INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Análisis de historias clínicas basado en similitud textual

Tesis que para obtener el grado de:

DOCTORADO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

CAROLINA FÓCIL ARIAS

DIRECTORES DE TESIS:

DR. GRIGORI SIDOROV

DR. ALEXANDER GELBUKH

CIUDAD DE MÉXICO **ENERO, 2019**

SIP-14 bis

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 10:00 horas del día 05 del mes de noviembre de 2018 Se reunieron los miembros de la Comisión Revisora de la Tesis en la sala designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Análisis de historias clínicas basado en similitud textual"

Presentada por la alumna:

FÓCIL	ARIAS	CAROLINA
Apellido paterno	Apellido materno	Nombre(s)

Con registro: B 1 5 1 2 6 8

aspirante de: **DOCTORADO EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de Tesis

Dr. Alexander Gelbukh

Dr. Grigori Sidorov

Dr. Sergio Suárez Guerra

Dr. Ildar Batyrshin

Dra. Olga Kolesnikova

Dr. Miguel Jesús Torres Ruiz

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Marco Antonio Ramírez Salinas

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN
DIRECCIÓN

INSTITUTO POLITÉECNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 4 del mes Diciembre del año 2018, el (la) que suscribe Carolina Fócil Arias alumno (a) del Programa de Doctorado en Ciencias de la Computación con número de registro B151268, adscrito a Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Grigori Sidorov y Dr. Alexander Gelbukh y cede los derechos del trabajo intitulado Análisis de historias clínicas basado en similitud textual, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección focil.carolina@gmail.com, sidorov@cic.ipn.mx, gelbukh@gelbukh.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Carolina Fócil Arias

Nombre y firma

Resumen

Recientemente, la extracción de eventos clínicos desde textos médicos no estructurados ha atraído mucha la atención en la comunidad científica. En este trabajo presentamos una metodología para la identificación y la clasificación de eventos clínicos con base en historias médicas de pacientes con cáncer de colón.

Primero, analizamos los textos clínicos y aplicamos técnicas de procesamiento de lenguaje natural para hacer un procesamiento cuidadoso de los textos médicos, de tal forma que se mejore la calidad de éstos.

Luego, proponemos el uso de características lingüísticas, léxicas, sintácticas, nivel de discurso, y de recursos externos con algoritmos de aprendizaje automático, como campo aleatorio condicional y el algoritmo de Viterbi, los cuales son muy populares para etiquetar secuencia de datos efectivamente.

Después, evaluamos los enfoques presentados en el estado del arte, y comparamos los resultados presentados en la competencia de SemEval 2016 - Tarea 12: TempEval 2016, llevando a cabo las subtareas: (i) detección de eventos, (ii) clasificación de eventos, (iii) clasificación de acuerdo con la polaridad del evento, y (iv) clasificación con base en el tipo de eventos.

Finalmente, analizamos los resultados de las tareas, y observamos que nuestra propuesta utilizando campo aleatorio condicional obtuvo buenos resultados en comparación con los trabajos presentados en la competencia de SemEval Tarea 12. Además, demostramos que el popular y mucho más simple algoritmo Viterbi (algoritmo de clasificación basado en el modelo de Markov oculto) puede producir resultados competitivos, cuando sus parámetros son ajustados utilizando técnicas de optimización tales como evolución diferencial y *hill climbing*.

Abstract

Recently, the extraction of clinical events from unstructured medical texts has attracted much attention of the research community. In this work, we propose a methodology to identify and to classify clinical events based on histories in colon cancer patients.

Firstly, we analyze the medical history and we apply natural language processing techniques for a better understanding and improving of the medical texts.

Then, we propose the usage of different features such as linguistic, word-forms, discourse level, lexical and external resources with machine learning algorithms: conditional random fields and Viterbi algorithm, which are very popular for solving the problem of sequence tagging effectively.

Afterwards, we evaluate the approaches based on the state-of-the-art and we compare the results presented in the task of extraction of medical events from the corpus developed for SemEval shared Task 12: Clinical TempEval (Temporal Evaluation) 2016, namely, for its subtasks: (i) event detection, (ii) event classification, (iii) polarity, and (iv) type based on contextual modality.

Finally, we analyze the results and observed that our proposal using conditional random fields achieves good results in comparison to the works presented in Task-12: Clinical TempEval 2016 challenges. We also show that the popular and much simpler Viterbi algorithm (hidden Markov model-based classification algorithm) can produce competitive results, when its parameters are tuned using specific optimization techniques using differential evolution and *hill climbing*.

Dedicatoria

A mis padres, Guillermo Fócil y Blanca Arias.

A mis hermanos, Angélica y Guillermo.

Agradecimientos

Este trabajo de investigación es un esfuerzo directo e indirecto de distintas personas, por lo tanto, quiero realizar un agradecimiento a todos aquellos que formaron parte de mi crecimiento profesional.

A dios por su ayuda espiritual, protección y guía en todo momento.

A mis padres, el señor Guillermo Fócil Zurita, y la señora Blanca Arias Hernández, quienes me inspiraron en todo momento a continuar con mis estudios. Sin el esfuerzo, sacrificio, apoyo, cariño, amor, educación y confianza de ellos, jamás hubiera sido posible la realización de este proyecto profesional. ¡Los amo!

A mis hermanos Angélica y Guillermo, quienes han estado a mi lado en todo este tiempo en que he trabajado en esta tesis de investigación.

A mi cuñado Adán Serrano.

A mis sobrinos Guillermo y Rafael.

A mis amigos por su amistad, ánimo y compañía, principalmente por esos momentos de risa extrema.

A mis directores de tesis el Dr. Grigori Sidorov y el Dr. Alexander Gelbukh por sus consejos, y asesoramiento durante el desarrollo de este trabajo de tesis.

A mis supervisores en el extranjero Jian Yun Nie y Paolo Rosso.

Al equipo de Menta Network por su confianza en mi profesionalismo.

Al equipo de SemEval.

A mi jurado de examen por sus opiniones y correcciones.

Al centro de Investigación en Computación y al Instituto Politécnico Nacional por ser mi segunda casa.

Al CONACYT y SIP por su apoyo económico.

¡Muchas gracias!

Carolina Fócil

Índice General

Índice de figuras

Índice de tablas

Índice de fórmulas

4.2
$$P(V_k|S_j) = \prod_{j=1}^{N} P(f_j|S_j)$$

4.3
$$A = \{a_{ij}\} = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} = a = [a_{11}, a_{12}, \dots, a_{NN}]$$

4.4
$$B = \{b_i(k)\} = \begin{bmatrix} b_{11} & \cdots & b_{1M} \\ \vdots & \ddots & \vdots \\ b_{N1} & \cdots & a_{NM} \end{bmatrix} = b = [b_{11}, b_{12}, \dots, b_{NN}]$$

4.5
$$ab = [a_{11}, a_{12}, \dots, a_{NN}, b_{11}, b_{12}, \dots, b_{NN}]$$

4.6
$$2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

4.7
$$BA = [a_{21}, a_{12}, \dots, a_{NN}, b_{21}, b_{12}, \dots, b_{NN}]$$ 4.6

4.8
$$BA_{i=0}^{i=n} = BA_i + f$$

4.9
$$BA_{i=0}^{i=n} = BA_i - f$$

5.1
$$\frac{\text{TP}}{\text{TP} + \text{FP}}$$

5.2
$$\frac{\text{TP}}{\text{TP} + \text{FN}}$$

5.3
$$2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

5.4
$$\frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

Capítulo 1. Introducción

En este trabajo de tesis se propone una metodología utilizando algoritmos de aprendizaje automático, técnicas de procesamiento de lenguaje natural y algoritmos de optimización para la detección de eventos médicos utilizando textos clínicos de pacientes con cáncer de colón.

1.1 Antecedentes

La extracción de información temporal desde textos clínicos no estructurados es un paso importante hacia la construcción de un historial clínico de un paciente; identificando y rastreando patrones que son cruciales en la toma de decisiones en la investigación médica [2].

Este trabajo de investigación está fuertemente relacionado con el el cáncer de colón, el cual es considerado la mayor causa de muerte en el mundo [2]. En el 2016, la Sociedad Americana de Cáncer reportó aproximadamente 95,270 pacientes con dicha enfermedad, y se estimaron 49,190 casos de muerte.

Debido a la importancia de esta enfermedad, se han realizado diversas investigaciones en todo el mundo, desde análisis de imágenes clínicas hasta análisis de reportes clínicos no estructurados para la identificación de patrones que permitan crear y evaluar un historial clínico.

El reconocimiento de entidades médicas o por su término en inglés de *Medical Entity Recognition,* se encargan de la detección y la extracción de eventos relacionados con el campo médico. Un ejemplo de esta tarea puede ser encontrado en el workshop

conocido como SemEval [1], la cual se encarga de extraer *span* de palabras, clasificar eventos médicos, determinar la polaridad del evento, entre otras.

Debido a la importance de este tema, se han realizado diversas investigaciones en todo el mundo, con el propósito de mejorar la calidad de vida de los pacientes. Entre las propuestas más utilizadas se encuentran campo aleatorio condicional, máquina de vectorwa soporte, y generación de reglas. Por ejemplo, el investigador Jay Urbain, presentaron el diseño y el análisis de sistemas de preprocesamiento de lenguaje natural para la detección de entidades nombradas [3].

Tourille et al. [4] proporcionó dos enfoques basados en aprendizaje automático. El primer enfoque está basado en redes neuronales recurrentes, y el segundo enfoque utilizó máquinas de vectores de soporte para evaluar los eventos clínicos, utilizando información como caracteres, y word2vec.

Los investigadores Leeuwenberg y Moens desarrollaron un sistema basado máquinas de vectores de soporte para la clasificación de los eventos clínicos. Este sistema está basado en el uso de características lingüisticas y representación de palabras [5].

MacAvaney et al. [6] entrenaron campo aleatorio condicional , árboles de decisión, y reglas utilizando dos dominios de adaptación: supervisado y no supervisado con características como n-gramas, palabras, técnicas de agrupación, word2vec, sintácticas entre otras, para la clasificación de eventos clínicos.

Con base en los diferentes estudios presentados en la actualidad , surge la idea de utilizar algoritmos de aprendizaje automático para mejorar los lineamientos médicos en la extracción y clasificación de eventos clínicos. La propuesta de esta tesis consiste en utilizar el algoritmo de campo aleatorio condicional, y el algoritmo de Viterbi junto con técnicas de optimización

Cabe mencionar que el algoritmo de Viterbi no ha sido utilizado en la tarea de identificación y clasificación de eventos clínicos. Por lo tanto, proponemos utilizarlo en conjunto con dos técnicas de optimización para mejorar el desempeño del algoritmo.

1.2 Planteamiento del problema

En esta sección de describe el planteamiento del problema:

Implementar una metodología que permita identificar todas aquellas palabras que sean eventos clínicos, es decir, medicinas, procedimientos, cirugías, síntomas, enfermedades, entre otros, desde textos clínicos. Así como también, una vez la clasificación de estos eventos de acuerdo con la polaridad, la modalidad contextual, y los aspectos contextuales.

1.3 Objetivo general

A continuación se detalla el objetivo general de la presente investigación.

Desarrollar una metodología para la extracción y la clasificación de eventos clínicos (medicinas, procedimientos, cirugías, síntomas, etc.), y hacer su evaluación aplicando algoritmos de aprendizaje automático, métodos de optimización y técnicas de procesamiento de lenguaje natural.

1.4 Objetivos específicos

Los objetivos especificos son descritos a continuación:

- Seleccionar uno o varios conjuntos de datos clínicos.
- Identificar los métodos de procesamiento de lenguaje natural y aprendizaje automático para la identificación y la clasificación de eventos clínicos.

- Revisar las metodologías reportadas en el estado del arte.
- Describir una metodología para la detección automática y clasificación de eventos clínicos, la cual pueda ser utilizada en diferentes tareas de procesamiento de lenguaje natural, y la cual consiste en el algoritmo de campo aleatorio condicional y Viterbi.
- Desarrollar un algoritmo de integración de los métodos utilizados a fin de identificar y clasificar los eventos clínicos con mayor precisión.
- Validar la metodología propuesta.

1.5 Resultados esperados

Los resultados esperando de esta investigación se detallan a continuación:

Planteamiento y desarrollo de una metodología basada en algoritmos de aprendizaje automático y técnicas de procesamiento de lenguaje natural para la extracción y clasificación de eventos clínicos (medicinas, procedimientos, cirugías, síntomas, enfermedades, etc.) usando el corpus clínico de SemEval 2016. Así como también publicaciones JCR en una revista, y artículos en congreso.

1.6 Aportaciones

En esta sección se presentan las aportaciones de la investigación:

- Una mejora del algoritmo de Viterbi utilizando métodos de optimización.
- Una metodología para la identificación y la clasificación de eventos clínicos.
- Evaluación de los algoritmos y características en la competencia de SemEval 2016.

1.7 Organización del documento

En este trabajo de investigación se estructura de la siguiente forma:

Capítulo 1. Introducción. En esta parte se detallan los antecedentes, el planteamiento del problema, el objetivo general, los objetivos específicos, las aportaciones, y los resultados esperados.

Capítulo 2. Estado del arte. Se describe el estado del arte en el área de aprendizaje automático para determinar si una palabra es un evento relacionado con el área clínica.

Capítulo 3. Materiales y métodos. Se establecen los materiales y los métodos que se requieren para la elaboración de este trabajo de tesis doctoral.

Capítulo 4. Metodología experimental. Se presenta la metodología desarrollada para elaborar este trabajo de tesis.

Capítulo 5. Resultados experimentales. Se muestran a detalle los experimentos y resultados junto con un análisis comparativo presentado en la competencia de SemEval 2016 Tarea 12.

Capítulo 6. Conclusiones y trabajo a futuro. Se presentan las conclusiones, las aportaciones, y el trabajo a futuro.

Capítulo 2. Estado del arte

En la actualidad, existe una gran cantidad de datos clínicos no estructurados, que al ser analizados pueden permitir la construcción de un historial clínico de un paciente, el conocimiento del avance de la enfermedad, entre otras. Por esta razón, la detección de entidades nombradas y el procesamiento de lenguaje natural han recibido mucha atención en los últimos años. Una prueba de esto se puede encontrar en el workshops de la tarea 12 de SemEval 2016, el cual se basa en la extracción y clasificación de eventos clínicos utilizando textos clínicos de pacientes con cáncer de colón.

En este capítulo, se presenta el estado del arte en el área clínica. En la primera sección, se exponen algunas de las principales técnicas aplicadas en el área de reconocimiento de entidades médicas, y las cuales han proporcionado resultados competitivos para mejorar la detección de eventos clínicos. En la segunda sección se detallan la definición y clasificación de eventos. Además, un resumen de las variables y clasificadores utilizados para estimar si una palabra pertenece a un evento clínico es presentado en el anexo A y B, respectivamente.

2.1. Reconocimiento de entidades médicas

El análisis eficiente de los textos clínicos es un paso importante y crucial para mejorar la calidad de vida de un paciente, y esto es posible a través del reconocimiento de entidades médicas. Con base en [7], el reconocimiento de entidades nombradas (MER) es caracterizada por ser:

- el proceso de extracción, detección, y delimitación de frases o palabras.
- la identificación de la clasificación semántica de las entidades localizadas.

Y puede ser clasificada en tres categorías [8]:

1. Modelos basados en diccionarios
2. Modelos basados en reglas
3. Modelos basados en algoritmos de aprendizaje automático

2.2 Modelos basados en diccionarios

Los modelos basados en diccionarios pueden extraer todas las entidades clínicas que se encuentren definidas en un diccionario [9]. Este tipo de modelos están fuertemente ligados con la detección de expresiones de tiempo y palabras con base en el contexto. Este método está enriquecido con diccionarios médicos y ontologías tales como [10]:

- *MedLee:* Es una forma de codificación de textos clínicos.
- *MetaMap:* Permite la detección de los conceptos que se encuentren en un sistema de lenguaje médico unificado, por sus siglas en inglés UMLS, *Unified Medical Language System,* el cual permite integrar palabras médicas con claves para hacer sistemas de información médicos más eficientes [11])
- *CTake:* Este sistema contiene una gran cantidad de diccionarios médicos y además, también contiene algoritmos de aprendizaje supervisado.

Robert Gaizauskas *et al.* [12] utilizó diccionarios y lemas para encontrar si un un término clínico era similar a lo existente en las herramientas empleadas.

2.3 Modelos basados en reglas

El uso de modelos basado en reglas es también incluido en el dominio clínico. Un modelo de este tipo comprende de un conjunto de reglas que pueden ser procesadas por herramientas de simulación y análisis de propósito general para lograr diferentes objetivos [13].

Sarath *et al.* [14], diseñaron un sistema utilizando una combinación de diccionarios, los cuales consisten en expresiones de tiempo y palabras que hacen referencia a períodos de tiempo; y formatos de sentencias que incluyen palabras como "*anual*", "*weekly*", "*monthly*", entre otras. Sin embargo, los resultados no son satisfactorios, por lo tanto, también aplicaron un enfoque de aprendizaje automático, conocido como Stanford CRF, el cual generó mejores resultados que el modelo propuesto originalmente.

El estudio de Arman Cohan *et al.* [15] proponen una combinación de algoritmos supervisados (campo aleatorio condicional y regresión logística), características de dominio específico utilizando *UMLS* (*Lenguaje Médico Unificado*) y reglas manuales para la detección de fechas. Los resultados utilizando la medida F_1 son de 0.881 para la detección de eventos, 0.827 para la clasificación de eventos, 0.866 para la polaridad, y 0.858 para el tipo de eventos.

Guergana *et al*. [13] construyeron y evaluaron un sistema de procesamiento de lenguaje natural para la extracción de información médica desde textos clínicos conocido como cTAKES, por su término en inglés, Clinical Text Analysis and Knowledge Extraction System.

2.4 Modelos basados en clasificación

Una de las técnicas más utilizadas en el área de aprendizaje automático es la clasificación. Esta técnica se aplica para identificar la relación que existe entre las variables dependientes (clases), y las variables independientes, (características) [16] . Una vez que los patrones son aprendidos bajo el esquema de entrenamiento, se asigna la clase correspondiente a los patrones que se encuentran en el área de prueba [15]. Es importante mencionar que existen tres importantes componentes para llevar acabo la

tarea de clasificación: un conjunto de prueba, un conjunto de entrenamiento y las clases predefinidas [17], [18] y [16].

En este capítulo, se presentan algunos de los principales modelos aplicados en el área de aprendizaje automático, y los cuales, han demostrado ser competitivos para la detección de eventos clínicos. Entre los modelos que destacan son campo aleatorio condicional, máquinas de vectores de soporte y redes neuronales. Además, las características más utilizadas en el ámbito de detección de secuencia de eventos en el ámbito clínico son presentadas en el anexo A, y un resumen de los algoritmos utilizados en la tarea de SemEval son presentados en el anexo B.

2.4.1 Campo aleatorio condicional

Campo aleatorio condicional o por su término en inglés *Conditional Random Fields* es un método gráfico probabilístico para el etiquetado de secuencias, principalmente utilizado en intervalos de eventos y expresiones de tiempo [19]. Este modelo es capaz de aprender de forma eficiente y rápida cuando una gran cantidad de datos es dada, y lo más importante es la capacidad de tomar en cuenta el contexto de una observación. Sin embargo, también ha demostrado algunas desventajas como el lenguaje y el dominio, los cuales son independientes. También el conocimiento es proporcional al tamaño del corpus [20]. A continuación se presentan algunos trabajos relacionados con este modelo.

De acuerdo con Veera Raghvendra [19], los algoritmos de campo aleatorio condicional, máquinas de vectores de soporte, y redes neuronales profundas son los enfoques más utilizados para resolver tareas de extracción de eventos. Los resultados presentados usando campo aleatorio condicional son 0.815, 0.746, 0.733, y 0787 para la extracción, modalidad, polaridad y tipo de eventos, respectivamente.

Abdulrahman Khalifa *et al.* [2] emplearon campo aleatorio condicional y máquinas de vectores de soporte para la identificación de eventos en la tarea de clasificación de secuencias. Además, indican que las características léxicas y sintácticas son eficientes para la tarea mencionada anteriormente. Los resultados son 0.892 en la extracción de eventos, 0.841 en la clasificación de eventos, 0.876 en polaridad, 70.866 en el tipo de eventos.

Tomasso Caselli y Roser Morante [21] presentaron un enfoque basado en campo aleatorio condicional utilizando características morfosintácticas y de conocimiento específico. Este sistema fue principalmente desarrollado para procesar textos y, posteriormente fue entrenado para procesar notas clínicas. Sin embargo, la falta de dominio específico y el post procesamiento de reglas afectaron la robustez del algoritmo. Los resultados evaluados utilizando la medida F_1 fueron de 0.847 en la tarea de detección de eventos y 0.776 en la clasificación de éstos.

Cyril Grouin *et al.* [22], realizaron una combinación entre el algoritmo de campo aleatorio condicional y un sistema basado en reglas, este último es principalmente utilizado para un básico post procesamiento de los datos, el cual es capaz de detectar adicionales eventos clínicos de una lista de 69 palabras más frecuentes. Entre las características empleadas para la detección de eventos se encuentran: token, longitud, presencia de puntuación o de dígitos, partes de un enunciado, (*Part Of Speech)*, entre otras. Los resultados del método propuesto son competitivos, ya que obtuvo 0.842 en la detección, 0.772 en la modalidad de eventos, 0.824 en polaridad, y 0.785 en el tipo de eventos, todos estos resultados son presentandos utilizando la medida F1.

Otro trabajo relacionado con la detección de eventos clínicos es presentado por Charlotte Hansart *et al.* [20], en el cual se integran enfoques lingüísticos y estadísticos, como el lenguaje médico unificado, por sus siglas en inglés, UMLS *(Unified Medical Language System),* donde es posible encontrar 150 vocabularios, y reglas lingüísticas

para la detección de sentencias negativas. Los resultados de F_1 utilizando este método fueron de 0.847 en detección, y 0.776 en clasificación de eventos médicos.

En el trabajo realizado por Marcia Barros *et al.* [23], aplicaron un *framework* conocido como IBEnt, el cual es utilizado para la detección de entidades químicas. Este *framework* consiste de dos módulos: el primero identifica entidades químicas y, la segunda, identifica los pares de identidades que representa una iteración química. Este sistema fue adaptado para la extracción de eventos clínicos utilizando campo aleatorio condicional y reglas. Marcia Barros *et al.* reportaron sólo resultados en la detección de eventos con una medida F_1 de 0.807.

Ben *et al.* [24] utilizó un sistema usando campo aleatorio condicional con características lingüísticas para identificar y clasificar el nombre de medicamentos. El sistema superó todos los equipos participantes de SemEval 2013 con una medida F_1 de 0.80, y 0.658 para la detección y clasificación, respectivamente.

Forsyth *et al.* [25] usó campo aleatorio condicional para la extracción de información desde textos clínicos, cuyo contenido está relacionado con el cáncer de mama. El sistema alcanzó una medida F_1 de 0.81.

2.4.2 Máquinas de vectores de soporte

Las máquinas de vectores de soporte, por sus siglas en inglés SVM (*Support Vector Machines)* permiten representar los patrones en un espacio dimensional más amplio que en el espacio original. La principal idea de este algoritmo es encontrar un hiperplano que sea capaz de separar dos conjuntos de datos mediante la maximización de la distancia o el margen entre los dos conjuntos de datos. Algunos de los trabajos que utilizan esta técnica son presentados a continuación.

Un gran número de atributos, por ejemplo léxicos, sintácticos, representación de palabras y recursos externos son utilizados con éxito para muchas tareas de reconocimiento de entidades nombradas o por sus siglas en ingles NER (*Named entity recognition*). Además, herramientas como cTAKES, clamp, openNLP, clearTK y clearNLP reciben gran atención debido a la importancia que el procesamiento de datos [26].

Hee-Jin Lee *et al.* [26], utilizaron las características mencionadas anteriormente, y también utilizaron los resultados de un etiquetador de Stanford como una característica más, adaptando técnicas del estado del arte para la detección de entidades nombradas. Los resultados demostraron que el uso de estas características para la identificación de eventos clínicos, son competitivos en relación con otros estudios, ya que en la medida F_1 obtuvo 0.903 en la detección de eventos, 0.855 en la clasificación de eventos con base en su modalidad contextual, 0.887 para la polaridad, y finalmente, 0.882 para el tipo de eventos.

El uso del sistema de cTAKES también es parte de la extracción de eventos. El sistema consiste de dos tareas utilizando máquinas de vectores de soporte para detectar eventos y expresiones temporales de tiempo. Los resultados presentados por Artuur Leeuwenberg y Maríe-Francine Moens [27], indicaron que el uso de las dependencias entre las características, no mejora el rendimiento del sistema, y decidieron agregar características como palabras embebidas y n-gramas.

Julien Tourille *et al.* [28] aplicó un algoritmo de estrategia *grid* para determinar el algoritmo de aprendizaje automático más apropiado. Entre los algoritmos seleccionados se encuentran: *Random Forests, Linear Support Vector Machine* y *Support Vector Machine*. Los mejores resultados fueron obtenidos por máquinas de vectores de soporte con 0.845 y 0.775, en la tarea de detección de eventos y clasificación de eventos, respectivamente.

2.4.3 Redes neuronales

Los recientes avances en redes neuronales profundas también involucran la identificación de entidades nombradas. Las redes neuronales consisten de muchos procesos conocidos como neuronas, las cuales producen una secuencia de activación de valor real [29].

Debido a los avances en la extracción de eventos, el estudio realizado por Veera Raghavendra [19], propone hacer una comparación entre campo aleatorio condicional, máquinas de vectores de soporte y redes neuronales profundas para la identificación de eventos. Las características empleadas son: *tokens, stem, pos, chunk*, ortográficas, medical *stopwords* y diccionario de eventos. La idea es utilizar palabras embebidas como entrada de palabras, y posteriormente, entrenar los datos con *deepnl*.

Veera Raghvendra utilizó dos enfoques: una combinación de campo aleatorio condicional y máquinas de vectores de soporte, y redes neuronales profundas. Los resultados demostraron que ambos enfoques pueden ser utilizados para la extracción de eventos. Sin embargo, campo aleatorio condicional obtuvo un mejor desempeño en la medida F_1 (0.815 en la detección de eventos, 0.746 en la modalidad, 0.733 en la polaridad, y 0.787 en el tipo de eventos) que las redes neuronales profundas con 0.811, 0.754, 0.788, y 0.789 en la detección, modalidad, polaridad y tipo de eventos, respectivamente.

De acuerdo con los avances en la extracción de eventos, Jason Alan [30] propuso una red neuronal recurrente con palabras embebidas (las palabras embebidas son palabras que se mapea a un vector de números reales) usando dos colecciones de documentos clínicos: MIMIC-III y UICH. También, propone el uso de *DeepDive*, el cual consiste de conocimiento de Stanford, y es capaz de vincular variables aleatorias con factores que codifican directamente el conocimiento del dominio.

Peng Li *et al.* [31] propuso un sistema de procesamiento de lenguaje natural basado en el uso de redes neuronales profundas para la extracción de información clínica. Los resultados obtenidos son competitivos con 0.855, 0.789, 0.792 y 0.833 en la medida F_1 para la detección de span, modalidad, polaridad, y tipo, respectivamente. La idea de este trabajo fue aprovechar módulos básicos de procesamiento de lenguaje natural, como la separación de las palabras (*tokenization*) y partes del enunciado (*POS)*.

2.4.4 Otros enfoques

Lauren *et al.* [32] entrenó un modelo (ELM, *extreme learning machine*) basado en kernels con múltiples análisis discriminantes utilizando skip-gramas y bolsa de palabras. Los experimentos mostraron que ELM proporcionó unbuen desempeño cuando es comparado con máquina de vectores de soporte y multilayer perceptron.

Bisaso *et al.* [32] analizaron estudios relaciones con el VIH. De acuerdo con el estudio, se observó una tendencia creciente en la aplicación de aprendizaje automático en la investigación del VIH utilizando textos electrónicos. Además, también presentaron un aumento en el uso de fuentes genómicas de datos y métodos de aprendizaje automático no paramétricos de alto rendimiento con un enfoque en la lucha contra la resistencia a la terapia antirretroviral (TAR).

2.5 SemEval

La evaluación semántica, SemEval, es un workshop considerado como la principal competencia internacional en la extracción de información temporal utilizando textos clínicos de pacientes con cáncer. A continuación, se describen brevemente los estudios que obtuvieron los mejores resultados.

2.5.1 SemEval 2016: Clinical TempEval

Hee-Jin Lee, *et al.* [26] obtuvieron el mejor desempeño en esta tarea utilizando el algoritmo de máquinas de vectores de soporte. Este enfoque usó variables como léxicas, sintácticas, nivel de discurso, representación de palabra y recursos externos. Además usaron herramientas para el preprocesamiento de los datos como: CLAMP (para la separación de las palabras), OpenNLP (para la etiqueta de los datos), y ClearNLP (para parsear dependencias). El sistema utilizó modelos ocultos de Markov con máquinas de vectores de soporte y obtuvo el primer lugar con una medida F_1 de 0.903 para la extracción de eventos, 0.855 para la modalidad contextual, 0.887 para la polaridad, y 0.882 para el tipo de evento.

Khalifa *et al.* [2] utilizaron campo aleatorio condicional, usando características léxicas (tokens, lema, partes de un enunciado, etiquetas, entre otras) utilizando la herramienta Ctake, junto con la forma de las palabras (minúsculas, númericas, entre otras) utilizado la herramienta ClearTK. Los resultados alcanzados en la medida F_1 son 0.892 para la extracción de eventos, 0.841 para la modalidad contextual, 0.876 para la polaridad, 0.866 para el tipo.

Finalmente, Hansart *et al.* [20] ocupó el tercer lugar en la extracción de eventos, y sólo participaron en esta tarea. Los investigadores integraron técnicas utilizadas previamentes en el procesamiento de noticias en Francia. El sistema alcanzó una medida F_1 de 0.885.

2.5.2 SemEval 2017: Clinical TempEval

Tourille *et al.* [4] presentó dos enfoques: uno supervisado y otro no supervisado. El primer enfoque está basado en redes neuronales recurrentes y características tales como palabras y caracteres embebidos, y alcanzó el mejor desempeño con una medida F_1 de 0.72 para la extracción de eventos, 0.64 para la modalidad contextual, 0.69 para

la polaridad, y 0.70 para el tipo de eventos. El segundo enfoque utilizó máquinas de vectores de soporte junto con características como tokens, y etiquetas de partes del enunciado, y alcanzó una medida F_1 de 0.76 para la detección de eventos, 0.69 para la modalidad contextual, 0.75 para la polariad y 0.75 por el tipo de evento.

MacAvaney *et al.* [6] entrenó campo aleatorio condicional, árboles de decisión y reglas junto con características tales como n-gramas, tokens, forma del tokens, agrupación de palabras, partes del enunciado, sintácticas, árboles de dependencia, semánticas, y UMLS (Unified Medical Language Systems). Ellos evaluaron sus resultados utilizando dos dominios: supervisado y no supervisado; el primer dominio alcanzó una medida F_1 de 0.71, 0.56, 0.65 y 0.68 para la extracción de eventos, modalidad contextual, polaridad, y tipo de evento, respectivamente. El segundo dominio alcanzó desempeños competitivos con una medida F_1 de 0.74 para la extracción de eventos, 0.66 para la modalidad contextual, 0.58 para la polaridad, y 0.72 para el tipo de eventos.

Finalmente, Leeuwenberg y Moens [5] utilizaron máquinas de vectores de soporte sobre características lingüisticas y diversas formas de un token. El sistema alcanzó un desempeño de F_1 de 0.68 para la extracción de eventos, 0.62 para la modalidad contextual, 0.67 para la polaridad, y 0.66 para el tipo de eventos utilizando aprendizaje supervisado.

2.5.3 Eventos

De acuerdo con [33], un evento es cualquier palabra que está relacionada con el ámbito médico, y que puede dar indicios en el cuidado o la vida del paciente. Por ejemplo, enfermedades, procedimientos, tumores, síntomas, entre otros.

Algunos ejemplos de eventos son:

- *The patient has a [contusion]*
- *She has an approximately [tumor] of 3cm…*
- *The [cancer] was detected in left upper [lung]…*
- *He presents [rectal] [bleeding] and persistent abdominal [discomfort]…*
- *The patient has [nausea], [diarrhea], [vomiting]…*

Cabe mencionar que debido a la protección de los datos de la clínica Mayo, los ejemplos presentados anteriormente y los que se presentaran en cada apartado son inventados para evitar problemas de derechos de autor.

2.5.3 Tipos de eventos

Los eventos son clasificados de acuerdo con el contexto y son descritos a continuación [34]:

Genérico

En esta categoría se encuentran los eventos relacionados con situaciones donde el médico escribe justificaciones por las cuales decidió cambiar el cuidado o tratamiento del paciente, y dichas situaciones no son reportadas en el historial clínico del paciente.

Ejemplos:

- *I highly recommended seeking second opinion for [breast cancer]…*
- *A [surgery] and [chemotherapy] is highly effective at killing cancer cells…*
- *For high-risk patient, the recommendation is a regular screening [mammograms]…*

Hipotético

Los eventos con esta modalidad son todos aquellos que son relevantes pero que son poco probables que ocurran, ya que su fundamento está fuertemente basado en suposiciones.

Ejemplos:

- *If he has [fever]…*
- *If she has a [runny nose]…*
- *If he has a really bad [headache]…*

Hedged

Esta modalidad implica sentencias que contenga las palabras como *"seems", "likely", "suspicious", "possible", "consistent with"* o frases como "*I suspect that*…", "*It would seem likely that*…", etc. Estos eventos no pueden ser justificados como reales por el médico debido a la falta de evidencia.

Ejemplos:

- *I suspect that an approximately [tumor] of 2.5 cm is growing in the R breast*
- *He presents fever not inconsistent with [varicella]*
- *It would seem likely that the patient presents [bulimia] due to ….*

Actual

Estos tipos de eventos son los más utilizados en los textos clínicos y son todos aquellos elementos que están ocurriendo o están siendo programados en el historial clínico de los pacientes.

Ejemplos:

- *The patient reports [threw up]*
- *She has [diarrhea] and also, presents [bleeding]*
- *She presents a new [tumor]*

Capítulo 3. Materiales y métodos

En este capítulo, se presentan los conceptos, métodos y materiales que se requieren para el desarrollo de la metodología que se presentará en el capítulo 4.

3.1 Modelo oculto de Markov

El modelo oculto de Markov es un método de estados finitos estocásticos, es decir, permite el movimiento de un estado a otro [35]. Fue desarrollado por Baum y Petrie, y es principalmente utilizado para representar la probabilidad de distribución sobre secuencia de observaciones [36]. Los siguientes elementos caracterizan a un modelo oculto de Markov:

1. N, número de estados en el modelo
2. $S = \{S_1, \dots, S_N\}$, es el conjunto de estados ocultos. Un estado en el tiempo t, es denotado como q_t.
3. $\pi = \{\pi_i\}$, es el conjunto de probabilidades iniciales, es decir, la probabilidad que inicie en un estado Si en el t = 1
4. T, es el número de observaciones dadas, y una observación de secuencias es denotada como $O = \{O_1, .., O_T\}$.
5. $A = \{a_{ij}\}$, define la matriz de probabilidades de transiciones entre el estado i en el tiempo t, al estado j en el tiempo $t+1$. Esta matriz es de tamaño $N*N$.
6. $B = \{b_i(k)\}$, define la matriz de probabilidades entre las observaciones dadas y los estados existentes. b_j es el símbolo de observación de la probabilidad en el estado j. Esta matriz es de tamaño $M*N$.
7. M, define el número de observaciones.

El modelo oculto de Markov utiliza la siguiente información:

$$\Lambda = \{A, B, \pi\} \quad 3.1$$

Existen tres principales problemas que pueden ser resueltos con los modelos ocultos de Markov [35]:

1. Evaluación: Dada la secuencia de observaciones $O = \{O_1, .., O_T\}$ y un modelo oculto de Markov λ, ¿Cómo calcular eficientemente la $P(O|\lambda)$?
2. Decodificación: Dada la secuencia de observaciones $O = \{O_1, .., O_T\}$ y un modelo oculto de Markov λ, ¿Cómo encontrar una secuencia óptima de una secuencia de estados $O = \{O_1, .., O_T\}$?
3. Entrenamiento: Dada la secuencia de observaciones $O = \{O_1, .., O_T\}$, ¿Cómo encontrar un modelo oculto de Markov λ que maximice la probabilidad de las observaciones $P(O|\lambda)$?

Para el problema de evaluación se ha utilizado el algoritmo de *forward backward recursive*, para el problema de decodificación es comúnmente utilizado Viterbi. Finalmente, el tercer problema, se puede resolver con el algoritmo de Baum-Welch [35].

De acuerdo con la definición del modelo oculto de Markov [37], la fórmula se muestra a continuación.

$$\hat{y} = argmax_y(y_0) \prod_{i=1}^{n+1} P(y_i|y_{i-1}) \prod_{i=1}^{n} p(x_i|y_i), \quad 3.2$$

y_0 se refiere a las etiqueta “inicio”, y y_{n+1} a una etiqueta final “stop”. $P(y_i|y_{i-1})$ representa la probabilidad de transición entre un estado y otro, y $p(x_i|y_i)$ es la probabilidad de un estado dado una observación.

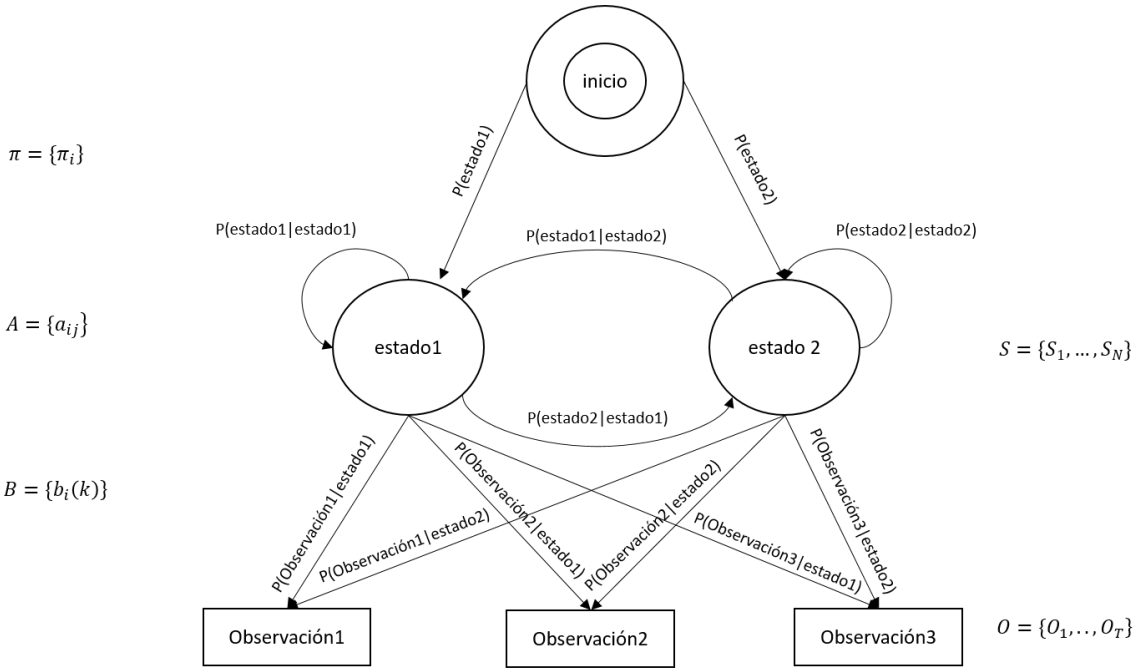

Figura 3. 1. Ejemplo de un modelo oculto de Markov con tres observaciones y dos estados

3.1.1. Algoritmo de Viterbi

El algoritmo de Viterbi fue propuesto en 1967 por Andrew Viterbi como un método de decodificación de códigos convolucionales [38]. Este método determina para cada observación, el mejor posible estado, y basa su entrenamiento en un modelo oculto de Markov de la siguiente forma [39]:

$$V_k(t) = P(I(t)|k)max_i(V_i(t-1) * T_{ik}, \quad 3.3$$

La ecuación 3.3 es descrita a continuación:

- $P(I(t)|k)$ es la probabilidad de una observación en el tiempo t en el estado k.
- T_{ik} es la probabilidad de transición entre un estado previo i a un nuevo estado k.

- $V_i(t-1)$ representa la probabilidad obtenida en la ruta anterior (*t-1*), y se caracteriza al inicio por:

$$V_{1,k} = P(l_1|k) * \pi_k, \qquad 3.4$$

donde

- $P(l_1|k)$ representa la probabilidad de la primera observación, es decir, primer tiempo.
- π_k es la probabilidad inicial.

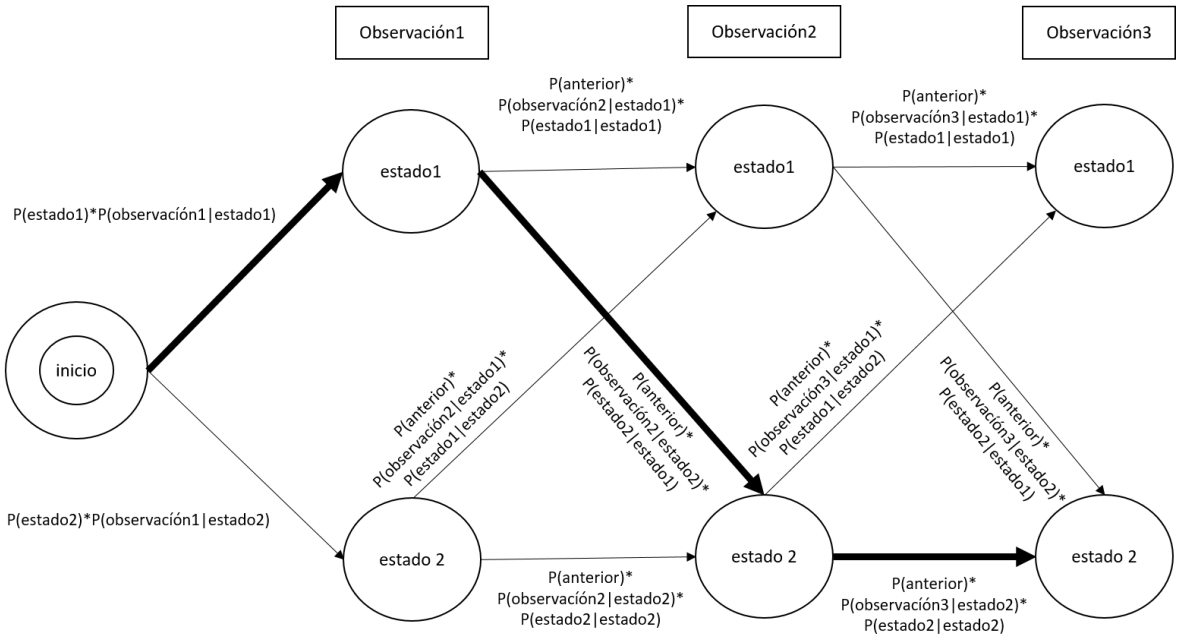

Figura 3. 2. Algoritmo de Viterbi

En la Figura 3.2 se observa el funcionamiento del algoritmo de Viterbi.

3.2 Evolución diferencial

El algoritmo de evolución diferencial fue presentado por primera vez por Price y Storn en 1995 [40]. La idea principal de este método es la optimización. Es decir, maximizar las propiedades de un sistema y reducir al mínimo sus características

indeseables [41]. Es conocido como un modelo estocástico para encontrar una solución óptima [42].

Este algoritmo consta de cuatro operaciones: inicialización, mutación, cruce y selección, y son descritos a continuación [40].

3.2.1 Inicialización

Una población de NP individuos es generada al azar utilizando la siguiente forma:

$$x_{ij} = x_j^{min} + (x_j^{max} - x_j^{min}) \cdot rand(0,1), \qquad 3.5$$

donde *i*= 1, 2, 3…, NP, *j*=1,2,…, D; x_j^{max}, x_j^{min} son los límites inferiores y superiores de los parámetros *j.*

3.2.2 Mutación

Una vez inicializada la población, el algoritmo de evolución diferencial muta y recombina la población para producir una población de NP vector. Es decir, suma un valor escalar (un número positivo seleccionado aleatoriamente, donde $F \in (0,1+)$) y la diferencia de dos vectores seleccionados al azar [41]. El vector mutante se genera de la siguiente forma:

$$V_i = x_a + F(x_b - x_c), \qquad 3.6$$

Los vectores X_a, X_b , X_c son vectores seleccionados aleatoriamente, y son seleccionados sólo una vez por mutante. El factor de mutación escalar $F \epsilon$ [0,2], es un

número real y constante, el cual nos permite controlar la amplificación de la variación diferencial [37].

3.2.3 Cruce

Con el objetivo de complementar la mutación diferencial, el algoritmo de evolución diferencial implementa una discreta recombinación entre el vector de prueba (vector mutante) y el vector padre, con el objetivo de intercambiar información. y es implementado de la siguiente forma [40][41]:

$$u_{ij} = \begin{cases} v_{ij} \ si \ rand\ [0,1]_j \leq Cr \vee j == j_{rand} \\ x_{ij} \qquad\qquad de\ otra\ forma \end{cases}, \tag{3.7}$$

Y es descrita a continuación:

- j = 1,2,3…., D, $rand[0,1]_j$ es un número real al azar entre [0,1]
- $j_{rand} \epsilon \{1,2,3, \dots, D\}$ es un índice seleccionado al azar, lo cual nos asegura que el vector de prueba obtenga al menos un parámetro del vector mutante.
- Cr es una constante predefinida dentro del rango [0,1] y controla la fracción de parámetros copiados del vector mutante.

3.2.4 Selección

Una vez realizada la operación anterior, el vector de prueba es comparado con el vector objetivo a través de una selección, la cual se lleva acabo de la siguiente forma:

$$x_i^* = \begin{cases} u_i, if\ (u_i) < f(x_i) \\ x_i, \ \ de\ otra\ forma' \end{cases} \tag{3.8}$$

Donde,

f(x) denota el objetivo de la solución *x* y x_i^* es un descendiente correspondiente al vector objetivo x_i [40].

3.3 *Hill climbing*

Hill climbing es un enfoque heurístico que encuentra un mínimo local óptimo [43]. Figura 3.3 muestra la arquitectura del algoritmo usado en esta tesis de investigación. El objetivo de este método de búsqueda local es mejorar una solución candidata [44]. En este caso la función *fitness* es el algoritmo de Viterbi usando la metrica F_1.

La idea de este algoritmo es comenzar a encontrar una solución óptima al comenzar a subir la colina. Esta técnica contiene las siguientes fases [45]:

1. Construir una solución subóptima obedeciendo las restricciones del problema.
2. Mejorar la solución paso a paso hasta que no sea posible más mejora con base en la función *fitness* seleccionada.
3. La técnica de *hill climbing* se utiliza principalmente para resolver problemas difíciles de cálculo. Se ve sólo en el estado actual y el estado futuro inmediato. Por lo tanto, esta técnica es eficiente en memoria ya que no mantiene un árbol de búsqueda.

El algoritmo de *hill climbing* se describe a continuación.

1. Evaluar un estado inicial
2. Repetir hasta que se encuentre una solución o no haya nuevos operadores pendientes de aplicar
3. Evaluar el nuevo estado
    a. Si el estado es mejor entonces remplazar el anterior estado

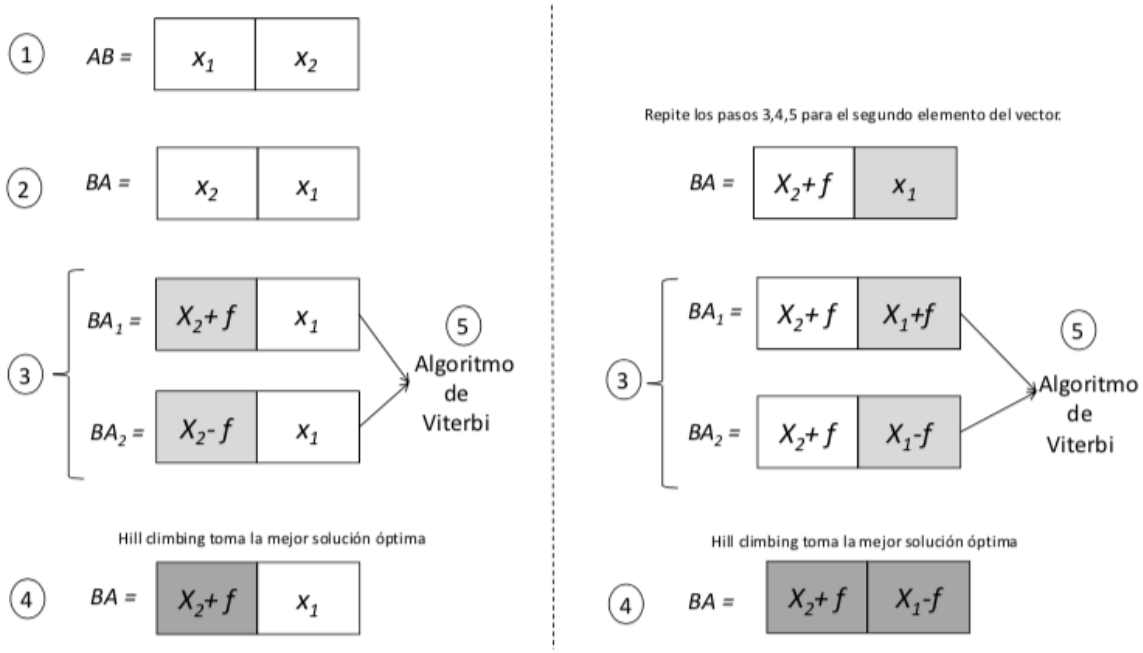

Figura 3.3. Ejemplo del uso de *hill climbing*, cada elemento del vector está modificado sumando y restando un valor random *f*, y es evaluado por el algoritmo de Viterbi para encontrar la mejor solución

3.4 Campo aleatorio condicional

El algoritmo de campo aleatorio condicional es un modelo de aprendizaje automático, para etiquetar y segmentar secuencias de datos. Este algoritmo basa su funcionamiento en una secuencia de palabras como entradas y produce una secuencia de etiquetas como salida [10], y es conocido como un algoritmo probabilístico basado en grafos.

La definición de este algoritmo es presentado en [46], y es detallado a continuación.

Definición 3.4. *Sea G = (V, E) un grafo tal que* $\boldsymbol{Y} = (\boldsymbol{Y_v})_{\boldsymbol{v \in V}}$, *de modo que* ***Y*** *está indexada por los vértices de G. Entonces,* ***(X, Y)*** *es un modelo de campo aleatorio*

condicional en caso, cuando se condiciona en $\boldsymbol{X}$*, la variable* $\boldsymbol{Y_v}$*, obedece las propiedades de Markov, de acuerdo con el grafo:* $\boldsymbol{p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v)}$*, donde* $w \sim v$ *significa que w y v son vecinos en G.*

Por lo tanto, se puede determinar que este algoritmo es un campo aleatorio globalmente condicionado a la variable X.

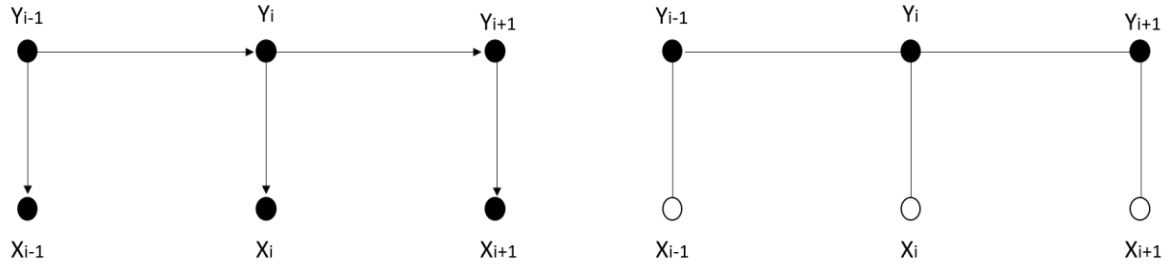

Figura 3. 4. Estructura gráfica de un simple modelo oculto de Markov (izquierda) y de un caso de campo aleatorio condicional (derecha)

El modelo representa la probabilidad de una secuencia de estados ocultos dando algunas observaciones [3], y ha sido aplicado en diferentes dominios y aplicaciones, tales como visión por computadora, bioinformática, y en tareas de procesamiento de lenguaje natural [47] , en la Figura 3.4, se observa que las variables en círculo blanco no son generadas por el modelo.

Una cadena lineal de campo aleatorio condicional es una distribución de probabilidad, y es mostrada a continuación [47]:

$$P(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_{\tau=1}^{T} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_\tau, y_{\tau-1}, \vec{x}\right), \tag{3.9}$$

En la ecuación 3.1, los diferentes términos son detallados a continuación:

- $\vec{x}$, representa una secuencia de palabras de tamaño τ.
- $\vec{y}$, representa una secuencia de etiquetas de tamaño τ.

- θ_k, es el conjunto de pesos asociados a cada función característica f_k, y el valor de k se encuentra entre (k = 1, 2, 3,…,K), donde, K es el número de características.
- f_k , es el conjunto de características, estás características establecen la correspondencia entre los elementos de entrada y salida, y forman la probabilidad de distribución, debido a la salida en tiempo τ-1 ($y_{\tau-1}$) está relacionada con la salida en tiempo τ(y_τ).
- $Z(\vec{x})$, es una función de normalización que permite asegurar una adecuada probabilidad de distribución. Y es formulada a continuación [48]:

$$\boldsymbol{Z(\vec{x})} = \sum_{\boldsymbol{y}} \prod_{\boldsymbol{\tau}}^{\boldsymbol{T}} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_\tau, y_{\tau-1}, \vec{x}\right) \tag{3.10}$$

Capítulo 4. Metodología experimental

En este capítulo, se presentan los métodos y materiales que se han utilizado durante los avances de esta tesis doctoral. Además, se describe el corpus utilizado para la predicción de eventos clínicos.

4.1 Metodología

En el desarrollo de esta tesis se propone el uso de una metodología para el análisis de textos clínicos, con base en el algoritmo de Viterbi y evolución diferencial debido a que estos algoritmos no han sido utilizados en el estado del arte para resolver la detección y clasificación de eventos clínicos.

1. Análisis del corpus clínico
2. Preprocesamiento de los datos a través de las técnicas de procesamiento de lenguaje natural
3. Implementación y validación del algoritmo de Viterbi y otros algoritmos, en este caso campo aleatorio condicional
4. Aplicación de evolución diferencial (Véase sección 3.2)
5. Selección de la mejor matriz de transición $A = a_{ij}$ y mejor matriz de emisión de probabilidades $B = \{b_i(k)\}$ que producen el mejor valor de la medida F_1.
6. Si el resultado del modelo no es competitivo, utilizar el algoritmo más empleado en el área de reconocimiento de entidades nombradas.

4.1.1 Análisis del corpus clínico

Los textos clínicos a utilizar provienen de la clínica Mayo, la cual es una institución médica sin fines de lucro, comprometida con la práctica clínica, la educación y la

investigación. Su sede principal se encuentra en Minnesota, y sus dependencias para la investigación están ubicadas en Florida, y Arizona.

Los datos se recolectaron a través del proyecto THYME *("Temporal Histories of Your Medical Events"*), y están enfocados en notas clínicas y patológicas de pacientes con cáncer de colón. Cabe mencionar que el corpus THYME fue obtenido a través del registro en el *International Workshop on Semantic Evaluation* (*SemEval*) [33].

En el análisis del corpus, se puede observar que contiene una gran cantidad considerable de signos de puntuación que no permiten separar correctamente una palabra para ser tokenizada. Además, en algunos casos no existe un espacio en blanco entre las palabras. Por lo tanto, el uso de técnicas de procesamiento de lenguaje natural y expresiones regulares son indispensables para mejorar la calidad del corpus.

4.1.2 Técnicas de procesamiento de lenguaje natural

Las técnicas de preprocesamiento de lenguaje natural ayudan a mejorar la precisión y eficiencia de los procesos de análisis de datos.

En este corpus se realizó el siguiente preprocesamiento:

- Separación de palabras (*Tokenization*)
- Eliminación de *stopwords* y signos de puntuación
- Normalización de palabras en minúsculas
- Lematización, utilizando el lematizador de NLTK
- Identificación de las partes del enunciado (*Part of speech*),
- Etiquetas chunk, de acuerdo con las etiquetas (BIO, Begin, Inside, Outside)
- Uso de UMLS
- Aplicación de bolsa de palabras y bigramas
- Separación de cada uno de los textos por:

- sección
- sentencias, utilizando el sistema de NLTK
- Palabras

4.1.3 Implementación y validación del algoritmo de Viterbi

En la validación del modelo, el corpus está dividido en tres subconjuntos: entrenamiento, validación y prueba, a fin de evitar el fenómeno de sobreajuste *(overfitting).*

En la implementación del modelo, se encontró una limitante del algoritmo de Viterbi, la cual es la asunción que las observaciones son de una dimensión, por lo tanto, se empleó una de las soluciones presentadas por Phil Blunson [49]. Esta propuesta consiste en usar un modelo donde se asuma que las características de las observaciones son independientes.

$$v_k = (f_1, \dots, f_N) \tag{4.1}$$

$$P(v_k|s_j) = \prod_{j=1}^{N} P(f_j|S_j) \tag{4.2}$$

Este modelo es fácil de implementar y computacionalmente simple, y es aplicado en el desarrollo del algoritmo de Viterbi (Véase Figura 4.1).

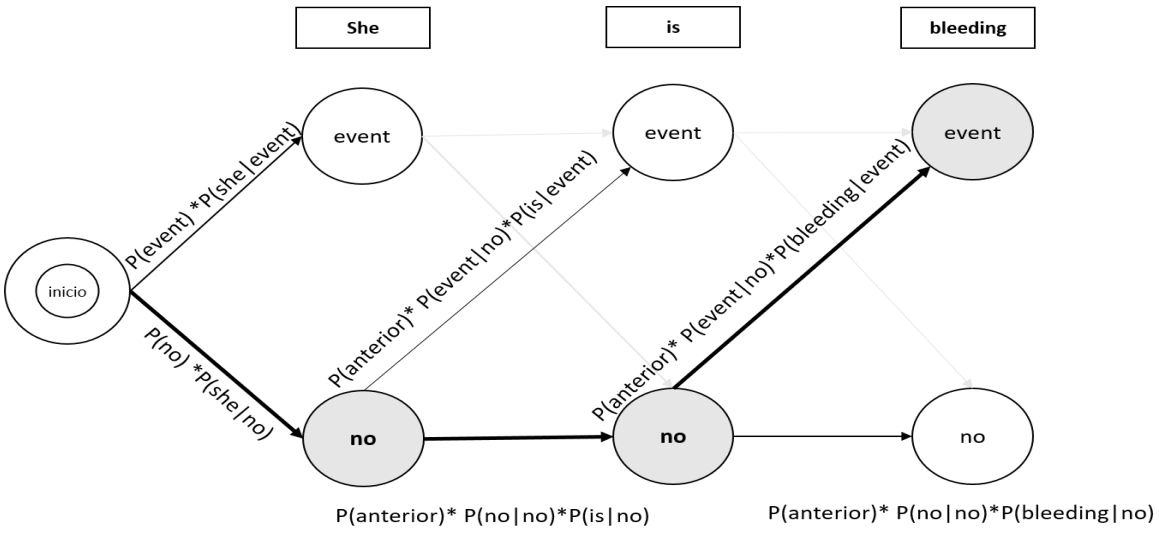

Figura 4. 1. Aplicación del algoritmo de Viterbi en una oración con tres palabras

4.1.3.1 Aplicación de evolución diferencial

El proceso de esta algoritmo es presentando a continuación (Véase Figura 5.2)

a) Inicialización: Durante la inicialización, la matriz de emisión de probabilidades $\boldsymbol{B} = \{\boldsymbol{b_i}(\boldsymbol{k})\}$ y la matriz de transición $\boldsymbol{A} = \{\boldsymbol{a_{ij}}\}$, son transformadas en un sólo vector **ab**, de una dimensión, de la siguiente forma [50]:

$$A = \{a_{ij}\} = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} = [a_{11}, a_{12}, \dots, a_{NN}] \quad 4.3$$

$$B = \{b_i(k)\} = \begin{bmatrix} b_{11} & \cdots & b_{1M} \\ \vdots & \ddots & \vdots \\ b_{N1} & \cdots & a_{NM} \end{bmatrix} = [b_{11}, b_{12}, \dots, b_{NN}] \quad 4.4$$

$$AB = [a_{11}, a_{12}, \dots, a_{NN}, b_{11}, b_{12}, \dots, b_{NN}] \quad 4.5$$

El vector **AB** forma el primer elemento de la población y representa un cromosoma.

Posteriormente, la población inicial de padres es formada multiplicando cada elemento $\mathbf{AB_i}$ del vector **AB** por un valor $x \epsilon R^+$ seleccionado al azar.

Los parámetros del algoritmo de evolución son definidos como:

- D, representa la longitud del vector **AB**
- CR = 0.8, es la constante de cruce.
- F, es un número $x \epsilon R^+$
- NP = 150, representa el tamaño de la población
- G = 120, es el número de generaciones

b) Función *fitness*: De acuerdo con el algoritmo de Viterbi que permite seleccionar el mejor estado posible a cada observación dada (Véase sección 3.1.1). La medida F_1 fue seleccionada, para que exista un balance en las clases, la cual es calculada de la siguiente forma:

$$2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \qquad 4.6$$

c) Mutación. Selecciona al azar tres vectores de la población, para calcular la Ecuación 3.6 (Véase sección 3.2.2).

d) Cruce. Con el fin de intercambiar información entre el vector mutante y vector padre, se lleva a cabo la Ecuación 3.7, y recuérdese que durante la experimentación el valor de la constante CR es de 0.8.

e) Condición de paro. El algoritmo de evolución diferencial termina el proceso si el error de criterio ha sido alcanzado o bien, el número de iteraciones ha terminado.

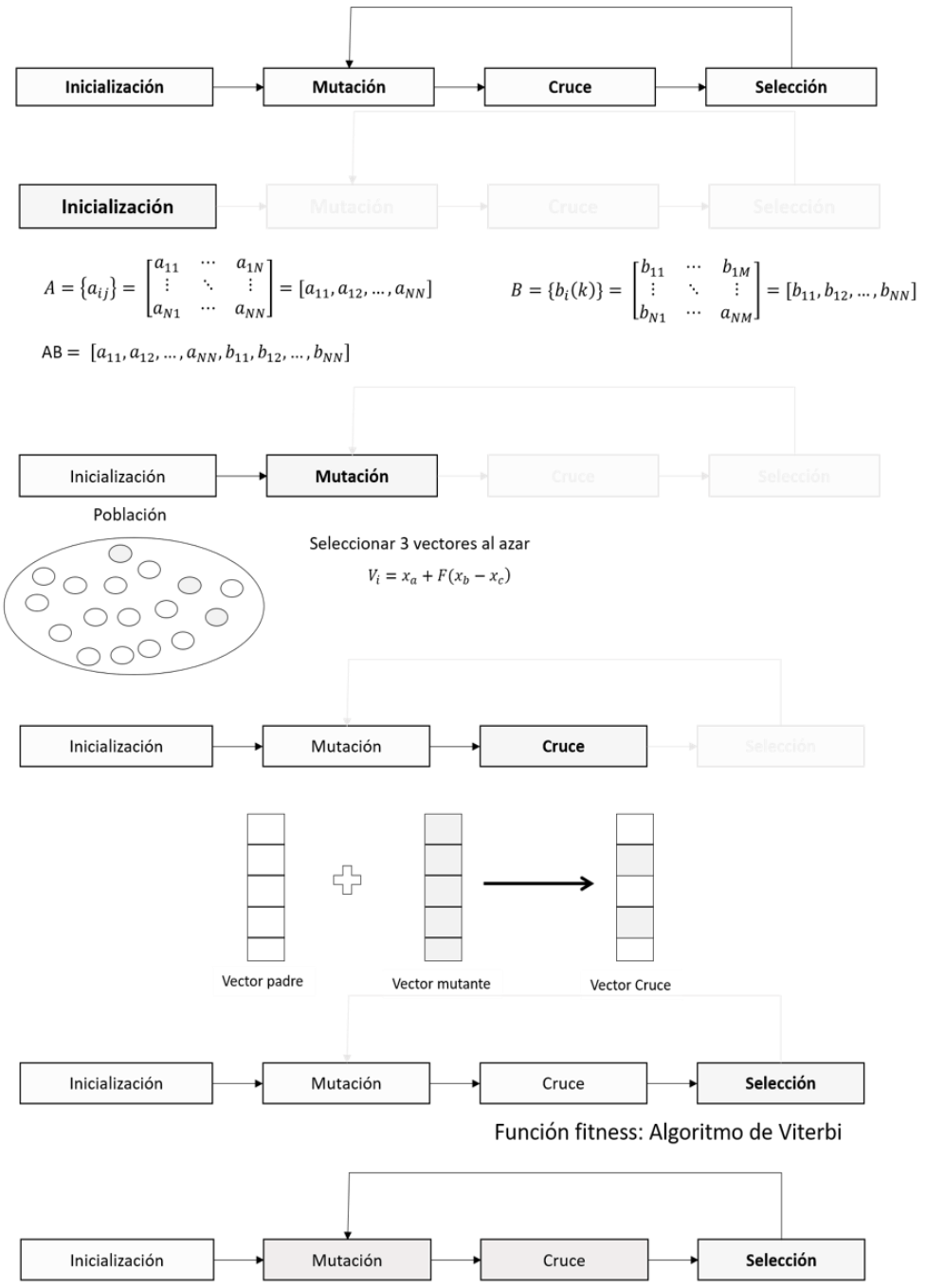

Figura 4. 2. Descripción gráfica del algoritmo de evolución diferencial para la optimización de los parámetros del algoritmo de Viterbi

4.1.3.2 Aplicación de hill climbing

El algoritmo se describe a continuación, y una descripción gráfica es presentada en la Figura 4.3.

1. Usando los parámetros de Viterbi, la matriz de transición y la matriz de emisión de probabilidad (A_{ij} y B_{ij}) son concatenados en un vector de probabilidad ***AB***, como se presenta a continuación [50].

$$A = \{a_{ij}\} = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} = [a_{11}, a_{12}, \dots, a_{NN}] \quad 4.3$$

$$B = \{b_i(k)\} = \begin{bmatrix} b_{11} & \cdots & b_{1M} \\ \vdots & \ddots & \vdots \\ b_{N1} & \cdots & a_{NM} \end{bmatrix} = [b_{11}, b_{12}, \dots, b_{NN}] \quad 4.4$$

$$AB = [a_{11}, a_{12}, \dots, a_{NN}, b_{11}, b_{12}, \dots, b_{NN}] \quad 4.5$$

Los parámetros del algoritmo de evolución son definidos como:

- D, representa la longitud del vector ***AB***
- F, es un número $x \epsilon R^+$

2. Determinar el umbral de pausa, es decir, el error mínimo alcanzado, *u*.
3. Los elementos del vector ***AB*** son reordenados al azar dentro de un arreglo llamado, ***BA***, con el objetivo de reducir el tiempo de cómputo.

$$BA = [a_{21}, a_{12}, \dots, a_{NN}, b_{21}, b_{12}, \dots, b_{NN}] \quad 4.7$$

4. Para cada elemento del vector, ***BA***, sumar y restar simultáneamente un valor aleatorio *f.*

$$BA_{i=0}^{i=n} = BA_i + f \quad (4.8)$$

$$BA_{i=0}^{i=n} = BA_i - f \quad (4.9)$$

5. La metodología propuesta evalúa los dos resultados usando la función *fitness* de Viterbi, y selecciona el valor que produce la menor tasa de error en la clasificación, de acuerdo con la medida F_1. En este caso la función *fitness* se basa en la medida F_1 para que exista un balance en las clases, la cual se calcula de la siguiente forma:

$$2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (4.6)$$

Recuérdese que el funcionamiento de Viterbi se basa en la selección del mejor estado posible de acuerdo con la probabilidad más alta. (Véase sección 3.1.1).

5.1 Si uno de los valores produce una tasa de error más pequeña, asignar este nuevo valor al vector $\boldsymbol{BA_i}$

5.2 Si ambos elementos proporcionan el mismo resultado, entonces continuar con la siguiente posición en el vector $\boldsymbol{BA_i}$, y repetir los pasos 4 y 5.

6. Finalmente, *hill climbing* se detiene cuando todos los valores en el vector, ***BA***, son evaluados para encontrar la mejor solución, o bien se haya alcanzado el umbral de pausa, *u*.

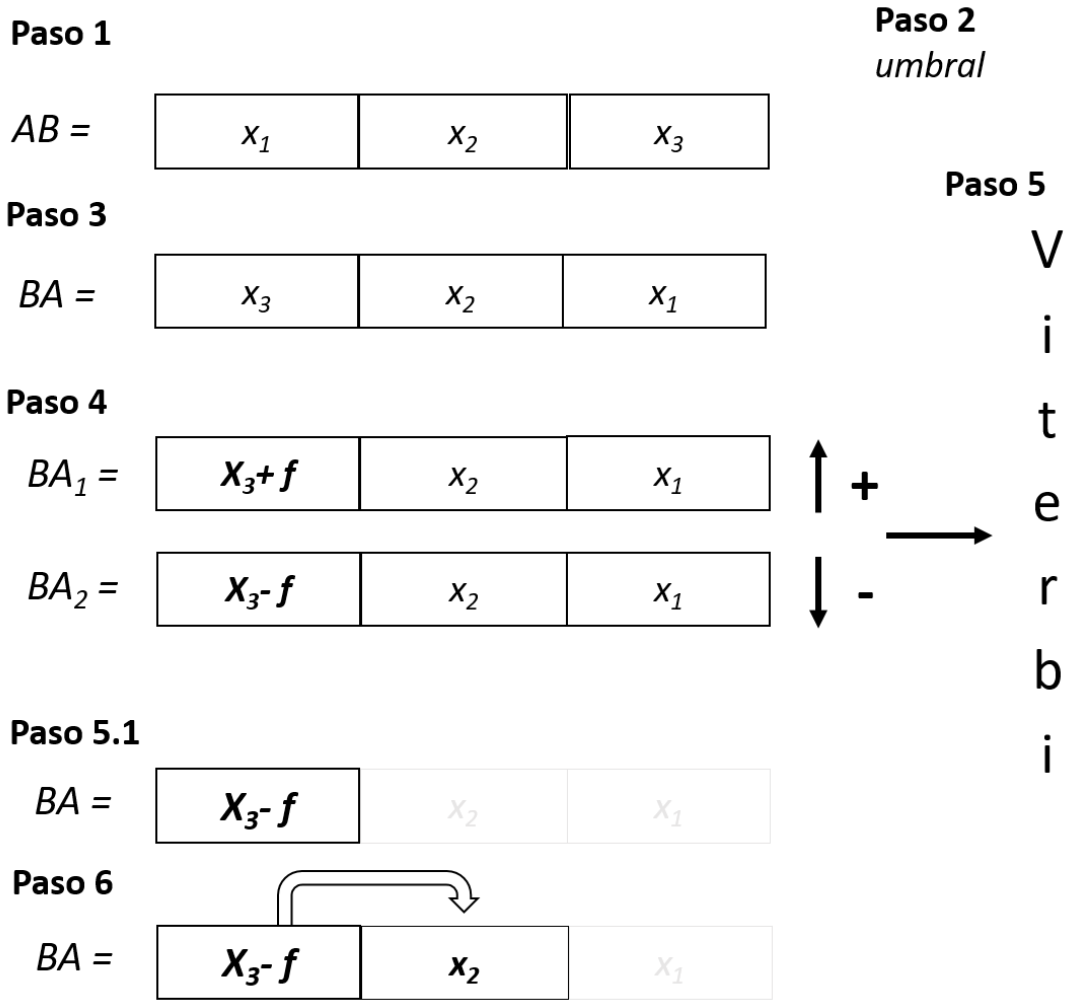

Figura 4. 3. Descripción gráfica del algoritmo de *hill climbing* para la optimización de los parámetros de Viterbi

4.1.4 Implementación y validación de campo aleatorio condicional

De acuerdo con el conjunto de datos, el algoritmo de campo aleatorio condicional resulta ser el adecuado para la identificación y clasificación de eventos clínicos debido a que permite asignar pesos a las características durante el proceso de entrenamiento; además de ser flexible y fácil de utilizar aplicando la herramienta de CRF Suite [51].

Con base en los resultados que se presentan en el capítulo 5, confirman que efectivamente fue el clasificador indicado para resolver las tareas presentes en la competencia de SemEval 2016.

En la validación del modelo, el corpus está dividido en tres subconjuntos: entrenamiento, validación y prueba, a fin de evitar el fenómeno de sobreajuste *(overfitting),* y se entrenó con la combinación de los subconjuntos de entrenamiento y validación.

4.1.5 Resultados

Después de aplicar los métodos de procesamiento de lenguaje natural, y de la validación de los modelos seleccionados, se realizará una serie de comparaciones entre los algoritmos de campo aleatorio condicional, Viterbi y evolución diferencial para determinar si los resultados de estos modelos son competitivos. Posteriormente, serán analizados a través del sistema SemEval 2016, y se realizará una comparación entre los modelos clásicos reportados en el estado del arte.

Cabe mencionar que el algoritmo de campo aleatorio condicional no fue implementado, se utilizó la herramienta CRF Suite [51].

En este capítulo se presentó la metodología propuesta en este trabajo de investigación, la cual parte del trabajo presentado en la competencia de SemEval 2016, y cuyos pasos se basan en el análisis y la comprensión del corpus clínico, hasta la aplicación y la validación de los algoritmos de clasificación. Es preciso indicar que el algoritmo de Viterbi, y las técnicas de optimización (evolución diferencial y *hill climbing*) no habían sido considerados en el estado del arte para detectar y clasificar eventos clínicos.

Capítulo 5. Resultados experimentales

En este capítulo, se presentan los resultados de las pruebas realizadas utilizando los algoritmos de Viterbi, campo aleatorio condicional, y las técnicas de optimización: evolución diferencial y *hill climbing*. Además, una comparación con los resultados obtenidos en este estudio, y los algoritmos reportados en el workshop de SemEval 2016: Tarea 12-Clinical TempEval es presentada. Además, esta sección se presenta de acuerdo con las publicaciones derivadas de este trabajo de tesis.

5.1 Descripción del corpus

El corpus clínico utilizado en este estudio fue recabado por el proyecto THYME, por su término en inglés "*Temporal Histories of Your Medical Events*", y está constituido por 750 reportes clínicos divididos por documentos patológicos y clínicos de pacientes con cáncer de colón de la clínica Mayo. En la Tabla 5.1 se presenta un resumen del corpus clínico después de aplicar técnicas de procesamiento de datos.

Tabla 5. 1. Descripción del corpus clínico

Conjunto	# documentos	# sentencias	Entrenamiento	Validación
Train + Dev	597	15,809	202,803	17,741
Test	153	18,157	221,547	18,946

De acuerdo con [52], el conjunto de datos de entrenamiento se utiliza para ajustar los parámetros de los modelos seleccionados, el conjunto de validación se emplea para estimar los errores de predicción de los modelos, y el conjunto de test se utiliza para evaluar la tasa de error para generalizar nuevas instancias. En este caso, los conjuntos. *dev* y *train*, son utilizados para entrenar los modelos.

5.2 Resultados experimentales

De acuerdo con el diseño de los experimentos, los algoritmos fueron entrenados con los conjuntos de entrenamiento y validación. Posteriormente, los algoritmos son comprobados con el conjunto de prueba. Cabe mencionar que la evaluación de los algoritmos se llevará a cabo a través de las siguientes medidas:

$$\text{Precisión} \qquad \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.1}$$

$$\text{Recall} \qquad \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.2}$$

$$F_1 \qquad 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{5.3}$$

$$\text{Accuracy} \qquad \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{5.4}$$

Además con el objetivo de hacer una comparación con los trabajos presentados en la competencia de *SemEval 2016: Task 12-Clinical TempEval* [53], se utilizó el sistema proporcionado por dicha competencia.

5.2.1. *Aplicación de campo aleatorios condicionales y ajustes de los parámetros de las cadenas de Markov*

En esta publicación se utilizaron los algoritmos de campo aleatorio condicional, y modelos de optimización como evolución diferencial y *hill climbing* para mejorar los resultados de una variante de los modelos ocultos de Markov, conocido como el algoritmo de Viterbi, para resolver las tareas de detección y clasificación de eventos.

5.2.1.1 Descripción de experimentos

En este experimento, sólo se realizaron dos tareas: detección de eventos y clasificación de los eventos con base en su modalidad contextual

1. **Detección de eventos.**

   En este experimento, se identifican las palabras que son eventos clínicos, es decir, procedimientos, enfermedades, síntomas, riesgos, entre otros.

2. **Modalidad contextual**

   Durante este experimento, las palabras detectadas como eventos son clasificadas dentro de las siguientes categorías:

- **Protección,** indica eventos que están fuertemente ligados a situaciones donde no existe suficientes evidencias, seguridad o responsabilidades.
  Hipotético, consiste en eventos, cuyo fundamento está basado en asunciones.
- **Genérico,** indica que el evento está relacionado a situaciones donde un médico proporciona justificaciones por las acciones y decisiones tomadas.
- **Actual,** representa a los eventos que ya ocurrieron, o bien están programados en un futuro.

5.2.1.2 Características

Las características que se emplearon en este experimento son presentadas a continuación.

- **Palabra:** Palabra
- **Minúscula:** Palabra en minúscula
- **Lema:** Lema de la palabra que se está evaluando
- **Partes del enunciado:** Forma de una palabra en una sentencia: sujeto, pronombre, adjetivo, verbo, adverbio preprosiciones, conjunciones, etc.

- **Chunk:** Extracción de frases usando etiquetas BIO (Begin, Inside, Outside)
- **Tipo:** Extracción de frases usando etiquetas BIO (Begin, Inside, Outside)
- **Número de sección:** Sección a la que pertenece la palabra que se está evaluando

Tabla 5. 2. Variables utilizadas en el experimento 1

word	lower	lemma	POS	chunk	type	section_id
Patient	patient	patient	NNP	Begin-NNP	word	20107
is	is	be	VERB	Inside-V	word	20107
bleeding	bleeding	bleed	VERB	Inside-V	word	20107
.	.	.	.	.	punct	20107

5.2.1.3 Experimentos

1. **Detección de eventos.**

En la Figura 5.1 se muestran los resultados obtenidos utilizando los algoritmos de campo aleatorio condicional, Viterbi, y las técnicas de optimización: evolución diferencial y *hill climbing*.

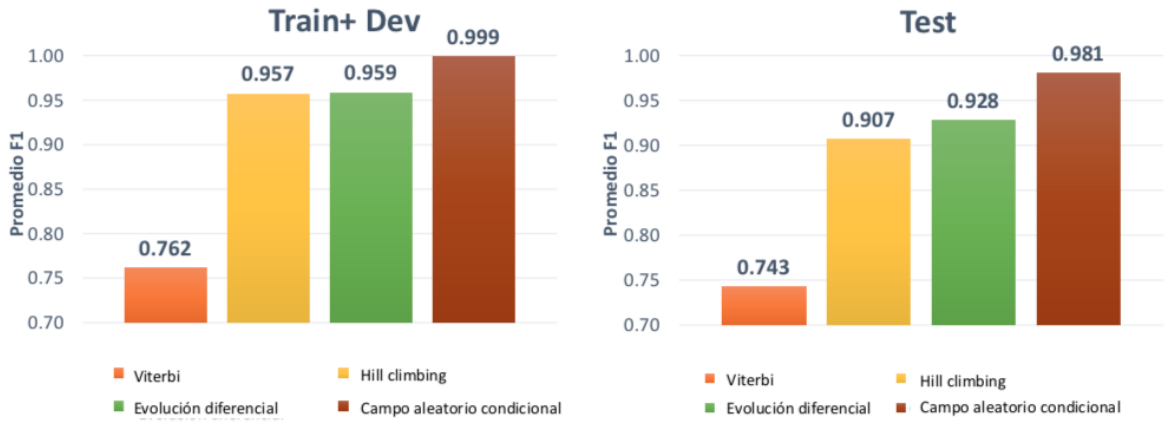

Figura 5. 1. Resultados comparativos entre el clasificador de campo aleatorio condicional, y el algoritmo de Viterbi con dos técnicas de optimización: evolución diferencial y *hill climbing*

El resultado de los algoritmos en el corpus de entrenamiento son los siguientes: campo aleatorio condicional obtuvo una promedio en la medida F_1 de 0.999, seguido de Viterbi con evolución diferencial, y *hill climbing* con un desempeño de 0.959, y 0.957, respectivamente. Es evidente que el algoritmo de Viterbi sin ninguna técnica de optimización obtuvo el peor desempeño en el corpus de entrenamiento.

Con base en el resultado en el corpus de prueba, el algoritmo de campo aleatorio condicional obtuvo un promedio en la medida F_1 de 0.981, y supera los resultados presentados por Viterbi junto con las técnicas de optimización de evolución diferencial, y *hill climbing* con 0.928, y 0.907 en la medida F_1. Finalmente, Viterbi presentó una medida F_1 de 0.743, lo cual demuestra que las técnicas de optimización mejoraron considerablemente el desempeño de este algoritmo de clasificación.

La Tabla 5.3 detalla los resultados de los algoritmos empleados en la detección de eventos clínicos.

Tabla 5. 3. Resultados obtenidos para la detección de eventos clínicos

Clasificador	Clase	Train + Dev			Test		
		Precisión	Recall	F_1	Precisión	Recall	F_1
Campo aleatorio condicional	**Evento**	**0.999**	**0.999**	**0.999**	**0.981**	**0.950**	**0.965**
	No evento	0.999	0.999	0.999	0.995	0.998	0.997
Viterbi con evolución diferencial	Evento	0.930	0.920	0.925	0.874	0.864	0.869
	No evento	0.992	0.993	0.996	0.987	0.988	0.987
Viterbi con *hill climbing*	Evento	0.944	0.901	0.922	0.811	0.854	0.832
	No evento	0.990	0.995	0.992	0.986	0.980	0.983
Baseline: Viterbi	Evento	0.429	0.974	0.595	0.402	0.943	0.563
	No evento	0.997	0.869	0.929	0.994	0.862	0.923

2. Modalidad contextual

En esta sección se presentan los resultados de los algoritmos: campo aleatorio condicional, y Viterbi con las técnicas de optimización seleccionadas para determinar si un evento es clasificado como actual, hipotético, genérico, o protección.

En la Figura 5.2, se presentan los resultados en los corpus de entrenamiento y prueba. Campo aleatorio condicional obtuvo un promedio en la medida F_1 de 0.985 seguido de Viterbi con evolución diferencial con una promedio en la medida F_1 de 0.604, y *hill climbing* de 0.522. Finalmente, Viterbi obtuvo los peores resultados con 0.482 en la medida F_1.

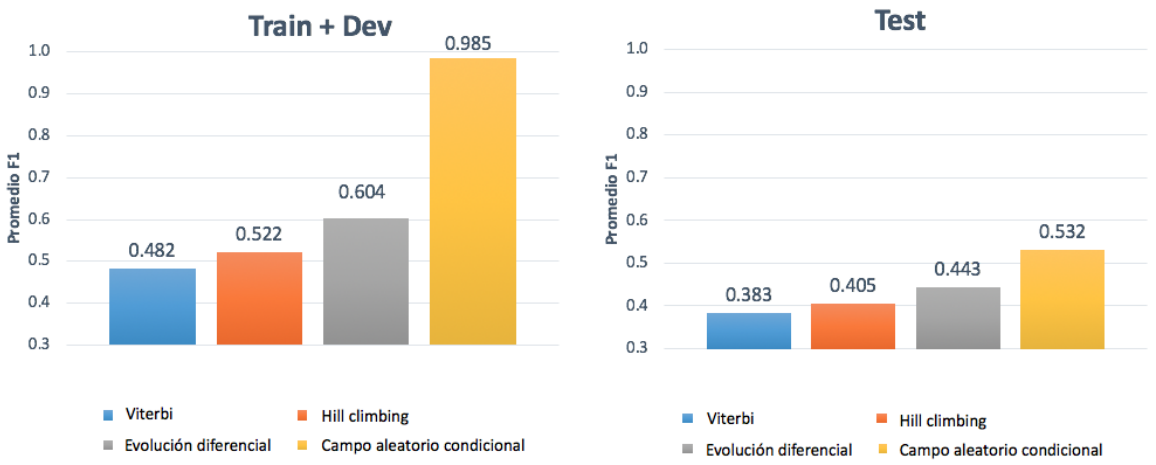

Figura 5. 2. Representación gráfica de los resultados obtenidos en términos de F_1 de campo aleatorio condicional, y Viterbi con evolución diferencial y *hill climbing* en la tarea de clasificación con base en la modalidad contextual

De acuerdo con los experimentos en el corpus de prueba, se observó que campo aleatorio condicional es el mejor algoritmo para la clasificación de eventos. Sin embargo, los resultados demostraron una baja efectividad de la clasificación, ya que los resultados son un promedio en la medida F_1 de 0.532, seguido por Viterbi con evolución

diferencial de 0.443, Viterbi con *hill climbing* de 0.405, y Viterbi con 0.383, por lo cual, se sospecha que los algoritmos sufren de "*overfitting*", ya que, éstos no son capaces de generalizar correctamente los nuevos datos presentados en el corpus de entrenamiento.

En la Tabla 5.4 se muestran los resultados en términos de precision, recall y F_1 cuando el corpus clínico es usado para el entrenamiento y prueba. Estos resultados demostraron que los algoritmos tuvieron problemas para generalizar los nuevos datos en las clases: hipotético, protección y genérico. Además, se observa que aunque el promedio general de los resultados de campo aleatorio condicional superó los resultados de los otros algoritmos propuestos, no existe diferencia significativa entre los resultados de Viterbi con evolución diferencial, y Viterbi con *hill climbing*.

Tabla 5. 4. Evaluación de los resultados para la tarea de clasificación de eventos con base en la modalidad contextual

Clasificador	Clase	Train + Dev			Test		
		Precisión	Recall	F_1	Precisión	Recall	F_1
Campo aleatorio condicional	N/A	**0.999**	**0.999**	**0.999**	**0.995**	**0.998**	**0.997**
	Actual	0.997	0.998	0.998	0.918	0.946	0.932
	Hipotético	0.981	0.963	0.972	0.724	0.237	0.357
	Protección	0.979	0.956	0.967	0.443	0.073	0.125
	Genérico	0.993	0.993	0.990	0.607	0.154	0.246
Viterbi con evolución diferencial	N/A	0.992	0.939	0.965	0.987	0.924	0.954
	Actual	0.571	0.896	0.697	0.490	0.847	0.621
	Hipotético	0.473	0.518	0.494	0.337	0.242	0.282
	Protección	0.410	0.329	0.347	0.189	0.139	0.160
	Genérico	0.441	0.567	0.496	0.180	0.223	0.199
Viterbi con *hill climbing*	N/A	0.988	0.911	0.948	0.982	0.898	0.938
	Actual	0.448	0.770	0.566	0.389	0.718	0.505

	Hipotético	0.292	0.655	0.404	0.195	0.322	0.243
	Protección	0.214	0.578	0.312	0.118	0.266	0.164
	Genérico	0.258	0.706	0.378	0.114	0.361	0.173
Baseline: Viterbi	N/A	0.997	0.867	0.928	0.994	0.857	0.921
	Actual	0.383	0.857	0.530	0.348	0.826	0.489
	Hipotético	0.237	0.657	0.349	0.157	0.335	0.213
	Protección	0.184	0.583	0.280	0.101	0.270	0.147
	Genérico	0.208	0.716	0.322	0.089	0.361	0.142

5.2.1.4 Conclusiones

En este trabajo se llevaron a cabo dos experimentos: i) detección de eventos y ii) modalidad contextual, y para la evaluación de los resultados, se utilizó el corpus y el sistema proporcionados por la tarea 12 de SemEval [53]. Los resultados demostraron que el método propuesto, es decir, campo aleatorio condicional superó los resultados presentados en la competencia en las dos tareas realizadas.

También es importante, resaltar que para la extracción de las características, se extrajeron características simples tales como bolsa de palabras, partes del enunciado, entre otras, las cuales superaron sofisticados sistemas que usaron características como UMLS (sistemas médicos de lenguaje unificado), agrupaciones Brown, o usando herramientas como CLAMP (*Clinical Language Annotation, Modeling, and Processing Toolkit*).

Otra de las contribuciones en este trabajo, es el estudio del desempeño de otro popular clasificador conocido como Viterbi. Este algoritmo es mucho más simple de implementar que campo aleatorio condicional. Sin embargo, es conocido por producir resultados inferiores que campo aleatorio condicional o máquinas de vectores de soporte.

No obstante, se demostró que si los parámetros de Viterbi son optimizados utilizando técnicas como evolución diferencial, o *hill climbing* puede lograr resultados competitivos de acuerdo con el estado del arte. Cabe mencionar que estas técnicas de optimización son costosas en términos de procesamiento, y en caso de evolución diferencial, memoria.

5.2.2 *Aplicación de campo aleatorio condicional con características adicionales*

En esta publicación se consideraron más características lingüísticas, léxicas, sintñacticas, nivel de discuso, entre otras. Además, sólo se tomó en consideración el algoritmo de campo aleatorio condicional y se anexaron tareas como polaridad y tipo de eventos.

5.2.2.1 Descripción de experimentos

Adicionalmente a las tareas presentadas en la sección 5.2.1.1, en esta publicación se anexaron las siguientes tareas:

3. **Polaridad**

Esta tarea mide dentro de dos categorías (positivo o negativo) la polaridad del evento con base en el contexto de la sentencia.

4. **Tipo**

El objetivo de esta tarea, es determinar el tipo de información aspectual en la que el evento es detectado.

- **n/a,** representa el valor por default del evento, y no debe confundirse con palabras que no son eventos.
- **aspectual,** relaciona eventos que pueden continuar o reaparecer durante el tratamiento, o evolución del cáncer.
- **evidencial,** proporciona detalles del evento con base en evidencias o demostraciones obtenidas desde imágenes, pruebas u observaciones humanas.

5.2.2.2 Características

Las características que se emplearon en este experimento son presentadas a continuación.

- **Lingüísticas:** Identifica el lema y las partes del enunciado (*part-of-speech*)
- **Palabras:** Toma en consideración la forma simple del token que se está evaluando.
- **Léxicas:** Incluye prefijo y sufijo de cada palabra extraída.
- **Nivel de discurso:** Representa el tamaño de la palabra, tipo (palabra, números, símbolos), e información de la sección a la que pertenece el token en evaluación.
- **Recursos externos:** para obtener una clave de los términos en el corpus clínico, nosotros usamos *UMLS* por su término en inglés *Unified Medical Language System*, para asociar recursos y servicios de información biomédica de manera más eficiente [54]
- **Representación de palabras:** los espacios semánticos se obtuvieron con *word2vec*, donde las dimensiones utilizadas durante estos experimentos son 20, 100 y 300, con un tamaño de vocabulario de aproximadamente 202,000 palabras.

Con base en las características presentadas anteriormente, se realizó la siguiente combinación:

- **LWD:** Lingüística (lema y pos), forma simple de una palabra (palabra, palabra en minúscula), y nivel de discurso (tamaño de la palabra, tipo e información de la sección). Una representación de esta combinación es presentada en la Tabla 5.3.
- **LWDLE:** LWD más características léxicas (prefijo y sufijo), y recursos externos (UMLS).
- **WR:** Representación de palabras con diferentes dimensiones (20, 100 y 300) utilizando la herramienta de *word2vec*.

Tabla 5. 5. Ejemplo de las variables utilizadas en la combinación de características LWD

word	lowercase	lemma	POS	size	token_type	section_id
sections	sections	section	NNS	8	word	5000
.	.	.	NN	.	numeric	.

5.2.2.3 Experimentos

1. **Detección de eventos**

En la Figura 5.3, se observa una gráfica comparativa utilizando exclusivamente campo aleatorio condicional y diversas combinaciones de características (LWD, LWDLE, y WR). Posteriormente, los resultados de cada una de las propuestas es presentada en la Tabla 5.6.

De acuerdo con los resultados en el corpus de prueba, la combinación de características LWDLE, proporcionó el mejor desempeño, alcanzando un promedio en la medida F1 de 0.926, seguido de LWD y WR con un promedio en F1 de 0.925 y 0.915, respectivamente. Sin embargo, LWDLE no proporcionó una significativa mejora en el desempeño cuando es comparado con LWD y WR con 300 dimensiones (Véase Figura 5.3).

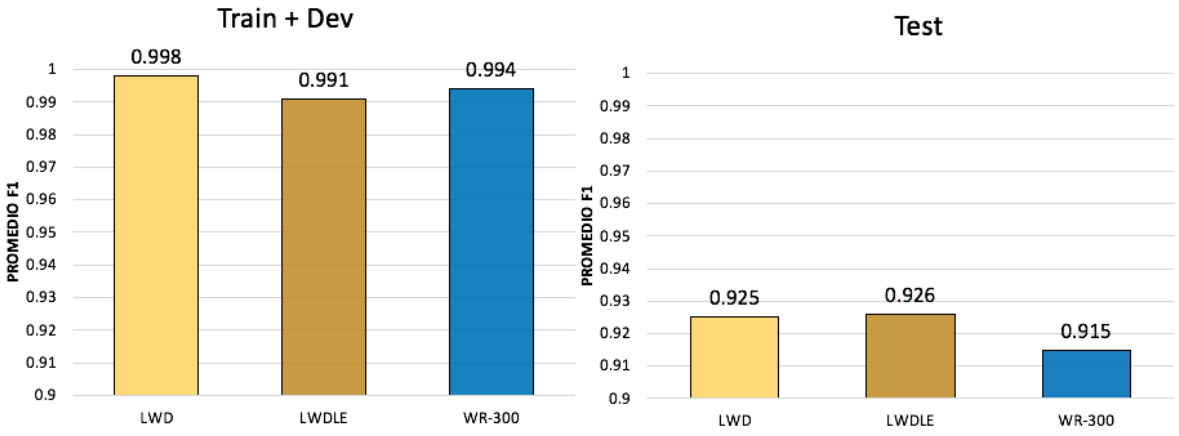

Figura 5. 3. Resultados comparativos del algoritmo de campo aleatorio condicional con las diferentes características en la tarea de detección de eventos

Tabla 5. 3. Resultados detallados del algoritmo de campo aleatorio condicional utilizando diversas características

Clasificador	Clase	Train + Dev			Test		
		Precisión	Recall	F_1	Precisión	Recall	F_1
LWD	**Evento**	**0.985**	**0.981**	**0.998**	**0.875**	**0.851**	**0.863**
	No evento	0.998	0.998	0.998	0.985	0.987	0.986
LWDLE	Evento	0.985	0.982	0.983	0.881	0.851	0.866
	No evento	0.998	0.998	0.998	0.985	0.988	0.986
WR-300	Evento	0.955	0.995	0.994	0.980	0.990	0.985
	No evento	0.994	0.995	0.994	0.889	0.805	0.845

2. Modalidad contextual

En este experimento, los mejores resultados son proporcionados por la combinación de características LWDLE con un promedio en la medida F_1 de 0.508, seguido por WR-300 y LWD con un promedio en la medida F1 de 0.501 y 0.500, respectivamente. En la Tabla 5.7, se encontró que campo aleatorio condicional tuvo un resultado significante para clasificar la clase “actual”, sin embargo, no ocurrió lo mismo en las clases “hipotético”, “protección” y “genérico”.

Tabla 5. 4. Resultados predictivos para la tarea de clasificación de eventos con base en la modalidad contextual

Clasificador	Clase	Train + Dev			Test		
		Precisión	Recall	F_1	Precisión	Recall	F_1
LWD	N/A	0.998	0.998	0.998	0.984	0.988	0.986
	Actual	0.981	0.980	0.980	0.828	0.842	0.835
	Hipotético	0.980	0.952	0.966	0.679	0.224	0.337
	Protección	0.976	0.948	0.961	0.442	0.253	0.094
	Genérico	0.973	0.980	0.977	0.512	0.164	0.249

LWDLE	N/A	0.998	0.998	0.998	0.984	0.989	0.987
	Actual	0.981	0.980	0.980	0.835	0.841	0.838
	Hipotético	0.980	0.952	0.966	0.666	0.229	0.340
	Protección	0.973	0.953	0.963	0.465	0.062	0.110
	Genérico	0.973	0.980	0.950	0.517	0.180	0.268
WR-300	N/A	0.994	0.995	0.995	0.980	0.990	0.985
	Actual	0.936	0.935	0.935	0.845	0.791	0.816
	Hipotético	0.896	0.730	0.804	0.534	0.236	0.328
	Protección	0.865	0.614	0.718	0.238	0.092	0.133
	Genérico	0.927	0.811	0.865	0.365	0.180	0.241

Como se puede observar en la Figura 5.4, campo aleatorio condicional proporciona mejores resultados en el corpus de entrenamiento que en el corpus de prueba, con un promedio en la medida F1 de 0.977, 0.971 y 0.863 usando LWD, LWDLE, y WR-300, respectivamente, Sin embargo, no proporcionó resultados favorables en el conjunto de prueba. Por lo tanto, creemos que el problema de "*overfitting*" es ocasionado por el sesgo en el conjunto de datos. Esto significa, que las clases se encuentran desbalanceadas, y por lo tanto, el algoritmo de campo aleatorio condicional no es capaz de generalizar los nuevos datos.

3. Polaridad

Este experimento se basa en la clasificación de los eventos de acuerdo con su polaridad, es decir, positivo o negativo. Recuérdese que en estos experimentos sólo se empleó el algoritmo de campo aleatorio condicional con diversas características (Véase Figura 5.5).

A continuación, se muestra una tabla comparativa entre las características utilizadas y campo aleatorio condicional en términos de precisión, recall y F_1 (Véase Tabla 5.8).

Figura 5. 4. Promedio de F1 alcanzado por campo aleatorio condicional para la tarea de clasificación de eventos con base en la modalidad contextual

Tabla 5. 5. Resultados predictivos para la tarea clasificación de eventos con base en su polaridad

Clasificador	Clase	Train + Dev			Test		
		Precisión	Recall	F_1	Precisión	Recall	F_1
LWD	No evento	0.998	0.998	0.998	0.984	0.988	0.986
	pos	0.982	0.981	0.981	0.858	0.834	0.846
	neg	0.990	0.977	0.984	0.862	0.693	0.768
LWDLE	No evento	0.998	0.998	0.998	0.984	0.988	0.986
	pos	0.982	0.980	0.981	0.854	0.833	0.843
	neg	0.991	0.975	0.983	0.862	0.699	0.772
WR-300	No evento	0.994	0.996	0.995	0.980	0.990	0.985
	pos	0.943	0.931	0.937	0.863	0.780	0.820
	neg	0.932	0.876	0.903	0.772	0.621	0.688

Figura 5.5 muestra que el mejor resultado alcanzado sobre el conjunto de prueba fue obtenido por LWD y LWDLE, las cuales presentaron una mejora de 3.6% en detectar la polaridad de los eventos sobre WR-300.

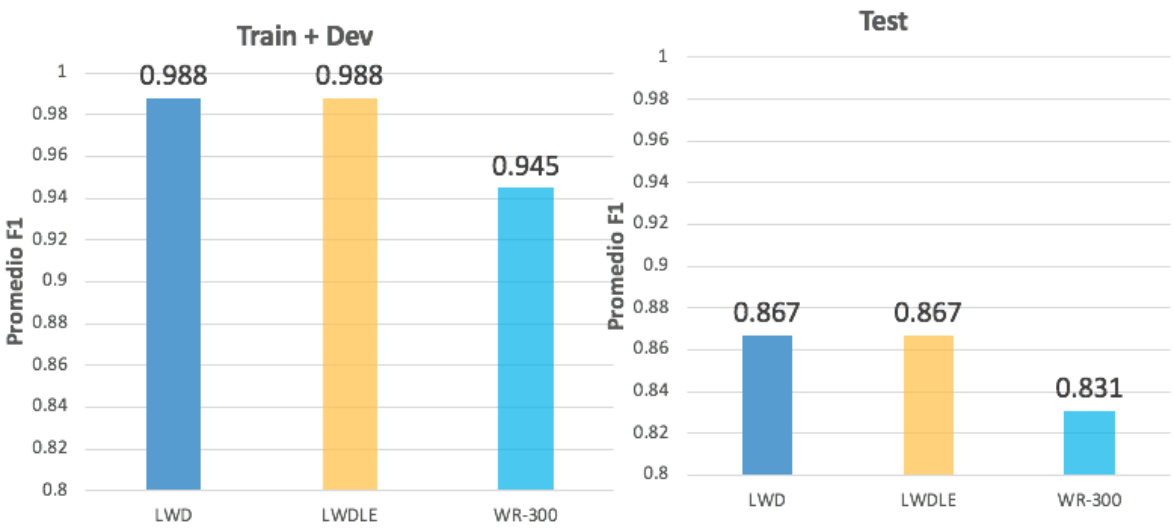

Figura 5. 5. Promedio de F_1 alcanzado por campo aleatorio condicional para la tarea de clasificación de eventos con base en la polaridad

4. Tipo

Un cuidadoso análisis de los resultados presentados en la Tabla 5.9 indica que WR-300 proporcionó los mejores resultados que LWD y LWDLE. Además, se observa que esta es la única tarea donde WR proporcionó los mejores resultados, considerando que el promedio en F_1 es de 0.778.

Tabla 5. 6. Resultados predictivos para la tarea de clasificación de eventos de acuerdo con el tipo

Clasificador	Clase	Train + Dev			Test		
		Precisión	Recall	F_1	Precisión	Recall	F_1
LWD	No evento	0.998	0.998	0.998	0.984	0.988	0.986
	n/a	0.982	0.977	0.980	0.858	0.832	0.845
	aspectual	0.991	0.974	0.982	0.690	0.222	0.337

	evidencial	0.956	0.979	0.967	0.750	0.690	0.702
LWDLE	No evento	0.998	0.998	0.998	0.984	0.988	0.986
	n/a	0.982	0.977	0.980	0.864	0.831	0.847
	aspectual	0.991	0.978	0.984	0.661	0.204	0.312
	evidencial	0.956	0.980	0.968	0.751	0.661	0.704
WR-300	No evento	0.994	0.996	0.995	0.980	0.990	0.985
	n/a	0.950	0.905	0.927	0.733	0.436	0.547
	aspectual	0.950	0.905	0.927	0.733	0.436	0.547
	evidencial	0.898	0.927	0.912	0.738	0.767	0.752

Analizando los resultados de la Tabla 5.9 y la Figura 5.6, se observa que la diferencia en desempeño entre el conjunto de prueba y entrenamiento es enorme, debido al problema de "*overfitting*". Esto se debe a que campo aleatorio condicional ajusta muy bien los datos de entrenamiento, pero falla cuando nuevos datos son tomados en consideración.

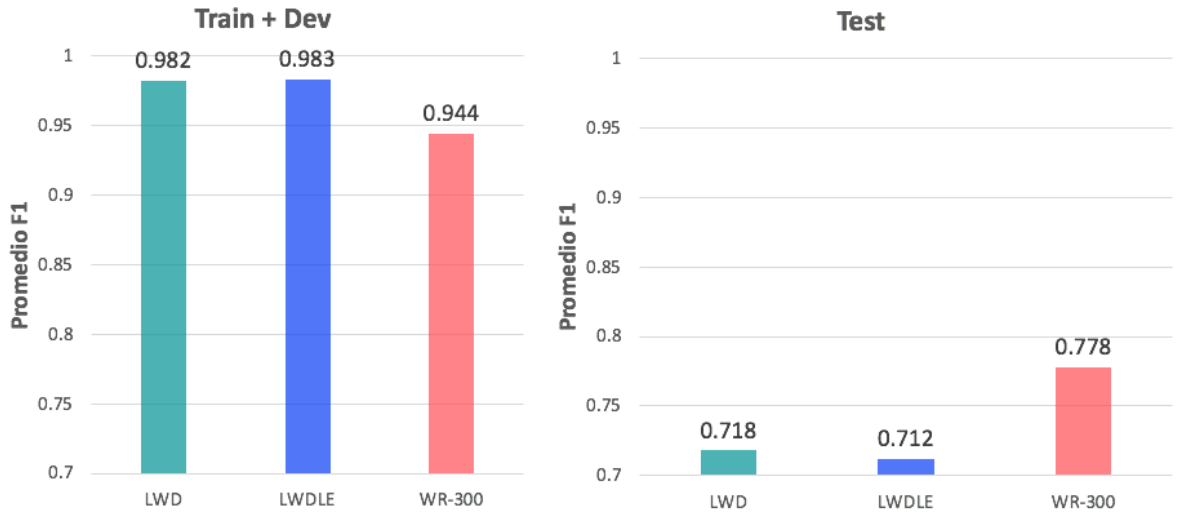

Figura 5. 6. Promedio de F_1 alcanzado por campo aleatorio condicional para la tarea de clasificación de eventos con base en el tipo

5.2.2.4 Conclusiones

Como se presentó anteriormente, los mejores desempeños en las tareas: detección de eventos, modalidad contextual y polaridad se obtuvieron con la combinación de características LWDLE. Por lo tanto, se encontró que características simples pueden ser más efectivas que *word2vec*, el cual, sólo alcanzó los mejores resultados en la tarea de clasificación de eventos de acuerdo con su tipo, no obstante al utilizar el sistema de SemEval no proporcionó los mejores resultados.

Una contribución sustancial de este trabajo es que campo aleatorio condicional y la selección de características puede predecir mejores resultados que los trabajos presentados en SemEval 2016 para las tareas de polaridad y tipo, donde LWDLE tomaron un papel importante para lograr mejorar el desempeño de las tareas mencionadas anteriormente. Sin embargo, no se logró mejorar los desempeños de campo aleatorio condicional para las tareas de detección de eventos, y clasificación de acuerdo con su modalidad contextual.

5.2.3 Resultados de SemEval 2016 - Task 12: Clinical TempEval

Como se presentó anteriormente, los mejores resultados en el experimento 1 fueron alcanzados por campo aleatorio condicional para las tareas de detección de eventos y clasificación de eventos con base en la modalidad contextual, seguido de Viterbi con evolución diferencial y Viterbi con *hill climbing*. Finalmente, el peor algoritmo fue Viterbi cuando no se aplicó ninguna técnica de optimización. De acuerdo con el experimento 2, existieron variaciones en los resultados al aplicar las diferentes combinaciones de características.

Es importante mencionar que para poder realizar una comparación directa con los trabajos presentados en *SemEval 2016: Task 12- Clinical TempEval*, se utilizó el sistema proporcionado por la competencia para la evaluación de cada una de las tareas (Véase Tabla 5.10).

Tabla 5. 7. Resultados obtenidos utilizando el sistema de SemEval en la publicación 1

Tarea	Algoritmo	Total	Predichos	Correctos	P	R	F_1
Detección	**Campo aleatorio condicional**	**18,989**	**18,352**	**18,008**	**0.981**	**0.948**	**0.965**
	Evolución diferencial	18,989	18,723	16,367	0.874	0.862	0.868
	Hill climbing	18,989	19,967	16,187	0.854	0.810	0.831
	Baseline: Viterbi	18,989	44,510	17,875	0.401	0.943	0.563
Clasificación	**Campo aleatorio condicional**	**18,989**	**17,896**	**16,328**	**0.912**	**0.860**	**0.885**
	Evolución diferencial	18,989	31,301	15,045	0.480	0.792	0.598
	Hill climbing	18,989	35,272	12,971	0.367	0.683	0.478
	Baseline: Viterbi	18,989	45,345	14,851	0.327	0.782	0.461

Tabla 5. 8. Resultados obtenidos utilizando el sistema de SemEval en la publicación 2.

Tarea	Algoritmo	Total	Predichos	Correctos	P	R	F_1
Detección	LWD	18,989	18,431	16,124	0.875	0.849	0.862
	LWDLE	**18,989**	**18,309**	**16,127**	**0.881**	**0.849**	**0.865**
	HWR-300	18,989	17,143	15,245	0.889	0.803	0.844
Clasificación	**LWD**	**18,989**	**15,905**	**14,892**	**0.936**	**0.784**	**0.854**
	LWDLE	18,989	15,897	14,881	0.936	0.784	0.853
	HWR-300	18,989	15,126	14,028	0.927	0.739	0.822
Polaridad	**LWD**	**18,989**	**15,984**	**15,550**	**0.973**	**0.819**	**0.889**
	LWDLE	18,989	16,002	15,561	0.972	0.819	0.889
	HWR-300	18,989	15,127	14,505	0.958	0.762	0.849
Tipo	**LWD**	**18,989**	**15,951**	**15,463**	**0.969**	**0.814**	**0.885**
	LWDLE	18,989	15,929	15,436	0.969	0.813	0.884
	HWR-300	18,989	15,153	14,738	0.973	0.774	0.862

5.2.4 Comparativa de los resultados con los trabajos presentados en la tarea 12 de la evaluación semántica

De acuerdo con *SemEval 2016: Task 12- Clinical TempEval*, los mejores resultados son presentados en la Tabla 5.12, y se incluyen los resultados de ambas publicaciones.

Tabla 5. 9. Comparación de nuestros resultados con los presentados en SemEval 2016: Task 12- Clinical TempEval en las tareas de detección de eventos y clasificación de eventos con base en la modalidad contextual. Las posiciones en la competencia están basadas en la medida F_1. Algunos sistemas son omitidos, ver [1] para una lista completa

Rank	Team	Detección de eventos		
		P	R	F1
-	**Our-CRF**	**0.981**	**0.948**	**0.965**
1	UTHEALTH-1	0.915	0.891	0.903
2	UTHEALTH-2	0.903	0.886	0.895
3	UTAHBMI-1	0.897	0.886	0.892
4	UTAHBMI-2	0.902	0.883	0.892
…	…	…	…	…
9	UTA-5	0.900	0.850	0.874
-	**Our: CRFs-LWDLE**	**0.881**	**0.849**	**0.865**
-	**Our: CRFs-LWD**	**0.875**	**0.849**	**0.862**
-	**Our: Viterbi con evolución diferencial**	**0.874**	**0.862**	**0.868**
10	VUACLTL	0.868	0.828	0.847

Rank	Team	Modalidad contextual		
		P	R	F1
-	**Our-CRF**	**0.912**	**0.860**	**0.885**
1	UTHEALTH-1	0.866	0.843	0.855
-	**Our: CRFs-LWD**	**0.936**	**0.784**	**0.854**
-	**Our: CRFs-LWDLE**	**0.936**	**0.784**	**0.853**
2	UTHEALTH-2	0.855	0.839	0.847
3	UTAHBMI-1	0.850	0.832	0.841
4	UTAHBMI-2	0.841	0.831	0.836
…	…	…	…	…
7	GUIR-1	0.830	0.817	0.824
-	**Our: CRFs-WR-300**	**0.927**	**0.739**	**0.822**
8	UTA-4	0.876	0.812	0.842

-	**Our: CRFs-WR-300**	**0.889**	**0.803**	**0.844**
…	…	…	…	…
13	LIMSI-2	0.869	0.816	0.842
-	**Our: Viterbi con *hill climbing***	**0.854**	**0.810**	**0.831**
…	**…**	…	…	…
16	Brundlefly	0.883	0.660	0.755
-	**Our baseline: Viterbi**	**0.401**	**0.943**	**0.563**

…	…	…	…	…
15	Brundlefly	0.819	0.612	0.701
-	**Our: Viterbi con evolución diferencial**	**0.480**	**0.792**	**0.598**
-	**Our: Viterbi con *hill climbing***	**0.367**	**0.683**	**0.478**
-	**Our baseline: Viterbi**	**0.327**	**0.782**	**0.461**

Tabla 5. 10. Comparación de nuestros resultados con los presentados en SemEval 2016: Task 12- Clinical TempEval en las tareas de polaridad y tipo. Las posiciones en la competencia están basadas en la medida F_1. Algunos sistemas son omitidos, ver [1] para una lista completa

Rank	**Team**	**Polaridad**		
		P	**R**	**F1**
-	**Our-CRFs-LWD**	**0.973**	**0.819**	**0.889**
-	**Our-CRFs-LWDLE**	0.972	0.819	0.889
1	UTHEALTH-1	0.900	0.875	0.887
2	UTHEALTH-2	0.888	0.872	0.880
3	UTAHBMI-1	0.879	0.869	0.874

Rank	**Team**	**Tipo**		
		P	**R**	**F1**
-	**Our-CRFs-LWD**	**0.969**	**0.814**	**0.885**
-	**Our: CRFs-LWDLE**	**0.969**	**0.813**	**0.884**
1	UTHEALTH-1	0.894	0.870	0.882
2	UTHEALTH-2	0.880	0.863	0.871
3	UTAHBMI-1	0.875	0.857	0.866

…	…	…	…	…
7	GUIR-1	0.869	0.855	0.862
-	**Our: CRFs: WR-300**	**0.959**	**0.764**	**0.850**
8	UTA-4	0.876	0.812	0.842
…	…	…	…	…
16	Brundlefly	0.856	0.640	0.733

-	**Our: CRFs: WR-300**	**0.973**	**0.776**	**0.863**
4	UTAHBMI-2	0.854	0.843	0.849
…	…	…	…	…
16	Brundlefly	0.829	0.620	0.709

El sistema de SemEval evaluó dos tipos de situaciones:

- Run1: Acceso a los textos clínicos sin gold standard.
- Run2: Acceso a los textos clínicos y acceso al gold standard.

De acuerdo con las Tablas 5.12 y 5.13, se observa que el mejor resultado fue obtenido por el equipo UTHEALTH [26] con una medida de F_1 de 0.903 en detección de eventos, 0.855 en la modalidad contextual, 0.887 en la polaridad, y 0.882 en el tipo de eventos. Sin embargo, estos resultados fueron superados por nuestra propuesta.

En la tarea de detección de eventos, el algoritmo de campo aleatorio condicional del experimento 1, logró una tasa de clasificación de 18,008 eventos de los 18,989 eventos presentes en el corpus con un promedio en la medida F_1 de 0.965, mostrando una significativa tasa de predicción con respecto al ganador del concurso, el cual obtuvo 0.903.

En la tarea de clasificación de eventos con base en la modalidad contextual, el algoritmo de campo aleatorio condicional presentó una mejor tasa de predicción con un promedio en la medida F_1 de 0.885 que el equipo ganador con 0.855.

En la tarea de clasificación de eventos con base en su polaridad, la combinación de características LWDLE obtuvo un promedio en la medida F_1 de 0.889, el cual presento 0.2 % de mejora que el equipo ganador.

Finalmente, en la tarea de clasificación de eventos con base en su tipo, la combinación de LWDLE presentó una mejora del 0.3% sobre el equipo ganador, quien obtuvo 0.882 en la medida F_1.

Es importante mencionar que aunque las otras características no proporcionaron el primer lugar en la competencia, nos permiten identificar los algoritmos y características adecuadas (resultados superiores) y competitivos (resultados inferiores).

Capítulo 6. Conclusiones y trabajo a futuro

En este capítulo se presentan las conclusiones del estudio realizado en este trabajo de tesis. Además se proponen ideas que pueden llevarse a cabo para posibles trabajos futuros.

6.1 Conclusiones

Con base en los objetivos en este trabajo de investigación, se describen las siguientes conclusiones.

1. De acuerdo con el estado del arte, los algoritmos utilizados en el análisis de historias clínicas para la extracción de eventos son campo aleatorio condicional, redes neuronales, reglas de asociación y máquinas de vectores de soporte.
2. Debido al poco acceso de la información en México, se empleó un corpus clínico de la clínica Mayo, cuyo contenido está basado en pacientes con cáncer de colón.
3. Con base en el análisis realizado al conjunto de datos, se aplicaron dos técnicas de análisis más empleadas: análisis sintáctico y léxico.
4. Se desarrolló una metodología competitiva, utilizando el algoritmo de Viterbi y dos técnicas de optimización conocidas como evolución diferencial y *hill climbing*. Así como también, el uso de combinaciones de características para obtener una mejor precisión.
5. Se realizaron experimentos con el algoritmo de campo aleatorio condicional y se lograron desempeños competitivos entre los modelos clásicos presentados en el estado del arte para la extracción de eventos clínicos desde textos médicos. Los resultados de los experimentos realizados en este trabajo superaron los resultados presentados por la competencia de SemEval 2016-Tarea 12: Clinical TempEval.

6. Se mejoró el algoritmo de Viterbi, para la identificación de dos tareas: detección y clasificación de eventos.

6.2 Publicaciones derivadas de este trabajo de tesis

De este trabajo se realizaron los siguientes artículos en la revista "Journal of Intelligent & Fuzzy Systems (JIFS)"

1. *Extracting medical events from clinical records using conditional random fields and parameter tuning for hidden Markov models*
2. *Medical events extraction to analyze clinical records with conditional random fields*

Además la aplicación de la técnica de *hill climbing* para resolver otros problemas fue presentado en la siguiente publicación, presentado en la revista "*Evolving systems".*

3. *Dendrite ellipsoidal neurons based on k-means optimization*

6.3 Trabajos a futuro

Se recomiendan:

1. Seleccionar otros corpus clínicos, con el objetivo de utilizar el algoritmo de campo aleatorio condicional, y comparar los resultados obtenidos con los algoritmos presentados en el estado del arte.
2. Recopilar datos de pacientes mexicanos para aplicar el estudio a nivel nacional
3. Utilizar métodos para tratar el problema de "*overfitting*".

4. Modificar el código de evolución diferencial y *hill climbing* para reducir los tiempos de ejecución. Así como también, aplicar técnicas de optimización de los pesos.
5. Etiquetar el corpus clínico de pacientes con cáncer cerebral.
6. Utilizar otros algoritmos como máquina de vectores de soporte.
7. Aplicar significancia estadística

6.4 Contribuciones del trabajo

En esta sección se presentan las contribuciones:

- Una mejora del algoritmo de Viterbi utilizando métodos de optimización como evolución diferencial y *hill climbing*, con el objetivo de maximizar las propiedades y reducir la tasa de error en el algoritmo de Viterbi.
- Una metodología que permite la identificación y la clasificación de eventos clínicos con resultados superiores y competitivos.
- Evaluación de los métodos y las características extraídas en la competencia de SemEval 2016 Tarea 12, donde obtuvimos resultados competivos y superiores en las tareas de detección, clasificación, polaridad y tipo de eventos.

Referencias

[1] *SemEval-2016 The 10th International Workshop on Semantic Evaluation Proceedings of the Workshop San Diego , California , USA*. 2016.

[2] A. AAl Abdulsalam, S. Velupillai, and S. Meystre, "UtahBMI at SemEval-2016 Task 12: Extracting Temporal Information from Clinical Text," *Proc. 10th Int. Work. Semant. Eval.*, pp. 1256–1262, 2016.

[3] J. Urbain, "Mining heart disease risk factors in clinical text with named entity recognition and distributional semantic models," *J. Biomed. Inform.*, vol. 58, pp. S143–S149, 2015.

[4] J. Tourille, O. Ferret, X. Tannier, and A. Névéol, "LIMSI-COT at SemEval-2017 Task 12: Neural Architecture for Temporal Information Extraction from Clinical Narratives," *Semeval*, pp. 595–600, 2017.

[5] A. Leeuwenberg and M.-F. Moens, "KULeuven-LIIR at SemEval-2017 Task 12: Cross-Domain Temporal Information Extraction from Clinical Records," *Proc. 11th Int. Work. Semant. Eval.*, no. Phase I, pp. 1029–1033, 2017.

[6] S. MacAvaney, A. Cohan, and N. Goharian, "GUIR at SemEval-2017 Task 12: A Framework for Cross-Domain Clinical Temporal Information Extraction," *Proc. 11th Int. Work. Semant. Eval.*, pp. 1021–1026, 2017.

[7] A. Ben Abacha and P. Zweigenbaum, "Medical entity recognition: a comparison of semantic and statistical methods," *2011 Work. Biomed. Nat. Lang. Process.*, no. ii, pp. 56–64, 2011.

[8] C. Sun, Y. Guan, X. Wang, and L. Lin, "Rich features based Conditional Random Fields for biological named entities recognition," *Comput. Biol. Med.*, vol. 37, no. 9, pp. 1327–1333, 2007.

[9] M. Gridach, "Character-Level Neural Network for Biomedical Named Entity Recognition," *J. Biomed. Inform.*, vol. 70, pp. 85–91, 2017.

[10] A. Pérez, R. Weegar, A. Casillas, K. Gojenola, M. Oronoz, and H. Dalianis, "Semi-supervised medical entity recognition: a study on Spanish and Swedish clinical corpora," *J. Biomed. Inform.*, vol. 71, pp. 16–30, 2017.

[11] “UMLS,” 2009. .

[12] K. Gaizauskas, Robert; Demetriou, George; Humphreys, “Term Recognition and Classification in Biological Science Journal Articles.”

[13] C. G. C. Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, “Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications,” *J. Am. Med. Informatics Assoc.*, vol. 17, no. 5, pp. 507–513, 2010.

[14] S. P R, M. R, and Y. Niwa, “Hitachi at SemEval-2016 Task 12: A Hybrid Approach for Temporal Information Extraction from Clinical Notes,” *Proc. 10th Int. Work. Semant. Eval.*, vol. 2015, pp. 1231–1236, 2016.

[15] A. Cohan, K. Meurer, and N. Goharian, “GUIR at SemEval-2016 task 12: Temporal Information Processing for Clinical Narratives,” *Proc. 10th Int. Work. Semant. Eval.*, pp. 1248–1255, 2016.

[16] C. Focil Arias, “Cómputo no convencional para la estimación del desempeño académico,” Centro de Investigación en Computación, 2015.

[17] F. Fócil Arias, Carolina; Sidorov, Grigori; Gelbukh, Alexander; Arce, “Extracting medical events from clinical records using conditional random fields and parameter tuning for hidden Markov models,” *J. Intell. Fuzzy Syst.*, vol. 34, no. 5, pp. 2935–2947, 2018.

[18] Z. Liu *et al.*, “Automatic de-identification of electronic medical records using token-level and character-level conditional random fields,” *J. Biomed. Inform.*, vol. 58, pp. S47–S52, 2015.

[19] V. R. Chikka, “CDE-IIITH at SemEval-2016 Task 12 : Extraction of Temporal Information from Clinical documents using Machine Learning techniques,” pp. 1237–1240, 2016.

[20] C. Hansart, D. De Meyere, P. Watrin, A. Bittar, and C. Fairon, “CENTAL at SemEval-2016 Task 12: a linguistically fed CRF model for medical and temporal information extraction,” *Proc. 10th Int. Work. Semant. Eval.*, pp. 1286–1291, 2016.

[21] T. Caselli and R. Morante, “VUACLTL at SemEval 2016 Task 12: A CRF Pipeline to Clinical TempEval,” *Proc. 10th Int. Work. Semant. Eval.*, pp. 1241–1247, 2016.

[22] C. Grouin and V. MORICEAU, “LIMSI at SemEval-2016 Task 12: machine-learning

and temporal information to identify clinical events and time expressions," *Proc. 10th Int. Work. Semant. Eval.*, pp. 1225–1230, 2016.

[23] M. Barros, A. Lamurias, M. Antunes, J. Teixeira, A. Pinheiro, and F. M. Couto, "ULISBOA at SemEval-2016 Task 12 : Extraction of temporal expressions , clinical events and relations using IBEnt," *Proc. 10th Int. Work. Semant. Eval. (SemEval 2016)*, pp. 1263–1267, 2016.

[24] P. Abacha, A.B.; Chowdhury, M.F.M.; Karanasiou, A.; Mrabet, Y.; Zweigenbaum, "Text mining for pharmacovigilance: Using machine learning for drug name recognition and drug–drug interaction extraction and classifi- cation," *J. Biomed. Inform.*, vol. 58, pp. 122–132, 2015.

[25] C. Forsyth, A. W. ; Barzilay, R.; Hughes, K. S.; Lui, D.; Lorenz, K. A.; Enzinger, A.; Tulsky, J.A.; Lindvall, "Machine learning methods to extract documentationof breast cancer symptons from electronic health records," *J. Pain Symptom Manage.*, 2018.

[26] H.-J. Lee *et al.*, "UTHealth at SemEval-2016 Task 12: an End-to-End System for Temporal Information Extraction from Clinical Notes," *Proc. 10th Int. Work. Semant. Eval.*, pp. 1292–1297, 2016.

[27] A. Leeuwenberg and M.-F. Moens, "KULeuven-LIIR at SemEval 2016 Task 12: Detecting Narrative Containment in Clinical Records," *Proc. 10th Int. Work. Semant. Eval.*, pp. 1280–1285, 2016.

[28] J. Tourille, O. Ferret, A. Névéol, and X. Tannier, "LIMSI-COT at SemEval-2016 Task 12: Temporal relation identification using a pipeline of classifiers," *Proc. 10th Int. Work. Semant. Eval.*, pp. 1136–1142, 2016.

[29] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[30] J. A. Fries, "Brundlefly at SemEval-2016 Task 12: Recurrent Neural Networks vs. Joint Inference for Clinical Temporal Information Extraction," *Proc. 10th Int. Work. Semant. Eval.*, pp. 1274–1279, 2016.

[31] P. Li and H. Huang, "UTA DLNLP at SemEval-2016 Task 12: Deep Learning Based Natural Language Processing System for Clinical Information Identification from Clinical Notes and Pathology Reports," *Proc. 10th Int. Work. Semant. Eval.*, pp. 1268–1273, 2016.

[32] A. Lauren, P., Qu, G., Zhang, F., and Lendasse, "Discriminant document embeddings with an extreme learning machine for classifying clinical narrative," *Neurocomputing*, vol. 277, pp. 128–138, 2018.

[33] M. A. Gulcelik, H. Dincer, D. Sahin, O. F. Demir, and E. Yenidogan, "THYME Annotation Guidelines." 2010.

[34] W. Styler, G. Savova, M. Palmer, J. Pustejovsky, T. O. Gorman, and P. C. De, "THYME Annotation Guidelines Developed," 2014.

[35] H. Uğuz, A. Arslan, and İ. Türkoğlu, "A biomedical system based on hidden Markov model for diagnosis of the heart valve diseases," *Pattern Recognit. Lett.*, vol. 28, no. 4, pp. 395–404, 2007.

[36] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi, "Hidden Markov models for cancer classification using gene expression profiles," *Inf. Sci. (Ny).*, vol. 316, pp. 293–307, 2015.

[37] Y. Wang *et al.*, "Supervised methods for symptom name recognition in free-text clinical records of traditional Chinese medicine: An empirical study," *J. Biomed. Inform.*, vol. 47, pp. 91–104, 2014.

[38] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[39] W. Timp, J. Comer, and A. Aksimentiev, "DNA base-calling from a nanopore using a viterbi algorithm," *Biophys. J.*, vol. 102, no. 10, pp. L37–L39, 2012.

[40] W. Xiang, X. Meng, M. An, Y. Li, and M. Gao, "An Enhanced Differential Evolution Algorithm Based on Multiple Mutation Strategies," *Comput. Intell. Neurosci.*, vol. 2015, pp. 1–15, 2015.

[41] K. V: Price, R. M. Stron, and J. A. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization*, vol. 1. 2008.

[42] D. M. Diab and K. M. El Hindi, "Using differential evolution for fine tuning naïve Bayesian classifiers and its application for text classification," *Appl. Soft Comput. J.*, vol. 54, pp. 183–199, 2017.

[43] G. Wang, S.; Hussein, M. A.; Baudoin, "Optimal sizing of generalized memory polynomialmodel structure based on hill climbing heuristic," in *46th European Microwave Conference (EuMC)*, 2016, pp. 190–193.

[44] Brownlee, *Clever Algorithms: Nature-Inspired Programming Recipes*. 2011.

[45] “Design and Analysis of Algorithms.” .

[46] J. Lafferty, A. Mccallum, and F. Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data,” in *ICML*, 2001, vol. 2001, no. June, pp. 282–289.

[47] A. Álvarez, C. D. Martínez-Hinarejos, H. Arzelus, M. Balenciaga, and A. del Pozo, “Improving the automatic segmentation of subtitles through conditional random field,” *Speech Commun.*, vol. 88, pp. 83–95, 2017.

[48] C. Sutton, “An Introduction to Conditional Random Fields,” *Found. Trends® Mach. Learn.*, vol. 4, no. 4, pp. 267–373, 2012.

[49] P. Blunsom, “Hidden Markov Models,” *Lect. notes, August*, pp. 1–7, 2004.

[50] Â. A.R. Sá, A. Andrade O., and A. B. Soares, “Estimation of Hidden Markov Models Parameters using Differential Evolution,” *AISB 2008 Conv. Commun. Interact. Soc. Intell.*, vol. 1, pp. 51–56, 2008.

[51] Na. Okazaki, “CRFsuite.” .

[52] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, vol. 1. 2009.

[53] S. Bethard, W. Chen, J. Pustejovsky, L. Derczynski, and M. Verhagen, “SemEval-2016 Task 12 : Clinical TempEval,” pp. 1052–1062, 2016.

[54] “Unified Medical Language System,” 2009. [Online]. Available: https://www.nlm.nih.gov/research/umls/#.

Anexo A

Variables utilizadas en los trabajos presentados en SemEval 2016 para la detección de eventos clínicos.

Referencias	Léxicas	Sintácticas	Semánticas	Discourse	Recursos	Palabras	Minúsculas	Lema	Partes del	Chunk	Tipo de	Forma de	Sección	Patrón	N-grams	Tam. de	Prefijo	Sufijo	Agrupación	Stemming	Word2vec	UMLS
[26]	x	x		x	x	x																
[2]						x	x	x	x	x	x	x	x	x								
[14]												x			x	x	x	x				
[20]	x						x		x		x											
[22]						x			x		x					x			x	x	x	x
[19]					x	x			x	x												
[30]																					x	
[21]			x					´x	x													
[15]							x	x	x		x						x	x	x			
[28]			x					x	x		x											

Anexo B

Algoritmos reportados en el SemEval 2016 para la extracción de eventos clínicos.

Referencia	CRF	SVM	Reglas	Regresión Logística	RF	DL
[2]	X					
[14]	X		X			
[15]	X			X		
[19]	X	X				X
[20]	X					
[21]	X					
[22]	X					
[26]		X				
[28]		X			X	
[30]						X