línea. Asociación de Lingüística Computacional.

David McClosky, Mihai Surdeanu y Christopher D. Manning. 2011. Extracción de eventos como análisis de dependencias. En Actas de la 49.ª Reunión Anual de la Asociación de Lingüística Computacional, páginas 1626–1635.

Sneha Mehta, Bahareh Azarnoush, Boris Chen, Avneesh Saluja, Vinith Misra, Ballav Bihani y Ritwik Kumar. 2020. [Simplifique y luego traduzca: preprocesamiento automático para traducción automática de caja negra](#).

Thien Huu Nguyen, Kyunghyun Cho y Ralph Grishman. 2016. [Extracción conjunta de eventos a través de neural recurrente redes](#). En Actas de la Conferencia de 2016 del Capítulo de América del Norte de la Asociación de Lingüística Computacional: Tecnologías del Lenguaje Humano, páginas 300–309, San Diego, California. Asociación de Lingüística Computacional.

Adam Pauls y Dan Klein. 2012. Modelado de lenguaje [sintáctico a gran escala con treelets](#). En Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volumen 1: Documentos extensos), páginas 959–968, Isla de Jeju, Corea. Asociación de Lingüística Computacional.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev y Percy Liang. 2016. [SQuAD: más de 100 000 preguntas para Comprensión automática de texto](#). En Actas de la Conferencia de 2016 sobre métodos empíricos en el procesamiento del lenguaje natural, páginas 2383–2392, Austin, Texas. Asociación de Lingüística Computacional.

Naren Ramakrishnan, Patrick Butler, Sathappan Muthiah, Nathan Self, Rupinder Khandpur, Parang Saraf, Wei Wang, Jose Cadena, Anil Vullikanti, Gizem Korkmaz, Chris Kuhlman, Achla Marathe, Liang Zhao, Ting Hua, Feng Chen, Chang Tien Lu, Bert Huang, Aravind Srinivasan, Khoa Trinh, Lise Getoor, Graham Katz, Andy Doyle, Chris Ackermann, Ilya Zavorin, Jim Ford, Kristen Summers, Youssef Fayed, Jaime Arredondo, Dipak Gupta y David Mares. 2014. ['superando las noticias' con ascuas: pronóstico de disturbios civiles utilizando indicadores de fuente abierta](#). En Actas de la 20.ª Conferencia internacional ACM SIGKDD sobre descubrimiento de conocimientos y minería de datos, KDD '14, página 1799–1808, Nueva York, NY, EE. UU. Asociación de Maquinaria de Computación.

Parang Saraf y Naren Ramakrishnan. 2016. Embers autogr: codificación automatizada de eventos de disturbios civiles. En Actas de la 22ª Conferencia Internacional ACM SIGKDD sobre Descubrimiento de Conocimiento y Minería de Datos, páginas 599–608.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi y Hannaneh Hajishirzi. 2017. Flujo de atención bidireccional para la comprensión de máquinas. ArXiv, abs/1611.01603.

Lei Sha, Jing Liu, Chin-Yew Lin, Sujian Li, Baobao

Chang y Zhifang Sui. 2016. [RBPB: método de balanceo de patrones basado en la regularización para la extracción de eventos](#). En Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volumen 1: Documentos extensos), páginas 1224–1234, Berlín, Alemania. Asociación de Lingüística Computacional.

David Wadden, Ulme Wennberg, Yi Luan y Hannaneh Hajishirzi. 2019. [Extracción de entidades, relaciones y eventos con representaciones de tramo contextualizadas](#). En Actas de la Conferencia sobre métodos empíricos en el procesamiento del lenguaje natural de 2019 y la 9.ª Conferencia conjunta internacional sobre procesamiento del lenguaje natural (EMNLP-IJCNLP), páginas 5784–5789, Hong Kong, China. Asociación de Lingüística Computacional.

Yuhao Zhang, Peng Qi y Christopher D. Manning. 2018. [Convulación de gráficos sobre árboles de dependencia podados mejora la extracción de relaciones](#). En Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, páginas 2205–2215, Bruselas, Bélgica. Asociación de Lingüística Computacional.

Yang Zhao, Zhiyuan Luo y Akiko Aizawa. 2018. Un evaluador basado en un modelo de lenguaje para la comprensión de oraciones. En Actas de la 56.ª Reunión Anual de la Asociación de Lingüística Computacional (Volumen 2: Artículos breves), páginas 170–175.

Un algoritmo de RUSS

Algoritmo 1: Algoritmo de simplificación de oraciones –
Entrada RUSS:

```

oración := s, preguntas = {q1, ..qn}
Salida: simplificación := c Función
RUSS(s): maxIter ←
M for iter
maxIter do
    candidatos ← generarCandidatos(c) puntajes ←
    maxScore
    ← 0 para cand
    candidatos hacen
        puntuaciones ←
            partituras va lm b v v vrc rc entidad pred rolei
    end
    currMax ← max(puntuaciones)
    if currMax > maxScore then
        maxScore ← currMax c ←
        candidatos[argmax(puntuaciones)] end
    end
final
retorno c

```

El algoritmo tarda 5 horas en 1 núcleo de CPU y 1 GPU. Sin embargo, al paralelizar el cálculo en 5 núcleos, ese tiempo puede reducirse a 1 hora.

B Estadísticas del conjunto de datos

La Tabla 6 describe la distribución de diferentes tipos de eventos utilizados en el conjunto de datos ICEWS utilizado.

Algoritmo 2: Algoritmo de generación de candidatos

```

Entrada: frase := s
Salida: candidatos
Función generar candidatos(s): parseT ree
← getParseT ree(s) toRemove ←
extracciones
← candidatos
← frasesT ags
← getV alidP hraseT ags() for pos parseT
ree.positions do if parseT ree[pos]
    frasesT ags then toRemove ←
        toRemove parseT ree[pos].hojas
    fin
    si pos.label [S, SBAR] entonces
        extracciones ←
            extracciones parseT ree[pos].hojas
    fin
fin
de la frase toRemove do
    candidato ← s.replace(frase, ) si
    candidato.longitud > t entonces
        candidatos ← candidatos candidato
    fin
fin
para extracciones de frase
do if frase.longitud > t entonces
    candidatos ← candidatos candidato
fin
acabar
con los candidatos de retorno

```

Detalles de entrenamiento

Para entrenar el algoritmo RUSS, usamos el modelo TransformerQA disponible a través de la API de predictores de la biblioteca [allennlp](https://github.com/allenai/allennlp-models/blob/main/allennlp_models/rc/models/transformer_qa.py). ejecutando el

[9https://github.com/allenai/allennlp-models/blob/main/allennlp_models/rc/models/transformer_qa.py](https://github.com/allenai/allennlp-models/blob/main/allennlp_models/rc/models/transformer_qa.py)

Tabla 5: La tabla enumera los tipos de eventos ICEWS utilizados y sus predicados correspondientes que se identificaron para generar plantillas de preguntas.

Tipo de evento	Código CAMEO	predicados
Secuestrar, secuestrar o tomar rehenes	181	secuestrado, raptar, secuestrado, capturado
Acusar	112	culpar, culpar, acusado, alegado, acusando
Disculparse	55	disculpa, disculpa
Asesinar	186	llevó a cabo el asesinato de, asesinar
Traer demanda contra	115	está demandando a alguien, demandó, ha demandado, presentó una demanda contra
Demostrar o reunir	141	condenar, protestar, demostrar
Arrestar, detener o acusar con acción legal	173	arrestado, sentenciado, detenido, atrapado, capturado, arrestando, captura, encarcelado, rutinariamente arrestado, procesado, condenado
Usar la fuerza militar convencional	190	asesinado, descascarado, combatiendo, proyectiles, huelgas, huelga, matar

Tabla 6: La tabla muestra la distribución de los tipos de eventos en los conjuntos de datos de ICEWS Train and Test utilizados.

Tipo de evento	#Entrenamiento de registros	#Prueba de registros
Secuestrar, secuestrar o tomar rehenes	3473	193
Acusar	8856	651
Disculparse	181	11
Arrestar, detener o acusar con acción legal	9933	782
asesinar	146	12
Presentar demanda contra	206	18
Demostrar o reunir	2890	175
Usar fuerza militar convencional	12209	1111