Universidad de La Habana
Facultad de Matemática y Computación

Generación automática de reportes textuales sobre enfrentamientos deportivos

Autor:
Abel Molina Sánchez

Tutores:

Dr. Yudivián Almeida Cruz
Lic. Manuel Santiago Fernández Arias

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencia de la Computación

Noviembre 2022

A mis padres.

Agradecimientos

En primer lugar quiero agradecer a todos los profesores de mi vida estudiantil. Especial agradecimiento a mis tutores Yudivián Almeida y Manuel Santiago por su guía y su disposición en todo momento.

Quiero agradecer a mis padres, que han sido un ancla en todos los momentos de dificultad. Quién le iba a decir a una bioquímica y a un telecomunicador que iban a terminar revisando trabajos de cibernética, aunque solo sea la ortografía mamá. Un agradecimiento especial a mi Faby, por aguantar todos los momentos duros, por ser una inspiración, tú eres mi ejemplo de que siempre hay que seguir por mal dadas que vengan. Agradecimiento inmenso a mi abuela, siempre en función de que trabaje en la mayor calma posible. A mi hermano, por despertarme todos los días a las seis de la mañana, puedes estar seguro de que otro año hubieras ido caminando al servicio. Agradecer a mis hermanos de la vida, Ale, David, Flaco, con ustedes al lado no hay moral que caiga. Ale, tus audios levantan a un muerto. David, los apagones de tesis en el Naútico ya no nos los quita nadie, para la historia quedan. A los amigos que me regaló esta facultad, Ana, Javier, Marcos, agradecerles las mañanas, tardes, noches, madrugadas compartidas, entre gente linda todo fluye mejor. Agradecer al Dano, que como amigo me aguantó todas las dudas durante los cuatro años de carrera. Agradezco también de forma especial al profe Carlos, que cuando peor iba todo mantuvo el barco a flote. Quiero agradecer a Claudia, Laura, Coto, Cri, Sandrita, Dariel, por estar ahí siempre. Agradecer a todos mis amigos y familiares que han formado parte de este camino, me es imposible mencionarlos a todos, los llevo conmigo.

Opinión del tutor

El estudiante Abel Molina Sánchez desarrolló satisfactoriamente el trabajo de diploma titulado "Generación automática de reportes textuales sobre enfrentamientos deportivos". En este trabajo el estudiante propuso el diseño de un sistema para la generación automática de notas textuales sobre eventos deportivos. La idea general seguida fue la definición de un esquema general, o meta-esquema, de tuplas (4-tuplas) de conocimiento sobre las cuales se pudieran implementar, siguiendo una metodología genérica, funciones de realización particulares.

Para validar esta propuesta, el estudiante propuso esquemas particulares para dos deportes con características diferentes: el fútbol y el boxeo. Además, propuso un conjunto de funciones de realización lingüística para cada caso. Con ello pudo mostrar como se generaban distintos reportes para bases de conocimiento diferentes y así mostrar la validez y factibilidad de la propuesta.

Para poder afrontar el trabajo, el estudiante tuvo que revisar literatura científica relacionada con la temática así como soluciones existentes y bibliotecas de software que pueden ser apropiadas para su utilización. Todo ello con sentido crítico, determinando las mejores aproximaciones y también las dificultades que presentan.

Todo el trabajo fue realizado por el estudiante con una elevada constancia, capacidad de trabajo y habilidades, tanto de gestión, como de desarrollo y de investigación.

Por estas razones pedimos que le sea otorgada al estudiante Abel Molina Sánchez la máxima calificación y, de esta manera, pueda obtener el título de Licenciado en Ciencia de la Computación.

Dr. Yudivián Almeida Cruz

Resumen

La Generación de Lenguaje Natural, como subcampo de la Inteligencia Artificial y la lingüística computacional, ha despertado cada vez mayor interés por su impacto en la automatización de la generación de texto en distintos escenarios. Son varios los sistemas que buscan producir textos a partir de datos en el área del deporte. Aun así, no abundan los sistemas desarrollados en idioma español y que sean independientes de la fuente de los datos. En el presente trabajo de tesis se propone un diseño de sistema para la generación automática de resúmenes de enfrentamientos deportivos independiente de la fuente de datos. Se propone un esquema general para definir las entradas específicas por deporte siguiendo una estructura de tuplas de conocimiento. Se presenta una propuesta de diseño para los modelos específicos de los deportes. La propuesta se valida a través de la implementación de un sistema que genera resúmenes de partidos de fútbol y combates de boxeo. Los modelos de generación siguen el estándar basado en reglas y plantillas.

Abstract

Natural Language Generation, as a subfield of Artificial Intelligence and computational linguistics, has aroused increasing interest due to its impact on the automation of text generation in different scenarios. There are several systems that seek to produce texts from data in the area of sport. Even so, there are not many systems developed in Spanish and that are independent of the source of the data. In this thesis work, a system design is proposed for the automatic generation of summaries of sports matches independent of the data source. A general scheme is proposed to define the specific entries by sport following a structure of knowledge tuples. A design proposal for specific sports models is presented. The proposal is validated through the implementation of a system that generates summaries of soccer matches and boxing matches. The generation models follow the standard based on rules and templates.

Índice general

Índice de figuras

Introducción

Desde hace varios años la Inteligencia Artificial (IA) viene revolucionando e impactando significativamente en muchas esferas de la vida del hombre. Dentro de la IA uno de los campos que más actividad tiene es el Procesamiento de Lenguaje Natural (PLN). Este abarca el conjunto de técnicas computacionales que tienen como objetivo el trabajo con el lenguaje humano, que van desde la extracción de entidades en textos hasta modelos de comprensión y generación de textos. Tanto la generación de texto a texto (*text-to-text*, en inglés) como la generación de datos a texto (D2T por sus siglas en inglés, *data-to-text*) son instancias de la Generación de Lenguaje Natural (GLN). Reiter y Dale [38] caracterizan la GLN como el subcampo de la IA y la lingüística computacional que se ocupa de la construcción de sistemas informáticos que pueden producir textos comprensibles a partir de alguna representación no lingüística subyacente de la información. Esta definición se adapta fácilmente a los sistemas cuya entrada consiste en datos y es asumida en este trabajo para referirse a los sistemas de GLN.

Existe consenso en la forma en que la salida de un sistema de GLN debe presentarse: texto. Sin embargo, no hay establecido un estándar en cuanto a la forma en que se presentan los datos para su procesamiento, variando de un sistema a otro [39, 15]. Como regla general el texto producido por estos sistemas debe mantener fidelidad a los datos que lo originan y debe ser consecuente con su intención comunicativa [39], no siendo lo mismo un sistema para la generación de diálogos que uno que tiene como objetivo describir resúmenes biográficos. Propuestas tempranas de sistemas como Ana [24] para generar reportes financieros, o como FoG [16], generador de reportes climáticos, comenzaron a mostrar las capacidades de estos modelos a la hora de dotar de interpretabilidad y relevancia datos que se presentaban de forma repetitiva y tediosa.

En un contexto donde la producción de datos se ha acelerado como resultado de los avances tecnológicos y la digitalización de los sistemas industriales, se ha hecho necesario para las empresas el manejo y la interpretación de los mismos. Por esa razón, algunas empresas se han especializado y han comenzado a brindar estos servicios a

otras [8]. Un ejemplo es la compañía *Automated Insights*[1] que ha enfocado su negocio en brindar soluciones a otras corporaciones con vista a automatizar sus procesos de producción de texto. Entre los casos de uso clásico de estas soluciones encontramos, en el marco del comercio electrónico, la generación de descripciones de productos a partir de sus fichas técnicas. El periodismo ha sido otra de las esferas beneficiadas de estos avances en la GLN. La generación automática de noticias (conocida como periodismo robótico) está cada vez más extendida, al punto de que importantes editoriales como *The Washington Post* han creado sus propios sistemas para la generación de texto a partir de datos[2]. En este caso, su sistema se apoda *Heliograf* y les permite cubrir todos los partidos de fútbol americano de las escuelas secundarias del área de Washington DC cada semana.

Motivaciones

A partir de las perspectivas que se abren en el campo de la GLN, surge la motivación del presente trabajo. Siendo el deporte un campo que despierta tanto interés y que es fuente de entretenimiento de muchas personas, sentar las bases del diseño de un sistema para la generación de resúmenes en español de eventos deportivos es un reto estimulante. Por su carácter estadístico y su gran audiencia, el deporte es una de las esferas que se puede beneficiar claramente del desarrollo de los sistemas de generación automática de texto. Muchos trabajos se han enfocado en este tema [45, 26, 19]. Y aunque es relativamente sencillo encontrar en la web los datos o tablas estadísticas de un determinado enfrentamiento deportivo, la gran variedad de eventos que se suceden constantemente hacen que sea humanamente imposible darle cobertura a cada uno de ellos a nivel de narración y resumen. Por esta razón es motivante desarrollar sistemas que sean capaces de cubrir muchas esferas del deporte.
Asimismo, este trabajo se presenta en el marco de las líneas de investigación existentes en el grupo de Inteligencia Artificial de la Facultad de Matemática y Computación de la Universidad de La Habana (MATCOM) y puede servir de base para futuros trabajos que sigan ampliando y profundizando en la GLN.

Antecedente

La intención de adentrarse en la investigación de los sistemas de GLN dentro del departamento de IA de MATCOM comenzó con la propuesta, en 2019, de realizar un modelo GLN capaz de dar cobertura a las actuaciones destacadas de los peloteros cubanos en las Grandes Ligas de Béisbol. La misma derivó en el trabajo de tesis de

[1] https://automatedinsights.com/

[2] https://www.washingtonpost.com/pr/wp/2017/09/01/the-washington-post-leverages-heliograf-to-cover-high-school-football/

Roberto Balboa González [3]. Este primer acercamiento a la GLN sirvió para buscar nuevos escenarios a abarcar desde el punto de vista de esta disciplina.

Problemática

Aún con el desarrollo de los sistemas de GLN y su mayor asimilación en diversos ámbitos, siguen siendo absolutamente predominantes los sistemas que tienen el inglés como idioma de referencia a la hora de generar el texto de salida. En la literatura consultada no abundan las soluciones en lenguaje español en este campo, entre otras razones influenciado por la gran complejidad estructural de este lenguaje, así como el menor número de herramientas específicas para este. De la misma forma esto impacta directamente en los sistemas que tienen como objetivo comunicativo los eventos deportivos. Siendo el deporte un objeto de mucho interés en la comunidad hispanohablante en general y en Cuba en particular, se hace necesario ampliar los escenarios existentes desde esta perspectiva.

Los sistemas analizados en la literatura consultada en su mayoría se basan en un conocimiento explícito del dominio a tratar así como en una estructuración predefinida de los datos en base al dominio. A su vez, son muchas las propuestas que parten de la obtención de los datos desde la fuente como parte propia del sistema y no propiciados por el usuario. Son pocos los sistemas funcionales, fuera de la industria, capaces de desacoplarse de las fuentes de datos y de abarcar, desde una misma estructura de entrada de los datos, distintos dominios.

Teniendo en cuenta los antecedentes, las problemáticas, y partiendo de la hipótesis de que es posible definir un esquema general para representar los datos que describen los enfrentamientos deportivos, se arriba al objetivo del presente trabajo.

Objetivo

Proponer el diseño de un sistema para la generación de resúmenes o reportes de eventos deportivos independiente de la fuente de datos del dominio.

Para la consecución del objetivo es necesario:

- Analizar el dominio y determinar características comunes y relevantes de los eventos deportivos.
- Crear un esquema general a partir del cual poder definir esquemas específicos para la entrada de datos de los distintos deportes.
- Proponer un diseño general para los modelos de generación de resúmenes.

- Validar la propuesta con dos modelos de generación de texto basados en reglas y plantillas para el fútbol y el boxeo.

Estructura del trabajo

El resto del trabajo se organiza de la siguiente manera. El capítulo 1 aborda los problemas generales de los sistemas de GLN y las distintas técnicas relevantes en la literatura para su solución. En el capítulo 2 se presenta la propuesta de meta esquema para la representación de los datos de entrada, la metodología para definir los esquemas específicos, así como la propuesta de diseño para la general de los modelos de generación de resúmenes. En el capítulo 3 se presenta la validación de la propuesta a través de la exposición de los esquemas de definición y los modelos generadores para el fútbol y el boxeo. Mientras en el capítulo 4 se analizan los detalles de la implementación del prototipo de sistema creado y se muestran los resultados textuales de los modelos tratados en el capítulo 3. El trabajo concluye con la presentación de las conclusiones y recomendaciones, así como la bibliografía consultada.

Capítulo 1

Estado del Arte

Los sistemas de GLN normalmente reciben los datos de forma estructurada, pero esta estructura puede variar de un sistema a otro. La naturaleza de los datos es diferente con lo cual sus representaciones pueden ser desde registros de bases de datos, bases de conocimiento, grafos de conocimiento, o estructuras intermedias que utilizan representaciones del tipo clave-valor como pudiera ser el formato JSON, entre otras. Por ejemplo, en algunos obtienen los datos en forma de series temporales [48], mientras que otros los trabajan a partir de una base de datos de registros financieros [24].

En cuanto a la arquitectura, el diseño de sistemas de GLN es un campo abierto donde no existe un amplio consenso. Hay una diversidad de propuestas e implementaciones que varían en dependencia del desarrollador y del problema para el cual se crea el sistema de GLN. En este sentido, es difícil identificar elementos comunes y proporcionar una abstracción completa que sea aplicable a la mayoría de éstos sistemas [36].

Sin embargo, la arquitectura propuesta por Riter y Dale en [38] se ha tomado como el estandar de facto dentro de este campo, lo que no implica que muchos sistemas no plantearan estructuras diferentes [32]. Esta arquitectura de flujo (*pipeline* en inglés), o modular, establece que los sistemas de GLN pueden dividirse en módulos que encapsulen distintas tareas dentro del proceso de generación del texto.

1.1. Tareas principales de los sistemas de Generación de Lenguaje Natural

Hasta hoy, considerado como el punto de referencia y texto más completo en el campo de GLN [15], el libro de Ehud Riter y Robert Dale: Building Natural Language Generation Systems [39], sienta las bases y define las que se han asumido como tareas

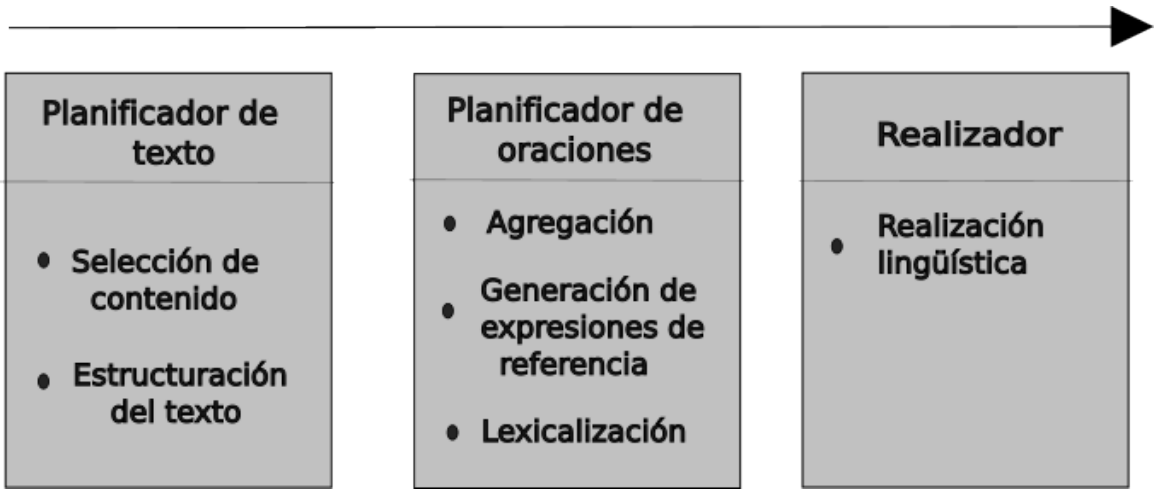


Figura 1.1: Arquitectura modular [38].

principales de un sistema de GLN:

- Determinación del contenido: decidir qué información es relevante.
- Estructuración del texto: determinar en qué orden se presentará la información.
- Agregación: decidir qué información se presenta en cada oración.
- Lexicalización: encontrar las palabras y frases adecuadas para expresar información.
- Generación de expresiones de referencia: selección de las palabras y frases para identificar entidades del dominio.
- Realización lingüística: conformación del texto con oraciones bien formadas.

Estas tareas naturalmente se pueden dividir entre las que envuelven el proceso de elegir qué decir y de qué forma (planificación del contenido) a partir de los datos, y las que tienen un carácter más orientado al proceso lingüístico de elegir los términos adecuados para la expresión en sí (realización) [15]. Cada una de estas tareas, aunque no presentes en su conjunto en todos los sistemas, responden a distintos objetivos dentro de los mismos y existen distintas técnicas para cada una de ellas en la literatura.

1.1.1. Determinación del contenido

Cada sistema de GLN tiene un objetivo comunicativo, lo cual hace necesario determinar qué información del dominio es relevante y debe estar representada en el texto final [39]. Los sistemas pueden presentar una especificación de los datos de la entrada o seleccionar un subconjunto de los mismos [39].

El reconocimiento de patrones es una de las técnicas utilizadas con este fin. SumTime-Turbine [48] es un ejemplo de sistema funcional para crear informes sobre un mecanismo de turbinas de gas. En un contexto donde la cantidad de datos es muy grande y en su mayoría irrelevante, los autores con ayuda de los expertos del dominio crean un modelo que permite reconocer patrones en los datos así como luego, utilizando una base de datos de patrones antiguos determinar los patrones relevantes a ser expresados en el informe.

Existen muchos sistemas de GLN que utilizan modelos basados en reglas como única metodología para la selección de contenido [39, 32]. Bouayad-Agha, Casamayor y Wanner [5] presentaron el diseño teórico de un sistema para generar resúmenes de partidos de fútbol de la Liga Española. Para la conformación del mismo describieron la construcción de una gran base de conocimientos del dominio así como un estricto sistema de reglas para la selección de contenido. Es relevante este enfoque, ya que con un conocimiento del dominio a tratar y de la intención comunicativa del texto a producir es posible implementar reglas que den lugar a la selección del contenido relevante [38, 39].

Como el proceso de creación de reglas puede ser tedioso y es dependiente del dominio [38], aparecieron iniciativas para la automatización de esta tarea. Un enfoque basado en técnicas de aprendizaje automático es el presentado por Duboué y McKeown [11]. En este trabajo plantearon un modelo de aprendizaje supervisado que a través de un corpus de resultados deseados (texto escrito por humanos), aparejados con los tipos lingüísticos de la entrada (distintos tipos de datos que recibe el sistema), determina un conjunto de constantes. Estas constantes expresan si determinado dato de entrada debe aparecer o no reflejado en la salida y bajo qué condiciones. Este sistema se utilizó para la generación de descripciones biográficas cortas que resumen hechos importantes sobre personajes famosos.

1.1.2. Estructuración del texto

La estructuración del texto o planificación del discurso es el proceso donde se da orden y estructura al conjunto de mensajes a expresar en el texto producido. En un texto la información se presenta en un orden particular y, por lo general, hay una estructura subyacente a la presentación. La complejidad de la estructura de los textos puede variar de un sistema a otro. Una buena estructuración puede hacer que un texto sea mucho más fácil de leer [38].

Los primeros enfoques para la estructuración de documentos se basaron en reglas estructuradas hechas a mano dependientes del dominio [15]. A este enfoque se le conoce como el enfoque basado en esquemas, nombre que se derivó del trabajo de Kathleen R. McKeown [28] cuando acuñó el término *"schematta"*. En la construcción de su sistema TEXT [28], McKeown, luego del análisis de muchos ejemplos del dominio, concluyó que dado un objetivo comunicativo, la información tiende a transmitirse en el mismo orden. En base a esto definió estructuras (esquemas) que determinan posibles combinaciones de atributos, formando patrones y plantillas. De esta forma, el sistema, dada una intención comunicativa, puede seleccionar un esquema que defina la forma de transmitir la información. La mayoría de los sistemas que siguen este enfoque utilizan las Matrices de Valores de Atributos (AVMs por sus siglas en inglés, *Attribute Value Matrics*) [32].

Las estructuras retóricas son otro de los mecanismos que se utilizan para la planificación. Estas se derivan del trabajo de Mann y Thompson quienes introdujeron la Teoría de la Estructura Retórica(RST por sus siglas en inglés, *Rhetorical Structure Theory*) [27]. Las estructuras retóricas constituyen un método lingüístico para la descripción de texto caracterizando las estructuras primarias del mismo y estableciendo relaciones funcionales entre sus distintas partes. La RST tiene como base los conceptos de núcleo y satélite que definen las partes del texto entre las que se establece una relación. Los distintos sistemas que utilizan las estructuras retóricas para la estructuración del texto definen qué tipo de relaciones establecen. Ejemplos de relaciones lingüísticas planteadas por Mann y Thompson en su trabajo son: motivación, causa, condición, circunstancia, entre otros [27].

1.1.3. Agregación

En un texto, cada parte de información no tiene que estar presente en oraciones independientes. Hay escenarios donde es deseable que distintos mensajes sean transmitidos en una misma oración. La agregación puede permitir crear textos de mayor calidad o eliminar repeticiones innecesarias que vayan en contra de la fluidez del mismo [15]). Reape y Mellish [37] realizaron un estudio sobre la tarea de agregación dentro de los sistemas de generación de texto, distinguiendo entre la agregación a nivel semántico (más dependiente del dominio) y a nivel sintáctico. Muchos de los primeros trabajos sobre agregación fueron dependientes del dominio, centrados en la aplicación de reglas (por ejemplo, "si un jugador marca dos goles consecutivos, exprésalo en la misma frase").

Un ejemplo de agregación, en el dominio del fútbol, describiendo el hecho de dos anotaciones consecutivas de un jugador, pudiera ser:

- (1) Ronaldo anotó para el Real Madrid en el minuto 2. Ronaldo anotó para el Real Madrid en el minuto 8.

- (2) Ronaldo anotó dos veces para el Real Madrid antes del minuto 8.

En el segundo caso, se evita la repetición y la información se presenta de forma más fluida y natural al lector.

Los primeros trabajos de este tipo se basaron generalmente en la aplicación de reglas hechas a mano (ejemplo [42]). Con el tiempo aparecieron propuestas que utilizan enfoques de aprendizaje automático. SPoT [46] constituyó uno de los primeros sistemas entrenados que incluye la tarea de agregación. Los autores plantearon una metodología basada en la producción de varios textos para una misma entidad informativa utilizando diferentes cláusulas de agregación asociadas al dominio. Después utilizaron un modelo entrenado para dar un valor a cada una de las salidas estableciendo un ranking a partir del cual hacer la selección. Mientras, Barzilay y Lapata [4] plantearon el problema en términos de optimización global. Realizan una clasificación inicial sobre pares de entradas de la base de datos que determina si deben agregarse o no en función de su similitud por pares. Posteriormente, seleccionan un conjunto globalmente óptimo de entradas relacionadas en función de un grupo de restricciones.

Con la agregación sintáctica podría decirse que es más factible definir reglas independientes del dominio para eliminar la redundancia [15]. Esto podría lograrse identificando las frases verbales paralelas en las dos oraciones conjuntas y eliminando el sujeto y el verbo en la segunda. Por ejemplo, convertir (3) en (4):

- (3) Ronaldo marcó en el minuto 2 y marcó de nuevo en el minuto 8.
- (4) Ronaldo marcó en el minuto 2 y de nuevo en el 8.

1.1.4. Lexicalización

La lexicalización es un proceso muy importante dentro de un sistema de GLN. Es el proceso durante el cual se seleccionan la palabra o palabras que expresan un concepto o relación [38]. Una de las complicaciones del proceso de lexicalización está dada porque una misma relación puede ser expresada de distintas formas. Por ejemplo, el evento de la anotación de un gol en un partido de fútbol puede ser expresado como: ¨marcar un gol¨, ¨poner el balón en la red¨, ¨conseguir una anotación¨. La complejidad de este proceso depende en gran medida del número de alternativas que el sistema pueda o quiera contemplar. Las restricciones contextuales también juegan un papel importante a la hora de expresar un mensaje. Por ejemplo, la expresión ¨marcó un gol¨ es desafortunada si el evento descrito es un gol en propia puerta [15].

El proceso de lexicalización puede seguir dos vertientes principales. Una sería la realización de la lexicalización de la forma más simple posible, lo cual se lleva a cabo generalmente utilizando técnicas para el llenado de plantillas lexicalizadas. De otra forma se puede realizar este proceso en mayor profundidad utilizando técnicas

más complejas que permitan, por ejemplo: la eliminación de palabras innecesarias, la selección de vocablos que maximicen la efectividad del objetivo comunicativo del texto o la unión de términos que se aparejan frecuentemente en el dominio [32].

Los enfoques basados en reglas son de los más utilizados, pudiendo variar en complejidad entre un sistema y otro. *EasyText*[10] es un ejemplo de sistema que utiliza reglas para la lexicalización, pero que da un paso más allá pues consume de una base de datos léxica creada, principalmente, por lingüistas. Esta alternativa permite una lexicalización más avanzada, pero a su vez es altamente costosa en recursos.

La utilización de ontologías también está presente en este proceso de los sistemas de GLN. El uso de ontologías permite al sistema ganar en adaptabilidad, ya que encontrar una ontología para un dominio determinado es más sencillo que encontrar un corpus para el mismo. Asimismo, ofrecen una mayor cobertura de las representaciones semánticas que los corpus [32]. Cimiano [7] introdujo un modelo que utiliza este enfoque presentado en el dominio de las recetas de cocina.

1.1.5. Expresiones de referencia

Robert Dale y Ehud Rither describieron la generación de expresiones de referencia dentro de un sistema de GLN como la tarea de identificar la expresión a utilizar, comprensible de cara al usuario, para identificar a una instancia del dominio [39, 15]. Los primeros métodos para la selección de referencias fueron los algoritmos generativos que en común presentan la necesidad de tener conocimiento contextual y de propiedades de las entidades [15]. De este orden es el algoritmo incremental cuya base se planteó en [9]. El algoritmo, conociendo la entidad a referenciar (objetivo), el resto de entidades (llamadas distractores) y el grupo de propiedades que definen entidades en el dominio, busca determinar un conjunto de propiedades únicas que definan al objetivo y lo diferencien del resto.

Muchos de los trabajos que versan sobre este tema hacen énfasis en un tipo determinado de referencia [14]. La selección puede ser de un pronombre (él/ella), una descripción (Simón, el jugador cubano) o en la generación de nombres propios (Frederich Cepeda/Cepeda). Rither y Dale [39] plantean una diferencia entre una referencia temprana (primera vez que se menciona una entidad en el texto) y una tardía (cuando se refiera a una entidad mencionada anteriormente). Plantearon el uso de los nombres propios a la hora de introducir una entidad, para luego, a través de claúsulas, seleccionar un pronombre apropiado, ejemplo:

```
si el referente fue mencionado en la oración anterior;
entonces utiliza un pronombre
```

Es necesario también considerar los escenarios donde la generación de expresiones de referencia pudiera llevar a ambigüedades:

- Benzema anotó dos para el Real Madrid mientras Luka Modric brindó dos asistencias. Él fue elegido el jugador del partido...

Siddharthan y col. [43] realizaron un estudio empírico del comportamiento de las referencias hacia personas basadas en su nombre propio en el contexto de los artículos de noticias. Para ello utilizaron un corpus de noticias en inglés de diferentes agencias de prensa y cuantificaron las diferentes formas de referencia según el momento referencial de la instancia (temprana o tardía). Como resultado de este trabajo arrojaron que el nombre completo de la entidad suele utilizarse como primera referencia en prácticamente totalidad de los casos, mientras que en su mayoría el apellido se utiliza cuando se trata de una referencia tardía.

1.1.6. Realización lingüística

El proceso de realización lingüística es el que da lugar a la formación final del texto expresado en oraciones con una estructura gramatical correcta y coherente con el mensaje a transmitir. Esta tarea implica ordenar los constituyentes de una oración, así como generar las formas morfológicas correctas (incluidas las conjugaciones y la concordancia de los verbos). A menudo, los realizadores también necesitan insertar palabras funcionales (como verbos auxiliares y preposiciones) y signos de puntuación [15].

Una de las técnicas más utilizadas es la que incluye el uso de plantillas predefinidas para expresar mensajes [15]. Las plantillas, aunque requieren una carga intensiva de trabajo para lograr mayor variabilidad en el texto a producir, permiten un control total sobre la calidad y la correctitud del texto a elaborar. Un ejemplo de plantilla:

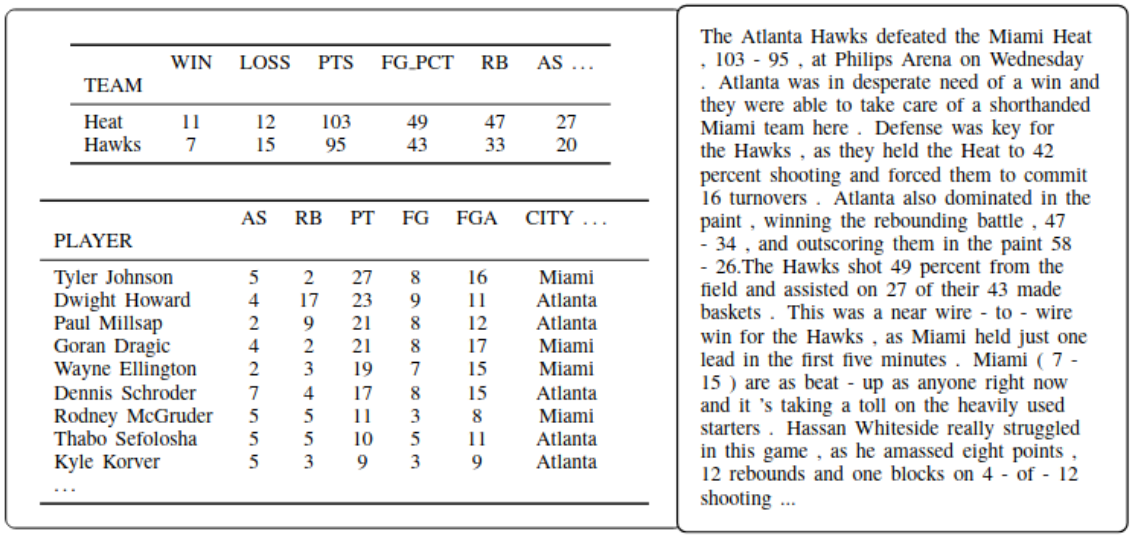
```
$equipo_1 venció $pts_equipo_1 a $pts_equipo_2 a $equipo_2
```

Dicha plantilla, que representa el resultado de un enfrentamiento entre dos equipos se completa con la información extraída de los datos y a la hora de la realización el resultado pudiera ser el siguiente:

```
Industriales venció 10 a 1 a Granma
```

1.2. Propuestas para la generación de texto basadas en redes neuronales

Al igual que en otras áreas del procesamiento de lenguaje natural, el dominio de la GLN se ha visto impactado por el auge de las soluciones basadas en redes neuronales [15, 41]. Mientras los sistemas tradicionales de D2T siguen una estructura modular



TEAM	WIN	LOSS	PTS	FG_PCT	RB	AS ...
Heat	11	12	103	49	47	27
Hawks	7	15	95	43	33	20

PLAYER	AS	RB	PT	FG	FGA	CITY ...
Tyler Johnson	5	2	27	8	16	Miami
Dwight Howard	4	17	23	9	11	Atlanta
Paul Millsap	2	9	21	8	12	Atlanta
Goran Dragic	4	2	21	8	17	Miami
Wayne Ellington	2	3	19	7	15	Miami
Dennis Schroder	7	4	17	8	15	Atlanta
Rodney McGruder	5	5	11	3	8	Miami
Thabo Sefolosha	5	5	10	5	11	Atlanta
Kyle Korver	5	3	9	3	9	Atlanta
...						

The Atlanta Hawks defeated the Miami Heat , 103 - 95 , at Philips Arena on Wednesday . Atlanta was in desperate need of a win and they were able to take care of a shorthanded Miami team here . Defense was key for the Hawks , as they held the Heat to 42 percent shooting and forced them to commit 16 turnovers . Atlanta also dominated in the paint , winning the rebounding battle , 47 - 34 , and outscoring them in the paint 58 - 26.The Hawks shot 49 percent from the field and assisted on 27 of their 43 made baskets . This was a near wire - to - wire win for the Hawks , as Miami held just one lead in the first five minutes . Miami (7 - 15) are as beat - up as anyone right now and it 's taking a toll on the heavily used starters . Hassan Whiteside really struggled in this game , as he amassed eight points , 12 rebounds and one blocks on 4 - of - 12 shooting ...

Figura 1.2: Instancia de RotoWire [47].

con etapas bien definidas (estructuración, realización, etc), los modelos neuronales variaron el camino hacia estructuras *end-to-end* (de extremo a extremo, en español), unificando en muchos casos, varias tareas en un solo paso de entrenamiento.

Los trabajos que siguieron este enfoque [25, 29, 47] plantearon modelos *Seq2Seq* (de secuencia a secuencia, en español) que adoptan la influyente estructura de codificador-decodificador [44]. Utilizan una red neuronal recurrente (RNN, *Recurrent Neural Network*, en inglés) para codificar la entrada de un vector de representación el cual sirve a su vez como entrada auxiliar de otra RNN que hace de decodificador y produce el texto, y no tienen módulos específicos para mejorar la calidad del resultado más allá de los mecanismos genéricos de atención y copia [2, 17]. La popularidad de los modelos *end-to-end* se vio impulsada y a la vez creó la necesidad de contar con corpus de datos paralelos que permitieran evaluar la calidad de las propuestas. Estos conjuntos de datos tenían que tener la suficiente cantidad de instancias para poder entrenar modelos de estas características. RotoWire [47] es un punto de referencia ampliamente utilizado, que se construyó con cerca de cinco mil pares de datos estadísticos de partidos de baloncesto y su correspondiente resumen descriptivo escrito por profesionales. MLB [34] es otro corpus de reciente creación, cuenta con aproximadamente 25 mil instancias de estadísticas aparejadas con descripciones, en este caso, en el dominio del beisbol. Otros conjuntos de datos hechos a mano como E2E [31] se han construido para analizar tareas específicas como la capacidad de realización lingüística.

Weiseman y col. [47] mostraron la capacidad de estos modelos de producir un texto dotado de mayor fluidez en comparación con propuestas tradicionales. A su vez,

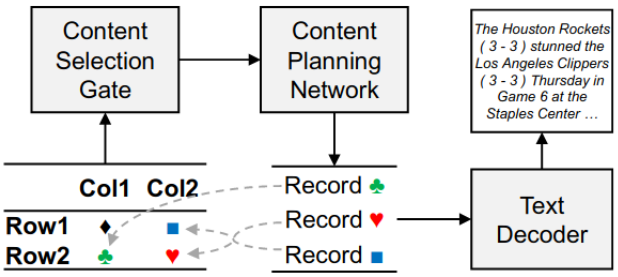


Figura 1.3: Arquitectura del modelo presentado por Puduppully y col. [33].

además de ser propensos a la alucinación (es decir, generan texto que no es compatible con la entrada), mostraron deficiencias a la hora de seleccionar el contenido y/o estructurar el documento (por ejemplo, en el marco de una descripción biográfica, la omisión de la fecha de nacimiento es un relevante, mientras que de aparecer no es apropiado que lo haga en la última oración del escrito). Para mejorar esta situación, Puduppully y col. [33] presentaron una arquitectura que incorpora selección y planificación de contenido sin sacrificar el entrenamiento *end-to-end*. Descompusieron la tarea de generación en dos etapas. Dado el corpus de registros de datos (junto con las descripciones), primero generaron un plan de contenido destacando qué información debía mencionarse y en qué orden para luego generar el documento teniendo en cuenta dicho plan. Un enfoque similar lo plantearon Chen y col. [6], mientras que Moryossef y col. [30] desacoplaron por completo ambas fases, dejando solo el modelo neuronal para la realización del texto una vez desarrollada la estructura informativa.

La utilización de modelos preentrenados en tareas de generación de texto-a-texto para producir texto a partir de datos en base al reentrenamiento fue de los enfoques que se exploraron recientemente. Kale y Rastogi [22] utilizaron el modelo T5 (*"Text-to-Text Transfer Transformer"*, su nombre en inglés) [35] preentrenado para tareas de generación de resúmenes, traducción entre idiomas, clasificación de texto, entre otras. Sus resultados fueron alentadores sobre todo desde el punto de vista de la generalización. Este modelo no solo tuvo una buena actuación en cuanto a la calidad del texto generado, sino que también mostró buena adaptación a dominios distintos a los del reentrenamiento, lo cual constituyó una diferencia respecto a los modelos neuronales tratados anteriormente.

Como quedó patente en varios trabajos [47, 13, 12, 41] los sistemas basados en redes neuronales son capaces de producir texto que supera en fluidez y en evaluaciones humanas, como la naturalidad, a los sistemas tradicionales. Sin embargo, estos modelos siguen lidiando con su principal problema que es la presencia de alucinaciones en las salidas. Esta deficiencia es objeto de interés de los investigadores en este campo

[21]. Por esta razón, Robert Dale, en su *Natural language generation: The commercial state of the art in 2020* [8] planteó que estas soluciones, aunque prometedoras, se encuentran en fase principalmente académica y que en la industria, entorno donde la fidelidad de los datos es crucial, su adopción general no ha llegado aún.

1.3. Sistemas para la generación de texto a partir de datos

Al referirse al campo de GLN, se puede afirmar que el mismo es ya un campo investigativo consolidado dada la cantidad de sistemas que han sido implementados y la variedad de los dominios de su aplicación práctica. FOG (*Forecasting Generator*, en inglés) [16] es de los primeros sistemas funcionales para la generación de texto a partir de datos. Este, al igual que SumTime [40], genera breves pronósticos del tiempo a partir de los valores de variables meteorológicas.

En la escena industrial, propiciado por el desarrollo constante de herramientas para el análisis y captura de datos, hubo empresas que detectaron en la generación de texto un nicho de mercado no cubierto y prometedor [8]. Compañías como *Ax Semantic*[1] y *Narrativa*[2] ofrecen soluciones para automatizar la descripción de productos para el comercio electrónico. *Automated Insights* creó su plataforma para la generación de lenguaje natural, *Wordsmith*. Este software da la posibilidad al usuario de crear un conjunto de plantillas basadas en reglas que describen los datos que se desea tratar. A partir de las mismas, y en dependencia de su sofisticación los textos generados pueden llegar a ser indistinguibles de los textos redactados por un humano. Precisamente en colaboración con *Automated Insights*, la gran corporación de prensa *Associated Press (AP)* generó las vistas previas de los partidos de una temporada regular de baloncesto, liberando a los periodistas de este trabajo. A su vez, *Yahoo!* utiliza esta tecnología para crear informes de jugadores y resúmenes de partidos de fútbol del juego *Fantasy Football* que es de gran interés para sus usuarios. Otro caso de uso dentro de la automatización de contenido en la prensa lo encontramos en *PostData*[3], un sitio de periodismo de datos cubano, donde utilizan un modelo propio, *ArmandBot* [3], para generar reportes sobre las actuaciones de los peloteros cubanos en grandes ligas.

En la literatura hay propuestos varios modelos de sistemas para la GLN en el ámbito del deporte. Por ejemplo, Kanerva y col. [23] apoyados en la construcción de un corpus de 2000 encuentros de hockey sobre hielo, proponen un modelo de secuencia a secuencia, para generar narrativas sobre esta práctica deportiva. Hasan [20] propuso un sistema basado en plantillas que sigue la arquitectura tradicional

[1] https://en.ax-semantics.com/
[2] https://www.narrativa.com/
[3] http://www.postdata.club/index.html

basada en módulos para generar texto en inglés y bengalí[5] para generar resúmenes de juegos de críquet a partir de datos estructurados obtenidos de la web. Siguiendo el mismo enfoque, se presentó otro sistema basado en plantillas para el críquet, en este caso al estilo propio de Sri Lanka [17].

En el caso del fútbol, una de las principales propuestas fue GoalGetter [45], un sistema de datos a voz en idioma neerlandés, que consta de dos módulos, uno para transformar los datos en texto y otro para luego llevar el texto a sonido. GoalGetter tomó los datos de una página de Telexto que contenía la información de uno o más partidos de fútbol. Una característica de este sistema es que propuso una arquitectura diferente al estándar modular de otros sistemas. En este caso, utilizó un solo módulo para la generación de texto que consumía una base de conocimiento con los nombres de los jugadores y equipos junto a un sistema de plantillas sintácticas. Basado en este trabajo se propuso GameRecapper [1], un sistema capaz de generar resúmenes de partidos de fútbol de la liga portuguesa en idioma portugués. Este sistema obtenía los datos de *www.zerozero.pt*, una página web que contenía la información de los partidos de cada jornada. Al módulo de generación agregaron a su vez una base de conocimiento y un conjunto de funciones léxico semánticas para mejorar la calidad del texto. De este trabajo resaltó el hecho de que hicieron una caracterización más amplia del evento principal del fútbol, el gol, en su sistema de plantillas. Para cada escenario distinto propusieron la creación de plantillas oracionales específicas (primer gol del partido, último gol, gol que empata, entre otros).

PASS [26] es una propuesta también basada en GoalGetter que busca diferenciar el texto producido en base a la audiencia del mismo. Para ello establecen un sistema de plantillas que abarca distintos posibles escenarios de resultado de un partido: victoria o derrota del equipo local, victoria o derrota del equipo visitante o empate. La elección de cuál plantilla utilizar va en dependencia de a qué afición irá dirigido el texto.

La gran mayoría de estos sistemas cuentan con un dominio específico de aplicación, así como cuentan con una fuente de datos de la cual extraer la información a desarrollar. Por tanto, y aunque fuera posible la adaptabilidad, existe una dependencia de la fuente de información. A su vez, el tener un dominio bien definido permite agregar bases de conocimiento previo del mismo que enriquezcan los modelos. Predominan los sistemas basados en plantillas y reglas, variando entre sí su nivel de complejidad. Estos sistemas son especialmente adaptables y prácticos para idiomas distintos del inglés debido a que las herramientas de generación específicas desarrolladas son menores en comparación [19].

[5]El bengalí es el idioma nacional y el idioma oficial de la República Popular de Bangladesh

Capítulo 2

Propuesta

El objetivo de proponer el diseño de un sistema para generar resúmenes de eventos deportivos independiente de la fuente de datos planteó distintos retos. El primero de estos fue la necesidad de definir un esquema que permitiera abstraer las características generales del conjunto de deportes de enfrentamiento (enfrentamiento dos a dos). Junto con dicho esquema se necesitó definir una estructura común para la entrada de los datos que permitiera expresar los conocimientos del dominio. Se seleccionó una estructura basada en tuplas de cuatro elementos (cuatro-tuplas en lo adelante). A partir de esta estructura y en base al esquema de definición general se buscó poder determinar esquemas específicos para cada deporte que se fuera a incluir en el sistema. Cada uno de estos esquemas específicos son los que se encargan de definir un deporte de forma individual dentro del sistema. Luego se realiza una propuesta general de diseño para concebir los modelos generadores de un deporte.

En segunda instancia, se construye, en base a la propuesta, un esquema y su modelo correspondiente para generar resúmenes de partidos de fútbol. Lo mismo se realizó con un deporte de naturaleza diferente como el boxeo.

2.1. Propuesta de Esquema General

Los deportes se pueden clasificar en la categoría de individuales o colectivos. En los deportes colectivos, las representaciones del enfrentamiento ocurren en base a equipos que agrupan a individuos. A su vez, en los deportes individuales son dos los contendientes. Esta es la primera diferencia que se extrae en el análisis del conjunto de deportes. Las modalidades analizadas están representadas en 2.1, clasificadas en individuales o colectivas.

De cada uno de estos deportes se analizó:

- Naturaleza de decisión: La mayoría de los deportes se definen como juegos ad-

Tabla 2.1: Deportes analizados

Colectivos	Individuales
Béisbol, Voleibol, Fútbol, Tenis Dobles, Baloncesto, Waterpolo, Balonmano, Hockey	Tenis, Esgrima, Boxeo, Judo, Lucha libre, Taekwondo

versariales por acumulación de puntos. La entidad con mayor puntuación gana. Otros, como el tenis y el voleibol se definen por cantidad de etapas ganadas (sets), y cada etapa se gana por puntos. A su vez, en el boxeo la definición se deriva de votaciones de árbitros.

- Posibilidad de empate: Hay deportes como el fútbol en el que, según la competición o la fase de ésta, existe la posibilidad de definirse sin ganadores ni perdedores.
- División de los eventos: La mayoría de los eventos se divide por etapas de tiempo. constante. Una excepción es el judo que ocurre de forma continua durante cuatro minutos.
- Alargues de tiempo: La mayoría de los deportes, en caso de no definición en su tiempo reglamentario, presentan etapas adicionales en forma punto de oro (ej. judo), tiempos extras (ej. béisbol, fútbol), tiebreak (desempate, ej. voleibol, tenis).
- Roles: Dentro de los deportes los participantes ejercen roles, como puede ser su posición en los deportes colectivos. En los deportes individuales estos roles no son tan explícitos.
- Acciones principales: La definición de los eventos son las acciones relevantes que ocurren durante el tiempo de juego.

Del análisis también se extrajeron un conjunto de características que son comunes a los enfrentamientos deportivos: la sede, el público, la fecha. Asimismo, los enfrentamientos normalmente se encuadran dentro de un torneo, y existen distinciones entre categorías lo mismo sea de edad, sexo, u de otro tipo (ej. peso).

A partir del análisis se definió un meta esquema general de tipos de entradas basado en una estructura de cuatro-tuplas de conocimiento.

Tabla 2.2: Meta esquema general para definir las entradas de cada deporte

Tipo de Entrada	Estructura
SEDE	(TipoSEDE, Nombre Asistencia Capacidad)
TORNEO	(TipoTORNEO Nombre Expresión de Género Expresión de Categoría)
ENFRENTAMIENTO	(TipoENFRENTAMIENTO Entidad_1 Entidad_2 Expresión de Fecha)
ROLENJUEGO	(TipoROLENJUEGO Entidad del Rol Entidad Complementaria Rol Complementario)
RESULTADOPARCIAL	(TipoRESULTADOPARCIAL Entidad Indicador de parcial Expresión de puntuación)
RESULTADOFINAL	(TipoRESULTADOFINAL Entidad Expresión de puntuación Descriptor de resultado)
EVENTO	(TipoEVENTO Expresión de Tiempo Entidad Protagonista Entidad Complementaria)

Cada cuatro-tupla tiene en la primera posición el tipo de entrada. El resto de los valores constituyen la base de información. Con cada tipo de entrada se encapsula un subconjunto de la información que se muestra, común al conjunto de deportes estudiados.

Ejemplos abstractos de formación de entradas

Se presenta una meta representación de entradas basadas en el esquema general y su interpretación en el contexto del sistema.

- (SEDE, A, 1450, 1700) : El enfrentamiento ocurre en la sede de nombre A, con capacidad para 1700 espectadores, asistieron 1450.
- (TORNEO, B, F, categoría_1) : El enfrentamiento pertenece al torneo B, femenino, en la categoría categoría_1.
- (ENFRENTAMIENTO, contrincante_A, contrincante_B, 11-11-2022) : Se enfrentan contrincante_A y contrincante_B el 11 de noviembre de 2022.
- (ROLENJUEGO, individuo_A, entidad_A, segundo_rol) : El individuo_A desarrolla primer_rol y segundo_rol respecto a entidad_A.
- RESULTADOPARCIAL:
  - (RESULTADOPARCIAL, contrincante_A , P, X): En el parcial P, contrincante_A tiene X puntos.
  - (RESULTADOPARCIAL, contrincante_B , P, Y): En el parcial P, contrincante_B tiene Y puntos.
- RESULTADOFINAL:
  - (RESULTADOFINAL, contrincante_A, X, Derrota): El contrincante_A perdió con X puntos.
  - (RESULTADOFINAL, contrincante_B, Y, Victoria): El contrincante_B ganó con Y puntos.
- EVENTO:
  - (EVENTO, tiempo_x , individuo_A, “”): En el tiempo_x, el individuo_A protagonizó el EVENTO
  - (EVENTO, tiempo_y, individuo_A, individuo_B): En el tiempo_y, individuo_A protagonizó el EVENTO en complemento de (en oposición de, en beneficio de, en perjuicio de, en relación con, respecto a) individuo_B.

A partir del meta esquema general es posible definir el diseño de los esquemas específicos de cada deporte, con sus tipos particulares para cada entrada y su forma de interpretar cada uno de los valores. Los esquemas de cada deporte tienen que ser capaces de expresar la información del mismo y, a través de ella, generar textos que describan el enfrentamiento.

2.2. Metodología para la conformación de los esquemas específicos

Primero, es necesario tener en cuenta que el meta esquema planteado anteriormente busca la abstracción de conceptos comunes. Estos conceptos necesitan, al menos los referentes a los roles, eventos y resultados, ser llevados a su expresión específica dentro de una modalidad deportiva. A su vez, se deben diferenciar los conceptos de: capacidad de representación y obligación de representación. Que el esquema permita definir un tipo de información determinada no significa que todos los deportes expresen necesariamente ese concepto. Además, es posible que se conciban distintos esquemas específicos para un mismo deporte. Esto depende de cómo cada modelo sea representado.

La representación de un deporte en un esquema específico a partir del esquema general propuesto necesita del análisis de sus características. Se debe realizar un estudio que permita detallar las situaciones que presenta el deporte y representarlas basado en eventos. A partir de estudiar las reglamentaciones se separa al deporte en cuanto a su categoría: individual o colectivo.

Para los deportes colectivos la expresión de los roles de los deportistas se encuentra mínimamente definida a partir del concepto de alineación. Esta serie de deportes tienen un conjunto de individuos que inician las disputas de los encuentros y otros que ingresan a raíz de decisiones que se toman durante el transcurso de los mismos. Además, se pueden expresar conceptos como las disposiciones que ocupa cada deportista dentro del equipo. En este tipo de información, la *entidad complementaria* que define la tupla de *ROLENJUEGO* sería el equipo del deportista. En el caso de los deportes individuales, los roles no se expresan tan claramente. Aun así, es posible identificar roles de representación, ya sea de un país, una delegación, un equipo multi categoría. Un ejemplo fuera de los deportes de enfrentamiento se encuentra en la fórmula 1, donde los competidores representan a escuderías durante las carreras.

En lo referido a los parciales, se necesita determinar las etapas en las que trascurre un enfrentamiento en caso de que este ocurra por etapas. La información de los parciales permite al sistema desambiguar situaciones que ocurren durante los enfrentamientos, así como da la posibilidad de dotar de más información la narrativa. Para conformar las tuplas de *RESULTADOPARCIAL*, es necesario determinar si existe

uno o más tipos de segmentación dentro del enfrentamiento, así como lograr una expresión identificativa que sea única para cada una.

La expresión de los eventos es la que dota principalmente de capacidad descriptiva a los modelos. Los eventos, acciones que se suceden en un deporte, son la esencia de este y por esa razón son la información fundamental que sobre ellos se transmite, más allá del resultado. Para expresar los eventos es necesario en primera instancia determinar cuáles son los que existen dentro de la modalidad seleccionada. A partir de esto, definir si para su expresión es necesario el concepto de antagonista como sujeto no protagonista en la acción. También se debe determinar una expresión temporal que identifique de forma única y cronológica la secuencia de eventos. De esta forma no se generan ambigüedades a la hora de que los modelos interpreten los mismos.

Los datos referentes a la sede, el público, el torneo, el resultado y las categorías se expresan de forma más directa. Queda en decisión del realizador del esquema y su modelo específico determinar qué informaciones constituyen un requerimiento en el contexto de la generación del resumen y cuáles son complementos informativos. Es decir, el modelo sería capaz de lidiar con la ausencia de determinados datos.

2.3. Propuesta de diseño para los modelos generadores

A partir de la definición de un esquema para la conformación de las tuplas de conocimiento de un deporte determinado se debe concebir un modelo que transforme esa entrada en un resumen textual. En la sección se propone un enfoque adaptable para los modelos de generación siguiendo los requerimientos de los sistemas de GLN y las propuestas presentes en la literatura para idiomas distintos al español.

Como paso previo necesario para la conversión de los datos en resúmenes textuales se define qué información es relevante a incluir en la salida y bajo qué estructura. Como los enfrentamientos deportivos son eventos repetitivos, cuyo conocimiento está bien definido, el enfoque basado en corpus es adaptable para la planificación del contenido. Reiter y Dale [39] plantearon que tras el análisis de un conjunto de textos que aborden el dominio es posible determinar la estructura subyacente, así como la información relevante a incluir. En los reportes deportivos es posible identificar una estructura con una presentación, donde se incluye el resultado e información general, seguido de una mención de los eventos de mayor importancia.

La arquitectura general planteada se presenta en 2.1 y constituye una adaptación de la propuesta de Aires [1] que a su vez sigue un enfoque basado en la propuesta de Theune y col. [45]. A diferencia de las mismas, el sistema propuesto no consume bases de conocimiento específicas, ya que busca adaptarse al hecho general del deporte a abordar. Un sistema sencillo para generar expresiones de referencia se incluye con el

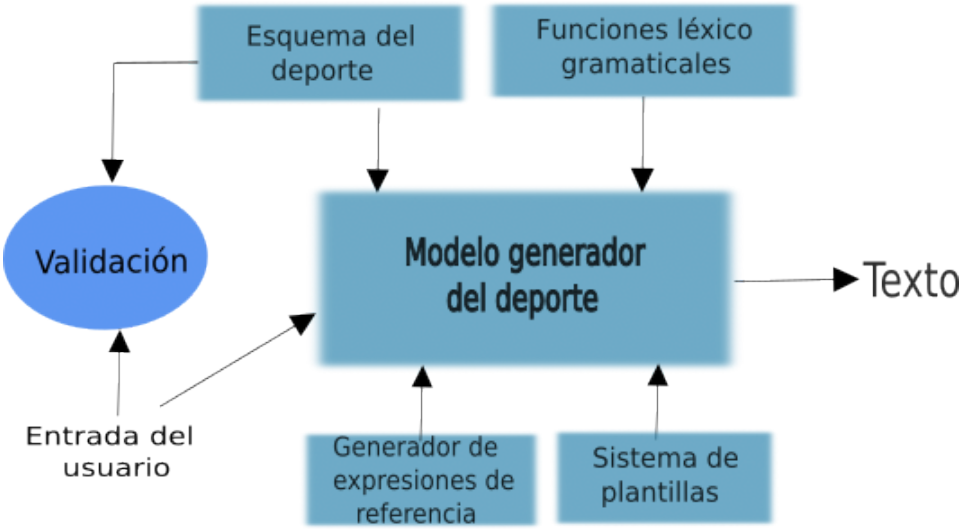


Figura 2.1: Arquitectura de modelo propuesta

objetivo de que los modelos generen un texto más fluido.

Como se analizó en el capítulo anterior (1), las tareas de los sistemas de generación de texto a partir de datos se pueden dividir en dos etapas: la referente al contenido a representar y su estructura (planificación del contenido) y la de realización lingüística. La planificación del contenido viene dada por la estructura extraída de los textos analizados. Para la etapa de realización, se propone un sistema basado en reglas y plantillas.

Realización lingüística

En la etapa de realización del texto se ven unificadas las tareas de carácter lingüístico tal y como se presentó en 1. En este proceso se determinan las palabras y expresiones con las que exponer el contenido seleccionado bajo la estructura definida. El enfoque basado en reglas y llenado de plantilla es de los más utilizados por el control que otorga sobre la producción del texto (1.1.4). Tiene la ventaja de que se asegura la calidad estructural del texto producido, así como permite dotar de variabilidad a las salidas tal y como se discutió en el capítulo anterior (1.1.6).

Las plantillas se utilizan dentro del modelo para conformar estructuras más complejas en forma de oraciones. Cada conjunto de plantillas dentro del sistema pertenece a una parte de la estructura del texto. Para cada contexto se conciben varias plantillas que brindan opciones para expresar la información relativa a una idea.

Las plantillas pueden o no presentar ranuras para completar con información. Las ranuras se utilizan de la forma: $< dato >$ donde *dato* se sustituye por el valor de la

variable que representa. Para dotar de facilidad al sistema a la hora de adaptarse a eventos de ambos géneros, se puede utilizar dentro de las plantillas expresiones como *$ expresión dependiente del género $*, donde *"expresión dependiente del género"* se sustituye por su expresión de género correcta dentro del contexto a través de una de las funciones léxicas. El caracter "@" también se utiliza dentro de las plantillas para hacer distinciones de género. Por ejemplo en la frase: *amb@s se golpearon*, el "@" se sustituye por "a" o por "o" en dependencia del género.

Funciones lingüísticas

Las funciones lingüísticas tienen el objetivo de mejorar la calidad del texto producido. Asimismo, buscan asegurar su correcta estructura gramatical. Una vez constituida una oración a partir de unificar plantillas de expresiones, una función se encarga de dotar de un formato a la misma. Se colocan las mayúsculas correspondientes al inicio, así como los puntos finales. Otra función se emplea para eliminar errores gramaticales o de estructura que se presentan durante la unión de las plantillas. Se eliminan los espacios en blanco múltiples, se corrigen los signos de puntuación así como los artículos repetidos y las construcciones mal formadas, como por ejemplo "de el".

Otras funciones léxicas se utilizan para dar mayor fluidez al texto, como las que transforman expresiones numéricas en texto, por ejemplo, "3" por "tres".

Generador de expresiones de referencia

Las expresiones de referencia son las que permiten identificar unívocamente a una entidad dentro de un contexto [39, 15]. Un sistema de expresiones de referencia debe determinar si la entidad a referenciar ha sido mencionada previamente, ya que existe una diferenciación entre la introducción de una entidad y su referencia tardía, tal y como se trató en el capítulo anterior (1.1.5). A su vez, la expresión seleccionada para referirse a una entidad debe tener en cuenta el resto de entidades del dominio. Por ejemplo, en un evento donde "Alejandro González" y "Pedro González" sean protagonistas, la expresión "González" resulta ambigua. Se propone un generador de expresiones de referencia basado en los nombres propios de los integrantes del enfrentamiento, utilizando los apellidos como referencias tardías y el nombre completo en la introducción.

Capítulo 3

Esquema y modelo para el fútbol y el boxeo

Para validar la propuesta presentada en el capítulo anterior, se planteó la concepción de dos modelos generadores que partieran de la representación de los datos a partir del meta esquema propuesto. Se seleccionaron el fútbol y el boxeo como deportes sobre los cuales concebir el sistema. Se selecciona el fútbol porque es un deporte colectivo, con una dinámica de acciones constantes, con muchos eventos de distinta naturaleza y distintas etapas de definición del resultado. Mientras, el boxeo es una contraparte en cuanto a que es un deporte individual, de carácter más monótono respecto a las acciones que lo definen así como presenta un sistema de definición variado y de mayor complejidad respecto a otros deportes.

Para ambos casos se modeló un esquema de entrada de datos siguiendo la metodología propuesta.

3.1. Definición y modelo generador para el fútbol

Es necesario, antes de la introducción del modelo, determinar el esquema específico que se utilizó para definir y validar las posibles entradas del sistema.

3.1.1. Esquema de entrada de datos

La experiencia del autor sobre el dominio a tratar, así como la consulta del sitio web *whoscored.com* sirvieron para realizar la caracterización del fútbol como deporte. Trabajos previos que también abordaron este deporte para la generación de texto [45, 1, 26] se tuvieron en cuenta. El sitio *whoscored.com* publica estadísticas y anotaciones de los eventos que ocurren durante un partido de fútbol y dispone de información de miles de encuentros.

Tabla 3.1: Esquema de definición del fútbol

Tipo de Entrada	Valores en el Esquema del Fútbol
SEDE	Estadio
TORNEO	Torneo
ENFRENTAMIENTO	PartidoPresentación
ROLENJUEGO	Titular Suplente
RESULTADOPARCIAL	Tiempos TiemposExtras Penaltis
RESULTADOFINAL	ResultadoFinal
EVENTO	PaseBueno,PaseFallado,PaseClave,PaseAsistencia, TiroPuerta,TiroNoPuerta,Gol,Atajada, EntradaConExito,EntradaFallada,BloqueoDeDisparo, Recuperación,FaltaFueraDelArea,ManoFueraDelArea, FaltaDentroDelArea,ManoDentroDelArea,PenaltiCometido, AtajadaPenalti,GolPenalti,TiroPenaltiFuera,TarjetaAmarilla, SegundaTarjetaAmarillaRoja,TarjetaRojaDirecta, GolTiroLibre,Autogol,CobraCorner,Cambio,FueraDeJuego

A partir de este análisis, se definió el esquema del fútbol como se muestra en 3.1

Cada esquema presenta sus especificaciones en cuanto a la intrepretación o la representación de los valores por tipo de entrada. Este esquema establece las siguientes especificaciones:

- Tupla **SEDE**: Las expresiones de *asistencia* y *capacidad* de no ser vacías deben ser expresiones numéricas (ej. “1200”).
- Tupla **TORNEO**: La expresión de *género* debe ser una entre “M” y “F” (masculino, femenino).
- Tupla **ENFRENTAMIENTO**: Las entidades son los nombres de los equipos que se enfrentan. La expresión de *fecha*, de ser incluida, debe seguir el siguiente formato: “AAAA-MM-DD HH:MM” (año-mes-día hora:minutos). Es posible que se provea solo el día o la hora, respetando sus formatos específicos.
- Tupla **ROLENJUEGO**: La entidad de Rol es el jugador, y la entidad complementaria su equipo. El *Rol Complementario* indica la posición que desempeña el jugador en el terreno. Debe ser uno entre: *“POR”* (portero), *“DEF”* (defensa), *“CEN”* (centrocampista) o *“DEL”* (delantero).

- Tupla **RESULTADOPARCIAL**: La entidad es el equipo al que se refiere, el *indicador de parcial* un valor entre 1 y 5 que se refiere al número de la etapa. Para los *Tiempos* los valores son 1 y 2, para los *TiemposExtra*, 3 y 4, y para *Penaltis*, el 5.
- Tupla **RESULTADOFINAL**: La entidad es el equipo al que se refiere, la *puntuación* es la cantidad de goles anotados por el equipo. El *descriptor de resultado* es uno entre "V" (victoria), "E" (empate) , "D" (derrota).
- Tupla **EVENTO**: La *expresión temporal* tiene el siguiente formato: "p_MM_SS", donde p es un indicador del segmento de juego en el que ocurrre la acción. La entidad *protagonista* es el jugador que, valga la redundancia, protagoniza el evento, mientras que el *complementario* es el jugador que complementa la información del evento y se requiere principalmente en acciones como el "Cambio", respondiendo a ¿de quién?, ¿por quién?.

Como se busca que el sistema pueda ser adaptable por el usuario a distintas fuentes de datos, no es obligatorio proveer todo el conjunto de información para obtener un resultado. En cambio, sí existe un conjunto minimal de información que el sistema requiere, como el resultado, el torneo y la estructura de los equipos. Una formación inadecuada de las tuplas de los eventos conllevará a incongruencias del texto producido respecto al partido en cuestión, pero no es objetivo de este trabajo evaluar la factibilidad de la obtención de los datos o su adaptabilidad a la estructura planteada.

3.1.2. Generación del reporte

Con el objetivo de generar reportes resumidos de partidos de fútbol a partir de las tuplas de conocimiento de los partidos, el sistema propuesto sigue las pautas planteadas en el capítulo anterior.

El módulo de generación hace uso de plantillas de expresiones creadas a mano. Con las plantillas se busca la construcción de oraciones que expresen distintos eventos relevantes que ocurren dentro del partido. Para la selección de los eventos y la determinación de las piezas importantes dentro de la estructura se analizó un conjunto de crónicas de partidos. Se busca que el texto producido tenga cierto grado de variabilidad y que sea correcto en cuanto a estructura y gramática.

Planificación de contenido

En orden de estructurar el texto a producir así como determinar los eventos relevantes de un partido se hizo un análisis basado en corpus [39]. Se seleccionaron 4

Tabla 3.2: Análisis de eventos reportados en crónicas de fútbol.
AS:Diario AS, ES: ESPN Deportes

	Resumen de 15 juegos			
Medio	**AS**		**ES**	
Eventos	Frecuencia en reportes	%	Frecuencia en reportes	%
Goles	52/52	100	52/52	100
Penaltis Marcados	6/6	100	6/6	100
Penaltis No marcados	2/2	100	1/2	50
Asistencias	25/40	62.5	27/40	67.5
Expulsiones	6/7	87.5	5/7	71.4

jornadas (las 22,23,24,25) de la Liga Española de Fútbol Profesional de la temporada 2021-2022. Asimismo, se seleccionaron dos medios de prensa que realizaron crónicas en español de estos partidos en su versión web. Estos medios fueron: el *Diario AS*[1], en su versión web *As.com*, y el sitio deportivo *ESPN Deportes*[2]. Para determinar qué eventos eran narrados en cada crónica se estudiaron los eventos ocurridos en cada partido anotados por una tercera fuente: el sitio estadístico *www.whoscored.com*. De los 40 juegos se hizo una selección de 15 que tuvieran presente la información en ambos medios para poder constrastar. Los resultados de las menciones de cada evento se pueden obeservar en la tabla (3.2).

Se determinan tres piezas comunicativas: la presentación del resultado, la descripción de los eventos relevantes y la mención de jugadores con una destacada actuación. En base a esto se definió la siguiente estructura:

- Presentación del partido: Una primera oración donde se presentan los equipos y el resultado final del encuentro.
- Mención de los eventos relevantes: Se van contando los eventos relevantes del partido en orden cronológico.
- Jugador más destacado: Se hace una mención del jugador más destacado del encuentro en base a las estadísticas extraídas de los eventos. En caso de no existir una actuación destacada se expresa este hecho.

[1]https://as.com/
[2]https://espndeportes.espn.com/

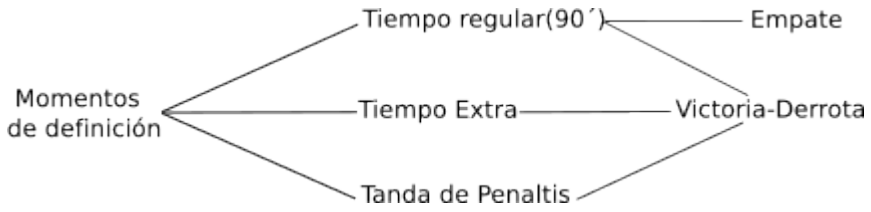


Figura 3.1: Momentos de definición de un partido de fútbol

Presentación

Los juegos de fútbol son disímiles en cuanto a los momentos de decisión. Estos juegos, lo mismo pueden pertenecer a eventos tipo Liga, donde los partidos duran 90 minutos y existen empates, o a eventos de eliminación directa donde los juegos deben tener una definición ya sea en tiempos extras o en tandas de penales (3.1). Para cada uno de estos tipos de definición se definen plantillas específicas a la hora de construir la presentación. Asimismo, en la presentación se incluye la información respectiva a la sede donde se desarrolla el encuentro.

Eventos relevantes

Se determinó como eventos relevantes a incluir en el resumen aquellos que tienen una influencia directa tanto en el resultado como en el transcurso del juego. Por tanto los eventos seleccionados son: los goles, las asistencias, las expulsiones y los fallos de penaltis. Cada uno de ellos se presenta en oraciones independientes y poseen plantillas de acuerdo a su clasificación. En la figura 3.2 se muestran las distintas clasificaciones de los eventos relevantes.

Jugador destacado

El conjunto de acciones que se pueden describir del juego (ver eventos en 3.1) se utilizaron para determinar el jugador más destacado del partido. Se planteó una heurística basada en asignar distinta aportación de valor a las acciones. Así mismo, acciones negativas se penalizan con evaluaciones negativas. En 3.3 se muestran los valores utilizados.

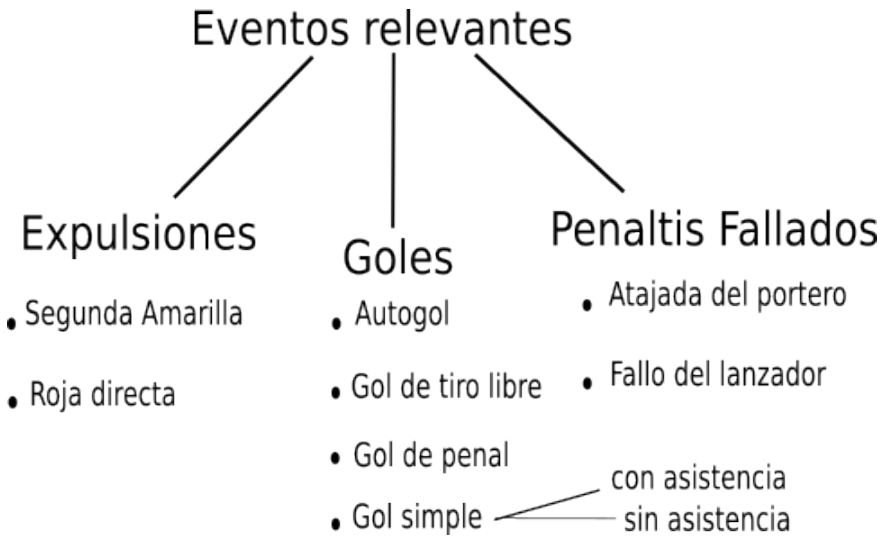


Figura 3.2: Clasificación de los eventos relevantes

Tabla 3.3: Valores que aporta cada estadística

Estadística	**Valor aportado**
Goles	x3
Autogoles	x(-3)
Asistencias	x1.5
Atajadas	x1
Penales atajados	x2
Pases claves	x0.25
Entradas buenas	0.1

Estadística	**Valor aportado**
Recuperaciones	x0.2
Pases buenos	x0.02
Pases malos	x(-0.1)
Tiros a puerta	x0.2
Tarjeta roja	x(-2)
Tarjeta amarilla	x(-0.5)

Realización

Siguiendo la propuesta planteada, la realización se realiza por cada una de las partes definidas en la estructura. Se definen los conjuntos de plantillas que lexicalizan las diferentes expresiones y a partir de ellas se conforman estructuras en forma de oraciones.

Plantillas de Presentación

En el texto de presentación se busca expresar la información concerniente al resultado del partido, su circunstancia, así como el lugar donde se realiza este. Se plantea en forma de oración que sigue una de estas estructuras de forma aleatoria:

- *plantilla de estadio* + ", " + *plantilla de resultado*
- *plantilla de resultado* + *plantilla de estadio*

Las *plantilla de estadio* se escogen en dependencia de si se conoce o no el valor de la asistencia al partido, pudiendo tener las siguientes expresiones:

- *con* < *asistencia* > *hinchas en las gradas del estadio* < *estadio* >
- *en el estadio* < *estadio* >

Las *plantilla de resultado* se construyeron en distintos grupos en dependencia del resultado del partido y el momento de definición:

- Se define en tiempo regular: Estos pueden ser empate o con ganador.
  - Si el resultado es empate, se tienen los subgrupos: empate con goles, empate con uno o dos goles, empate con más de dos goles.
  - Si hay un ganador definido, se tienen dos subgrupos: resultado normal, resultado por goleada. Se toma como resultado por goleada cuando la diferencia es de tres o más goles
- Se define en tiempo extra
- Se define en tanda de penaltis.

Ejemplos de construcción:

- *con* < *asistencia* > *hinchas en las gradas del estadio* < *estadio* >, *en tiempo extra* < *equipo_ganador* > *superó* < *equipo_ganador_goles* > *a* < *equipo_perdedor_goles* > *a* < *equipo_perdedor* >

- *< equipo_ganador > goleó a < equipo_perdedor > < equipo_ganador_goles > a < equipo_perdedor_goles > en el estadio < estadio >*

Plantillas de Eventos principales

Las expulsiones se expresan a través de un de las siguientes estructuras:

- *plantilla tiempo + plantilla tipo expulsión*
- *plantilla tipo expulsión + plantilla tiempo*

Las *plantillas de tiempo* son del tipo: *cuando corría el minuto < minuto >*
Los penaltis fallados siguen una estructura similar:

- *plantilla tiempo + plantilla tipo fallo*
- *plantilla tipo fallo + plantilla tiempo*

Ejemplos de construcciones:

- Expulsión por segunda amarilla: *una segunda amarilla a < jugador_expulsado > dejó a < equipo_jugador_expulsado > con < jugadores_restantes > sobre el terreno en el minuto < minuto >*
- Penalti atajado: *en el < minuto >, < lanzador > tuvo la oportunidad para < equipo_lanzador > desde el punto de penal pero < portero > le detuvo el lanzamiento*

Para la expresión de los goles se utilizan las plantillas: *tipo de gol*, *resultado del gol*, *asistencia*, *tiempo*. Para los goles se identifican cuatro tipos: autogol, gol de tiro libre, gol de penal y gol simple. A su vez, las circunstancias del partido condicionan a la hora de expresar el resultado de un gol. Se identifica que un gol puede ser el primer gol del partido, o el único gol del partido, o puede empatar un resultado, desempatar un resultado, aumentar la ventaja o disminuir la ventaja. Se escoge entre 14 estructuras de oraciones de forma aleatoria a la hora de realizar una acción de gol. Se presentan tres de estas estructuras, las restantes son variaciones estructurales de las mismas.

- *"con " + plantilla_tipo_gol + " " + plantilla_asistencia+ ", "+ autor_del_gol + " " + plantilla_resultado_de_gol + " " + plantilla_tiempo*
- *plantilla_tiempo + ", " + plantilla_tipo_gol + " de "+ autor_del_gol + ", " + plantilla_asistencia + ", " + plantilla_resultado_de_gol*

- *autor_del_gol + " con " + plantilla_tipo_gol + " " + plantilla_asistencia + ", " + plantilla_resultado_de_gol + " " + plantilla_tiempo*

Cuando un gol no tiene asistencia, este valor queda en blanco.
Ejemplos de construcciones:

- *con un libre directo, autor_del_gol aumentó la ventaja de < equipo_del_gol > en el minuto < minuto >*
- *autor_del_gol con un buen disparo a pase de < asistente >, disminuyó la distancia en el marcador a los < minuto > minutos*

Plantillas de Jugador destacado

La estructura de la oración del jugador más valioso utiliza: *plantilla más valioso*, *plantilla estadísticas* junto con el *nombre*, *equipo* y la expresión de la *posición* del jugador. Cuando hay un jugador destacado se elige aleatoriamente sobre cuatro variaciones de esta estructura:

- *nombre_del_jugador + ", " + posición_del_jugador + " de " + equipo + ", fue " + plantilla_más_valioso + " con " + plantilla_estadísticas*

Ejemplo de formación de la oración:

- *< nombre_del_jugador >, < posición_del_jugador > + de < equipo >, fue la figura del partido con < cantidad_goles > goles y una asistencia"*

Cuando no hay un jugador destacado, lo cual es casi garantía de que el partido tubo uno o ningún gol en base a la definición de la heurística planteada, se selecciona una *plantilla de partido sin jugadores destacados*, que es una oración en sí.

Ejemplo de oración de partido sin jugadores destacados: *en un partido parejo, no hubo ninguna actuación individual relevante*

3.2. Definición y modelo generador para el boxeo

A diferencia del fútbol que fue tratado anteriormente, el boxeo es un deporte individual, cuya estructura de acciones es más monótona. Asimismo, las crónicas que tratan el boxeo, al menos las analizadas por el autor, no son tan variables y poseen un matiz informativo más restringido y directo.

Tabla 3.4: Esquema de definición del boxeo

Tipo de Entrada	**Valores en el Esquema del Boxeo**
SEDE	Sede
TORNEO	Evento CombateOrganizado
ENFRENTAMIENTO	CombateContrincantes
ROLENJUEGO	RepresentaciónDePaís
RESULTADOPARCIAL	Asaltos
RESULTADOFINAL	VeredictoFinal
EVENTO	JabEfectivo, JabDerechaEfectivo, JabIzquierdaEfectivo, JabNoEfectivo, CrochetEfectivo, CrochetNoEfectivo, CrochetDerechaEfectivo, CrochetIzquierdaEfectivo DirectoEfectivo, DirectoNoEfectivo, DirectoDerechaEfectivo, DirectoIzquierdaEfectivo, GanchoEfectivo, GanchoNoEfectivo, GanchoDerechaEfectivo, GanchoIzquierdaEfectivo, SwingEfectivo, SwingNoEfectivo, SwingDerechaEfectivo, SwingIzquierdaEfectivo, GolpeDerechaEfectivo, GolpeIzquierdaEfectivo, Descalificado, NocautPropinado, NocautTécnicoSufrido, CuentaDeOchoSuperada, ConteoProtecciónSuperado, Rendición

3.2.1. Esquema de entrada de datos

Para analizar y hacer una abstracción de las características del boxeo y sus eventos más significativos se consultaron las crónicas sobre boxeo de *ESPN Deportes* y de *Fightnews*[1], un sitio web dedicado a la cobertura del boxeo desde 1999. Además, se consultaron y estudiaron las reglas del boxeo profesional. Se utilizó la página del Consejo Mundial de Boxeo[2] (*World Boxing Council*) como referencia para las distintas categorías de pesaje tanto en boxeo masculino como en boxeo femenino.

A partir de este análisis se plantea el esquema de definición para la entrada de datos específicos del boxeo que se puede observar en la figura 3.4.

Cada esquema plantea sus especificaciones en cuanto a los valores de entrada; éstas son las del esquema propuesto:

- Tupla **SEDE**: Las expresiones de *asistencia* y *capacidad* de no ser vacías deben ser expresiones numéricas (ej. “1200”).

[1]https://fightnews.com/
[2]https://wbcboxing.com/

- Tupla **TORNEO**: La expresión de *género* debe ser una entre "M" y "F" (masculino, femenino). La expresión de *categoría* se requiere y debe indicar una de las categorías de pesos permitidas en el boxeo profesional (ejemplo: mosca, superpluma, completo, entre otros). En su defecto, se puede expresar un indicador con el formato "xxx_lb" ("135_lb") del cual se infiere la categoría a partir del peso.

- Tupla **ENFRENTAMIENTO**: Las entidades son los nombres de los boxeadores que se enfrentan. La expresión de *fecha*, de ser incluida, debe seguir el siguiente formato: "AAAA-MM-DD HH:MM" (año-mes-día hora:minutos). Es posible que se provea solo el día o la hora, respetando sus formatos específicos.

- Tupla **ROLENJUEGO**: La entidad de rol es el boxeador, y la entidad complementaria el país que representa.

- Tupla **RESULTADOPARCIAL**: La entidad es el boxeador al que se refiere, el *indicador de parcial* un número positivo que indica el asalto. La expresión de *puntuación* debe seguir el formato "1_XX_2_YY_3_ZZ" (ejemplo: (1_10_2_09_3_10) que indica la puntuación de cada juez al boxeador en cuestión.

- Tupla **RESULTADOFINAL**: La entidad es el boxeador al que se refiere, la expresión de *puntuación* debe seguir el formato "1_XXX_2_YYY_3_ZZZ" ej(1_098_2_097_3_098) que representa las votaciones acumuladas de cada juez para el boxeador en cuestión. El *descriptor de resultado* tiene el formato "R_TR" donde "R" (Resultado) es uno entre "V" (victoria), "E" (empate), "D" (derrota) y "TR" (Tipo de resultado) uno entre "DU"(unánime), "DD" (dividida), "DM" (mayoritaria), "KO" (nocaut), "DS" (descalificación), "DT" (técnica), "DA" (abandono).

- Tupla **EVENTO**: La *expresión temporal* tiene el siguiente formato: "rr_sss" (ejemplo "04_060", al minuto del cuarto asalto) donde "rr" indica el asalto y "sss" los segundos transcurridos. La entidad *protagonista* es el boxeador que protagoniza el evento.

3.2.2. Generación del reporte

Para la generación de los resúmenes de las peleas de boxeo, el autor siguió, adaptando cuando fuera necesario, las pautas descritas durante la modelación. Se sigue un enfoque basado en reglas para la etapa de planificación. Para la realización lingüística también se utilizan un conjunto de plantillas predefinidas.

Planificación del contenido

A partir del análisis descrito en la sección anterior se determinó la estructura bajo la cual se va a realizar el reporte. De la lectura y estudio de las crónicas se obtuvo que muchas de ellas presentan una comunicación breve y directa en cuento a la información a brindar. Estos son dos ejemplos de reportes tomados desde cada uno de los medios:

- por *ESPN*: *"Alycia Baumgardner derrotó por decisión dividida a Mikaela Mayer en la pelea coestelar de una cartelera exclusiva de mujeres en la O2 Arena de Londres. Los jueces votaron (96-95, 96-95 y 93-97) a favor de Baumgardner, quien retuvo sus cinturones superpluma del Consejo Mundial de Boxeo (CMB) y (...)"*
- por *Fightnews*[1]: *" El invicto peso ligero Jamaine Ortiz (15-0-1, 8 KOs) venció por decisión unánime en diez asaltos a Nahir Albright (14-2, 7 KOs) el viernes por la noche en el Caribe Royale Resort en Orlando, Florida. Ortiz estuvo al mando todo el tiempo, ganando 98-92, 97-93, 97-93. Ortiz recogió el título vacante de la NABF."*

La estructura seleccionada para el reporte es la que se indica:

- Presentación: Se incluye la información referente a la sede del combate, la categoría, así como de los boxeadores y se indica el resultado y su naturaleza. Si se cuenta con información al respecto se da a conocer el tipo de golpeo que propicia las definiciones por nocaut.
- Votaciones de los jueces: Cuando el resultado se defina por las votaciones de los jueces, se incluye una oración con dicho veredicto.
- Derribos: Con el objetivo de dotar de más información el reporte, se incluyeron los derribos que ocurrieron durante los combates en caso de que los hubiera y estuvieran expresados en la entrada.

Presentación

El algoritmo del modelo de generación primeramente extrae los datos generales referentes a la sede, la categoría y los boxeadores. Luego, chequea el tipo de resultado que tuvo la pelea en función de lo expresado en las tuplas de resultado final. Comprueba si no es incongruente respecto a los eventos; ejemplo, no hay un nocaut en los eventos y el veredicto final indica que el combate se decidió por los jueces.

[1]Traducido por *Google Translate*

A partir de determinar el tipo de resultado es posible que se determine la plantilla correspondiente para expresarlo.

Votaciones de los jueces

A partir de conocer el tipo de resultado se determina si la decisión quedó en manos de los jueces. Si es así, se construye una expresión de la votación a partir de representar en una misma estructura los resultados expresados en las tuplas de resultados de ambos boxeadores.

Derribos

Para determinar los derribos ocurridos en el combate lo primero que se hace es ordenar por tiempo de ocurrencia de los eventos. A partir de identificar un *Conteo-ProtecciónSuperado*, se identifica el golpe previo y el asalto, datos con los cuales se identifica el derribo. Luego de identificados todos los derribos ocurridos, se utiliza la técnica de agregación basada en las reglas descritas (1.1.3). Si los dos boxeadores se derriban en el mismo asalto, este evento se realiza en una misma oración. A su vez, si un mismo boxeador derriba a su oponente en dos o más asaltos consecutivos estos derribos se expresan en la misma oración.

Realización

La realización se completa con la formación de las expresiones a partir de las plantillas predefinidas, hechas a mano. Para la realización el modelo se apoyó en funciones léxicas y gramaticales que dotan al texto de mejor calidad. Una de las funciones léxicas se utiliza para expresar el número de los asaltos en forma ordinal (ej. primer, segundo, tercer). A su vez, una de las funciones gramaticales se emplea para desambiguar las expresiones de género que se incluyen en las plantillas. Basado en el género de la modalidad y en el contexto de la expresión, esta función expresa el concepto en el género correspondiente (ej. boxeador y boxeadora, el púgil y la púgil).

Como se hizo con el modelo anterior, se realiza una breve descripción de la formación de las plantillas para la expresión de la información en las distintas oraciones.

Plantillas de Presentación

Para la presentación se utilizan las plantillas siguientes: *plantilla de categoría*, *plantilla de sede*, *plantilla de resultado*, *plantilla de tipo definición*, *plantilla de tipo de golpeo*. La forma de realizar cada tipo de resultado es diferente, porque, por ejemplo, la plantilla de *tipo de golpeo* solo se utiliza cuando se define el resultado por nocaut.

Para dotar al modelo de variabilidad no solo se construyeron al menos 4 plantillas de cada tipo, sino que además las oraciones se conforman con estructura variable jugando con las estructuras propias de las plantillas. Algunos ejemplos de estructura: “”

- *plantilla de sede* + “, ” + *plantilla de categoría* + “, ” + *plantilla de resultado* + “ ” + *plantilla de tipo definición*
- *plantilla de sede* + “, ” + *plantilla de categoría* + “, ” + *plantilla de tipo de golpeo* + “ ” + *plantilla de resultado*
- *plantilla de resultado* + “, ” + *plantilla de sede* + “, ” + *plantilla de tipo definición*+ “ ” + *plantilla de categoría*

Ejemplos de construcciones:

- *en el ring del < estadio >, en la categoría de peso < categoría >, < ganador > selló la victoria sobre < perdedor > por votación dividida de los jueces*
- *en el < estadio >, en la categoría de peso < categoría >, con un fuerte recto de derecha < ganador > noqueó a < perdedor >*

Con el objetivo de que el modelo tuviera más complejidad en cuanto a las estructuras de la salida, para la presentación se construyeron dos plantillas especiales con el fin de variar la forma de las oraciones. Estas fueron las *plantillas de presentación* y las *plantillas de postsede*. Ejemplos:

- *< boxeador_uno > y < boxeador_dos > se vieron las caras*
- *la velada tuvo como sede el < estadio >*

Este es un ejemplo de construcción utilizando estas plantillas:

- *< boxeador_uno > y < boxeador_dos > se vieron las caras en la categoría de peso < categoría >. Con un buen gancho < ganador > noqueó a < perdedor >. La velada tuvo como sede el < estadio >*

Plantillas de Votaciones de los jueces

La expresión de la votación de los jueces sigue una construcción más simple basada en solo dos grupos de plantillas. Las *plantillas de votación* y las *plantillas de complemento de votación*.

Las construcciones siguen una de estas estructuras:

- *plantilla de complemento de votación* + “ ” + *plantilla de votación*

- *plantilla de votación* + " " + *plantilla de complemento de votación*

Ejemplo de construcción:

- los jueces calificaron la pelea < votos > a favor de < ganador >

Plantillas de Derribos

Para expresar un derribo simple que ocurre en un determinado asalto, se utiliza: *plantilla de derribo*, *plantilla de asalto* y *plantilla de golpeo*. Se hacen variaciones sobre la siguiente estructura:

- *plantilla de asalto* + ", " + *plantilla de golpeo* + ", " + *plantilla de derribo*

Ejemplo de realización:

- en el < ordinal_round > asalto, de un derechazo, < protagonista > puso en conteo de protección a < antagonista >

Los derribos continuados se expresan con la *plantilla de derribos continuos* y la *plantilla de múltiples rounds*. Se estructuran:

- *plantilla de múltiples rounds* + " " + *plantilla de derribos continuos*
- *plantilla de derribos continuos* + " " + *plantilla de múltiples rounds*

Ejemplo de construcción:

- *< protagonista > hizo visitar la lona a < antagonista > en el segundo y cuarto asalto*

Para cuando existen derribos de ambos contrincantes en el mismo asalto, las *plantillas de derribo doble* se realiza como una única oración. Ejemplo: *amb@s púgiles se llevaron un conteo de protección en el < ordinal_round > asalto*

El comodín arroba ("@") se desambigua con una de las funciones gramaticales que utiliza el modelo.

Capítulo 4

Detalles de Implementación y Resultados

Para la validación de la propuesta planteada a partir del esquema general de definición de los datos de entrada se concibió un prototipo de sistema. Este implementa los modelos de generación para el fútbol y el boxeo y permite la interacción con los mismos. Para facilitar la interacción se implementó a su vez una interfaz gráfica sencilla. A continuación se presentan detalles generales del sistema, funciones de interés y una presentación de la interfaz creada. Finalmente se muestra el resultado del texto producido por los modelos implementados. Se utilizó *python* como lenguaje de programación.

4.1. Detalles generales del sistema

Para la representación de los esquemas y de los modelos se crearon dos clases abstractas, *SportSchema* y *SportModel*. En la figura (4.1) se representan cada una con la signatura principal.

Los esquemas, con el método *validate_signature*, validan la entrada en cuanto a su signatura, o sea, comprueban que su representación en cada uno de los valores se corresponda con la definición. La *definición* (método *definition*) de un esquema es el conjunto de valores admitidos por cada tipo principal (los tipos principales son los definidos en el meta esquema general). A su vez, el método *validate_requeriments* se utiliza para validar si en el conjunto de entrada existe un conjunto minimal de los datos a partir de los cuales es posible la generación de texto. El método *name* devuelve un nombre identificativo del esquema, que debe ser único. La funcionalidad de *validate* es una conjunción de los otros dos métodos de validación.

Los modelos para su correcto funcionamiento dependen de un primer llamado al método *build* previo a cualquier secuencia de *generate*. El método *build* de los modelos

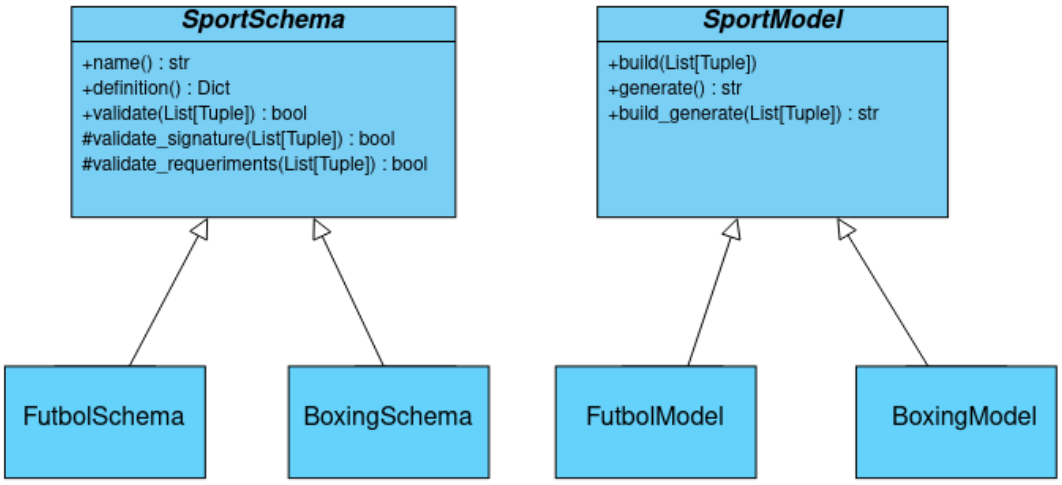


Figura 4.1: Definición de las clases principales

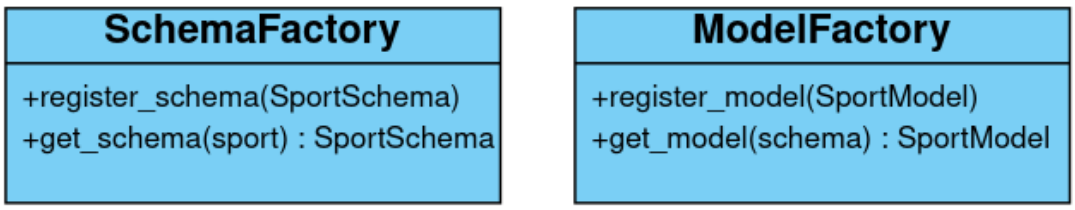


Figura 4.2: Definición de las clases *Factory*

es el que se encarga de procesar la entrada, hacer las transformaciones correspondientes de los datos. Durante su ejecución, los modelos completan toda la información necesaria para las distintas etapas de generación del texto que se definan. Para esto, se definen estructuras propias de cada modelo para representar la información. Con el uso de la información estructurada e interpretada en la construcción del modelo, con el método *generate* se generan las distintas piezas textuales que conforman el reporte. El método *build_generate* unifica un llamado a *build* seguido de uno a *generate*.

El flujo principal del programa se encuentra representado en el *Algoritmo Principal* a continuación. Se utilizó el patrón *Factory* (de Factoría, en español) para la selección de los modelos y los esquemas (Ver fig4.2). Se planteó de esta forma, ya que el prototipo podría adaptarse más fácilmente a nuevos modelos y esquemas sin necesidad de grandes cambios en el código.

```
Algoritmo principal

  Entrada: knowdlage_tuple:List[Tuple], selected_sport
  Salida : str

    schema = schema_factory.get_schema(selected_sport)
    if not schema.validate(knowdlage_tuple):
        mostrar error
        FIN
    model = model_factory.get_model(schema)
    summary = model.build_generate(knowdlage_tuple)
    return summary
```

4.1.1. Proceso de realización. Selección de plantillas

El proceso de realización lingüística se lleva a cabo utilizando un conjunto de funciones que permiten la expresión correcta de la información en forma de oraciones. Para la representación de los datos los modelos consumen un sistema de plantillas hechas a mano. Un ejemplo de plantilla:

$$< portero > atajó\ un\ penal\ a < lanzador > en\ el\ minuto < minuto >$$

Las expresiones entre $< >$ son las ranuras de las plantillas y se completan utilizando la información procesada de la entrada. Para facilitar la interpretación, a la expresión contenida entre $< clave >$ se le denominará clave de ranura.

El proceso de selección y llenado de las plantillas se realiza a través de la función *select_template* (Algoritmo de selección de plantilla). Esta recibe como parámetros el conjunto de posibles plantillas a emplear, el género a realizar, así como un diccionario de representación de la información donde las llaves coinciden con las posibles claves de ranura de las plantillas y los valores son los datos reales.

Como el sistema presenta cierto grado de sensibilidad ante la ausencia de algunos datos, es posible que haya plantillas para las que alguna ranura no tenga un valor real. La función *_is_valid_template* se utiliza para comprobar esto. El algoritmo selecciona aleatoriamente una plantilla del conjunto; si es válida, se selecciona, y si no, se descarta y se repite el proceso. Siempre se garantiza que habrá al menos una plantilla que sea funcional, ya que presentará las ranuras correspondientes al subconjunto minimal que admite el sistema en ese contexto. Para comprobar la validez de la plantilla primero se detectan las ranuras presentes en esta, utilizando la siguiente expresión regular: *(r'< (?P<key>[a-zA-ZáéíóúÁÉÍÓÚ_]*) >')*.

```
Algoritmo de selección de plantilla

  Entrada: template_group:List[str], data_values:Dict[str,str],
           genre:Genre.M | Genre.F
  Salida: str

    while not valid templat selected:
        template_selected = choice(tempalte_group)
        slots_values = []
        slots = slot_reggex.findall(template_group)
        for slot_value in slots:
            slots_values.append(slot_value)
        if __is_valid_template(slots_values, data_values):
            filled_template = fill_template(template_selected,
                                slots_values, data_values)
            return disambiguate_template(filled_template, gender)
    return
```

El método *fill_template* sustituye las ranuras de la plantilla por su valor real. Luego se pasa a desambiguar los términos de género. Como el sistema es adaptable tanto a la modalidad femenina como masculina de los deportes, es necesario realizar las expresiones en el género correcto. Las ranuras de género están presentes en algunas plantillas y tienen esta estructura: $ *clave* $. Estas ranuras se detectan igualmente utilizando una expresión regular similar a la vista para las ranuras comunes. El caracter "@" también se utiliza dentro de las plantillas para hacer distinciones de género. Este proceso se produce en la función *disambiguate_template*.

Funciones lingüísticas

Otras funciones se utilizan para mejorar el carácter léxico gramatical de las oraciones producidas. La función *ordinal* transforma expresiones numéricas en su expresión ordinal (primer, segundo, tercer, ..., vigésimo). La función *numeral* transforma un valor numérico en su expresión léxica (uno, dos, tres). Esta abarca los números desde el 1 al 20 y los múltiplos de 10 hasta 100. Una vez que todas las transformaciones sobre las plantillas han sido realizadas, la función *sentence_lexical_realization*, se utiliza para eliminar posibles errores, como espacios dobles, repetición de artículos ("el EL"), o corrige errores transformando expresiones como "de el" en su forma correcta "del".

Figura 4.3: Ejemplo de representación intermedia de una tupla de entrada en formato *json*

Generación de expresiones de referencia

La generación se expresiones de referencia llevada a cabo se realiza a partir de los nombres propios de los individuos. Se utiliza la clase *ReferentialExpressionGenerator* que se instancia con la lista de todos los nombres presentes. A partir de estos, utilizando el paquete de *python*, *nameparser* [18], se separa el nombre completo, en caso de que se brinde, en nombre y apellidos. Luego se determina qué referencias son ambiguas y se descartan. Cada vez que se solicite referencias a un nombre específico, el generador determina si esta es una primera referencia o una referencia tardía. Para las primeras referencias se emplea el nombre completo mientras que para segundas y terceras referencias se emplean los apellidos. En caso de muchas referencias a una misma persona, el generador podría utilizar el nombre si este no genera ambigüedad.

4.2. Interfaz gráfica

Para poder interactuar más fácil con la propuesta, se proporcionó una interfaz gráfica. Esta interfaz es sencilla y se concibió utilizando el *framework* de *python* *streamlit*. Se da la opción a los usuarios de aportar datos en forma de archivos, ya sea a partir de definir la ruta local o cargando directamente el archivo.

Para permitir esta interacción se concibió una estructura intermedia para poder representar las tuplas de entrada en archivos *.json* (ver fig4.3). Cada tupla se representa con las llaves: “1”, “2”,“3”,“4” donde cada valor correspondiente representa el valor de la tupla en esa posición. Si no se proporciona uno de los valores, esa posición se considera vacía.

Desde la primera interfaz se puede interactuar con el conjunto de datos de prueba que se concibieron junto a la propuesta. Para ello hay que seleccionar el deporte deseado y uno entre los eventos de prueba. El sistema mostrará el texto producido en pantalla, luego de presionar el botón “generar”.

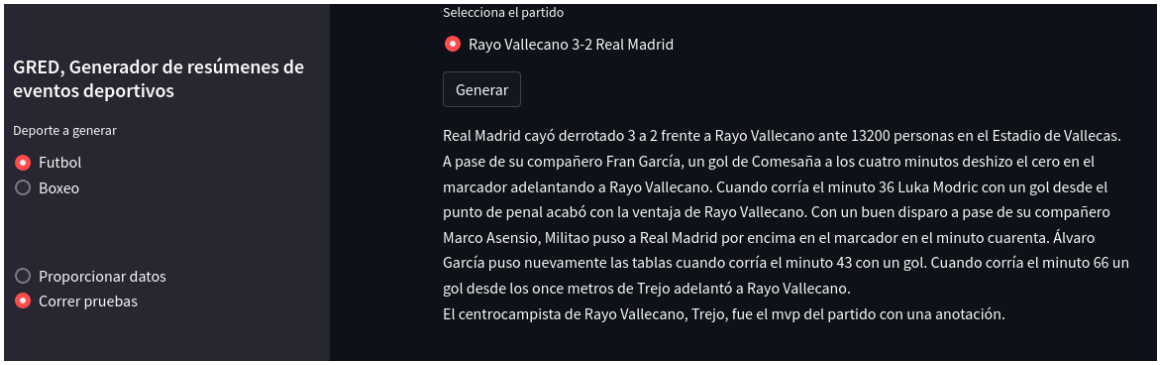


Figura 4.4: Interfaz del sistema

Figura 4.5: Muestra del resultado del partido y los goles por el *Diario AS*

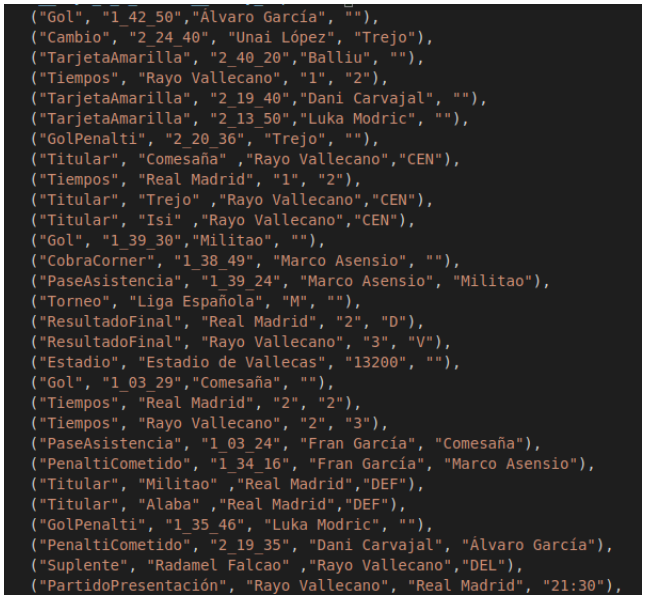
4.3. Resultados de la generación de texto

Para la validación de la propuesta se concibieron los modelos de fútbol y boxeo. Ambos debían, en base a su diseño generar resúmenes del evento a describir a partir de las tuplas de conocimiento del mismo. El texto debía mostrar cierto grado de variabilidad en la salida. O sea, distintos textos ante la misma instancia de datos. A continuación se muestran resultados de ambos modelos.

Fútbol

A continuación se muestran dos variantes distintas producidas por el modelo para un mismo partido cuyo resultado reflejado por el *Diario AS* se puede ver en figura 4.5. Un extracto de las tuplas que conforman la entrada se presenta en la figura 4.6.

- *En el Estadio de Vallecas con la presencia de 13200 espectadores, Rayo Vallecano logró la victoria sobre Real Madrid 3 a 2. Asistido por Fran García, un gol de Comesaña en el minuto cuatro deshizo el cero en el marcador adelantando a Rayo Vallecano. Luka Modric con un gol desde el punto de penal en el minuto 36 puso nuevamente las tablas. En el minuto cuarenta, un tiro preciso de Militao, a pase de Marco Asensio, dió la delantera en el marcador a Real Madrid. Cuando corría el minuto 43, un buen disparo de Álvaro García acabó*



```
("Gol", "1_42_50","Álvaro García", ""),
("Cambio", "2_24_40", "Unai López", "Trejo"),
("TarjetaAmarilla", "2_40_20","Balliu", ""),
("Tiempos", "Rayo Vallecano", "1", "2"),
("TarjetaAmarilla", "2_19_40","Dani Carvajal", ""),
("TarjetaAmarilla", "2_13_50","Luka Modric", ""),
("GolPenalti", "2_20_36", "Trejo", ""),
("Titular", "Comesaña" ,"Rayo Vallecano","CEN"),
("Tiempos", "Real Madrid", "1", "2"),
("Titular", "Trejo" ,"Rayo Vallecano","CEN"),
("Titular", "Isi" ,"Rayo Vallecano","CEN"),
("Gol", "1_39_30","Militao", ""),
("CobraCorner", "1_38_49", "Marco Asensio", ""),
("PaseAsistencia", "1_39_24", "Marco Asensio", "Militao"),
("Torneo", "Liga Española", "M", ""),
("ResultadoFinal", "Real Madrid", "2", "D"),
("ResultadoFinal", "Rayo Vallecano", "3", "V"),
("Estadio", "Estadio de Vallecas", "13200", ""),
("Gol", "1_03_29","Comesaña", ""),
("Tiempos", "Real Madrid", "2", "2"),
("Tiempos", "Rayo Vallecano", "2", "3"),
("PaseAsistencia", "1_03_24", "Fran García", "Comesaña"),
("PenaltiCometido", "1_34_16", "Fran García", "Marco Asensio"),
("Titular", "Militao" ,"Real Madrid","DEF"),
("Titular", "Alaba" ,"Real Madrid","DEF"),
("GolPenalti", "1_35_46", "Luka Modric", ""),
("PenaltiCometido", "2_19_35", "Dani Carvajal", "Álvaro García"),
("Suplente", "Radamel Falcao" ,"Rayo Vallecano","DEL"),
("PartidoPresentación", "Rayo Vallecano", "Real Madrid", "21:30"),
```

Figura 4.6: Extracto de las tuplas de entrada para el ejemplo del fútbol

con la ventaja de Real Madrid. En el minuto 66, con un gol de penal, Trejo puso a Rayo Vallecano en ventaja. Trejo, centrocampista de Rayo Vallecano, fue el mvp del partido con un gol.

- *Rayo Vallecano se llevó la victoria 3 a 2 frente a Real Madrid en el Estadio de Vallecas ante un público de 13200 espectadores. En el cuatro, Comesaña con un tiro preciso declinó la balanza inicial a favor de Rayo Vallecano, el pase fue de Fran García. Luka Modric niveló el encuentro con un tiro ajustado preciso desde el punto de penal cuando corría el minuto 36. Cuando el crono marcaba el minuto cuarenta, con un disparo ajustado a pase de su compañero Marco Asensio, Militao deshizo la igualdad en favor de Real Madrid. Cuando el crono marcaba el minuto 43 una buena definición de Álvaro García acabó con la ventaja de Real Madrid. Trejo deshizo la igualdad en favor de Rayo Vallecano en el 66 con un gol desde los once metros. Con una anotación, el centrocampista de Rayo Vallecano, Trejo, fue el jugador más destacado del partido.*

El modelo desarrollado es capaz de generar textos que siguen la estructura elegida y cumple con el objetivo de crear un reporte de un partido de fútbol partiendo de la base de ser independiente de la fuente de datos utilizada.

```
("ConteoProtecciónSuperado","02_130", "Jeison Rosario",""),
("GolpeIzquierdaEfectivo","03_023","Jeison Rosario",""),
("GolpeIzquierdaEfectivo","02_083","Jeison Rosario",""),
("JabNoEfectivo","01_025","Jeison Rosario",""),
("CombateContrincantes", "Brian Mendoza", "Jeison Rosario","2022-11-06 18:00"),
("DirectoDerechaEfectivo","01_080","Jeison Rosario",""),
("GanchoEfectivo","05_020","Brian Mendoza",""),
("SwingNoEfectivo","03_036","Jeison Rosario",""),
("JabNoEfectivo","03_055","Brian Mendoza",""),
("NocautPropinado", "05_030", "Brian Mendoza",""),
("VeredictoFinal","Brian Mendoza","1_039_2_038_3_039","V_KO"),
("GanchoNoEfectivo","01_118","Brian Mendoza",""),
("SwingNoEfectivo","01_121","Jeison Rosario",""),
("JabEfectivo","02_053","Jeison Rosario",""),
("DirectoDerechaEfectivo","02_066","Brian Mendoza",""),
("GolpeDerechaEfectivo","02_069","Brian Mendoza",""),
("JabNoEfectivo","01_086","Brian Mendoza",""),
("CrochetNoEfectivo","01_088","Brian Mendoza",""),
("JabNoEfectivo","02_110","Jeison Rosario",""),
("GolpeIzquierdaEfectivo","02_118","Brian Mendoza",""),
("VeredictoFinal","Jeison Rosario","1_036_2_037_3_036","D_KO"),
("GolpeIzquierdaEfectivo","02_121","Brian Mendoza",""),
("JabNoEfectivo","03_033","Brian Mendoza",""),
("Sede", "Minneapolis Armory","",""),
("CombateOrganizado","Minneapolis, EE.UU","M","medio"),
("CrochetNoEfectivo","01_020","Brian Mendoza",""),
("JabNoEfectivo","01_020","Jeison Rosario",""),
```

Figura 4.7: Extracto de las tuplas de entrada para el ejemplo del boxeo

Boxeo

Se presentan dos muestras de reportes generados para una pelea entre Brian Mendoza y Jeison Rosario. Los datos se construyeron con el visionado de un extracto de la pelea en *youtube.com*. Un ejemplo de la entrada utilizada puede verse en la figura 4.7.

- *En el ring del Minneapolis Armory en la categoría de peso medio, con un imparable gancho Brian Mendoza obtuvo la victoria por nocaut frente a Jeison Rosario en el quinto asalto. Mendoza hizo besar la lona a Rosario con un derechazo en el segundo asalto.*

- *Ante los aficionados congregados en el Minneapolis Armory Brian Mendoza y Jeison Rosario cumplieron su cita en la categoría de peso medio. Con un gancho Brian Mendoza obtuvo la victoria al noquear a Jeison Rosario en el quinto asalto. Mendoza puso en conteo de protección a Rosario con un inesperado puñetazo de derecha en el segundo asalto.*

El modelo implementado fue capaz de generar textos que describían el combate de boxeo definido por los datos de entrada independiente de la fuente de obtención de los mismos. Mostró variabilidad, generando diferentes textos ante la misma entrada de datos. El texto generado respetó la estructura informativa definida.

Conclusiones

En el presente trabajo se definió la propuesta de un sistema para la generación de resúmenes de enfrentamientos deportivos, independiente de la fuente de datos del dominio, a partir del análisis de características comunes del conjunto de los deportes de enfrentamiento.

Se presentó una propuesta de meta esquema general con el cual definir la estructura de entrada de los datos al sistema y se validó que es posible, a partir de dicho esquema, definir un esquema de representación individual por deporte. Se concibió una propuesta de diseño para los modelos de generación de un deporte en base a su esquema. El fútbol y el boxeo demostraron ser muestras útiles para la validación del sistema, por ser disciplinas muy diferentes en cuanto a su naturaleza de ejecución y características.

Se logró implementar un prototipo de sistemas con los modelos de generación para generar reportes basados en los eventos de estas dos disciplinas deportivas. Los modelos, siguiendo un enfoque simple de reglas y plantillas, mostraron variabilidad en distintas ejecuciones frente a los mismos datos. A su vez, mostraron fidelidad en la información representada en la salida respecto a los datos de entrada y garantizaron la correcta estructura del texto producido. De esta forma, cumplieron con el requerimiento básico de los sistemas de generación de GLN funcionales.

Recomendaciones

A partir del análisis de los resultados obtenidos, se considera que el texto producido por los modelos presentados pudiera ser dotado de mayor fluidez. Agregar un mecanismo más potente para la generación y selección de expresiones de referencia, pudiera contribuir a esto, disminuyendo las repeticiones continuas del mismo nombre propio. Asimismo, la fluidez se puede mejorar definiendo reglas específicas para la agregación de eventos iguales consecutivos, o que presenten relación de causalidad. También aumentar el número de plantillas para distintos tipos de expresiones, contribuiría a aumentar la variabilidad de los modelos.

Se recomienda evaluar la adaptabilidad de los esquemas propuestos a una diversidad de fuentes proveedoras de datos para comprobar su factibilidad en la práctica. Asimismo, se recomienda buscar una propuesta para la generación automática de las tuplas de conocimiento a partir de un esquema específico.

Bibliografía

[1] João Pinto Barbosa Machado Aires. «Automatic generation of sports news». En: (2016) (vid. págs. 15, 21, 24).

[2] Dzmitry Bahdanau, Kyunghyun Cho y Yoshua Bengio. «Neural machine translation by jointly learning to align and translate». En: *arXiv preprint arXiv:1409.0473* (2014) (vid. pág. 12).

[3] Roberto Balboa. «Generación de noticias a partir de datos estructurados en el dominio del béisbol». En: (2020) (vid. págs. 3, 14).

[4] Regina Barzilay y Mirella Lapata. «Aggregation via Set Partitioning for Natural Language Generation». En: *NAACL*. 2006 (vid. pág. 9).

[5] Nadjet Bouayad-Agha, Gerard Casamayor y L. Wanner. «Content selection from an ontology-based knowledge base for the generation of football summaries». En: *ENLG*. 2011 (vid. pág. 7).

[6] Kai Chen y col. «Neural data-to-text generation with dynamic content planning». En: *Knowledge-Based Systems* 215 (2021), pág. 106610 (vid. pág. 13).

[7] Philipp Cimiano y col. «Exploiting ontology lexica for generating natural language texts from RDF data». En: *Proceedings of the 14th European Workshop on Natural Language Generation*. 2013, págs. 10-19 (vid. pág. 10).

[8] Robert Dale. «Natural language generation: The commercial state of the art in 2020». En: *Natural Language Engineering* 26.4 (2020), págs. 481-487 (vid. págs. 2, 14).

[9] Robert Dale y Ehud Reiter. «Computational interpretations of the Gricean maxims in the generation of referring expressions». En: *Cognitive science* 19.2 (1995), págs. 233-263 (vid. pág. 10).

[10] Laurence Danlos, Frédéric Meunier y Vanessa Combet. «EasyText: an operational NLG system». En: *ENLG 2011-13th European Workshop on Natural Language Generation*. 2011 (vid. pág. 10).

[11] Pablo Ariel Duboué y Kathleen McKeown. «Statistical Acquisition of Content Selection Rules for Natural Language Generation». En: *EMNLP*. 2003 (vid. pág. 7).

[12] Ondřej Dušek, Jekaterina Novikova y Verena Rieser. «Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge». En: *Computer Speech & Language* 59 (2020), págs. 123-156 (vid. pág. 13).

[13] Thiago Castro Ferreira y col. «Neural data-to-text generation: A comparison between pipeline and end-to-end architectures». En: *arXiv preprint arXiv:1908.09022* (2019) (vid. pág. 13).

[14] Thiago Castro Ferreira y col. «NeuralREG: An end-to-end approach to referring expression generation». En: *arXiv preprint arXiv:1805.08093* (2018) (vid. pág. 10).

[15] Albert Gatt y Emiel J. Krahmer. «Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation». En: *ArXiv* abs/1703.09902 (2018) (vid. págs. 1, 5, 6, 8-11, 23).

[16] Eli Goldberg, Norbert Driedger y Richard I Kittredge. «Using natural-language processing to produce weather forecasts». En: *IEEE Expert* 9.2 (1994), págs. 45-53 (vid. págs. 1, 14).

[17] Jiatao Gu y col. «Incorporating copying mechanism in sequence-to-sequence learning». En: *arXiv preprint arXiv:1603.06393* (2016) (vid. págs. 12, 15).

[18] Derek Gulbranson. *Python human name parser¶*. URL: `https://nameparser.readthedocs.io/en/latest/` (vid. pág. 43).

[19] MHDY Gunasiri. «Automated cricket news generation in Sri Lankan style using natural language generation». Tesis doct. 2021 (vid. págs. 2, 15).

[20] Fahim Muhammad Hasan. «Automatic generation of multilingual sports summaries». Tesis doct. Applied Science: School of Computing Science, 2011 (vid. pág. 14).

[21] Ziwei Ji y col. «Survey of hallucination in natural language generation». En: *arXiv preprint arXiv:2202.03629* (2022) (vid. pág. 14).

[22] Mihir Kale y Abhinav Rastogi. «Text-to-text pre-training for data-to-text tasks». En: *arXiv preprint arXiv:2005.10433* (2020) (vid. pág. 13).

[23] Jenna Kanerva y col. «Template-free data-to-text generation of Finnish sports news». En: *arXiv preprint arXiv:1910.01863* (2019) (vid. pág. 14).

[24] Karen Kukich. «Design of a knowledge-based report generator». En: *21st Annual Meeting of the Association for Computational Linguistics*. 1983, págs. 145-150 (vid. págs. 1, 5).

[25] Rémi Lebret, David Grangier y Michael Auli. «Neural Text Generation from Structured Data with Application to the Biography Domain». En: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, nov. de 2016, págs. 1203-1213. DOI: 10.18653/v1/D16-1128. URL: https://aclanthology.org/D16-1128 (vid. pág. 12).

[26] Chris van der Lee, Emiel Krahmer y Sander Wubben. «PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences». En: *Proceedings of the 10th International Conference on Natural Language Generation*. 2017, págs. 95-104 (vid. págs. 2, 15, 24).

[27] William C Mann y Sandra A Thompson. «Rhetorical structure theory: Toward a functional theory of text organization». En: *Text-interdisciplinary Journal for the Study of Discourse* 8.3 (1988), págs. 243-281 (vid. pág. 8).

[28] Kathleen R McKeown. «Discourse strategies for generating natural-language text». En: *Artificial intelligence* 27.1 (1985), págs. 1-41 (vid. pág. 8).

[29] Hongyuan Mei y col. «What to talk about and how? Selective Generation using LSTMs with Coarse-to-Fine Alignment». En: *Proceedings of NAACL-HLT*. 2016, págs. 720-730 (vid. pág. 12).

[30] Amit Moryossef, Yoav Goldberg e Ido Dagan. «Step-by-step: Separating planning from realization in neural data-to-text generation». En: *arXiv preprint arXiv:1904.03396* (2019) (vid. pág. 13).

[31] Jekaterina Novikova, Ondřej Dušek y Verena Rieser. «The E2E dataset: New challenges for end-to-end generation». En: *arXiv preprint arXiv:1706.09254* (2017) (vid. pág. 12).

[32] Rivindu Perera y Parma Nand. «Recent Advances in Natural Language Generation: A Survey and Classification of the Empirical Literature». En: *Comput. Informatics* 36 (2017), págs. 1-32 (vid. págs. 5, 7, 8, 10).

[33] Ratish Puduppully, Li Dong y Mirella Lapata. «Data-to-text generation with content selection and planning». En: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, págs. 6908-6915 (vid. pág. 13).

[34] Ratish Puduppully, Li Dong y Mirella Lapata. «Data-to-text generation with entity modeling». En: *arXiv preprint arXiv:1906.03221* (2019) (vid. pág. 12).

[35] Colin Raffel y col. «Exploring the limits of transfer learning with a unified text-to-text transformer.» En: *J. Mach. Learn. Res.* 21.140 (2020), págs. 1-67 (vid. pág. 13).

[36] Alejandro Ramos-Soto, Alberto Bugarín y Senén Barro. «On the role of linguistic descriptions of data in the building of natural language generation systems». En: *Fuzzy Sets and Systems* 285 (2016), págs. 31-51 (vid. pág. 5).

[37] Mike Reape y Chris Mellish. «Just what is aggregation anyway». En: *Proceedings of the 7th European Workshop on Natural Language Generation*. 1999, págs. 20-29 (vid. pág. 8).

[38] Ehud Reiter y R. Dale. «Building applied natural language generation systems». En: *Natural Language Engineering* 3 (1997), págs. 57-87 (vid. págs. 1, 5-7, 9).

[39] Ehud Reiter y Robert Dale. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press, 2000. DOI: 10.1017/CBO9780511519857 (vid. págs. 1, 5, 7, 10, 21, 23, 26).

[40] Ehud Reiter y col. «Choosing words in computer-generated weather forecasts». En: *Artificial Intelligence* 167.1-2 (2005), págs. 137-169 (vid. pág. 14).

[41] Mandar Sharma, Ajay Gogineni y Naren Ramakrishnan. «Innovations in Neural Data-to-text Generation». En: *arXiv preprint arXiv:2207.12571* (2022) (vid. págs. 11, 13).

[42] James Shaw. «Clause Aggregation Using Linguistic Knowledge». En: *INLG*. 1998 (vid. pág. 9).

[43] Advaith Siddharthan, Ani Nenkova y Kathleen McKeown. «Information status distinctions and referring expressions: An empirical study of references to people in news summaries». En: *Computational Linguistics* 37.4 (2011), págs. 811-842 (vid. pág. 11).

[44] Ilya Sutskever, Oriol Vinyals y Quoc V Le. «Sequence to sequence learning with neural networks». En: *Advances in neural information processing systems* 27 (2014) (vid. pág. 12).

[45] Mariët Theune y col. «From data to speech: a general approach». En: *Natural Language Engineering* 7.1 (2001), págs. 47-86 (vid. págs. 2, 15, 21, 24).

[46] Marilyn Walker, Owen Rambow y Monica Rogati. «SPoT: A trainable sentence planner». En: *Second Meeting of the North American Chapter of the Association for Computational Linguistics*. 2001 (vid. pág. 9).

[47] Sam Wiseman, Stuart Shieber y Alexander Rush. «Challenges in Data-to-Document Generation». En: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, sep. de 2017, págs. 2253-2263. DOI: 10.18653/v1/D17-1239. URL: https://aclanthology.org/D17-1239 (vid. págs. 12, 13).

[48] Jin Yu y col. «Choosing the content of textual summaries of large time-series data sets». En: *Natural Language Engineering* 13 (2006), págs. 25-49 (vid. págs. 5, 7).